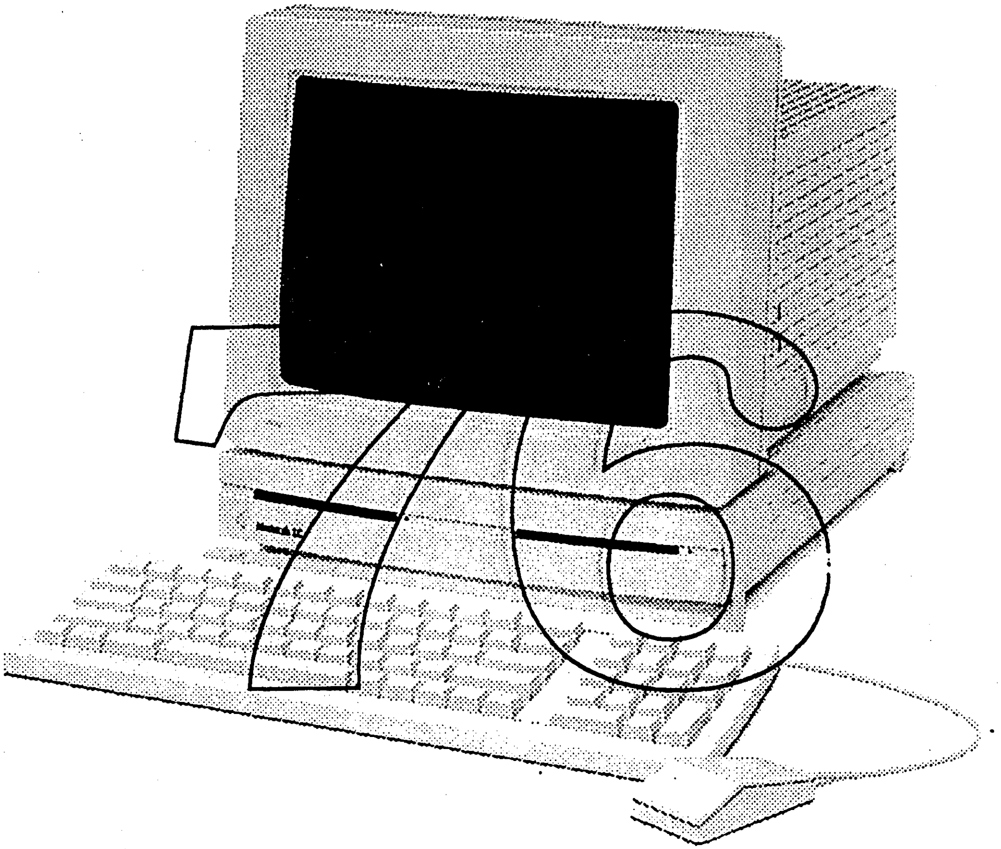
MACINTOSH HARDWARE OVERVIEW

Rev 2.0
February 11, 1991



John Atwood
(rev 1, 2)

Andy Poggio
(rev 1)

Architectural Investigations & Modeling
Functional Technology Group, Macintosh Products

Apple Computer Inc.
20705 Valley Green Dr. M-S 60AR
Cupertino, CA 95014

Apple Registered Confidential

76

TABLE OF CONTENTS

76

1. INTRODUCTION

This document is a summary of the Apple Macintosh CPU hardware capabilities and constraints. It can serve multiple purposes, including establishing a baseline Macintosh architecture, and to provide new hardware and software engineers at Apple a hardware review of the Macintosh. In a sense, this serves a similar purpose as the Macintosh Family Hardware Reference and the Inside Macintosh series. However, this is an Apple internal document, and thus can include proprietary information and make references to internal specifications.

The common hardware features of the Macintosh are documented, as well as major differences. To keep the document small, details are left to ERSs or specifications, which are referenced in the text. Since this document is being updated within a fixed schedule, not all areas have been given equal coverage. A goal is to continue updating this, so that after several revisions, all areas are well-covered. Your feedback, both in correcting errors, and in suggesting areas to emphasize, will be very helpful in improving the quality of this document.

Special notes or warnings that affect hardware engineers, software designers, or developers of the Mac are highlighted.

If you find any errors, inconsistencies, unclear areas, or omissions, please alert me (John Atwood, in the AIM Group) so that subsequent versions can be updated. Given time and demand, new releases should come out approximately every 6 to 12 months.

WARNING

This document contains highly confidential material, and should be treated with the utmost care. Do not make copies of this. Dispose of old copies in confidential bins.

