Developer Note

Power Macintosh 7200 Computer

Developer Press

Every effort has been made to ensure that the information in this manual is accurate. Apple is not responsible for printing or clerical errors.

Apple Computer, Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Simultaneously published in the United States and Canada.

LIMITED WARRANTY ON MEDIA AND REPLACEMENT

If you discover physical defects in the manual or in the media on which a software product is distributed, APDA will replace the media or manual at no charge to you provided you return the item to be replaced with proof of purchase to APDA.

ALL IMPLIED WARRANTIES ON THIS MANUAL, INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO NINETY (90) DAYS FROM THE DATE OF THE ORIGINAL RETAIL PURCHASE OF THIS PRODUCT.

Even though Apple has reviewed this manual, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS MANUAL IS SOLD “AS IS,” AND YOU, THE PURCHASER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS MANUAL, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Figures and Tables vii

Preface **About This Developer Note** ix

Contents of This Note ix
Supplemental Reference Documents x
  Apple Publications x
  Obtaining Information From APDA x
  Other Publications xi
Conventions and Abbreviations xi
  Typographical Conventions xi
  Standard Abbreviations xii

Chapter 1 **Introduction** 1

Appearance and Features 2
Comparison With Other Power Macintosh Computers 5
Configurations and Options 7
Compatibility 8
  Machine Identification 8
  Open Transport 8
  NuBus Expansion Cards 8
  Video Cards 8
  Power Supply 9

Chapter 2 **Architecture** 11

Main ICs and Subsystems 12
  Block Diagram 12
  PowerPC 601 Processor 12
  Read-Only Memory 12
  Random-Access Memory 13
  Second-Level Cache 14
  Platinum Memory and Video Controller IC 14
    Data Interleaving 14
    Bank Base Registers 14
    Iridium IC 14
  Bandit Bus Bridge IC 15
    Bus Clock Rates 15
    Big-Endian and Little-Endian Bus Addressing 15

Grand Central I/O System IC 15
Curio I/O Controller IC 16
Cuda Microcontroller IC 16
AWACS Sound IC 16
VRAM and VRAM Expansion 17

Chapter 3 I/O Features 19

I/O Ports 20
Serial Ports 20
Apple Desktop Bus (ADB) Port 21
Ethernet Port 22
SCSI Port 23
Sound Input Jack 25
Sound Output Jack 25
Video Monitor Connector 26
Disk Drives 28
Floppy Disk Drive 28
Internal Hard Disk Drive 29
CD-ROM Drive 29

Chapter 4 Expansion Features 31

RAM DIMMs 32
RAM DIMM Connectors 33
RAM Address Multiplexing 36
RAM Devices 38
RAM Refresh 38
RAM DIMM Dimensions 38
Second-Level Cache SIMM 40
PCI Expansion Slots 42

Chapter 5 Software Features 45

New Features 46
Large Partition Support 46
64-Bit Volume Addresses 46
System-Level Software 47
Application-Level Software 47
Limitations 48
Drive Setup 48

Open Transport 48
New Features of Open Transport 49
Compatibility 49
Open Firmware Startup 50
Enhanced Power Management Software 51
Features of the New Energy Saver Software 52
Performance Enhancements 53
Dynamic Recompilation Emulator 53
Resource Manager in Native Code 53
Math Library 54
New BlockMove Extensions 54
Hardware Support Features 55
PCI Bus Support 56
Removal of Slot Manager Dependencies 56
PCI Compatibility 56
QuickDraw Acceleration API 56
Display Manager 57
Support of Native Drivers 57

Chapter 6 Large Volume Support 59

Overview of the Large Volume File System 60
API Changes 60
Allocation Block Size 60
File Size Limits 61
Compatibility Requirements 61
The API Modifications 61
Data Structures 61
Extended Volume Parameter Block 61
Extended I/O Parameter Block 63
New Extended Function 65

Glossary 69

Index 73

Figures and Tables

Chapter 1 Introduction 1

Figure 1-1 Front view of the Power Macintosh 7200 computer 2
Figure 1-2 Back view of the Power Macintosh 7200 computer 3
Figure 1-3 Positioning the top chassis support foot 3
Figure 1-4 Unlocking the top chassis 4
Figure 1-5 Top chassis of the Power Macintosh 7200 computer in the open position 4
Figure 1-6 Voltage switch location 9

Table 1-1 Comparison with first-generation Power Macintosh computer and the Power Macintosh 7500 computer 6
Table 1-2 Configurations 7

Chapter 2 Architecture 11

Figure 2-1 Block diagram 13

Table 2-1 Display sizes and pixel depths 17

Chapter 3 I/O Features 19

Figure 3-1 Serial port connector 20
Figure 3-2 ADB connector 21
Figure 3-3 External video connector 26

Table 3-1 Pin assignments on the serial port connector 21
Table 3-2 Pin assignments on the ADB connector 22
Table 3-3 Pin assignments on the 14-pin Apple AUI Ethernet connector 22
Table 3-4 Pin assignments on the 8-pin 10baseT Ethernet connector 23
Table 3-5 Pin assignments on the SCSI connectors 24
Table 3-6 Pin assignments on the video connector 26
Table 3-7 Sense line codes for various monitors 27
Table 3-8 Pin assignments on the floppy disk connector 28

Chapter 4 Expansion Features 31

Figure 4-1 Dimensions of the RAM DIMM 39

Table 4-1 Memory sizes and configurations 32
Table 4-2 Pin assignments on the RAM DIMM connectors 33
Table 4-3 RAM DIMM signals 36
Table 4-4 Address multiplexing modes for various DRAM devices 37
Table 4-5 Address multiplexing in noninterleaved banks 38

Table 4-6 Pin assignments for L2 cache SIMM connector 40
Table 4-7 Signal descriptions for L2 cache SIMM connector 42
Table 4-8 PCI signals 43

Chapter 5 Software Features 45

Figure 5-1 Energy Saver application dialog box 52

Table 5-1 Summary of `BlockMove` routines 55

About This Developer Note

This developer note describes the Power Macintosh 7200 computer and compares it with the earlier Power Macintosh models, emphasizing the features that are new or different.

This developer note is intended to help experienced Macintosh hardware and software developers design compatible products. If you are unfamiliar with Macintosh computers or would simply like more technical information, you may wish to read the related technical manuals listed in the section "Supplemental Reference Documents," later in this preface.

This note is published in two forms: an online version included with the Apple Developer CD and a looseleaf paper version published by APDA. For information about APDA, see "Supplemental Reference Documents."

Contents of This Note

This developer note has six chapters.

- Chapter 1, "Introduction," gives a summary of the features of the Power Macintosh 7200 computer, describes its appearance, and lists the available configurations and options.
- Chapter 2, "Architecture," describes the internal organization of the computer. It includes a block diagram and descriptions of the main components of the logic board.
- Chapter 3, "I/O Features," describes the built-in I/O devices and the external I/O ports.
- Chapter 4, "Expansion Features," describes the memory expansion DIMMS and the PCI expansion slots in the Power Macintosh 7200 computer.
- Chapter 5, "Software Features," summarizes the new features of the ROM software and system software that accompany the Power Macintosh 7200 computer.
- Chapter 6, "Large Volume Support," describes the modified software that allows the Power Macintosh 7200 computer to support volume sizes larger than 4 GB.

Supplemental Reference Documents

The following documents provide information that complements or extends the information in this developer note.

Apple Publications

For information about the first generation of Power Macintosh computers, refer to *Macintosh Developer Note Number 8,* APDA catalog number R0566LL/A.

For information about PCI expansion cards, refer to *Designing PCI Cards and Drivers for Power Macintosh Computers.*

For information about the ADB and the serial ports, you may wish to refer to *Guide to the Macintosh Family Hardware,* second edition.

Developers may also need copies of the appropriate Apple reference books. You should have the relevant books of the *Inside Macintos*h series, particularly *Inside Macintosh: Devices*, *Inside Macintosh: QuickTime Components,* and *Inside Macintosh: Operating System Utilities.*

Obtaining Information From APDA

The Apple publications listed above are available from APDA. APDA is Apple's worldwide source for hundreds of development tools, technical resources, training products, and information for anyone interested in developing applications on Apple platforms. Customers receive the *APDA Tools Catalog* featuring all current versions of Apple development tools and the most popular third-party development tools. APDA offers convenient payment and shipping options, including site licensing.

To order products or to request a complimentary copy of the *APDA Tools Catalog*, contact

APDA
Apple Computer, Inc.
P.O. Box 319
Buffalo, NY 14207-0319

Telephone	1-800-282-2732 (United States) 1-800-637-0029 (Canada) 716-871-6555 (International)
Fax	716-871-6511
AppleLink	APDA
America Online	APDAorder
CompuServe	76666,2405
Internet	APDA@applelink.apple.com

Other Publications

For information about programming the PowerPC™ 601 microprocessor, developers should have copies of Motorola's *PowerPC 601 RISC Microprocessor User's Manual.*

For mechanical specifications of the 8-byte DIMM, refer to the MO-161 specification of the JEDEC JC-11 committee. Electrical specifications are defined by the JEDEC JC-42.5 committee in JEDEC standard No. 21-C.

For codec standards, refer to the *ASCO 2300 Audio-Stereo Codec Specification* from IT&T.

For information about the PCI expansion bus, refer to the *PCI Local Bus Specification*, Revision 2.0, and *PCI Bus Binding to IEEE 1275-1994.* You can obtain these documents from

PCI Special Interest Group
Intel Corporation
M/S HF3-15A
5200 NE Elam Young Parkway
Hillsboro, Oregon 97124-6497
Telephone 800-433-5177 (U.S.)
503-797-4207 (International)

For information about the open firmware startup process, see *1275-1994 Standard for Boot (Initialization, Configuration) Firmware,* IEEE part number DS02683. It is referred to in this developer note as IEEE Standard 1275. You can order a copy from

IEEE Standards Department
445 Hoes Lane, P.O. box 1331
Piscataway, NJ 08855-1331
Telephone 800-678-4333

Conventions and Abbreviations

This developer note uses the following typographical conventions and abbreviations.

Typographical Conventions

Computer-language text—any text that is literally the same as it appears in computer input or output—appears in `Courier` font.

Hexadecimal numbers are preceded by a dollar sign ($). For example, the hexadecimal equivalent of decimal 16 is written as $10.

Note
A note like this contains information that is interesting but not essential for an understanding of the text. ◆

IMPORTANT
A note like this contains important information that you should read before proceeding. ▲

Standard Abbreviations

When unusual abbreviations appear in this book, the corresponding terms are also spelled out. Standard units of measure and other widely used abbreviations are not spelled out. Here are the standard units of measure used in this developer note:

A	amperes	mA	milliamperes
dB	decibels	µA	microamperes
GB	gigabytes	MB	megabytes
Hz	hertz	MHz	megahertz
in.	inches	mm	millimeters
k	1000	ms	milliseconds
K	1024	µs	microseconds
KB	kilobytes	ns	nanoseconds
kg	kilograms	Ω	ohms
kHz	kilohertz	sec.	seconds
kΩ	kilohms	V	volts
lb.	pounds	W	watts

Other abbreviations used in this note include the following:

$*n*	hexadecimal value *n*
AC	alternating current
ADB	Apple Desktop Bus
A/D	analog to digital
ADC	analog-to-digital converter
AGC	automatic gain control
AGND	analog ground
API	application programming interface
AUI	attachment unit interface
AWACS	audio waveform amplifier and converter
CAS	column address strobe (a memory control signal)

continued

CD-ROM	compact-disc read-only memory
CISC	complex instruction set computing
CPU	central processing unit
DAC	digital-to-analog converter
DBDMA	descriptor-based direct memory access
DC	direct current
DIMM	Dual Inline Memory Module
DIN	Deutsche Industrie Normal
DMA	direct memory access
DRAM	dynamic RAM
FIFO	first in, first out
FPU	floating-point unit
FSM	File System Manager
GND	ground
GPi	general-purpose input
HFS	hierarchical file system
IC	integrated circuit
I/O	input and output
IR	infrared
ISA	Industry Standard Architecture
ISDN	Integrated Services Digital Network
JEDEC	Joint Electron Device Engineering Council
MACE	Media Access Controller for Ethernet
MMU	memory management unit
NBP	Name-Binding Protocol
n.c.	no connection
NMI	nonmaskable interrupt
NTSC	National Television System Committee (the standard system used for broadcast TV in North America and Japan)
NVRAM	nonvolatile random-access memory
PAL	Phase Alternate Lines (the standard for broadcast TV in most of Europe, Africa, South America, and southern Asia)
PB API	parameter-block application programming interface
PCI	Peripheral Component Interconnect, an industry-standard expansion bus
PRAM	parameter random-access memory
PROM	programmable read-only memory
PWM	pulse width modulation

continued

RAM	random-access memory
RAMDAC	random-access memory, digital-to-analog converter
RAS	row address strobe
RGB	red-green-blue, a video signal format with separate red, green, and blue color components
RISC	reduced instruction set computing
rms	root-mean-square
ROM	read-only memory
SCC	Serial Communications Controller
SCSI	Small Computer System Interface
SECAM	the standard system used for broadcast TV in France and the former Soviet countries
SIMM	Single Inline Memory Module
SNR	signal to noise ratio
SPR	special-purpose register
SVGA	super video graphics adapter
S-video	a type of video connector that keeps luminance and chrominance separate; also called a Y/C connector
SWIM	Super Woz Integrated Machine (custom IC that controls the floppy disk interface)
TTL	transistor-transistor logic (a standard type of device)
VCC	positive supply voltage (voltage for collectors)
VCR	video cassette recorder
VGA	video graphics adapter
VRAM	video RAM; used for display buffers
Y/C	a type of video connector that keeps luminance and chrominance separate; also called an S-video connector
YUV	a video signal format with separate luminance and chrominance components

CHAPTER 1

Introduction

The Power Macintosh 7200 computer is the lowest-cost multislot model in the new Power Macintosh computer line. It takes the place of the Power Macintosh 6100 computer for small to medium-sized businesses. It shares the compact desktop design and internal I/O subsystems of the Power Macintosh 7500 computer, but incorporates a simplified memory and video design. It has been designed to provide greater performance and flexibility than the Power Macintosh 6100 via a higher speed processor and three internal expansion slots.

This developer note describes the Power Macintosh 7200 computer with emphasis on the features that distinguish it from the previous Power Macintosh models and the Power Macintosh 7500 Power Macintosh.

Appearance and Features

The Power Macintosh 7200 computer has the compact desktop design and includes slots for three expansion cards. The case is similar to that of the Power Macintosh 7100 and Power Macintosh 7500 computers. Figure 1-1 and Figure 1-2 show front and back views of the computer.

Figure 1-1 Front view of the Power Macintosh 7200 computer

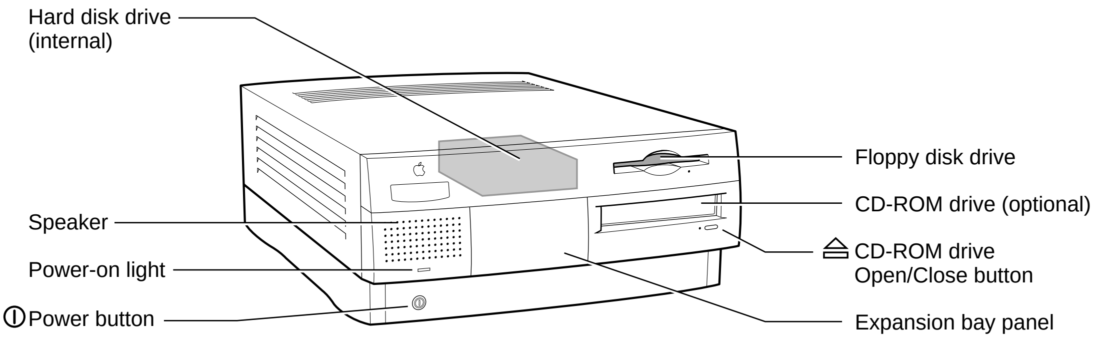
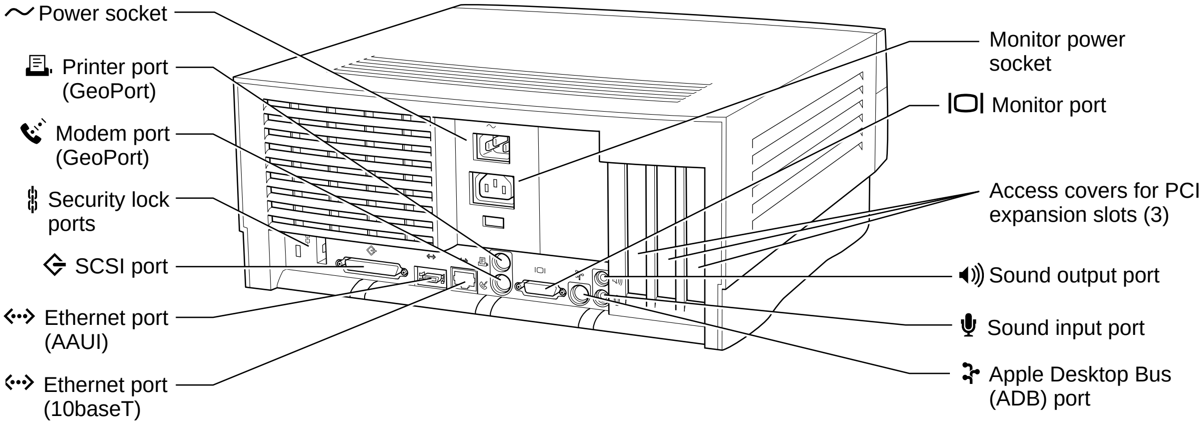


Figure 1-2 Back view of the Power Macintosh 7200 computer



The Power Macintosh 7200 computer has a hinged top chassis. The top chassis swings out of the way to provide access to the second-level cache DIMM slot, the DRAM DIMM slots, the VRAM DIMM slots, and the PCI card expansion slots on the main logic board located in the bottom chassis.

To access the expansion features, the support foot, shown in Figure 1-3, must be put into the locked position, and the top chassis release switches, shown in Figure 1-4, must be unlocked.

Figure 1-3 Positioning the top chassis support foot

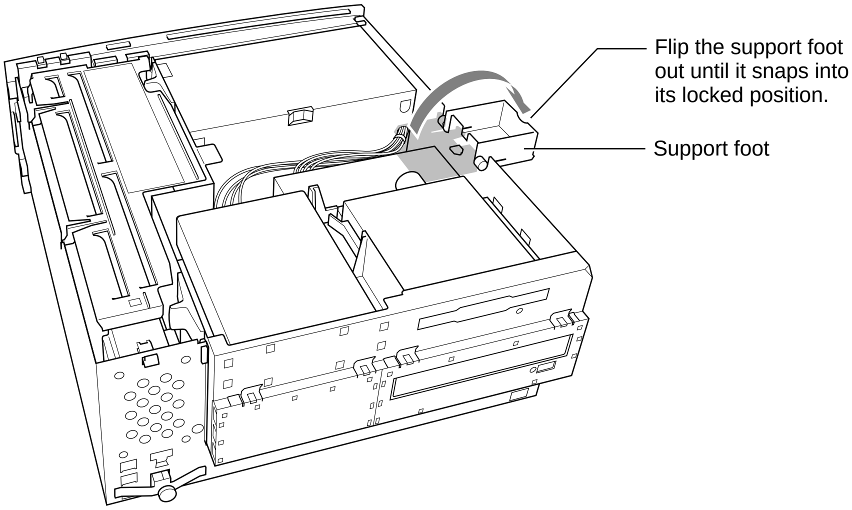
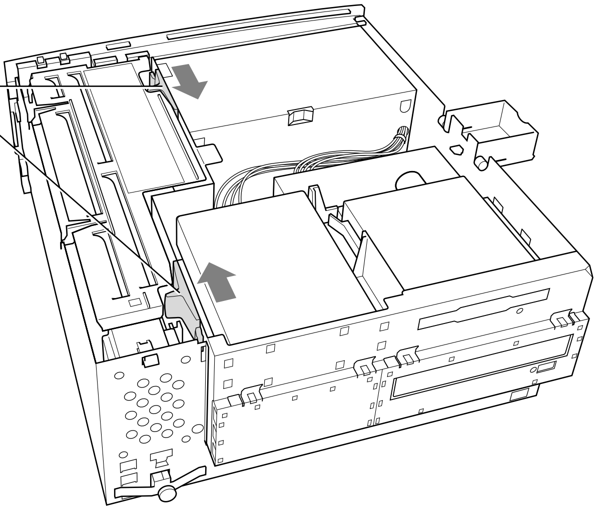

Flip the support foot out until it snaps into its locked position.
Support foot

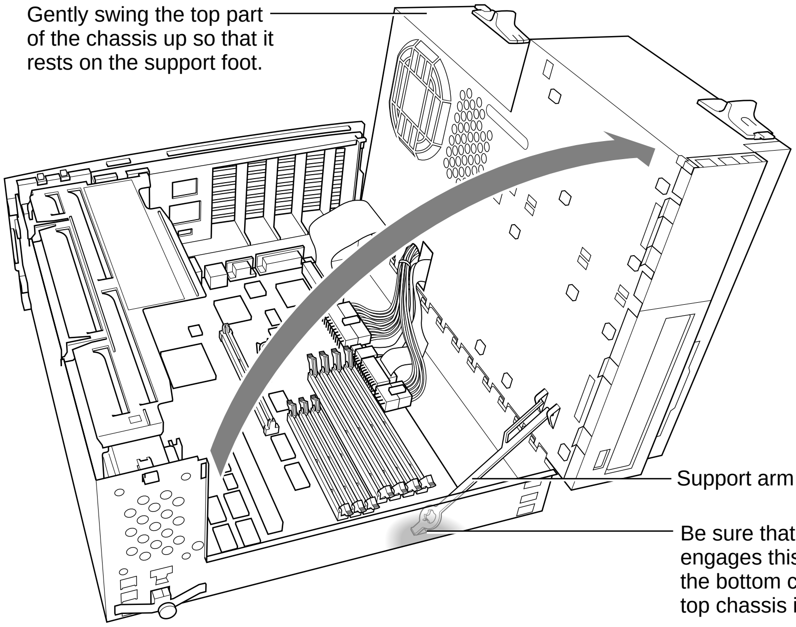
Figure 1-4 Unlocking the top chassis



Move the release switches toward the inside of the computer to unlock the top chassis.

Figure 1-5 shows the top chassis in the open position.

Figure 1-5 Top chassis of the Power Macintosh 7200 computer in the open position



Gently swing the top part of the chassis up so that it rests on the support foot.

Support arm

Be sure that the support arm engages this hole on the floor of the bottom chassis so that the top chassis is locked in place.

The following list is a summary of the hardware features of the Power Macintosh 7200 computer. Each of these features is described later in this developer note.

- **Processor.** The Power Macintosh 7200 computer has a PowerPC™ 601 microprocessor running at a clock frequency of 75 or 90 MHz.
- **Cache DIMM.** The computer has a cache DIMM slot for installing 256 KB, 512 KB, or 1 MB of optional second-level (L2) fast RAM cache.
- **RAM.** The computer has a minimum of 8 MB of main RAM.
- **RAM expansion.** The computer has four DIMM slots for RAM expansion up to 256 MB.
- **I/O expansion.** The computer has three expansion slots that conform to PCI V2.0 specifications.
- **Video monitor support.** The built-in video interface has 1 MB of VRAM, which provides up to 16 bpp at 832-by-624 resolution, or 8 bpp at 1152-by-870 resolution.
- **VRAM expansion.** Three VRAM DIMM slots allow optional VRAM expansion to 2 MB or 4 MB, which provides up to 24 bpp at 1024-by-768 resolution and up to 16 bpp at 1280-by-1024 resolution.
- **Standard I/O ports.** The computer has two 9-pin GeoPort serial ports, an ADB port, stereo sound input and output jacks, a SCSI port, and two Ethernet ports (AAUI and 10baseT).
- **SCSI.** The internal and external SCSI ports support transfer rates up to 5 MB/sec.
- **Floppy disk drive.** An internal 1.44 MB 3.5-inch floppy disk drive.
- **Hard disk.** The internal hard disk has a capacity of 500 MB to 1 GB.
- **CD-ROM.** An internal CD-ROM drive is optional.

Comparison With Other Power Macintosh Computers

This section summarizes the features of the Power Macintosh 7200 computer and compares them with the features of the Power Macintosh 6100, 7100, and 8100 computers and the Power Macintosh 7500 computer.

While the Power Macintosh 7200 computer has many of the same features as the Power Macintosh 6100, 7100, and 8100, it also has important new features. The major new features are

- a memory system using 64-bit DIMMs
- interfaces to I/O devices and expansion cards using the industry-standard PCI bus
- AAUI and 10baseT ethernet ports built-in
- new case design with easy to remove top cover and hinged chassis for access to expansion features on the main logic board

Table 1-1 compares the main features of the Power Macintosh 7200 computer with those of the Power Macintosh 6100, 7100, and 8100 computers and the Power Macintosh 7500 computer. The Power Macintosh 7200 computer shares many features with the Power Macintosh 7500 computer.

Table 1-1 Comparison with first-generation Power Macintosh computer and the Power Macintosh 7500 computer

Feature	First-generation Power Macintosh computers	Power Macintosh 7200 computer	Power Macintosh 7500 computer
Case design	Compact and tower	Compact	Compact
Processor and clock speed	PowerPC 601 66 to 110 MHz	PowerPC 601 75 or 90 MHz	PowerPC 601 at 100 MHz
Processor upgrade	None	None	Processor subsystem card
Second-level (L2) cache	Up to 256 KB	Optional, 256 KB–1 MB	Optional, 256 KB–4 MB
Type of RAM SIMMs	72-pin (32-bit data bus)	168-pin (8-byte data bus)	168-pin (8-byte data bus)
Maximum amount of RAM	72–264 MB	256 MB	1 GB
Maximum amount of VRAM	2 or 4 MB	2–4 MB	2–4 MB
Support for 21-inch monitors	None, 16 bpp, or 24 bpp	16 bpp with 2 MB VRAM, 16 bpp at 1280-by-1024 with 4 MB VRAM, 24 bpp at 1024-by-768 with 4 MB VRAM	16 bpp (2 MB VRAM), 24 bpp (4 MB VRAM)
DAV connector	Optional with AV card	None	Yes
Video input	Optional with AV card, 24 bpp	None	Built-in, 24 bpp
Video output	Optional with AV card	None	None
Sound	16-bit, 44.1 MHz, stereo input and output	Same	Same
Internal hard disk	160 MB to 1 GB	500 MB to 1 GB	500 MB to 1 GB
Additional internal drives	One 5.25-inch, one or two 3.5-inch on some models	One 5.25-inch, one 3.5-inch	One 5.25-inch, one 3.5-inch
CD-ROM drive	Built-in on some models	Optional (internal)	Optional (internal)

continued

Table 1-1 Comparison with first-generation Power Macintosh computer and the Power Macintosh 7500 computer (continued)

Feature	First-generation Power Macintosh computers	Power Macintosh 7200 computer	Power Macintosh 7500 computer
SCSI buses	1 fast internal on some models, 1 external	One bus, 5 MB per sec. transfer rates for internal and external SCSI devices	1 fast internal, 1 external
DMA for I/O devices	Yes	Yes	Yes
Network port	Ethernet (AAUI)	Ethernet (AAUI and 10baseT)	Ethernet (AAUI and 10baseT)
GeoPort	2 serial ports	Same	Same
Number and types of expansion slots	1–3 NuBus slots; DAV connector in some models	3 PCI slots	3 PCI slots, 1 DAV slot
QuickDraw acceleration	None	Built-in hardware for acceleration of some QuickDraw functions	None

Configurations and Options

Table 1-2 shows the standard configurations for the Power Macintosh 7200 computer at the time this developer note was written. All of the listed configurations include built-in video support.

Table 1-2 Configurations

Model	Processor speed	Installed DRAM	Size of hard disk	Size of L2 cache	Internal CD-ROM installed
Power Macintosh 7200/75 8/500 CD	75 Mhz	8 MB	500 MB	Optional	Yes
Power Macintosh 7200/90 8/500 CD	90 Mhz	8 MB	500 MB	Optional	Yes
Power Macintosh 7200/90 16/500 CD	90 Mhz	16 MB	500 MB	Optional	Yes
Power Macintosh 7200/90 16/1000 CD	90 Mhz	16 MB	1 GB	Optional	Yes
Power Macintosh 7200/90 8/500	90 Mhz	8 MB	500 MB	Optional	No

Compatibility

The Power Macintosh 7200 computer is a new entry-level model in the desktop Power Macintosh computer family. It incorporates many changes from the Power Macintosh 6100, 7100, and 8100 computers. This section highlights key areas you should investigate to ensure that your hardware and software works properly with the Power Macintosh 7200 computer. These topics are discussed in more detail in later parts of the developer note.

Machine Identification

Applications can find out which computer they are running on by using the Gestalt Manager routines; see *Inside Macintosh: Overview.* The `gestaltMachineType` value for the Power Macintosh 7200 computer is `$108`.

Open Transport

To preserve compatibility with older applications, new AppleTalk and TCP/IP protocol stacks accept AppleTalk and TCP/IP networking calls and reroute them to the Open Transport software. See the section “Open Transport” on page 48.

NuBus Expansion Cards

The Power Macintosh 7200 computer uses PCI cards for expansion and does not have any NuBus™ expansion slots. For users who must be able to run NuBus expansion cards, various third-party external NuBus expansion chassis are available. The expansion chassis has a cable that connects it to one of the computer’s PCI slots.

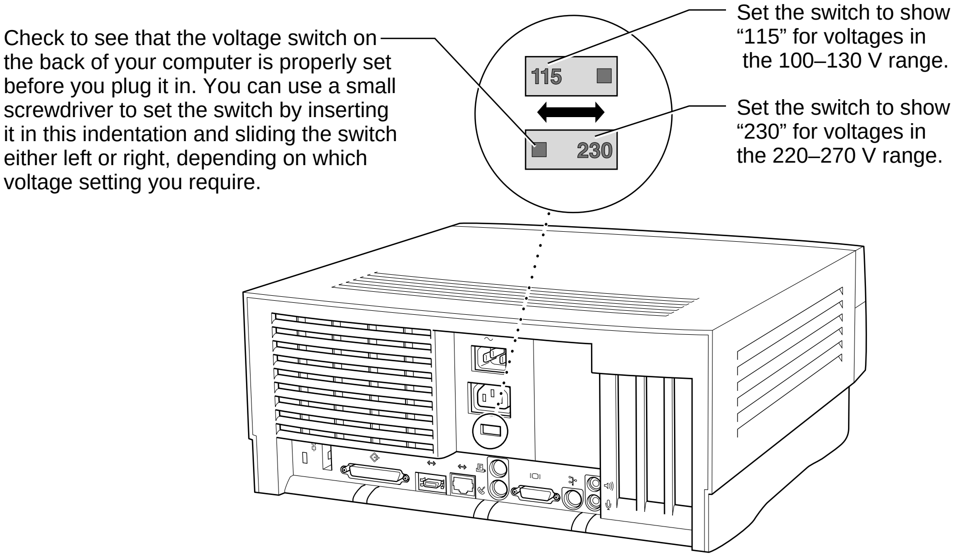
Video Cards

The Power Macintosh 7200 computer is compatible with video cards that comply with the *PCI Local Bus Specification*, Revision 2.0. The video card must include startup ROM that conforms to the IEEE Standard 1275. For information about obtaining these standards, see page xi.