76

CHANGES SINCE THE LAST RELEASE

Since the last version of this document was released, (version 1.0, April 19, 1990), eleven new processors have been added as official projects. These bring with them many new subsystems and ASICs. This release covers as much of this new information as possible in the time allotted to the update. In addition, the organization and emphasis has been changed. Chapters on general mechanical and manufacturing design philosophy have been dropped, with the emphasis focused on the computer architectural aspects. The original concentration on describing one representative machine from each design center has been dropped, being replaced by more references to the CPU feature tables (Tables 1.1 through 1.5). A chapter has been added on the upcoming new video overlay capability.

The entire document was reviewed, and minor corrections and updates have been made throughout. In addition, several areas were substantially changed or enlarged. These include:

- Virtual Memory and MMUs
- Memory Maps
- NuBus'90
- Cyclone and Oceanic's PSC-style DMA architecture
- The whole graphics chapter
- Floppy drive controllers
- Ethernet
- ChefCat
- VIA bits

76

The following areas need better coverage. Please foward your comments or ideas on these for inclusion in future releases of this document:

- Video Overlay
- SCSI and Hard Drives
- Power Management

ACKNOWLEDGEMENTS

Many people answered questions, supplied ERS's, reviewed draft versions of this document, and generally helped make this complete and correct. The team leaders and managers of the CPU design groups and the Blue Software group have been especially helpful. The following people have spent their time reviewing this release of the document:

Dale Adams
Paul Baker
Jim Cape
Toby Farrand
Brian Howard
Mary Johnson
Henry Kannapell
Ken Karakotsios
Tony Leung
Rodger Mohme
Michael O'Connor
Scott Sarnikowski
Jon Sedmak
Jim Stair
Mike Teener

76

The next five pages summarize the Macintosh family, with an emphasis on internal hardware features. These tables include information on future machines, which is subject to change.

	Mac Plus	Mac Classic	Mac SE	Mac SE/30	Mac LC
Intro Date	1/86	10/90	3/87	1/89	10/90
Processor: FPU: MMU:	8 MHz 68000 - -	8 MHz 68000 - -	8 MHz 68000 - -	16 MHz 68030 68882 '030	16 MHz 68020 - -
Processor Direct Slot (PDS):	-	-	68000 direct	68030 direct	68020 direct
ICache: DCache: Ext. Cache:	-	-	-	256 256 -	256 - -
ROM	128 Kbyte	256 Kbyte	256 Kbyte	256 Kbyte	512 Kbyte
RAM - minimum, speed, chip sizes	512 Kbyte, 150 ns, 256K, 1M SIMM	1 Mbyte, 150 ns, 256K, 1M SIMM	512 Kbyte, 150 ns, 256K, 1M SIMM	1 Mbyte, 120 ns, 256K, 1M, 4M, 16M SIMM	1 Mbyte, 100 ns, 1M, 4M SIMM
Memory/Video Control	PALs	BBU	BBU	GLUE	V8
Built-in Video	PALs, 512x342x1	BBU, 512x342x1	BBU, 512x342x1	PALs, 512x342x1	V8, 512x384x1,4 640x480x1, (560x384x4)
# NuBus Slots, NuBus Controller	-	-	-	-	-
SCSI Controller, Connector	5380 DB-25	85C80 Combo DB-25	5380 DB-25	5380 DB-25	85C80 Combo DB-25
Floppy Controller	IWM	SWIM	IWM, SWIM	SWIM	SWIM
Serial Ports	4 MHz 8530 SCC	4 MHz 8530 SCC	4 MHz 8530 SCC	4 MHz 8530 SCC	8 MHz 85C80 Combo
Other Networking	-	-	-	-	-
Sound Out - digital	PALs	BBU	BBU	ASC	V8
Sound Out -analog	1 x SONY	1 x SONY	1 x SONY	2 x SONY, TL071	DFAC, MC34119
Sound Input	-	-	-	-	DFAC, V8
ADB	ADB chip, 6522 VIA	ADB chip, 6523 VIA	ADB chip, 6523 VIA	ADB chip, 6523 VIA	Egret (68HC05)
Real Time Clock, Parameter RAM	RTC	RTC	RTC	RTC	Egret (68HC05)
VIA1	6522 VIA	6523 VIA	6523 VIA	6523 VIA	V8
VIA2	-	-	-	6523 VIA	V8
Hard Disk: Max size, Connector	-	3.5" 1/3 height 50-pin	3.5" 1/2 height 50-pin	3.5" 1/2 height 50-pin	3.5" 1/3 height 50-pin