Power Supply

The power supply in the Power Macintosh 7200 computer is not self configuring for different input voltages. The voltage switch on the Power Macintosh 7200 computer is delivered preconfigured for the input voltage of the region in which the unit is originally purchased. If the computer is moved to another location where the input voltage is different, a voltage switch must be adjusted to accomodate the voltage change. The switch has two positions that support voltage ranges of 100 to 130 V or 220 to 270 V. Figure 1-6 shows the location and appearance of the switch on the back of the computer.

Figure 1-6 Voltage switch location


Check to see that the voltage switch on the back of your computer is properly set before you plug it in. You can use a small screwdriver to set the switch by inserting it in this indentation and sliding the switch either left or right, depending on which voltage setting you require.
115
230
Set the switch to show "115" for voltages in the 100–130 V range.
Set the switch to show "230" for voltages in the 220–270 V range.

CHAPTER 2

Architecture

Architecture

This chapter describes the architecture of the Power Macintosh 7200 computer. It describes the major components on the main logic board: the microprocessor, the custom ICs, and the main memory.

Main ICs and Subsystems

The architecture of the Power Macintosh 7200 computer is based on two buses: the processor bus and the PCI bus. The processor bus connects the microprocessor and the memory; the PCI bus connect the expansion slots and the I/O devices.

Block Diagram

Figure 2-1 on page 13 is a simplified block diagram of the Power Macintosh 7200 computer showing the processor bus, PCI bus, and the major ICs on the main logic board.

PowerPC 601 Processor

The principal features of the PowerPC 601 microprocessor include

- full RISC processing architecture
- parallel processing units: one integer unit and one floating-point unit
- a branch manager that can usually implement branches by reloading the incoming instruction queue without using any processing time
- an internal memory management unit (MMU)
- a single built-in 32 KB cache for data and instructions

For complete technical details, see *PowerPC 601 RISC Microprocessor User's Manual.*

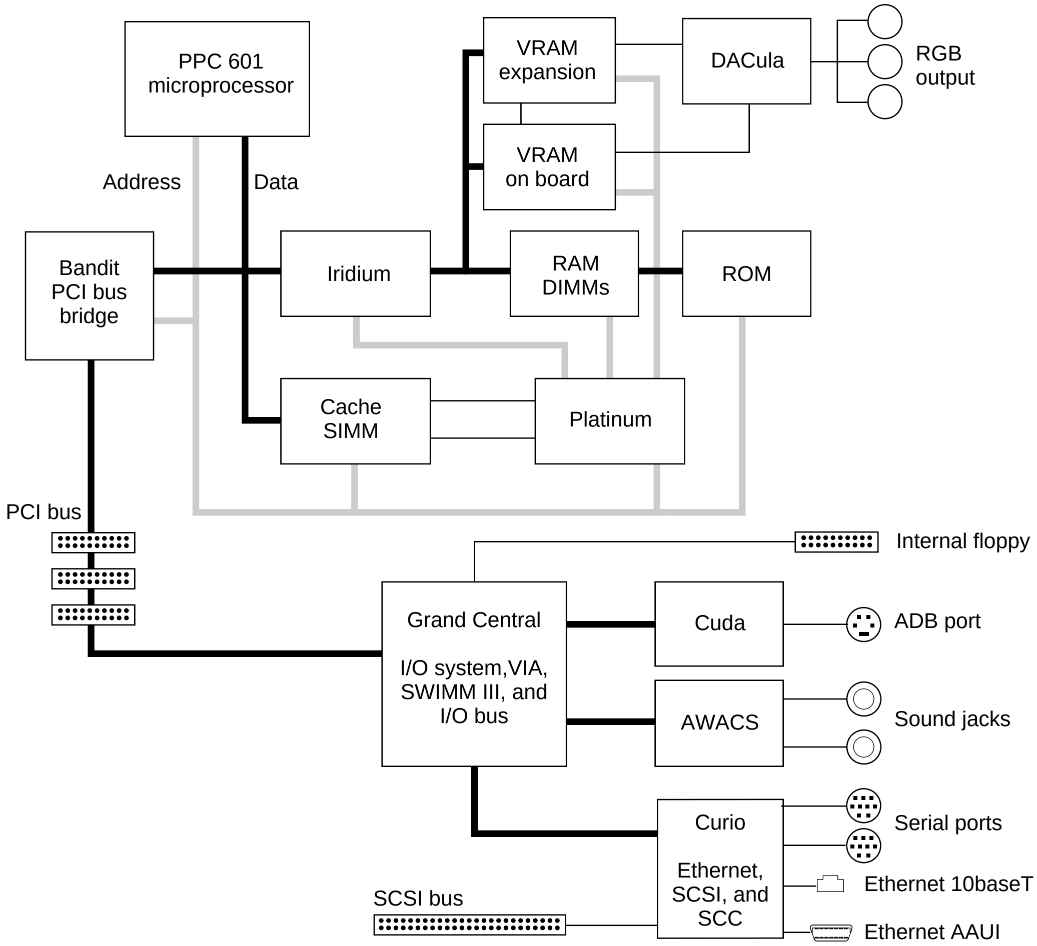
Note

The Power Macintosh 7200 computer does not have a processor subsystem card or associated processor card slot like that found on the Power Macintosh 7500 computer. The PowerPC 601 processor is soldered directly to the Power Macintosh 7200 main logic board. ◆

Read-Only Memory

The Power Macintosh 7200 computer uses a burst-mode masked ROM. The ROM is soldered to the main logic board. The burst-mode ROM contains 4 MB of ROM with 120 ns access time for the initial access and 60 ns access for the remaining three accesses of the burst.

Figure 2-1 Block diagram



Random-Access Memory

All RAM in the Power Macintosh 7200 computer is provided by DRAM devices on 8-byte DIMMs (Dual Inline Memory Modules). The computer has four RAM expansion slots that can provide 256 MB of memory if fully populated with 64 MB DIMMs.

The computer comes with 8 or 16 MB of RAM installed in the form one or of two DIMMs. The user can add more memory by installing one or more additional DIMMs. For more information, see “RAM DIMMs” beginning on page 32.

The Platinum custom IC contains bank base registers that are used to make RAM addresses contiguous, starting at address $0000 0000. See “Bank Base Registers” on page 14.

Second-Level Cache

The Power Macintosh 7200 computer has a slot for installing optional second-level (L2) cache, which provides 256 KB, 512 KB, or 1 MB of high-speed memory. The L2 cache is a organized as a write-through cache; it is direct mapped (single set) with allocate on read.

The cache's tag store is implemented with standard static RAM (SRAM) devices. The cache's data store is implemented with synchronous burst SRAM devices.

See "Second-Level Cache SIMM" beginning on page 40 for more information about the L2 cache connector.

Platinum Memory and Video Controller IC

The Platinum IC controls the memory and cache subsystem, which includes the system bus, the main memory, the ROM, the VRAM video frame buffer, and the L2 cache. The Platinum IC supports memory subsystem sizes up to 1 GB.

Data Interleaving

The Platinum memory controller IC, unlike the Hammerhead memory controller IC in the Power Macintosh 7500, Power Macintosh 8500, and Power Macintosh 9500 computers, supports linear memory organization only. It does not support data interleaving when like pairs of DIMM modules are installed in the RAM expansion slots.

Bank Base Registers

The Platinum IC contains a bank base register for each bank of main RAM. Each bank base register has space for a base address and control bits. The control bits set the address multiplexing mode.

The bank base address is used to make memory banks contiguous. The system software calculates the base addresses based on the amount of memory in each of the DIMMs. The base address for each bank is based on the sum of the sizes of all the lower-numbered banks.

Iridium IC

The Iridium custom IC works with the Platinum memory controller IC to optimize data throughput during ROM, DRAM, and VRAM data transactions. It also works with the Platinum IC to accelerate some QuickDraw display operations. For additional information about how software can take advantage of the QuickDraw acceleration hardware in the Power Macintosh 7200 computer, see "QuickDraw Acceleration API" on page 56.

Bandit Bus Bridge IC

The Bandit custom IC provides the interface between the processor and the PCI bus. It provides buffering and address translation from one bus to the other.

A separate logic device (gate array) provides the priorities for bus arbitration as follows:

1. Grand Central IC (I/O device controller), the highest priority
2. PCI slots and Bandit IC (bus master), in round-robin sequence (that is, each in turn, with no starting or ending place)

Bus Clock Rates

The two types of buses operate asynchronously: the PCI bus at a clock rate of 33 MHz and the processor bus at 37.5 or 45 MHz (one-half the processor speed). The Bandit IC supports burst transfers, in both directions, at up to 32 bytes in length, which is the size of a cache block.

Big-Endian and Little-Endian Bus Addressing

The Power Macintosh 7200 computer supports both big-endian and little-endian conventions for addressing bytes in a word. Byte order for addressing on the processor bus is big-endian and byte order on the PCI bus is little-endian. The Bandit IC performs the appropriate byte swapping and address translations between the two addressing conventions. For more information about the translations between big-endian and little-endian byte order, see Part One, “The PCI Bus,” in *Designing PCI Cards and Drivers for Power Macintosh Computers.*

Grand Central I/O System IC

The Grand Central custom IC provides an interface between the standard Macintosh I/O devices and the PCI bus. The Grand Central IC performs the following functions:

- support for the Cuda IC (the VIA registers)
- central system interrupt collection
- support for descriptor-based DMA for I/O devices
- floppy disk interface (SWIM III)

The Grand Central IC contains a DMA controller. It provides DBDMA support for all I/O transfers, including transfers through its internal I/O controllers as well as transfers through the Curio IC for other I/O devices.

The SWIM III floppy disk drive controller in the Grand Central IC is an extension of the SWIM II design used in earlier Macintosh models. The SWIM III controller supports DMA data transfers and does not require disabling of interrupts during floppy disk accesses.

The Grand Central IC provides bus interfaces for the following I/O devices:

- Cuda microcontroller IC
- AWACS sound input and output IC
- Curio multipurpose I/O IC

The Grand Central IC also provides a 16-bit bus to other devices, including the nonvolatile parameter RAM.

The Grand Central IC is connected to the PCI bus and uses the 33 MHz PCI bus clock.

Curio I/O Controller IC

The Curio IC is a multipurpose custom IC that contains a Media Access Controller for Ethernet (MACE), a SCSI controller, and a Serial Communications Controller (SCC).

The SCC section of the Curio includes 8-byte FIFO buffers for both transmit and receive data streams.

The functions of the Curio IC are described in the parts of Chapter 3 that deal with the external I/O ports it supports: “Serial Ports” on page 20, “Ethernet Port” on page 22, and “SCSI Port” on page 23.

Note
The Curio IC is also used in the Power Macintosh 6100, 7100, and 8100 computers. ◆

Cuda Microcontroller IC

The Cuda IC is a custom version of the Motorola MC68HC05 microcontroller. It provides several system functions, including

- program control of the power supply (soft power)
- management of system resets
- control of the Apple Desktop Bus (ADB)
- management of the real-time clock

AWACS Sound IC

The audio waveform amplifier and converter (AWACS) is a custom IC that combines a waveform amplifier with a 16-bit digital sound encoder and decoder (codec). It conforms to the IT&T *ASCO 2300 Audio-Stereo Codec Specification* and furnishes high-quality sound input and output. For information about the operation of the AWACS IC, see the *Power Macintosh DAV Interface for PCI Expansion Cards,* available on the developer CD-ROM.

The sound system in the Power Macintosh 7200 computer is similar to that in the 6100, 7100, and 8100 Power Macintosh computers. For more information about the sound features of the Power Macintosh 7200 computer, see “Sound Input Jack” on page 25 and “Sound Output Jack” on page 25.

VRAM and VRAM Expansion

The Power Macintosh 7200 computer has 1 MB of VRAM soldered to the main logic board and three VRAM expansion slots that expand the total VRAM size to either 2 MB or 4 MB. The slots are 112-pin connectors that accept 32-bit VRAM DIMMs (Dual Inline Memory Modules) with access times of 70 ns or faster. The larger amount of VRAM is needed to support large pixel depths on large monitors.

All VRAM is addressed as linear memory and appears as a single memory space. VRAM is organized in one half bank (32-bits wide), one, or two banks of 256K by 64 bits. Each full bank contains 2 MB of VRAM.

Note

The 32-bit VRAM DIMMs conform to the proposed standard currently under consideration by the JC-11 committee of the JEDEC. ◆

Table 2-1 shows the pixel depths supported by 1, 2, and 4 MB of VRAM on monitors with various display sizes.

Table 2-1 Display sizes and pixel depths

Monitor display size	Pixel depths		
	With 1 MB VRAM	With 2 MB VRAM	With 4 MB VRAM
512 by 384	8, 16, 32	8, 16, 32	8, 16, 32
640 by 480	8, 16	8, 16, 32	8, 16, 32
768 by 576	8, 16	8, 16, 32	8, 16, 32
800 by 600	8, 16	8, 16, 32	8, 16, 32
832 by 624	8, 16	8, 16, 32	8, 16, 32
1024 by 768	8	8, 16	8, 16, 32
1152 by 870	8	8, 16	8, 16
1280 by 960	—	8	8, 16
1280 by 1024	—	8	8. 16

CHAPTER 3

I/O Features

I/O Features

This chapter describes the I/O features of the Power Macintosh 7200 computer—both the built-in I/O devices and the interfaces for external I/O devices.

I/O Ports

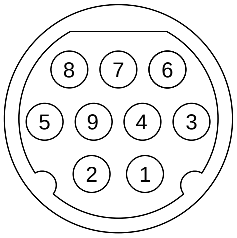
The Power Macintosh 7200 computer has the standard I/O ports found on other Macintosh models:

- two serial ports
- ADB port
- Ethernet port (AAUI and 10baseT)
- SCSI port
- sound input jack
- sound output jack
- video monitor connector

Serial Ports

The Power Macintosh 7200 computer has two serial ports on the back panel. Both ports use 9-pin circular mini-DIN sockets, as shown in Figure 3-1. The serial port sockets accept either 8-pin or 9-pin plugs.

Figure 3-1 Serial port connector



Either port can be independently programmed for asynchronous or synchronous communication formats, including AppleTalk and the full range of Apple GeoPort protocols. With external modules connected to the serial ports, the computer can communicate with a variety of ISDN and other telephone transmission facilities.

Table 3-1 gives the pin assignments for both serial port connectors.

Table 3-1 Pin assignments on the serial port connector

Pin	Name	Function
1	HSKo	Handshake output
2	HSKi	Handshake input or external clock (up to 920 Kbit/sec.)
3	TxD–	Transmit data negative
4	Gnd	Ground
5	RxD–	Receive data negative
6	TxD+	Transmit data positive
7	GPi	General-purpose input (wake up CPU or perform DMA handshake)
8	RxD+	Receive data positive
9	+5V	Power to external device (100 mA maximum per device)

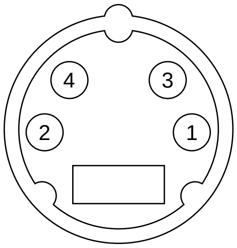
Pin 9 on each serial connector provides +5 V power from the ADB power supply. An external device should draw no more than 100 mA from that pin. The total current available for all devices connected to the +5 V supply for the ADB and the serial ports is 500 mA. Excessive current drain causes a circuit breaker to interrupt the +5 V supply; the breaker automatically resets when the load returns to normal.

Both serial ports include the GPi (general-purpose input) signal on pin 7. The GPi signal for each port connects to the corresponding data carrier detect input on the SCC portion of the Curio custom IC, which is described on page 16. For more information about the serial ports, see *Guide to the Macintosh Family Hardware,* second edition.

Apple Desktop Bus (ADB) Port

The Apple Desktop Bus (ADB) is an asynchronous communication bus used for relatively slow user-input devices such as the keyboard and the mouse. The Power Macintosh 7200 computer has a single ADB port on the back panel. The connector is a 4-pin mini-DIN socket, as shown in Figure 3-2.

Figure 3-2 ADB connector



The ADB is a single-master, multiple-slave serial communication bus that uses an asynchronous protocol. The custom ADB microcontroller (the Cuda IC) drives the bus and reads the status from the selected external device. Table 3-2 lists the ADB connector pin assignments.

Table 3-2 Pin assignments on the ADB connector

Pin	Name	Description
1	ADB	Bidirectional data bus
2	PSW	Power-on signal (generates reset and interrupt key combinations)
3	+5V	Power from the computer to external devices
4	GND	Ground from the computer

Note
The total current available for all devices connected to the +5V pins on the ADB and the modem port is 500 mA. Each device should use no more than 100 mA. ◆

For more information about the ADB, see *Guide to the Macintosh Family Hardware,* second edition. The software characteristics of the ADB are described in *Inside Macintosh: Devices.*

Ethernet Port

The Power Macintosh 7200 computer has a built-in Ethernet port with two connectors, one for a 10baseT cables and the other for the Apple Ethernet adapter for thicknet or thinnet cables. Either connector can be used for Ethernet communication, but when both connectors are occupied, only the 10baseT port is active. The electrical and mechanical characteristics of the Ethernet port are the same as on other Macintosh computers.

The Apple AUI Ethernet connector pin assignments are shown in Table 3-3.

Table 3-3 Pin assignments on the 14-pin Apple AUI Ethernet connector

Pin	Signal name	Description
1	+5 V	Fused +5 volts to power external drop box
2	DI+	Data in positive
3	DI–	Data in negative
4	GND	Ground
5	CI+	Clock in positive
6	CI–	Clock in negative
7	+5 V	Fused +5 volts to power external drop box

continued

Table 3-3 Pin assignments on the 14-pin Apple AUI Ethernet connector (continued)

Pin	Signal name	Description
8	+5 V	Fused +5 volts to power external drop box
9	DO+	Data out positive
10	DO–	Data out negative
11	GND	Ground
12	n.c.	No connection
13	n.c.	No connection
14	+5 V	Fused +5 volts to power external drop box

The 10baseT connector pin assignments are shown in Table 3-4.

Table 3-4 Pin assignments on the 8-pin 10baseT Ethernet connector

Pin	Signal name	Description
1	DO+	Data out positive
2	DO-	Data out negative
3	DI+	Data in positive
4	n.c.	No connection
5	n.c.	No connection
6	DI-	Data in negative
7	n.c.	No connection
8	n.c.	No connection

SCSI Port

The Power Macintosh 7200 computer uses a SCSI bus for external SCSI devices and for the internal CD-ROM drive. The external SCSI connector is a 25-pin D-type connector; the internal CD-ROM drive uses a 50-pin connector.

Table 3-5 shows the pin assignments on the internal and external SCSI connectors.

Table 3-5 Pin assignments on the SCSI connectors

Pin number (internal 50-pin)	Pin number (external 25-pin)	Signal name	Signal description
2	8	/DB0	Bit 0 of SCSI data bus
4	21	/DB1	Bit 1 of SCSI data bus
6	22	/DB2	Bit 2 of SCSI data bus
8	10	/DB3	Bit 3 of SCSI data bus
10	23	/DB4	Bit 4 of SCSI data bus
12	11	/DB5	Bit 5 of SCSI data bus
14	12	/DB6	Bit 6 of SCSI data bus
16	13	/DB7	Bit 7 of SCSI data bus
18	20	/DBP	Parity bit of SCSI data bus
25	–	n.c.	Not connected
26	25	TPWR	+5 V terminator power
32	17	/ATN	Attention
36	6	/BSY	Bus busy
38	5	/ACK	Handshake acknowledge
40	4	/RST	Bus reset
42	2	/MSG	Message phase
44	19	/SEL	Select
46	15	/C/D	Control or data
48	1	/REQ	Handshake request
50	3	/I/O	Input or output
20, 22, 24, 28, 30, 34, and all odd pins except pin 25	7, 9, 14, 16, 18, and 24	GND	Ground

The external SCSI port has automatic termination like that on the Power Macintosh 6100, 7100, and 8100 computers. When no external SCSI device is connected, the automatic termination is active. When one or more external SCSI devices are connected, the automatic termination is removed. As usual, the external SCSI device at the end of the SCSI bus requires termination.

The internal end of the SCSI bus is terminated by the internal hard disk drive. The internal CD-ROM drive does not include a terminator.

Sound Input Jack

The Power Macintosh 7200 computer has a stereo sound input jack for connecting an external microphone or a line-level source. The computer provides sound digitization and recording with 16-bit samples at sample rates of up to 44.1 KHz and supports Apple Computer's speech synthesis and recognition software.

The sound input jack is a stereo 1/8-inch phone jack with an additional contact to supply power to an Apple microphone. The sound input jack accepts either the Apple PlainTalk line-level microphone or a pair of line-level signals by way of 1/8-inch phone plug.

The sound input jack has the following electrical characteristics:

- input impedance: 8000 Ω
- maximum level: 2 V rms
- maximum gain: 22.5 dB
- signal-to-noise ratio: 82 dB

Note
The Apple PlainTalk microphone requires power from the main computer, which it obtains by way of an extra-long, 4-conductor plug that makes contact with a 5-volt pin inside the sound input jack. ◆

Sound Output Jack

The Power Macintosh 7200 computer has a stereo sound output jack for connecting externally powered speakers or other line-level devices. Inserting a plug into the jack disconnects the internal speaker.

The sound output jack is a standard 1/8-inch phone jack; it has the following electrical characteristics:

- output impedance: 37 Ω
- maximum level: 0.9 V rms
- maximum attenuation: 22.5 dB
- frequency response: 20 Hz to 20 kHz, plus or minus 2 dB
- harmonic distortion plus noise: less than 0.05 percent at 1 V rms input
- signal to noise ratio: 85 dB; no audible discrete tones

Note
Crosstalk degrades from -80 dB to -32 dB
when 32 Ω headphones are connected. ◆

Video Monitor Connector

The Power Macintosh 7200 computer requires an external video monitor for the graphics display. The video output from the Power Macintosh 7200 computer consists of standard R, G, and B signals at 75 Ω impedance. The video monitor connector is a DB-15 connector, as shown in Figure 3-3.

Figure 3-3 External video connector

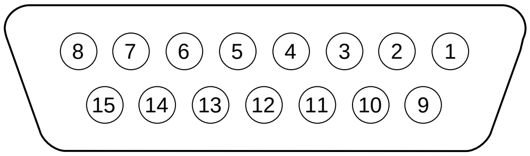


Table 3-6 shows the pin assignments on the DB-15 video connector.

Table 3-6 Pin assignments on the video connector

Pin number	Signal name	Description
1	RED.GND	Red video ground
2	RED.VID	Red video signal
3	/CSYNC	Composite synchronization signal
4	SENSE0	Monitor sense signal 0
5	GRN.VID	Green video signal
6	GRN.GND	Green video ground
7	SENSE1	Monitor sense signal 1
8	n.c.	Not connected
9	BLU.VID	Blue video signal
10	SENSE2	Monitor sense signal 2
11	GND	CSYNC and VSYNC ground
12	/VSYNC	Vertical synchronization signal
13	BLU.GND	Blue video ground
14	HSYNC.GND	HSYNC ground
15	/HSYNC	Horizontal synchronization signal
Shell	SGND	Shield ground

The video circuitry uses the sense lines to determine the type of monitor that is connected. Table 3-7 shows the sense line encodings supported by the Power Macintosh 7200 computer.

Table 3-7 Sense line codes for various monitors

	Sense lines			Extended sense codes		
Monitor type	(2)	(1)	(0)	(1, 2)	(0, 2)	(0, 1)
21-inch color	0	0	0	–	–	–
Portrait, grayscale	0	0	1	–	–	–
12-inch color	0	1	0	–	–	–
Two-page grayscale	0	1	1	–	–	–
Portrait, color	1	0	1	–	–	–
12-inch monochrome, 14-inch color	1	1	0	–	–	–
15-inch multiple scan	1	1	0	00	00	11
17-inch multiple scan	1	1	0	00	10	11
20-inch multiple scan	1	1	0	10	00	11
VESA	1	1	1	01	01	11
16-inch color	1	1	1	10	11	01
19-inch color	1	1	1	11	10	10
NTSC	1	0	0	–	–	–
PAL	1	1	1	11	00	00

The video circuitry also supports a Monitors control panel that allows the user to switch between monitor timings during operation. The switching happens immediately; the user does not need to restart the computer.

Disk Drives

The Power Macintosh 7200 computer has one internal floppy disk drive and one internal hard disk drive. Some configurations also have an internal CD-ROM drive.

Floppy Disk Drive

The Power Macintosh 7200 computer has one internal high-density floppy disk drive (Apple SuperDrive). The drive is connected to a 20-pin connector. Table 3-8 shows the pin assignments on the floppy disk connector.

Table 3-8 Pin assignments on the floppy disk connector

Pin	Name	Description
1	GND	Ground
2	PH0	Phase 0: state control line
3	GND	Ground
4	PH1	Phase 1: state control line
5	GND	Ground
6	PH2	Phase 2: state control line
7	GND	Ground
8	PH3	Phase 3: register write strobe
9	n.c.	Not connected
10	/WRREQ	Write data request
11	+5V	+5 volts
12	SEL	Head select
13	+12V	+12 volts
14	/ENBL	Drive enable
15	+12V	+12 volts
16	RD	Read data
17	+12V	+12 volts
18	WR	Write data
19	+12V	+12 volts
20	n.c.	Not connected

Internal Hard Disk Drive

The Power Macintosh 7200 computer has one internal hard disk drive. The drive capacity is either 500 MB or 1 GB.

The hard disk drive is connected to the end of the internal SCSI bus. For pin assignments on the internal SCSI hard disk connector, see Table 3-5 on page 24.

The internal end of the SCSI bus is terminated by the internal hard drive. The internal CD-ROM drive does not include a terminator.

CD-ROM Drive

The Power Macintosh 7200 computer has an optional quad-speed CD-ROM drive. The CD-ROM drive supports the worldwide standards and specifications for CD-ROM and CD-digital audio discs described in the Sony/Philips Yellow Book and Red Book. The drive can read CD-ROM, CD-ROM XA, CD-I, and PhotoCD discs as well as play standard audio discs.

The CD-ROM drive has a sliding tray to hold the disc. The drive features a 4X-speed mechanism that supports sustained data transfer rates of 600 KB per second and a data buffer that further enhances performance.

CHAPTER 4

Expansion Features

Expansion Features

This chapter describes the expansion features of the Power Macintosh 7200 computer: the RAM expansion DIMMs, the L2 cache SIMM, and the PCI expansion slots.

RAM DIMMs

The Power Macintosh 7200 computer has four RAM expansion slots. The RAM expansion slots accept a new type of memory module: the 8-byte DIMM (Dual Inline Memory Module). As its name implies, the 8-byte DIMM has a 64-bit-wide data bus.

The 8-byte DIMM is a new industry standard. Its mechanical design is defined by the MO-161 specification published by the JEDEC JC-11 committee; its electrical characteristics are defined by the JEDEC Standard No. 21-C. The 8-byte DIMM connector used in the Power Macintosh 7200 computer is Burndy Corporation's part number ELF168E5GC-3Z50 or equivalent.

IMPORTANT

The Power Macintosh 7200 computer does not have any main memory soldered to the logic board. At least one RAM DIMM must be present for the computer to operate. ▲

The smallest bank size supported by the Platinum IC is 4 MB and the largest is 128 MB; the largest DIMM supported is a two-bank DIMM holding 256 MB. Table 4-1 shows the DIMM configurations and sizes for a range of DRAM device sizes that are supported on the Power Macintosh 7200 computer.

Table 4-1 Memory sizes and configurations

Device size	DIMM configuration	DIMM size	Maximum memory with 4 DIMMs installed
4 MB	1 Mbit by 64 (1 bank)	8 MB	32 MB
4 MB	2 Mbits by 64 (2 banks)	16 MB	64 MB
16 MB	1 Mbit by 64 (1 bank)	8 MB	32 MB
16 MB	2 Mbits by 64 (2 banks)	16 MB	64 MB
16 MB	4 Mbits by 64 (1 bank)	32 MB	128 MB
16 MB	8 Mbits by 64 (2 banks)	64 MB	256 MB

The 8-byte DIMMs can be installed one or more at a time. The Power Macintosh 7200 computer supports only linear memory organization, therefore no performance gains are seen when two DIMMs of the same size are installed. Any size SIMM can be installed in any DIMM slot, and the combined memory of all of the DIMMs installed is configured as a contiguious memory space.

RAM DIMM Connectors

Table 4-2 gives the pin assignments for the RAM DIMM connectors.

Table 4-2 Pin assignments on the RAM DIMM connectors

Pin number	Signal name	Pin number	Signal name
1	VSS	28	/CAS(0)
2	DQ(0)	29	/CAS(2)
3	DQ(1)	30	/RAS(0)
4	DQ(2)	31	/OE(0)
5	DQ(3)	32	VSS
6	VCC	33	A(0)
7	DQ(4)	34	A(2)
8	DQ(5)	35	A(4)
9	DQ(6)	36	A(6)
10	DQ(7)	37	A(8)
11	Reserved	38	A(10)
12	VSS	39	A(12)
13	DQ(8)	40	VCC
14	DQ(9)	41	Reserved
15	DQ(10)	42	Reserved
16	DQ(11)	43	VSS
17	DQ(12)	44	/OE(2)
18	VCC	45	/RAS(2)
19	DQ(13)	46	/CAS(4)
20	DQ(14)	47	/CAS(6)
21	DQ(15)	48	/WE(2)
22	Reserved	49	VCC
23	VSS	50	Reserved
24	Reserved	51	Reserved
25	Reserved	52	DQ(16)
26	VCC	53	DQ(17)
27	/WE(0)	54	VSS

continued

Table 4-2 Pin assignments on the RAM DIMM connectors (continued)

Pin number	Signal name	Pin number	Signal name
55	DQ(18)	86	DQ(32)
56	DQ(19)	87	DQ(33)
57	DQ(20)	88	DQ(34)
58	DQ(21)	89	DQ(35)
59	VCC	90	VCC
60	DQ(22)	91	DQ(36)
61	Reserved	92	DQ(37)
62	Reserved	93	DQ(38)
63	Reserved	94	DQ(39)
64	Reserved	95	Reserved
65	DQ(23)	96	VSS
66	Reserved	97	DQ(40)
67	DQ(24)	98	DQ(41)
68	VSS	99	DQ(42)
69	DQ(25)	100	DQ(43)
70	DQ(26)	101	DQ(44)
71	DQ(27)	102	VCC
72	DQ(28)	103	DQ(45)
73	VCC	104	DQ(46)
74	DQ(29)	105	DQ(47)
75	DQ(30)	106	Reserved
76	DQ(31)	107	VSS
77	Reserved	108	Reserved
78	VSS	109	Reserved
79	PD(1)	110	VCC
80	PD(3)	111	Reserved
81	PD(5)	112	/CAS(1)
82	PD(7)	113	/CAS(3)
83	ID(0)	114	Reserved
84	VCC	115	Reserved
85	VSS	116	VSS

continued

Table 4-2 Pin assignments on the RAM DIMM connectors (continued)

Pin number	Signal name	Pin number	Signal name
117	A(1)	143	VCC
118	A(3)	144	DQ(54)
119	A(5)	145	Reserved
120	A(7)	146	Reserved
121	A(9)	147	Reserved
122	A(11)	148	Reserved
123	A(13)	149	DQ(55)
124	VCC	150	Reserved
125	Reserved	151	DQ(56)
126	B(0)	152	VSS
127	VSS	153	DQ(57)
128	Reserved	154	DQ(58)
129	Reserved	155	DQ(59)
130	/CAS(5)	156	DQ(60)
131	/CAS(7)	157	VCC
132	/PDE	158	DQ(61)
133	VCC	159	DQ(62)
134	Reserved	160	DQ(63)
135	Reserved	161	Reserved
136	DQ(48)	162	VSS
137	DQ(49)	163	PD(2)
138	VSS	164	PD(4)
139	DQ(50)	165	PD(6)
140	DQ(51)	166	PD(8)
141	DQ(52)	167	ID(1)
142	DQ(53)	168	VCC

Table 4-3 describes the signals on the RAM DIMM connector.