Table 1.1 – High-Volume Macintoshes

76

	Apollo	Columbia	Vail	Aspen	Mr. Ed
Processor: FPU: MMU:	16 MHz 68030 - '030	16 MHz 68030 - '030	16 MHz 68030 - '030	25 MHz 68030 - '030	16 MHz 68030 - '030
Processor Direct Slot (PDS):	-	68030 (same as LC)	68030 (same as LC)	68030 direct	"LC-Like"
ICache: DCache: Ext. Cache:	256 256 -	256 256 -	256 256 -	256 256 -	256 256 -
ROM	512 Kbyte	1 MByte TERROR	1 MByte TERROR	1 MByte TERROR	512 Kbyte, 512KB in SLIM
RAM - minimum, speed, chip sizes	1 Mbyte, 100 ns, 1M, 4M SIMM	1 Mbyte, 100 ns, 1M, 4M SIMM	1 Mbyte, 100 ns, 1M, 4M SIMM	1 Mbyte, 100 ns, 1M, 4M SIMM	512 Kbyte, 100 ns, soldered
Memory/Video Control	Eagle	Sonora	Sonora	Everest	V8/ME
Built-in Video	Eagle, 512x342x1	Sonora, 512x384x8	Sonora, 512x384x1,2,4,8,16 640x400x1,2,4,8,16 640x480x1,2,4,8 640x870x1,2,4 832x624x1,2,4,8	Everest, 512x384x1,2,4,8,16 640x400x1,2,4,8,16 640x480x1,2,4,8 640x870x1,2,4 832x624x1,2,4,8	V8/ME 512x384x16
NTSC, PAL, SVideo	-	-	-	-	460x300 (approx)
SCSI Controller, Connector	85C80 Combo DB-25	85C80 Combo DB-25	85C80 Combo DB-25	Curio DB-25	
Floppy Controller	SWIM	SWIM2	SWIM2	SWIM2, PIC?	SWIM2 in home card
Serial Ports	8 MHz 85C80 Combo	8 MHz 85C80 Combo	8 MHz 85C80 Combo	8 MHz 85C80 Curio	SCC in home card
Other Networking	-	-	-	Ethernet (MACE)	Ethernet (MACE)
Sound Out - digital	Eagle	Sonora	Sonora	Everest	V8/ME
Sound Out -analog	DFAC, MC34119	DFAC2, MC34119	DFAC2, MC34119		(CD Audio)
Sound Input	DFAC, VISA+	Sonora, VISA+	Sonora, VISA+		(CD Audio)
ADB	Egret (68HC05)	Cuda (68HC05)	Cuda (68HC05)	Cuda (68HC05)	Cuda w/ TouchPad Intfc
Real Time Clock, Parameter RAM	Egret (68HC05)	Cuda (68HC05)	Cuda (68HC05)	Cuda (68HC05)	Cuda (68HC05)
VIA1	Eagle	Sonora	Sonora	Everest	V8/ME
VIA2	Eagle	Sonora	Sonora	Everest	V8/ME
Mass Storage (HD unless stated)	3.5" 1/3 height	3.5" 1/3 height	3.5" 1/3 height	3.5" 1/3 height	LCCD (CD-ROM)

Table 1.2 – High-Volume Macintoshes (cont.)