Table 4-3 RAM DIMM signals

Signal name	Description
A(12:0), B(0)	Address inputs
/CAS(7:0)	Column address strobe signals
DQ(63:0)	Data input and output signals
ID(1:0)	Memory module identification (not used)
/OE(0, 2)	Output enable signals (always high)
PD(8:1)	Presence detect signals (not used)
/PDE	Presence detect enable signal (not used)
/RAS(0, 2)	Row address strobe signals
Reserved	Reserved, don't use.
VCC	+5 V power
VSS	Ground
/WE(0, 2)	Read/write input signals

RAM Address Multiplexing

Depending on their internal design and size, different types of DRAM devices require different row and column address multiplexing. The memory controller in the Platinum IC supports two addressing modes, selected individually for each bank of DRAM. The system software initializes the address mode bits in the bank base registers as part of the process of determining the amount of RAM installed in the computer.

Signals A(12–0) on each RAM DIMM make up a 13-bit multiplexed address bus that can support several different types of DRAM devices. Table 4-4 shows the address multiplexing modes used with several types of DRAM devices. The devices are characterized by their bit dimensions: for example, a 256K by 4-bit device has 256K addresses and stores 4 bits at a time.

Table 4-4 Address multiplexing modes for various DRAM devices

Device size	Device type	Size of row address	Size of column address	Address mode
1 Megabit	1M by 1 bit	10	10	1
1 Megabit	256K by 4 bits	9	9	1
4 Megabits	4M by 1 bit	11	11	1
4 Megabits	1M by 4 bits	10	10	1
4 Megabits	512K by 8 bits	10	9	1
4 Megabits	256K by 16 bits	9	9	1
4 Megabits	256K by 16 bits	10	8	0
16 Megabits	16M by 1 bit	12	12	0
16 Megabits	4M by 4 bits	11	11	1
16 Megabits	4M by 4 bits	12	10	1
16 Megabits	2M by 8 bits	11	10	1
16 Megabits	2M by 8 bits	12	9	0
16 Megabits	1M by 16 bits	12	8	0
16 Megabits	1M by 16 bits	10	10	1
64 Megabits	16M by 4 bits	13	11	0
64 Megabits	16M by 4 bits	12	12	0
64 Megabits	8M by 8 bits	13	10	1
64 Megabits	8M by 8 bits	12	11	0
64 Megabits	4M by 16 bits	12	10	1
64 Megabits	4M by 16 bits	11	11	1
64 Megabits	2M by 32 bits	12	9	0
64 Megabits	2M by 32 bits	11	10	1

Table 4-5 shows how the address signals to the RAM devices are multiplexed during the row and column address phases for noninterleaved banks.

Table 4-5 Address multiplexing in noninterleaved banks

	Individual signals on the DRAM_ADDR bus												
	A[12]	A[11]	A[10]	A[9]	A[8]	A[7]	A[6]	A[5]	A[4]	A[3]	A[2]	A[1]	A[0]
Address mode = 0													
Row address	5	9	10	20	11	12	13	14	15	16	17	18	19
Column address	4	5	7	6	8	21	22	23	24	25	26	27	28
Address mode = 1													
Row address	6	7	8	10	11	12	13	14	15	16	17	18	19
Column address	4	5	7	9	20	21	22	23	24	25	26	27	28

RAM Devices

The memory controller in the Platinum IC supports 1 MB, 4 MB, 16 MB, and 64 MB DRAM devices. The access time (T_{RAS}) of the DRAM devices should be 70 ns or faster.

Note

The Power Macintosh 7200 computer supplies +5 volts at VCC on the RAM expasion slot for DRAM DIMMs. Power for DRAM devices that require 3.3 volts is not supplied on the RAM expansion slot. ◆

IMPORTANT

The number of DRAM devices in a RAM DIMM for the Power Macintosh 7200 computer is constrained by the load limits of the unbuffered signals. On each DIMM, a maximum of two devices can be connected to each data line and a maximum of eight devices can be connected to each /RAS line. ▲

RAM Refresh

The Platinum IC provides a CAS-before-RAS refresh cycle every 15.6 µs. DRAM devices must be compatible with this refresh cycle; for example, this cycle will refresh 2K-refresh parts within 32 ms.

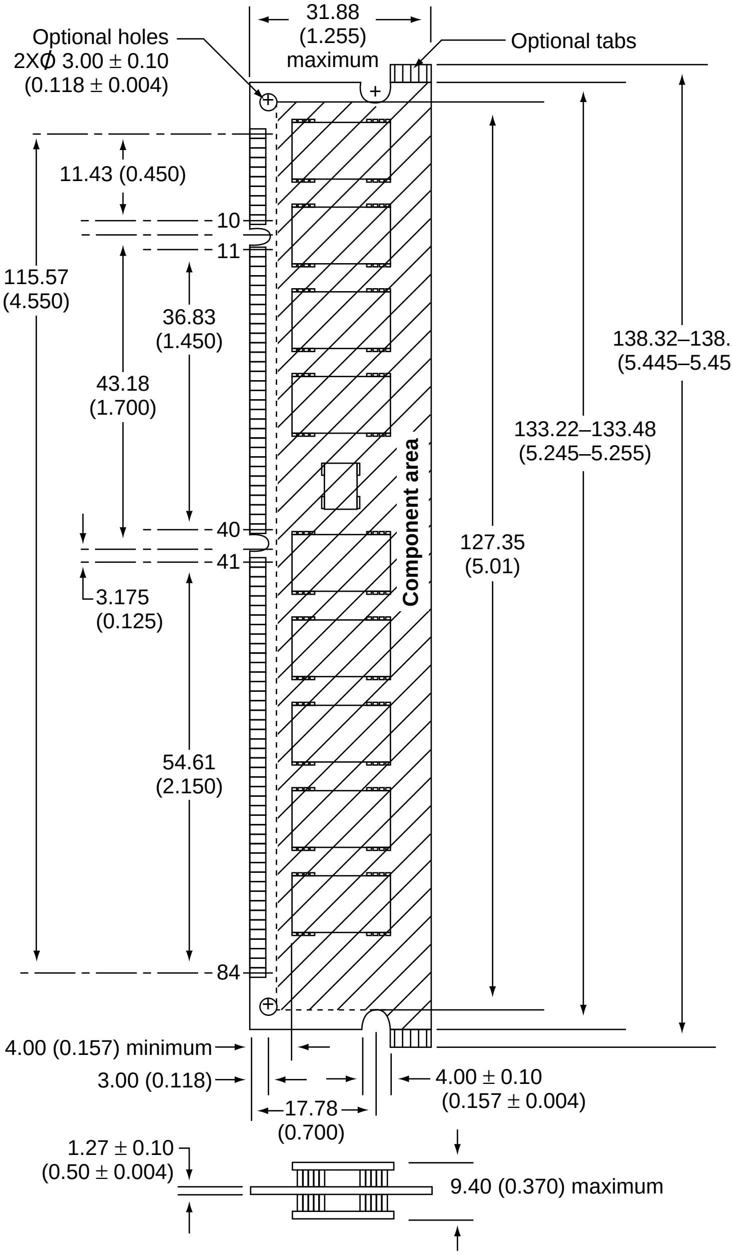
RAM DIMM Dimensions

Figure 4-1 shows the dimensions of the RAM DIMM.

IMPORTANT

The JEDEC MO-161 specification shows three possible heights for the 8-byte DIMM. The Power Macintosh 7200 computer accepts only the shorter two of the three specified sizes; the maximum height is 1.255 inches. ▲

Figure 4-1 Dimensions of the RAM DIMM



Note: dimensions are in millimeters (inches)

Second-Level Cache SIMM

The Power Macintosh 7200 computer has a slot for a second-level (L2) cache on a SIMM.

The L2 cache is organized as a write-through cache; it is direct mapped (single set) with allocate on read. The cache data store is implemented with synchronous burst static RAM devices; the cache tag store is implemented with standard static RAM devices. The maximum number of devices (or loads) is four devices for data store and two devices for tag store.

The L2 cache occupies a single SIMM. The size of the L2 cache can be from 256 KB to 1 MB.

The synchronous burst SRAM devices in the cache data store have an access time of 11 ns.

Table 4-6 shows the pin assignments on the L2 cache SIMM connector.

Table 4-6 Pin assignments for L2 cache SIMM connector

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	+5 V	18	D22	35	T14	52	A24
2	GND	19	GND	36	GND	53	A26
3	D0	20	D24	37	+5 V	54	A28
4	D2	21	D26	38	GND	55	GND
5	D4	22	D28	39	CLK	56	+5 V
6	D6	23	D30	40	GND	57	GND
7	GND	24	GND	41	/TOEN	58	D32
8	D8	25	+5 V	42	/TWEN	59	D34
9	D10	26	GND	43	/ADV	60	D36
10	D12	27	T0	44	A12	61	D38
11	D14	28	T2	45	CSIZ(1)	62	GND
12	GND	29	T4	46	A14	63	D40
13	+5 V	30	T6	47	A16	64	D42
14	GND	31	GND	48	A18	65	D44
15	D16	32	T8	49	A20	66	D46
16	D18	33	T10	50	GND	67	GND
17	D20	34	T12	51	A22	68	+5 V

continued

Table 4-6 Pin assignments for L2 cache SIMM connector (continued)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
69	GND	92	GND	115	T15	138	D35
70	D48	93	+3.3 V	116	GND	139	D35
71	D50	94	GND	117	+3.3 V	140	D37
72	D52	95	D17	118	GND	141	D39
73	D54	96	D19	119	CPRES	142	GND
74	GND	97	D21	120	A11	143	D41
75	D56	98	D23	121	/DOEN	144	D43
76	D58	99	GND	122	/DWEN	145	D45
77	D60	100	D25	123	/ADSC	146	D47
78	D62	101	D27	124	/CSIZ(2)	147	GND
79	GND	102	D29	125	GND	148	+3.3 V
80	+5 V	103	D31	126	A13	149	GND
81	+3.3 V	104	GND	127	A15	150	D49
82	GND	105	+3.3 V	128	A17	151	D51
83	D1	106	GND	129	A19	152	D53
84	D3	107	T1	130	GND	153	D55
85	D5	108	T3	131	A21	154	GND
86	D7	109	T5	132	A23	155	D57
87	GND	110	T7	133	A25	156	D59
88	D9	111	GND	134	A27	157	D61
89	D11	112	T9	135	GND	158	D63
90	D13	113	T11	136	+3.3 V	159	GND
91	D15	114	T13	137	D33	160	+3.3 V

Table 4-7 defines the signals on the L2 cache SIMM connector.

Table 4-7 Signal descriptions for L2 cache SIMM connector

Signal name	Description
+5 V	Power supply voltage of +5 volts for tag RAM (5% tolerance)
+ 3.3 V	Power supply voltage of +3.3 volts for data RAM (5% tolerance)
GND	Ground
A(13:28)	Processor address bus signals 13 through 27
D(63:0)	Processor data bus signals 0 through 63; sampled on the rising edge of the CLK signal during a write cycle
T(15:0)	Tag RAM data bits 0 through 15
/TOEN	Tag RAM output enable (active low signal)
/TWEN	Tag RAM write enable (active low signal)
/ADSC	Data RAM address select controller (active low signal); not used
/DOEN	Data RAM output enable (active low signal); sampled on the rising edge of the CLK signal
/DWEN	Data RAM write enable (active low signal); sampled on the rising edge of the CLK signal
/ADV	Data RAM advance address burst (active low); sampled on rising edge of the CLK signal
CLK	Processor bus clock used to clock data RAM
CPRES	Cache present (active low signal)
CSIZ(2:1)	Cache size

PCI Expansion Slots

The Power Macintosh 7200 computer uses the industry standard PCI bus for its I/O expansion bus. The PCI bus is a nonsplit bus with 32-bit multiplexed address and data. The PCI expansion slots in the Power Macintosh 7200 computer have a 33 MHz system clock.

The Power Macintosh 7200 computer has three PCI slots supported by one Bandit custom IC.

The Power Macintosh 7200 computer provides a total of 50 W of power for the three expansion slots. Both 5 V and 3.3 V are supplied; the total power at both voltages must not exceed the 50 W maximum.

The Power Macintosh 7200 computer accepts standard PCI cards as defined by the *PCI Local Bus Specification*, Revision 2.0. The cards are required to use the standard ISA fence and the 5 V signaling standard described in the specification.

The PCI slots support all the required PCI signals and certain optional PCI signals. The supported PCI signals are listed in Table 4-8.

Table 4-8 PCI signals

Signal name	Description
AD (31:0)	Address and data, multiplexed
C/BE (3:0)	Bus command and byte enable signals, multiplexed
PAR	Parity; used with AD and C/BE signals
/FRAME	Cycle frame; asserted to indicate a bus transaction
/TRDY	Target ready; selected device is able to complete the current phase
/IRDY	Initiator ready; master device is able to complete the current phase
/STOP	Stop; indicates the current target device is requesting the master to stop the current transaction
/DEVSEL	Device select; indicates that the driving device has decoded its address as the target of the current access
IDSEL	Initialization device select; used during configuration
/REQ	Request; indicates to the arbiter that the asserting agent requires use of the bus
/GNT	Grant; indicates to the agent that access to the bus has been granted
CLK	Clock; rising edge provides timing for all transactions
/RST	Reset; used to bring registers and signals to a known state
/INTA, /INTB, /INTC, /INTD	Interrupt request pins; wired together on each slot
/LOCK	Lock; indicates an operation that may require multiple transactions to complete
/PERR	Parity error; used to report data parity errors during PCI transactions excluding a special cycle transaction
/SERR	System error; used to report address parity errors, data parity errors during a special cycle, or any other system error that will be catastophic

The PCI slots in the Power Macintosh 7200 computer do not support the optional 64-bit bus extension signals or cache support signals.

For more information about the PCI expansion slot, refer to *Designing PCI Cards and Drivers for Power Macintosh Computers.*

CHAPTER 5

Software Features

The system software for the Power Macintosh 7200 computer is based on System 7.5 and is augmented by several new features.

IMPORTANT
Even though the software for the Power Macintosh 7200 computer incorporates significant changes from System 7.5, it is not a reference release: that is, it is not an upgrade for earlier Macintosh models. ▲

The system software includes changes of several kinds, including new features, performance enhancements, hardware support features, and power management software enhancements.

New Features

The system software for the Power Macintosh 7200 computer includes the following new features:

- large partition support
- Drive Setup
- transport-independent networking (Open Transport)
- Open Firmware startup
- enhanced power management software

Large Partition Support

The largest disk partition supported by System 7.5 is 4 GB. The new system software extends that limit to 2 terabytes.

IMPORTANT
The largest possible file is still just under 2 GB. ▲

The changes necessary to support the larger partition size affect many parts of the system software. The affected software includes system-level and application-level components.

64-Bit Volume Addresses

The current disk driver API has a 32-bit volume address limitation. This limitation has been circumvented by the addition of a new 64-bit extended volume function (`PBXGetVolInfo`) and 64-bit data types (`UnsignedWide`, `Wide`, `XVolumeParam`, and `XIOParam`).

For the definitions of the new API and data types, please see “The API Modifications” beginning on page 61.

System-Level Software

Several system components have been modified to use the 64-bit APIs to correctly calculate true volume sizes and read and write data to and from large disks. The modified system components are

- virtual memory code
- Disk Init
- File system manager
- Apple disk drivers
- HFS ROM code

Application-Level Software

Current applications do not require modification to gain access to disk space beyond the traditional 4 GB limit as long as they do not require the true size of the large partition. Applications that need to obtain the true partition size must be modified to use the new 64-bit APIs and data structures. Utilities for disk formatting, partitioning, initialization, and backup are examples of applications that need the true partition size.

The following application-level components of the system software have been modified to use the 64-bit APIs:

- Finder
- Finder extensions (AppleScript, AOCE Mailbox, and AOCE Catalogs)
- HDSC Setup (now called Drive Setup)
- Disk First Aid

In the past, the sum of the sizes of the files and folders selected in the Finder was limited to the largest value that could be stored in a 32-bit number—that is, 4 GB. By using the new 64-bit APIs and data structures, the Finder can now operate on selections whose total size exceeds that limit. Even with very large volumes, the Finder can display accurate information in Folder and Get Info windows and obtain the true volume size for calculating available space when copying.

The Finder extensions AppleScript, AOCE Mailbox, and AOCE Catalogs have been modified in the same way as the Finder because their copy-engine code is similar to that in the Finder.

The modified Drive Setup application is described on page 48.

Limitations

The software modifications that support large partition sizes do not solve all the problems associated with the use of large volumes. In particular, the modifications do not address the following:

- HFS file sizes are still limited to 2 GB or less.
- Large allocation block sizes cause inefficient storage. On a 2 GB volume, the minimum file size is 32 KB; on a 2 terabyte volume, the minimum file size would be a whopping 32 MB.
- Drives with the new large volume driver will not mount on computers running older versions of the Macintosh System software.

Drive Setup

The software for the Power Macintosh 7200 computer includes a new disk setup utility named Drive Setup. In addition to the ability to support large volumes, the Drive Setup utility has several other enhancements over the older HDSC Setup utility, including

- an improved user interface
- support for multiple partitions
- the ability to mount volumes from applications
- the ability to start up (boot) from any HFS partition
- support for removable media drives

Open Transport

Open Transport is the new communications and networking architecture that will become the standard for Macintosh networking and communications. Open Transport provides a mechanism for communications applications to operate independently from underlying networks such as AppleTalk, TCP, or IPX. Open Transport provides a code base and architecture that supports network stacks while eliminating many of the interrupt latency problems associated with AppleTalk.

Note
Open Transport runs native on the PowerPC microprocessors. ◆

Open Transport has two major aspects: the client interfaces and the environment for developing protocols and communications modules. The Open Transport client interfaces are a superset of the XTI interface from X/Open, a consortium of UNIX® vendors. XTI is a superset of TLI, a UNIX standard interface. By using the Open Transport interfaces, applications (called *clients*) can operate independently of the transport layer.

The environment for developing protocols and communications modules for Open Transport also uses industry standards. These standards are the UNIX standard Streams, and two other standards, Transport Provider Interface (TPI) and Data Link Provider Interface (DLPI).

Open Transport does not use the conventional .ENET-style drivers; instead it uses Streams-based DLPI drivers that are more appropriate for use with PCI devices. In addition to being consistent with industry standards, Streams-based DLPI drivers provide higher performance than .ENET-style drivers.

Apple Computer's Open Transport software includes new stack implementations for AppleTalk and MacTCP. Apple expects that third parties will provide implementations of DECnet™, IPX, and other network protocols.

New Features of Open Transport

The new features of Open Transport include

- a new API
- dynamic loading and shared code
- multihoming
- an optional static node number (AppleTalk)
- an optional NBP-to-catalog server (AppleTalk)
- IP multicasting
- dynamic retransmission timers

The Open Transport implementation of AppleTalk has a significant feature not found in the classic AppleTalk implementation for Macintosh computers. The Open Transport implementation supports multihoming (sometimes called *multiporting*), which makes it possible for AppleTalk to be active on more than one network port on the machine at a time.

The Open Transport implementation of TCP/IP is a replacement for MacTCP. It is designed for use under the Open Transport software interface.

Compatibility

Open Transport is compatible with existing AppleTalk networks and supports existing .ENET clients such as SoftWindows™ and DECnet.

Open Transport provides compatibility with 680x0-based computers by means of the following features:

- environment options
- 680x0-based APIs and stacks
- Open Transport APIs and stacks
- API compatibility glue
- use of parameter-block APIs with Open Transport stacks for 680x0-based applications

Open Transport provides compatibility with Power Macintosh computers by means of the following features:

- environment options
- 680x0-based APIs and stacks run in emulation mode
- Open Transport APIs and stacks run in native mode
- API compatibility glue runs in mixed mode
- 680x0-based applications can use parameter-block APIs with Open Transport stacks
- 680x0-based applications can use Open Transport APIs and stacks
- native applications can use parameter-block (PB) APIs with 680x0-based stacks
- native applications can use PB APIs with Open Transport stacks

Open Firmware Startup

The **Open Firmware startup process** in PCI-compatible Macintosh computers conforms to the IEEE Standard 1275 for boot firmware and the *PCI Bus Binding to IEEE 1275-1994* specification. These specifications are listed in "Other Publications" on page xi.

The Open Firmware startup process is driven by startup firmware (also called *boot firmware*) stored in the Macintosh ROM and in PCI card expansion ROMs. While the startup firmware is running, the Macintosh computer starts up and configures its hardware (including peripheral devices) independently of any operating system. The computer then finds an operating system in ROM or on a mass storage device, loads it into RAM, and terminates the Open Firmware startup process by giving the operating system control of the PowerPC main processor. The operating system may be the Mac OS or a different system, provided it uses the PowerPC instruction set.

The Open Firmware startup process includes these specific features:

- Startup firmware is written in the Forth language, as defined by the IEEE Standard 1275. Firmware code is stored in a tokenized representation called **FCode,** an abbreviated version of Forth in which most Forth words are replaced by single bytes or 2-byte groups. The startup firmware in the Power Macintosh ROM includes an FCode loader that installs FCode in system RAM so that drivers can run on the PowerPC main processor. Expansion card firmware can modify the Open Firmware startup process by supplying FCode that the computer's startup firmware loads and runs before launching an operating system.
- The startup firmware creates a data structure of nodes called a **device tree,** in which each PCI device is described by a **property list.** The device tree is stored in system RAM. The operating system that is ultimately installed and launched can search the device tree to determine what devices are available.

- Device drivers required during system startup (called **boot drivers**) are also stored in the expansion ROM on the PCI card. Plug-in expansion cards must contain all the driver code required during startup. The boot drivers are native drivers and are embedded in the FCode in the expansion ROM. The startup firmware in the Power Macintosh ROM installs the boot drivers in system RAM and lets them run on the PowerPC main processor.
- The startup firmware in the Power Macintosh ROM contains debugging facilities for both FCode and some kinds of operating-system code. These facilities can help expansion card designers develop the firmware for new peripheral devices compatible with Macintosh computers.

You can write PCI expansion ROM code in standard Forth words and then reduce the result to FCode by using an **FCode tokenizer,** a program that translates Forth words into FCode. The Forth vocabulary that you can use is presented in IEEE Standard 1275.

The burden on developers to provide Forth boot drivers need not be heavy. Developers can choose the level of support that they provide. The following are the three possible levels of support:

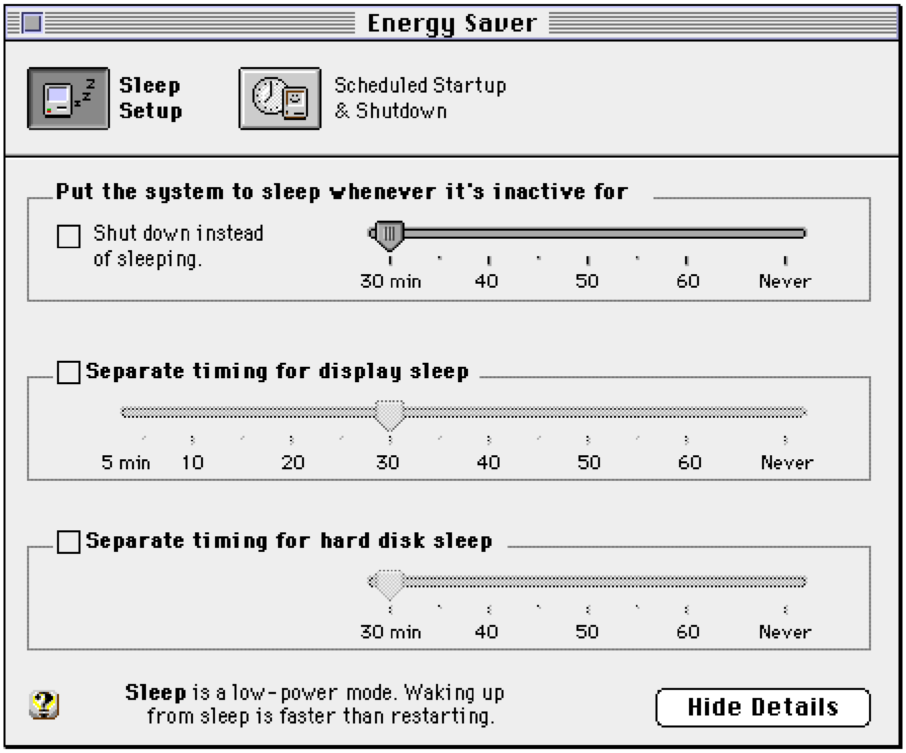
- **No driver.** The expansion ROM contains minimal FCode. The Open Firmware startup process recognizes the card and installs a node in the device tree, but no driver code is loaded and no device initialization occurs.
- **Runtime driver.** Only a small amount of Forth code is required to install an OS-dependent runtime driver in the device's property list. Sample code is provided in *Designing PCI Cards and Drivers for Power Macintosh Computers.*
- **Boot driver.** Expansion cards that need to be used at startup time must contain a boot driver with the required methods for the type of device (typically Open, Close, Read, and Write). Sample code is provided in *Designing PCI Cards and Drivers for Power Macintosh Computers.*

Enhanced Power Management Software

The system software for the Power Macintosh 7200 computer supports a superset of the Macintosh Energy Saver power-management software. The enhanced Energy Saver software fully employs the capabilities of Energy-Star compliant features in Macintosh computers and peripherals, such as hard disk drives and displays.

The Energy Saver control panel, shown expanded for Sleep preferences in Figure 5-1, provides desktop computer users access to power-saving features previously available only in portable computer power-management software. The user experience provided by the human interface to the new power management software is consitent across portable and desktop Macintosh computers.

Figure 5-1 Energy Saver application dialog box



Put the system to sleep whenever it's inactive for

□ Shut down instead of sleeping.

□ Separate timing for display sleep

□ Separate timing for hard disk sleep

Sleep is a low-power mode. Waking up from sleep is faster than restarting.

Features of the New Energy Saver Software

The Energy Saver power management software allows control of these features:

- Idle-time energy savings
  - computer sleep time
  - display sleep time
  - hard disk spin-down time
- Scheduled energy savings
  - scheduling of startup including bookmarking
  - scheduling of shutdown including document auto-save
- Wakeup preferences
  - blink power-on light when waking up
  - play user-defined sound when waking up
  - wake up when modem detects a telephone ring

The Energy Saver software provides users with a single control panel to configure all of the energy management features of their Macintosh computer system. The improved single control panel interface presents a clearer conceptual model of what energy saving features of the Macintosh computer can be managed. It also delivers a user experience that is as consistent as possible across desktop Macintosh computers and portable Macintosh computers. In addition, the new Energy Saver look and feel is appropriate for existing and future Macintosh system software models. Last but not least, Energy Saver satisfies the EPA requirements for Energy Star compliance.

To find out more about how to use all of the capabilities of the Energy Saver control panel, see the user guide and Macintosh Guide help files that accompany the shipping version of the Power Macintosh 7200 computer.

Performance Enhancements

The system software for the Power Macintosh 7200 computer includes the following performance enhancements:

- a new Dynamic Recompilation Emulator
- a Resource Manager completely in native code
- an improved math library
- new `BlockMove` extensions

Dynamic Recompilation Emulator

The Dynamic Recompilation Emulator (or **DR Emulator**) is an extension to the current interpretive emulator providing on-the-fly translation of 680x0 instructions into PowerPC instructions for increased performance.

The design of the DR Emulator mimics a hardware instruction cache and employs a variable size translation cache. Each compiled 680x0 instruction requires on average fewer than 20 PowerPC instructions. In operation, the DR Emulator depends on locality of execution to make up for the extra cycles used in translating the code.

The DR Emulator provides a high degree compatibility for 680x0 code. One area where compatibility isless than that of the current interpretive emulator is for self-modifying code that does not call the cache flushing routines. Such code also has compatibility problems on Macintosh Quadra models with the cache enabled.

Resource Manager in Native Code

The Resource Manager in the software for the Power Macintosh 7200 computer is similar to the one in the earlier Power Macintosh computers except that it is completely in native PowerPC code. Because the Resource Manager is intensively used both by system software and by applications, the native version provides an improvement in system performance.

The Process Manager has been modified to remove patches it formerly made to the Resource Manager.

Math Library

The new math library (MathLib) is an enhanced version of the floating-point library included in the ROM in the 6100, 7100, and 8100 Power Macintosh computers.

The new math library is bit compatible in both results and floating-point exceptions with the math library in the first Power Macintosh ROM. The only difference is in the speed of computation.

The new math library has been improved to better exploit the floating-point features of the PowerPC microprocessor. The math library now includes enhancements that assist the compiler in carrying out its register allocation, branch prediction, and overlapping of integer and floating-point operations.

Compared with the previous version, the new math library provides much improved performance without compromising its accuracy or robustness. It provides performance gains for often-used functions of up to 15 times.

The application interface and header files for the math library have not been changed.

New BlockMove Extensions

The system software for the Power Macintosh 7200 computer includes new extensions to the `BlockMove` routine. The extensions provide improved performance for programs running in native mode.

The new `BlockMove` extensions provide several benefits for developers.

- They're compatible with the new Dynamic Recompilation Emulator.
- They provide a way to handle cache-inhibited address spaces.
- They include new high-speed routines for setting memory to zero.

Note
The new `BlockMove` extensions do not use the string instructions, which are fast on the PowerPC 601 but slow on other PowerPC implementations. ◆

Some of the new `BlockMove` extensions can be called only from native code.

Except for `BlockZero` and `BlockZeroUncached`, the new `BlockMove` extensions use the same parameters as `BlockMove`. Calls to `BlockZero` and `BlockZeroUncached` have only two parameters, a pointer and a length, the same as the second and third parameters of `BlockMove`.

Table 5-1 summarizes the BlockMove routines according to two criteria: whether it is OK to use for moving 680x0 code, and whether it is OK to use with buffers or other uncacheable destination locations.

Table 5-1 Summary of BlockMove routines

BlockMove version	OK to use for moving 680x0 code	OK to use with buffers
BlockMove	Yes	No
BlockMoveData	No	No
BlockMoveDataUncached	No	Yes
BlockMoveUncached	Yes	Yes
BlockZero	—	No
BlockZeroUncached	—	Yes

IMPORTANT
The BlockMove versions for cacheable data use the dcbz instruction to avoid unnecessary pre-fetch of destination cache blocks. For uncacheable data, you should avoid using those routines because the dcbz instruction faults on uncacheable or write-through locations, making execution extremely slow. ▲

If you are developing driver software for more than one platform, you may be able to simplify your design by using the BlockCopy routine in the Driver Services Library. The BlockCopy routine is an abstraction that allows you to postpone binding the specific type of BlockMove operation until implementation time.

The Driver Services Library is a collection of useful routines that Apple Computer provides for developers working with the Power Macintosh computers that utilize the PCI bus. For more information, please refer to *Designing PCI Cards and Drivers for Power Macintosh Computers.*

Hardware Support Features

The system software for the Power Macintosh 7200 computer includes the following features to support the hardware:

- PCI bus support
- QuickDraw acceleration API
- Display Manager
- support of native drivers

PCI Bus Support

The Power Macintosh 7200 computer does not use NuBus™ slots for hardware expansion, but instead uses the industry standard PCI bus architecture. To support the Power Macintosh 7200 computer as well as future Macintosh models that do not use the NuBus architecture, new system software includes a bus-neutral expansion architecture used by system software in place of Slot Manager calls that are specific to NuBus.