76

	Portable	Aruba	Asahi	Tim LC	Tim	DBlite	Companion
Intro Date	9/89						
Processor: FPU: MMU:	16MHz 68HC000 - -	16MHz 68HC000 - -	16MHz 68HC000 - -	16 MHz 68020 - -	25 MHz 68030 68882 '030	16 MHz 68030 68882 (opt) '030	25 MHz 68030 68882 (opt) '030
Processor Direct Slot (PDS):	68HC000 direct	68HC000 direct		-	-	-	-
ICache: DCache: Ext. Cache:	-	-	-	256 256 -	256 256 -	256 256 -	256 256 -
ROM	256 Kbyte	256 Kbyte	256 Kbyte	1 Mbyte	1 Mbyte	1 MByte	1 MByte
RAM - minimum, speed, chip sizes	1 Mbyte, 100 ns SRAM, 256K, 1M, SIMM	1 Mbyte, 100 ns PSRAM, 256K, 1M, SIMM	2 Mbyte, 100 ns PSRAM, 256K, 1M, SIMM	2 Mbyte, 100 ns PSRAM, 4M on card	2 Mbyte, 100 ns PSRAM, 4M on card	2 Mbyte, 100 ns DRAM, 4M card	2 Mbyte, 100 ns DRAM, 4M, 16M card
Memory/Video Control	Normandy	Normandy	Normandy	Piranha	Piranha	MSC	BMC
Built-in Video	Omaha, 640x400x1 Active Matrix	Omaha II, 640x400x1 Active Matrix	DDC, 640x400x1 FSTN	DDC, 640x400x1 FSTN	DDC 640x400x1 Active Matrix	DDC 640x400x1	??? 640x480x1,16? AM, ST
External Video	-	-	-	-	-	MSC 512x384x1,4 640x480x1	
SCSI Controller, Connector	53C80 25 pin D	53C80 25 pin D	85C80 HDI-30	85C80 HDI-30	85C80 HDI-30	85C80 HDI-30	Curio HDI-30
Floppy Controller	SWIM	SWIM	SWIM	SWIM	SWIM	SWIM	New Age
Serial Ports	4 MHz 85C30 SCC	4 MHz 85C30 SCC	4 MHz 85C80 Combo	8 MHz 85C80 Combo	8 MHz 85C80 Combo	8 MHz 85C80 Combo	8 MHz ????? Curio
Other Networking	Optional Modem: 103, 212A, V.21, V.22, V.22bis, V.23, V.25	Optional Modem: 103, 212A, V.21, V.22, V.22bis, V.23, V.25	Optional Modem: 103, 212A, V.21, V.22, V.22bis, V.23, V.25	Optional Modem: 103, 212A, V.21, V.22, V.22bis, V.23, V.25	Optional Modem: 103, 212A, V.21, V.22, V.22bis, V.23, V.25	Modem: Rockwell 9623	Built-in V.32 -> B103, Ethernet (MACE)
Sound Out - digital	ASC	ASC	ASC	Batman	Batman	MSC	Cathedral
Sound Out -analog	2 x SONY	2 x SONY	1 xSONY	DFAC, TDA7072, TLE2062	DFAC, TDA7072, TLE2062	MSC, DFAC, MC34119	SAC, Singer
Sound Input	-	-	-	DFAC	DFAC	MSC, DFAC	Cathedral,SAC, Singer
ADB	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	PG&E	PG&E
Real Time Clock, Parameter RAM	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	M50753 PMGRµ	PG&E	PG&E
VIA1	6525 VIA	6525 VIA	6525 VIA	6525 VIA	6525 VIA	MSC	BMC
VIA2	-	-	-	ORCA	ORCA	MSC	BMC
Hard Disk: Capacity	3.5" 1/3 height 40 MB	3.5" 1/3 height 40 MB	2.5" 20 MB	2.5" 20 MB	2.5" 40 MB	2.5" 20 MB	2.5" 20 MB
Battery Type, Life	SLA 5AH 10 Hrs	SLA 5 AH 5 Hrs	SLA 2.5AH 3 Hrs	SLA 2.8AH 3 Hrs	SLA 2.8AH 3 Hrs	Flexible PGE 3 Hrs	Flexible PGE 3 Hrs
Volume, Weight	912 cu in ? lbs	912 cu in 15.5 lbs	169 cu in 5.5 lbs	220 cu in 6.7 lbs	220 cu in 6.7 lbs	100 cu in 4lbs	85 cu in 3.3 lbs