Removal of Slot Manager Dependencies

The system software that controls NuBus cards in current Macintosh models has many explicit dependencies upon the Slot Manager. The system software for models that use PCI bus slots requires changes to each of those dependencies so that PCI cards can operate with the system in the same fashion as NuBus cards.

The system software that formerly called the Slot Manager has been modified to use other services. The new Display Manager provides the means of obtaining video-specific information that was previously obtained by way of the Slot Manager. For example, QuickDraw currently calls the Slot Manager at startup time to check the consistency of the `'scrn'` resource. In the software for the Power Macintosh 7200 computer, QuickDraw calls the new Display Manager to check this consistency.

The following components formerly used the Slot Manager; they have been modified to use the services of the Display Manager:

- Monitors control panel
- QuickDraw
- Palette Manager
- Device Manager

PCI Compatibility

To support a third-party NuBus-to-PCI bridge product for PCI-based computers, it is important to retain Slot Manager capability. Also, several important applications (such as DECnet and SoftWindows) rely on Slot Manager calls to indicate the presence of networking cards. For compatibility, the new expansion architecture supports existing PCI-based cards by way of particular Slot Manager calls.

For more information about PCI expansion cards, please refer to *Designing PCI Cards and Drivers for Power Macintosh Computers.*

QuickDraw Acceleration API

The native QuickDraw acceleration API makes it easier for third-party card vendors and driver writers to produce video accelerator cards for the Power Macintosh 7200 computer.

The QuickDraw acceleration API is the current accelerator interface for the Power PC version of native QuickDraw. It allows a patch chaining mechanism for decisions on categories of block transfers, and also specifies the format and transport of the data to the accelerator.

IMPORTANT

This interface and design is intended only for the Power Macintosh Power Macintosh 7200, 7500, 8500, and 9500 computers and does not represent a new standard for future Macintosh models. ▲

Display Manager

Until now, system software has used the NuBus-specific Slot Manager to get and set information about display cards and drivers.The new system software removes this explicit software dependency on the architecture of the expansion bus. The Display Manager provides a uniform API for display devices regardless of the implementation details of the devices.

In a computer that uses PCI expansion cards, the Slot Manager is generally not available to provide information about display cards; instead, the Expansion Manager must be used. The Display Manager makes the actual calls to either the Slot Manager or the Expansion Manager, as appropriate, thus isolating the bus-specific calls to a single component and avoiding the need to change additional system software in the future. See the section “Removal of Slot Manager Dependencies” on page 56.

Support of Native Drivers

The Power Macintosh 7200 computer uses a new native-driver model for system software and device driver developers. Several components of system software are being modified to support native drivers. The modified components are

- Device Manager
- interrupt tree services
- driver loader library
- driver support library
- Slot Manager stubs
- Macintosh startup code
- interface libraries
- system registry

For more information, refer to *Designing PCI Cards and Drivers for Power Macintosh Computers.*

CHAPTER 6

Large Volume Support

This chapter describes the large volume file system for the Power Macintosh 7200 computer. The large volume file system is a version of the hierarchical file system (HFS) that has been modified to support volume sizes up to 2 tarabytes. It incorporates only the changes required to achieve that goal.

Overview of the Large Volume File System

The large volume file system includes

- modifications to the HFS ROM code, Disk First Aid, and Disk Init
- a new extended API that allows reporting of volume size information beyond the current 4 GB limit
- new device drivers and changes to the Device Manager API to support devices that are greater than 4 GB
- a new version of HDSC Setup (called Drive Setup) that supports large volumes and chainable drivers (chainable drivers are needed to support booting large volumes on earlier Macintosh models)

API Changes

The system software on the Power Macintosh 7200 computer allows all current applications to work without modifications. Unmodified applications that call the file system still receive incorrect values for large volume sizes. The Finder and other utility programs that need to know the actual size of a volume have been modified to use the new extended `PBXGetVolInfo` function to obtain the correct value.

The existing low-level driver interface does not support I/O to a device with a range of addresses greater than 4 GB because the positioning offset (in bytes) for a read or write operation is a 32-bit value. To correct this problem, a new extended I/O parameter block record has been defined. This extended parameter block has a 64-bit positioning offset. The new parameter block and the extended `PBXGetVolInfo` function are described in “The API Modifications” beginning on page 61.

Allocation Block Size

The format of HFS volumes has not changed. What has changed is the way the HFS software handles the allocation block size. Existing HFS code treats the allocation block as a 16-bit integer. The large volume file system uses the full 32 bits of the allocation block size parameter. In addition, any software that deals directly with the allocation block size from the volume control block must now treat it as a true 32-bit value.

Even for the larger volume sizes, the number of allocation blocks is still defined by a 16-bit integer. As the volume size increases, the size of the allocation block also increases. For a 2 GB volume, the allocation block size is 32 KB and therefore the smallest file on that disk will occupy at least 32 KB of disk space. This inefficient use of disk space is not addressed by the large volume file system.

The maximum number of files continues to be less than 65,000. This limit is directly related to the fixed number of allocation blocks.

File Size Limits

The HFS has a maximum file size of 2 GB. The large volume file system does not remove that limit because doing so would require a more extensive change to the current API and would incur more compatibility problems.

Compatibility Requirements

The large volume file system requires at least a 68020 microprocessor or a Power Macintosh model that emulates it. In addition, the file system requires a Macintosh IIci or more recent model. On a computer that does not meet both those requirements, the large volume file system driver does not load.

The large volume file system requires System 7.5 or later and a new Finder that supports volumes larger than 4 GB (using the new extended `PBXGetVolInfo` function).

The API Modifications

The HFS API has been modified to support volume sizes larger than 4 GB. The modifications consist of two extended data structures and a new extended `PBXGetVolInfo` function.

Data Structures

This section describes the two modified data structures used by the large volume file system:

- the extended volume parameter block
- the extended I/O parameter block

Extended Volume Parameter Block

In the current `HVolumeParam` structure, volume size information is clipped at 2 GB. Because HFS volumes can now exceed 4 GB, a new extended volume parameter block is needed in order to report the larger size information. The `XVolumeParam` structure contains 64-bit integers for reporting the total bytes on the volume and the number of free bytes available (parameter names `ioVTotalBytes` and `ioVFreeBytes`). In addition, several of the fields that were previously signed are now unsigned (parameter names `ioVAtrb`, `ioVBitMap`, `ioAllocPtr`, `ioVAlBlkSiz`, `ioVClpSiz`, `ioAlBlSt`, `ioVNxtCNID`, `ioVWrCnt`, `ioVFilCnt`, and `ioVDirCnt`).

```
struct XVolumeParam {
   ParamBlockHeader
   unsigned long     ioXVersion;      // XVolumeParam version == 0
   short             ioVolIndex;      // volume index
   unsigned long     ioVCrDate;       // date & time of creation
   unsigned long     ioVLsMod;        // date & time of last modification
   unsigned short    ioVAtrb;         // volume attributes
   unsigned short    ioVNmFls;        // number of files in root directory
   unsigned short    ioVBitMap;       // first block of volume bitmap
   unsigned short    ioAllocPtr;      // first block of next new file
   unsigned short    ioVNmAlBlks;     // number of allocation blocks
   unsigned long     ioVAlBlkSiz;     // size of allocation blocks
   unsigned long     ioVClpSiz;       // default clump size
   unsigned short    ioAlBlSt;        // first block in volume map
   unsigned long     ioVNxtCNID;      // next unused node ID
   unsigned short    ioVFrBlk;        // number of free allocation blocks
   unsigned short    ioVSigWord;      // volume signature
   short             ioVDrvInfo;      // drive number
   short             ioVDRefNum;      // driver reference number
   short             ioVFSID;         // file-system identifier
   unsigned long     ioVBkUp;         // date & time of last backup
   unsigned short    ioVSeqNum;       // used internally
   unsigned long     ioVWrCnt;        // volume write count
   unsigned long     ioVFilCnt;       // number of files on volume
   unsigned long     ioVDirCnt;       // number of directories on volume
   long              ioVFndrInfo[8];  // information used by the Finder
   uint64            ioVTotalBytes;   // total number of bytes on volume
   uint64            ioVFreeBytes;    // number of free bytes on volume
};
```

Field descriptions

`ioXVersion` Version number of the extended volume parameter block.

`ioVolIndex` An index for use with the `PBHGetVInfo` function.

`ioVCrDate` The date and time of volume initialization.

`ioVLsMod` The date and time the volume information was last modified. (This field is not changed when information is written to a file and does not necessarily indicate when the volume was flushed.)

`ioVAtrb` The volume attributes.

ioVNmFls The number of files in the root directory.

ioVBitMap The first block of the volume bitmap.

ioAllocPtr The block at which the next new file starts. Used internally.

ioVNmAlBlks The number of allocation blocks.

ioVAlBlkSiz The size of allocation blocks.

ioVClpSiz The default clump size.

ioAlBlSt The first block in the volume map.

ioVNxtCNID The next unused catalog node ID.

ioVFrBlk The number of unused allocation blocks.

ioVSigWord A signature word identifying the type of volume; it's $D2D7 for Macintosh File System (MFS) volumes and $4244 for volumes that support HFS calls.

ioVDrvInfo The drive number of the drive containing the volume.

ioVDRefNum For online volumes, the reference number of the I/O driver for the drive identified by ioVDrvInfo.

ioVFSID The file-system identifier. It indicates which file system is servicing the volume; it's zero for File Manager volumes and nonzero for volumes handled by an external file system.

ioVBkUp The date and time the volume was last backed up (it's 0 if never backed up).

ioVSeqNum Used internally.

ioVWrCnt The volume write count.

ioVFilCnt The total number of files on the volume.

ioVDirCnt The total number of directories (not including the root directory) on the volume.

ioVFndrInfo Information used by the Finder.

ioVTotalBytes Total number of bytes on volume.

ioVFreeBytes Total number of free bytes on volume.

Extended I/O Parameter Block

The extended I/O parameter block is needed for low-level access to disk addresses beyond 4 GB. It is used exclusively by PBRead and PBWrite calls when performing I/O operations at offsets greater than 4 GB. To indicate that you are using an XIOParam record, you should set the kUseWidePositioning bit in the ioPosMode field.

Because file sizes are limited to 2 GB, the regular IOParam structure should always be used when performing file level I/O operations. The extended parameter block is intended only for Device Manager I/O operations to large block devices at offsets greater than 4 GB.

The only change in the parameter block is the parameter `ioWPosOffset`, which is of type `int64`.

```
struct XIOParam {
   QElemPtr      qLink;       // next queue entry
   short         qType;       // queue type
   short         ioTrap;      // routine trap
   Ptr           ioCmdAddr;   // routine address
   ProcPtr       ioCompletion;// pointer to completion routine
   OSErr         ioResult;    // result code
   StringPtr     ioNamePtr;   // pointer to pathname
   short         ioVRefNum;   // volume specification
   short         ioRefNum;    // file reference number
   char          ioVersNum;   // not used
   char          ioPermssn;   // read/write permission
   Ptr           ioMisc;      // miscellaneous
   Ptr           ioBuffer;    // data buffer
   unsigned long ioReqCount;  // requested number of bytes
   unsigned long ioActCount;  // actual number of bytes
   short         ioPosMode;   // positioning mode (wide mode set)
   int64         ioWPosOffset;// wide positioning offset
};
```

Field descriptions

`ioRefNum` The file reference number of an open file.

`ioVersNum` A version number. This field is no longer used and you should always set it to 0.

`ioPermssn` The read/write permission access mode.

`ioMisc` Depends on the routine called. This field contains either a new logical end-of-file, a new version number, a pointer to an access path buffer, or a pointer to a new pathname. Because `ioMisc` is of type `Ptr`, you'll need to perform type coercion to interpret the value of `ioMisc` correctly when it contains an end-of-file (a `LongInt` value) or version number (a `SignedByte` value).

`ioBuffer` A pointer to a data buffer into which data is written by read calls and from which data is read by write calls.

`ioReqCount` The requested number of bytes to be read, written, or allocated.

`ioActCount` The number of bytes actually read, written, or allocated.

ioPosMode The positioning mode for setting the mark. Bits 0 and 1 of this field indicate how to position the mark; you can use the following predefined constants to set or test their value:

```
CONST
fsAtMark = 0;     {at current mark}
fsFromStart = 1;  {from beginning of file}
fsFromLEOF = 2;   {from logical end-of-file}
fsFromMark = 3;   {relative to current mark}
```

You can set bit 4 of the ioPosMode field to request that the data be cached, and you can set bit 5 to request that the data not be cached. You can set bit 6 to request that any data written be immediately read; this ensures that the data written to a volume exactly matches the data in memory. To request a read-verify operation, add the following constant to the positioning mode:

```
CONST
rdVerify = 64;    {use read-verify mode}
```

You can set bit 7 to read a continuous stream of bytes and place the ASCII code of a newline character in the high-order byte to terminate a read operation at the end of a line.

ioPosOffset The offset to be used in conjunction with the positioning mode.

New Extended Function

This section describes the extended PBXGetVolInfo function that provides volume size information for volumes greater than 4 GB.

Before using the new extended call, you should check for availability by calling the Gestalt function. Make your call to Gestalt with the gestaltFSAttr selector to check for new File Manager features. The response parameter has the gestaltFSSupports2TBVolumes bit set if the File Manager supports large volumes and the new extended function is available.

PBXGetVolInfo

You can use the PBXGetVolInfo function to get detailed information about a volume. It can report volume size information for volumes up to 2 terabytes.

```
pascal OSErr PBXGetVolInfo (XVolumeParam paramBlock,
                            Boolean async);
```

paramBlock A pointer to an extended volume parameter block.

async A Boolean value that specifies asynchronous (true) or synchronous (false) execution.

An arrow preceding a parameter indicates whether the parameter is an input parameter, an output parameter, or both:

Arrow	Meaning
→	Input
←	Output
↔	Both

Parameter block

→	`ioCompletion`	`ProcPtr`	Pointer to a completion routine.
←	`ioResult`	`OSErr`	Result code of the function.
↔	`ioNamePtr`	`StringPtr`	Pointer to the volume’s name.
↔	`ioVRefNum`	`short`	On input, a volume specification; on output, the volume reference number.
→	`ioXVersion`	`unsigned long`	Version of `XVolumeParam` (value = 0).
→	`ioVolIndex`	`short`	Index used for indexing through all mounted volumes.
←	`ioVCrDate`	`unsigned long`	Date and time of initialization.
←	`ioVLsMod`	`unsigned long`	Date and time of last modification.
←	`ioVAtrb`	`unsigned short`	Volume attributes.
←	`ioVNmFls`	`unsigned short`	Number of files in the root directory.
←	`ioVBitMap`	`unsigned short`	First block of the volume bitmap.
←	`ioVAllocPtr`	`unsigned short`	Block where the next new file starts.
←	`ioVNmAlBlks`	`unsigned short`	Number of allocation blocks.
←	`ioVAlBlkSiz`	`unsigned long`	Size of allocation blocks.
←	`ioVClpSiz`	`unsigned long`	Default clump size.
←	`ioAlBlSt`	`unsigned short`	First block in the volume block map.
←	`ioVNxtCNID`	`unsigned long`	Next unused catalog node ID.
←	`ioVFrBlk`	`unsigned short`	Number of unused allocation blocks.
←	`ioVSigWord`	`unsigned short`	Volume signature.
←	`ioVDrvInfo`	`short`	Drive number.
←	`ioVDRefNum`	`short`	Driver reference number.
←	`ioVFSID`	`short`	File system handling this volume.
←	`ioVBkUp`	`unsigned long`	Date and time of last backup.
←	`ioVSeqNum`	`unsigned short`	Used internally.
←	`ioVWrCnt`	`unsigned long`	Volume write count.
←	`ioVFilCnt`	`unsigned long`	Number of files on the volume.
←	`ioVDirCnt`	`unsigned long`	Number of directories on the volume.

continued

←	`ioVFndrInfo[8]`	`long`	Used by the Finder.
←	`ioVTotalBytes`	`uint64`	Total number of bytes on the volume.
←	`ioVFreeBytes`	`uint64`	Number of free bytes on the volume.