76

Table 1.3 - Portable Macintoshes

	Mac II	Mac IIx	Mac IIcx	Mac IIci	Mac IIfx	Mac IIsi
Intro Date	3/87			9/89	3/90	10/90
Processor: FPU: MMU:	16 MHz 68020 68881 68851 (opt)	16 MHz 68030 68882 '030	16 MHz 68030 68882 '030	25 MHz 68030 68882 '030	40 MHz 68030 68882 '030	20 MHz 68030 68882 (opt) '030
Processor Direct Slot (PDS):	–	–	–	68030 cache slot	68030 direct	68030 direct
ICache: DCache: Ext. Cache:	256 – –	256 256 –	256 256 –	256 256 any size (opt)	256 256 32Kbytes	256 256 –
ROM	256 Kbyte	256 Kbyte	256 Kbyte	512 Kbyte	512 Kbyte	512 Kbyte
RAM - minimum, speed, chip sizes	1 Mbyte, 120 ns, 256K, 1M, 4M SIMM	1 Mbyte, 120 ns, 256K, 1M, 4M SIMM	1 Mbyte, 120 ns, 256K, 1M, 4M, 16M SIMM	1 Mbyte, 80 ns fast pg mode, 256K, 1M, 4M, 16M SIMM	1 Mbyte, 80 ns fast pg mode, 1M, 4M, 16M SIMM	1 Mbyte, 100 ns fast pg mode, 256K, 1M, 4M, 16M SIMM
Memory/Video Control	GLUE	GLUE	GLUE	MDU	FMC	MDU
Built-in Video	–	–	–	RBV, 512x384x1,2,4,8 640x480x1,2,4,8 640x870x1,2,4	–	RBV, 512x384x1,2,4,8 640x480x1,2,4,8 640x870x1,2,4
I/O Buffering	–	–	–	–	Buffers	–
I/O Coprocessor	–	–	–	–	PICs for SCC, SWIM	–
# NuBus Slots, NuBus Controller	6, NUCHIP	6, NUCHIP	3, NUCHIP	3, NUBUS30	6, BIU2, BIU30	1 (optional)
SCSI Controller, Connector	53C80 DB-25	53C80 DB-25	53C80 DB-25	53C80 DB-25	SCSI DMA DB-25	85C80 DB-25
Floppy Controller	IWM	SWIM	SWIM	SWIM	SWIM/PIC	SWIM
Serial Ports	4 MHz 8530 SCC	4 MHz 8530 SCC	4 MHz 8530 SCC	8 MHz 8530 SCC	8 MHz 8530 SCC, PIC	8 MHz 85C80 Combo
Other Networking	–	–	–	–	–	–
Sound Out - digital	ASC	ASC	ASC	ASC	ASC	ASC
Sound Out -analog	2 x SONY	2 x SONY	2 x SONY	2 x SONY	2 x SONY	2 x SONY
Sound Input	–	–	–	–	–	Discrete A/D + PALs
ADB	ADB chip, 6523 VIA	ADB chip, 6523 VIA	ADB chip, 6523 VIA	ADB chip, 6523 VIA	SWIM - PIC	Egret (68HC05)
Real Time Clock, Parameter RAM	RTC	RTC	RTC	RTC	RTC	ADBμ (68HC05)
VIA1	6523 VIA	6523 VIA	6523 VIA	6523 VIA	6523 VIA	6523 VIA
VIA2	6523 VIA	6523 VIA	6523 VIA	RBV	OSS	RBV
Hard Disk	5.25" 1/2 height 50-pin	5.25" 1/2 height 50-pin	3.5" 1/2 height 50-pin	3.5" 1/2 height 50-pin	5.25" 1/2 height 50-pin	3.5" 1/3 height 50-pin