DESCRIPTION

The `PBXGetVolInfo` function returns information about the specified volume. It is similar to the `PBHGetVInfo` function described in *Inside Macintosh: Files* except that it returns additional volume space information in 64-bit integers.

ASSEMBLY-LANGUAGE INFORMATION

The trap macro and routine selector for `PBXGetVolInfo` are

Trap macro	Selector
`_HFSDispatch`	$0012

RESULT CODES

`noErr`	0	Successful completion, no error occurred
`nsvErr`	–35	No such volume
`paramErr`	–50	No default volume

Glossary

680x0 code Instructions that can run on a PowerPC processor only by means of an emulator. Compare **native code.**

ADB See **Apple Desktop Bus.**

APDA Apple Computer's worldwide direct distribution channel for Apple and third-party development tools and documentation products.

API See **application programming interface.**

Apple AV technologies A set of advanced I/O features for Macintosh computers that includes video input and output, sophisticated 16-bit stereo sound input and output, and speech recognition and synthesis.

Apple Desktop Bus (ADB) An asynchronous bus used to connect relatively slow user-input devices to Apple computers.

Apple SuperDrive Apple Computer's disk drive for high-density floppy disks.

AppleTalk Apple Computer's local area networking protocol.

application programming interface (API) A set of calls, instructions, and data structures in system software or a processor instruction set that application software can use to program the computer.

arbitration The process of determining which of several contending subsystems gains control of a bus at any given time.

audio waveform amplifier and converter (AWAC) A custom IC that combines a waveform amplifier with a digital encoder and decoder (codec) for analog sound signals, including speech.

autoconfiguration A method of integrating peripheral devices into a computer system that includes a mechanism for configuring devices during system startup and requires that vendors include expansion ROMs on plug-in cards.

AV card A PDS card that provides the video I/O features of the Apple AV Technologies on first-generation Power Macintosh computers.

AWAC See **audio waveform amplifier and converter.**

Bandit A custom IC that provides the interface between the system bus and the PCI bus in Power Macintosh 7200, 7500, 8500, and 9500 computers.

big-endian Data formatting where fields are addressed by pointers to their most significant bytes or bits. See also **little-endian.**

block transfer Data transfers of more than one longword at a time.

boot driver A device driver that is used during the Open Firmware startup process. It must be written in FCode and is usually loaded from the expansion ROM on a PCI card.

branch unit The part of a PowerPC microprocessor that handles branch instructions, usually without adding any time to program execution.

CISC See **complex instruction set computing.**

codec A digital encoder and decoder.

color depth The number of bits required to encode the color of each pixel in a display.

complex instruction set computing (CISC) A technology of microprocessor design in which machine instructions have nonuniform formats and are executed through different processes.

CPU bus The bus connected directly to the main processor.

Cuda A microcontroller IC that manages the ADB and real-time clock, maintains parameter RAM, manages power on and reset, and performs other general system functions.

Curio A custom IC that provides I/O interfaces for Ethernet, SCSI, SCC, and LocalTalk.

DAC See **digital-to-analog converter.**

data burst Multiple longwords of data sent over a bus in a single, uninterrupted stream.

data cache In a PowerPC microprocessor, the internal registers that hold data being processed.

DAV interface See **digital audio/video interface.**

descriptor-based direct memory access (DBDMA) A DMA technique using DMA descriptor lists that are read from memory by the IC performing the DMA transfers.

device node In a device tree, a node that serves one hardware device.

device tree A software structure, generated during the Open Firmware startup process, that assigns a node to each PCI device available to the system.

digital audio/video (DAV) interface A connector on the AV card that lets a NuBus card with a ribbon cable on it transfer digital sound and unscaled YUV video data directly to the AV card.

digital-to-analog converter (DAC) A device that produces an analog electrical signal in response to digital data.

DIMM See Dual Inline Memory Module.

direct memory access (DMA) A process for transferring data rapidly into or out of RAM without passing it through a processor or buffer.

DLPI Data Link Provider Interface, the standard networking model used in Open Transport.

DMA See **direct memory access.**

DRAM See **dynamic random-access memory.**

DR Emulator The Dynamic Recompilation Emulator, an improved 680x0-code emulator for the Power PC.

Dual Inline Memory Module (DIMM) A DRAM expansion card. On the Power Macintosh 7200 computer this is a 128-pin card with a 64-bit wide data path. The card has one or multilple banks of 70 ns or faster DRAM devices that are used to expand the dynamic memory capacity of the computer. See also **dynamic random access memory.**

dynamic random-access memory (DRAM) Random-access memory in which each storage address must be periodically interrogated ("refreshed") to maintain its value.

Ethernet A high-speed local area network technology that includes both cable standards and a series of communications protocols.

exception An error or other unusual condition detected by a processor during program execution.

Expansion Manager A set of toolbox routines that provides bus-neutral support for expansion devices. The Expansion Manager is a superset of the Slot Manager.

fat binary An execution module that contains native code and 680x0 code for the same program.

FCode A tokenized version of the Forth programming language, used in PCI card expansion ROMs.

FCode tokenizer A utility program that translates Forth source code into FCode.

floating-point format A data format that stores the magnitude, sign, and significant digits of a number separately.

GeoPort A software and hardware solution for digital telecom and wide-area connectivity using the serial port.

Grand Central A custom IC that provides core I/O services in second-generation Power Macintosh computers.

input/output (I/O) Parts of a computer system that transfer data to or from peripheral devices.

instruction queue The part of a PowerPC microprocessor that holds incoming instructions.

I/O See **input/output.**

Iridium A custom IC that works with the Platinum memory controller IC to optimize data throughput during ROM, DRAM, and VRAM data transactions.

little-endian Data formatting where fields are addressed by pointers to their least significant bytes or bits. See also **big-endian.**

LocalTalk The cable terminations and other hardware that Apple supplies for local area networking from Macintosh serial ports.

MACE See **Media Access Controller for Ethernet.**

MC68000 The model number of a family of microprocessor ICs manufactured by Motorola.

Media Access Controller for Ethernet (MACE) Circuitry within the Curio IC that supports Ethernet I/O.

MFM See **Modified Frequency Modulation.**

mini-DIN An international standard form of cable connector for peripheral devices.

multihoming A feature of Open Transport that makes it possible for AppleTalk to be active on more than one network port on the machine at a time. Also called *multiporting.*

native code Instructions that run directly on a PowerPC processor. Compare **680x0 code.**

nonvolatile RAM RAM that retains its contents even when the computer is turned off; also known as *parameter RAM.*

NuBus A bus architecture in Apple computers that supports plug-in expansion cards.

Open Firmware startup process The startup process by which PCI-compatible Power Macintosh computers recognize and configure peripheral devices connected to the PCI bus.

Open Transport A networking architecture that allows communications applications to run independently of the underlying network; formerly known as *Transport-Independent Interface (TII).*

PCI Peripheral Component Interconnect, an industry-standard expansion bus architecture.

pipelining The technique of sending instructions through multiple processing units in such a way that each unit handles one instruction per clock cycle.

pixel Contraction of *picture element*; the smallest dot that can be drawn on a display.

Platinum A custom IC that controls the memory and cache subsystem in the Power Macintosh 7200 computer.

PowerPC Tradename for a family of RISC microprocessors. The PowerPC 601, 603, and 604 microprocessors are used in Power Macintosh computers.

processing unit The part of a PowerPC microprocessor that executes instructions.

property list An element of a device tree that contains information about a device on a PCI bus.

reduced instruction set computing (RISC) A technology of microprocessor design in which all machine instructions are uniformly formatted and are processed through the same steps.

RISC See **reduced instruction set computing.**

SCC See **Serial Communications Controller.**

SCSI See **Small Computer System Interface.**

Serial Communications Controller (SCC) Circuitry on the Curio IC that provides an interface to the serial data ports.

SIMM See **Single Inline Memory Module.**

Single Inline Memory Module (SIMM) A plug-in card for memory expansion, containing several RAM ICs and their interconnections.

Small Computer System Interface (SCSI) An industry standard parallel bus protocol for connecting computers to peripheral devices such as hard disk drives.

Streams The standard UNIX-based networking model used in Open Transport.

S-video A video format in which chroma and luminance are transmitted separately; also called *Y/C.* It provides higher image quality than composite video.

SWIM III A custom IC that controls the Apple SuperDrive floppy disk drive.

Versatile Interface Adapter (VIA) The interface for system interrupts that is standard on most Apple computers.

VIA See **Versatile Interface Adapter.**

video RAM (VRAM) Random-access memory used to store both static graphics and video frames.

VRAM See **video RAM.**

Y/C Same as **S-video.**

YUV A data format for each pixel of a color display in which color is encoded by values calculated from its native red, green, and blue components.

Index

A

abbreviations xii
ADB 21–22
ADB connector 21–22
address multiplexing, RAM 36–38
APDA information x
appearance 2
Apple AUI connector 22
Apple Desktop Bus. *See* ADB
Apple PlainTalk microphone 25
Apple SuperDrive 28
AppleTalk stack for Open Transport 49
AWAC custom IC 16

B

Bandit custom IC 15
bank base registers 14
big-endian addressing 15
`BlockCopy` routine 55
block diagram 13
`BlockMoveData` routine 55
`BlockMoveDataUncached` routine 55
`BlockMove` extensions 54–55
`BlockMove` routine 55
`BlockMoveUncached` routine 55
`BlockZero` routine 55
`BlockZeroUncached` routine 55
boot drivers 51
bus arbitration 15
bus clock rates 15
byte order 15

C

CD-ROM drive 29
codec 16
compatibility
  and large volume file system 61
  of Open Transport 49
  of PCI and NuBus 56
compatibility issues 8
connectors
  ADB 21–22
  Ethernet 22
  floppy disk 28
  SCSI port 23
  serial port 20–21
  sound input 25
  sound output 25
  10baseT 22
  video monitor 26
Cuda custom IC 16
Curio custom IC 16
custom ICs
  AWAC 16
  Bandit 15
  Cuda 16
  Curio 16
  Grand Central 15
  Iridium 14
  Platinum 14

D

data bus 14
data interleaving 14
DBDMA 15
`dcbz` instruction 55
descriptor-based DMA 15
Device Manager 63
device tree 50
DIMMs. *See* RAM DIMMs; VRAM DIMMs
Display Manager 56, 57
  components modified for 56
DLPI drivers compared with `.ENET` drivers 49
DMA 15
DRAM devices 38–39
DR Emulator 53
Driver Services Library 55
Drive Setup utility 48
Dual Inline Memory Modules. *See* RAM DIMMs; VRAM DIMMS
Dynamic Recompilation Emulator 53

E

Emulator, Dynamic Recompilation 53
`.ENET` drivers compared with DLPI drivers 49
Ethernet port 22
Ethernet port connectors 22
expansion bus. *See* PCI expansion bus
Expansion Manager 57
expansion slots. *See* PCI expansion slots
extended I/O parameter block 63–65
extended volume parameter block 61–63

F

FCode 50
FCode tokenizer 51
features
  hardware, list of 5
  software 46–53
Finder modifications for large volume support 47, 60
floating-point library 54
floppy disk connector 28
floppy disk drive 28
Forth language 50
  vocabulary reference 51

G

GeoPort protocols 20
`Gestalt` function 65
GPi (general purpose input) signal 21
Grand Central custom IC 15

H

hard disk drive 29
hexadecimal numbers xi
HFS volume format 60

I, J, K

interpretive emulator 53
I/O parameter block, extended 63–65
I/O ports 20
Iridium custom IC 14

L

L2 cache 14
L2 cache SIMM 40
large partition support 46
large volume support 46, 60, 63
  allocation blocks 61
  extended API 46
  extended data structures 61–65
  extended function 65–67
  extended parameter block 62
  limitations 48
  maximum file size 61
  modified applications 47
  requirements 61
  64-bit addresses 46
level-2 cache 14, 40
little-endian addressing 15

M

MACE (Media Access Controller for Ethernet) 16
MacTCP stack for Open Transport 49
main memory 13
main processor. *See* PowerPC 601 microprocessor
math library 54
MC68HC05 microcontroller 16
Media Access Controller for Ethernet (MACE) 16
memory
  bank base registers for 14
  contiguous banks 14
  sizes and configurations of 32
memory expansion 13
memory management unit 12
microphone 25
MMU 12
modem port 21
monitor connector 26
monitor sense lines 27
monitors. *See* video monitors
multihoming, in Open Transport 49
multiplexing, RAM address 36–38

N

native drivers 57
network port 22–23
NuBus and PCI compatibility 8, 56

O

Open Firmware startup process 50–51
  boot drivers 51
  device tree 50
  property list 50
  standards for xi, 50
Open Transport 48–50
  AppleTalk stack for 49
  client interface 48
  compatibility issues 49
  development environment 48
  DLPI drivers 49
  features of 49
  MacTCP stack for 49

P

`PBXGetVolInfo` function 65
PCI expansion bus 12, 42
  Slot Manager dependencies 56
  software support for 56
  standards for xi
PCI expansion slots 42–43
PlainTalk microphone 25
Platinum custom IC 14
Power Macintosh computers, comparison with earlier models 5
power management software 51–53
PowerPC 601 microprocessor 12
processor bus 12
property list 50

Q

QuickDraw 56
QuickDraw acceleration API 56

R

RAM devices 38–39
  access time of 38
  refresh operation 38
RAM DIMMs 14, 32–39
  address multiplexing for 36–38
  connectors 33
  connector type 32
  devices in 36, 38
  dimensions of 38
  installation of 32
  signal descriptions 36
  specifications 32
reference documents
  from Apple x
  other than Apple xi
Resource Manager in native code 53
ROM 12

S

SCSI bus termination 24
SCSI port connectors 23, 24
second-level cache 14
serial port connectors 20–21
serial ports 20–21
Slot Manager 56
  compatibility with existing PCI cards 56
  relationship to Display Manager 57
  removal of dependencies 56
software features 46
sound IC 16
sound input and output 25
sound input jack 25
sound output jack 25
SRAM devices 14

startup firmware. *See* Open Firmware startup process
Streams network protocol 49
SWIM III floppy disk controller 15
System 7.5 46

T, U

10baseT connector 23

V, W

video monitor connector 26
video monitors
detecting type of 27
and VRAM sizes 17
volume control block allocation block size 60
volume parameter block, extended 61–63
VRAM 17
VRAM DIMMs 17

X, Y, Z

XIOParam data structure 63
XTI interface, with Open Transport 48
XVolumeParam parameter block 62

THE APPLE PUBLISHING SYSTEM

This Apple manual was written, edited, and composed on a desktop publishing system using Apple Macintosh computers and FrameMaker software. Proof pages and final pages were created on an Apple LaserWriter Pro 630 printer. Line art was created using Adobe Illustrator™ and Adobe Photoshop™. PostScript™, the page-description language for the LaserWriter, was developed by Adobe Systems Incorporated.

Text type is Palatino® and display type is Helvetica®. Bullets are ITC Zapf Dingbats®. Some elements, such as program listings, are set in Apple Courier.

WRITER
Steve Schwander

DEVELOPMENTAL EDITOR
Laurel Rezeau

ILLUSTRATORS
Sandy Karr, Tom Llewellyn

Special thanks to Allen Watson, Michael Dhuey, Dale Adams, Buddy Snow, Jacqui Gray, Bonnie Woo, and Kevin Peoples.