76

Table 1.4 – High-End Macintoshes

	Eclipse	Spike	Buccaneer	Oceanic	Cyclone
Intro Date					
Processor: FPU: MMU:	33 MHz 68040 '040 '040	25 MHz 68040 '040 '040	40 MHz 68030 68882(?) '030	25 MHz 68040 '040 '040	33 MHz 68040 '040 '040
Processor Direct Slot (PDS):	68040 direct	68040 direct	68030 direct	68040 direct(?)	–
ICache: DCache: Ext. Cache:	4Kbytes 4Kbytes –	4Kbytes 4Kbytes –	256 bytes 256 bytes 8 Kbytes	4Kbytes 4Kbytes –	4Kbytes 4Kbytes –
ROM	2 Mbyte	2 Mbyte	2 Mbyte	2 Mbyte	2 Mbyte
RAM - minimum, speed, chip sizes	4 Mbyte, 80 ns fast pg mode, 1M, 4M, 16M SIMM	4 Mbyte, 80 ns fast pg mode, 1M, 4M, 16M SIMM	4 Mbyte, 80 ns fast pg mode, 1M, 4M, 16M SIMM	4 Mbyte, 80 ns fast pg mode, 1M, 4M, 16M SIMM	4 Mbyte, 60 ns fast pg mode, 1M, 4M, 16M SIMM
Memory Controller	Orwell	Orwell	MDU	MMC	MMC
Built-In Video	DAFB, 512x384x1,2,4,8,32 640x480x1,2,4,8,32 640x870x1,2,4,8 832x624x1,2,4,8,32 1152x870x1,2,4,8	DAFB, 512x384x1,2,4,8,32 640x480x1,2,4,8,32 640x870x1,2,4,8 832x624x1,2,4,8,32 1152x870x1,2,4,8	RBV, 512x384x1,2,4,8 640x480x1,2,4,8 640x870x1,2,4	MMC, 512x384x1,2,4,8,16 640x480x1,2,4,8,16 832x624x1,2,4,8,16 640x870x1,2,4,8 1152x870x1,2,4,8	MMC, 512x384x1,2,4,8,16 640x480x1,2,4,8,16 832x624x1,2,4,8,16 640x870x1,2,4,8 1152x870x1,2,4,8
I/O Buffering	JDB, Relayer	JDB, Relayer	–	PSC	PSC
I/O Coprocessor	PICs for SCC, SWIM	–	–	PSC DMA	PSC DMA
# NuBus Slots, NuBus Controller	5, yancc	2, yancc	1 (opt), NuChip30	1 (opt), MUNI	2, MUNI
SCSI Controller, Connector	ESP236, 53C96 DB-25	53C96 DB-25	85C80 DB-25	53C94 (Curio) DB-25	53C94 (Curio) DB-25
Floppy Controller	SWIM	SWIM	SWIM	SWIM	New Age
Serial Ports	8530 SCC, PIC	8530 SCC	8 MHz 85C80 SCC	8 MHz Curio	8 MHz Curio
Other Networking	Ethernet (SONIC)	Ethernet (SONIC)	Ethernet (SONIC	Ethernet (MACE)	Ethernet (MACE)
Sound Out - digital	BATMAN	BATMAN	BATMAN	Cathedral (opt), <V8 equiv>, Singer	Cathedral, Singer
Sound Out - analog	TDA1543 A/D, SPORTY, TDA7056	TDA1543 A/D, SPORTY, TDA7056	Sporty	SAC	ExLax
Sound Input	DFAC, BATMAN	DFAC, BATMAN	DFAC, BATMAN	Cathedral (opt)/ Singer	Cathedral/ Singer
ADB	SWIM - PIC	ADB Chip, 6523 VIA	Egret (68HC05)	Cuda (68HC05)	Cuda (68HC05)
Real Time Clock, Parameter RAM	Egret (68HC05)	Egret (68HC05)	Egret (68HC05)	Cuda (68HC05)	Cuda (68HC05)
VIA1	6523 VIA	6523 VIA	6523 VIA	PSC	PSC
VIA2	6523 VIA	6523 VIA	RBV	PSC	PSC
Hard Disk	5.25" full height 50-pin	3.5" 1/3 height 50-pin	3.5" 1/3 height 50-pin	3.5" 1/2 height 50-pin	3.5" 1/2 height 50-pin

76

Table 1.5 – High-End Macintoshes (cont.)

2. PROCESSOR

Microprocessor

The Macintosh is based on the Motorola 68000 microprocessor family. Table 1 summarizes the CPU types, speeds, and related support chips for the entire Macintosh product line.

The Macintosh takes advantage of the 68000 features, including rich instruction set, many addressing modes, and seamless address space. Full hardware and software specs for the Motorola processors are given in the following Motorola publications:

General:	MC68000PM/AD, M86000 Programmer's Reference Manual
68000:	M68000UM/AD MC68000 User's Manual , 1989.
68020:	MC68020 32-bit Microprocessor User's Manual, 2nd Ed., 1985
68030:	MC68030UM/AD MC68030 User's Manual, 1987
68040:	MC68040UM/AD, MC68040 User's Manual, 1989
	MC68040DH/AD, MC68040 Designer's Handbook, 1989

Programmers Note: To provide downward compatibility with the original 68000 processor, not all the modes and instructions of the 68020 and 68030 can be used. These, and other compatibility issues are covered in Inside Macintosh, Vol. V, pp. 2-5. Other good techniques to insure compatibility between machines are covered in Macintosh Technical Note #117: *Compatibility: Why & How.*

Hardware Designer's Note: The 68000 TAS (Test and Set) instruction requires an indivisible read-modify-write memory cycle. On early Macintoshes, this hardware capability was not supplied, but, starting with the Mac SE machine, is has been, and should continue to be supported on new designs.

76

Bus Interfaces

Various bus interfaces are supported by the 68000-family processors. The following table summarizes the differences:

	Min. mem.	Bus	Dynamic	Data	Burst	Coherency
Processor	clocks	Width	Bus Sizing	alignment	Mode	mechanism
68000	4 (async)	16	no	word/long word	no	N/A
68020	3 (async)	32	yes	any	no	none
68030	2(sync)/3(async)	32	yes	any	cache fills*	none
68040	2 (sync)	32	no	any	MOVE16, cache fills, cache writes	bus snooping

* – Cache fill bursts not supported on Mac II, IIx, SE/30

All instructions must be word-aligned. On the 68020 and 68030, dynamic bus sizing permits 8 or 16 bit peripheral devices to be accessed on arbitrary byte or word boundaries. **Hardware Designer's Note:** Since the 68000 and 68040 do not support dynamic bus sizing, addresses of

I/O registers should be long-word aligned. All processors, except the 68000, permit data (operands) to be on any byte boundary, however, words or long words that are not aligned to word or long word boundaries (respectively) will be transferred with multiple memory cycles.

The 68030 and 68040 permit 16-byte burst transfers. However, the Mac IIx, IIcx, and SE/30 utilize the 68020 memory controller, so cannot run these burst transfers. The Mac IIci, Mac IIfx, and Mac IIsi and other new '030 machines can run burst reads for cache fills. Because memory on the NuBus cannot be guaranteed to be coherent with the on-chip cache, NuBus memory is not cached, thus no burst reads are done from the NuBus to the 68030.

The 68040 can be made to watch the bus for any transfers to cached memory space. If data in the caches are affected, the 68040 supplies correct data (for reads) and either feeds the cache new data or marks the cache line invalid (for writes). This bus snooping allows external bus masters to deal with cached data and maintain cache coherency. **Hardware Designer's Note:** For bus snooping to work, the bus cycles must be visible to the 68040 pins. Beware of local transfers on buffered busses. Bus snooping causes a considerable speed penalty. Giving other masters non-cached buffers is an alternative.

The 680XX family of processors use "Big-Endian" byte ordering. This means that in a word or long word, the most significant byte is at the lowest address. Be careful when interfacing with "Little-Endian" machines (MSbyte at the highest address) that the byte orders are correctly translated. The documentation standard for 680XX processors is that bits are numbered with 0 = LSB, and that data and addresses are in binary or hexadecimal.

76

Virtual Memory Management

All 68030- and 68040-based computers provide a built-in Memory Management Unit (MMU). The Mac II can use the optional Motorola 68851 PMMU chip to provide the MMU function. The MMU supports the use of virtual memory by mapping virtual to physical addresses using translation tables stored in memory. The MMU keeps address mappings in an address translation cache. Virtual memory is used by AU/X, Blue (7.0 and later), and Pink software. In machines using the MDU chip (Mac IIci, Mac IIsi), the MMU is also used to make RAM contiguous, as seen by the processor. See the MC68030 Enhanced 32-bit Microprocessor User's Manual for details of the MMU functionality, and the Atlantic and Pacific Theory of Operation for information on how Mac IIci uses the MMU to implement its memory mapping. A detailed analysis of virtual memory on the Macintosh is given in the PAR Technical Note # 5, *Virtual Memory Implementations at Apple*, by Henry Kannapell, the highlights of which are given below.

There are currently three MMUs in use by the Macintosh family: the 68851, the 68030 MMU and the 68040 MMU. Of these, the 68040's MMU has the least capabilities, so sets the floor that system software is coding to. Since the 68040 does not have early termination, limits, or initial shift, all active page tables have to be completely filled out. The 68030, which has these features, can have a smaller page table. In a typical example, the 68040's tables take 60% more space. On single tasking systems, such as System 7.0 VM, this is not much of a problem. However, on multitasking systems such as Pink or A/UX, each process requires the larger page tables, resulting in higher RAM usage. The characteristics of the different Motorola MMUs are given in Table 2.1, and the MMU requirements of the different Macintosh OSs are given in Table 2.2.

	68851	68030	68040	68050 **	88110
Address Width	32	32	32	32	32
Supervisor Root Pointer	Yes	Yes	Yes	Yes	Yes
Early Termination	Yes	Yes	No	No	No
Variable page hierarchy	Yes	Yes	No	No	No
Number of page levels	variable	variable	3	3	2
Virtual address table indexes	variable	variable	7-7-5 (8K) 7-7-6 (4K)	7-7-5 (8K) 7-7-6 (4K)	10-10
Address Translation Cache	64 entry fully associative	22 entry fully associative	64 entry 4 way set associative	64 entry 4 way set associative	56 entry fully associative
Cache Inhibit	Yes	Yes	Yes *	Yes *	Yes *
Indirect Page Descriptors	Yes	Yes	Yes	Yes	Yes
Limits	Yes	Yes	No	No	No
Initial Shift	Yes	Yes	No	No	No
Global Bit	Yes	Yes	Yes	Yes	Yes
Transparent Translation	Yes (Early Termination)	Yes (Early Termination)	Yes	Yes	Yes

* Cache control ~~is 2 lines~~

Table 2.1. Motorola MMUs

	Pre 7.0	System 7.0 VM	A/UX 2.0	Pink
I/O Translation		------		
Address Translation Cache				
Fully Associate Cache		+	+	+
Supervisor Root Pointer		------		
Early Termination		+(1)	+(1)	+(1)
Page levels (min)		2+Initial Shift	3	2
Page size	32K	4K	4K	512/4K
Cache Inhibit				
Used bits				
Modified bits				
Valid descriptor Bits				
R/W Protection		------		
Supervisor Protection		------		
Global Bit (not flushed)		------	------	------
Indirect Page Descriptors		------		
Limits		------ (1)	------ (1)	------ (1)
Initial Shift		------ (1)	------	------
Transparent Translation		+ (Can be Early implemented with Termination)	+ (Can be implemented with Early Termination)	(Can be implemented with Early Termination)
Ability to modify unused bits in Page tables				
Coprocessor Interface		------	------	------

 = required + = desirable ------ = non-essential

Notes: (1) Even though Limits, Early Termination, and Initial Shift are not required, they can greatly reduce page table size, which could be significant for high volume machines.

Table 2.2. MMU Requirements

Device drivers are affected by the virtual memory implementation. It is possible for I/O processing to span page faults and disk access times without careful planning. The result can be an unbounded period where interrupts are disabled, which could adversely affect real-time devices. Only A/UX has a solid solution to this problem, through the use of re-entrant device drivers. Pink separates the device controllers form the User tasks (which is not possible in System 7.0 VM), but can still suffer from I/O handling spanning page faults, although interrupt handling does not span page faults.

All virtual memory systems require that I/O be mapped into User space, primarily due to the requirements of accessing graphics video buffers. A/UX and System 7.0 VM require access to the ROM as well.

Floating Point Assistance

All current and future mid-range and higher machines have hardware floating point assistance. The Mac II uses the Motorola 68881, and all 68030-based machines use the 68882. A reduced set of floating point operators is built into the 68040. Other floating point operations on the 68881/2 are emulated in software. The 68882 coprocessor implements the ANSI-IEEE 754-1985 floating point standard in a 96-bit extended precision format, as well as large integer and transcendental functions. It is upwardly compatible with the 68881 coprocessor used in the Mac II. Normally, the 68881/2 is used in conjunction with the Apple SANE floating point routines, which, among other things, converts the 96-bit 68881/2 format to the SANE 80-bit format, alters boundary conditions, extends the accuracy of transcendental functions, and adds several functions that do not exist on the 68881/2.

76

Programmers have the choice of using the SANE routines, which work on all Macintoshes, or accessing the 68881/2 directly. The disadvantages of direct access include the need to have separate code for 68881/68882 machines and the need to convert numeric formats to and from SANE formats. The advantage, though, is a performance improvement of from 5 to 50 times over the SANE routines **Warning:** Programs that access the 68881 or 68882 directly will not run on Macs using the 68000 processor or machines such as the Mac LC and IIsi that may not have an optional FPU board installed. The definition of a Mac II does not include always having an FPU!

The hardware and software specifications of the 68882 chip are in the MC68881/MC68882 *Floating-Point Coprocessor's User's Manual,* Motorola Inc, 1987. Macintosh Technical Note #236: *Speedy the Math Coprocessor* summarizes the Motorola manual. Macintosh Technical Note #235: *Cooperating with the Coprocessor,* explains handling interrupts and exceptions from the floating point unit. SANE's definition of its interactions with the floating point coprocessor is in the *Apple Numeric Manual,* 2nd Edition.

The 68040 CPU has built-in floating point hardware for commonly-used operations such as FADD, FMUL, FCMP, FMOVE, etc. Other floating point instructions, such as the transcendentals, trap to software routines.

Interrupts

The 68000 family of microprocessors supports 7 levels of prioritized interrupts. When an interrupt occurs, the processor reads the interrupt flag register, which indicates the source of the interrupt. In some cases, additional registers need to be read to select the exact interrupt source. The priority level and interrupt flag register bit assignment is dependent on the processor hardware, but, generally a similar format is followed in each class of machine to enhance software compatibility. Details of the interrupt level assignments and the registers needed to determine the interrupt sources are given in the ERSs for each processor. A general outline of interrupt priorities is given in figure 2-1 to summarize differences. Interrupts in Macintoshes are always auto-vectored; interrupting devices do not respond to IACK cycles generated by the processor.

In all machines except the high-end machines (Mac IIfx and Eclipse), the interrupt priorities are fixed. In the Mac IIfx, any one of 16 possible interrupts can be assigned any priority. See the spec for *Operating System Support* by Steven Ray for details on Mac IIfx interrupts. In practice, different priorities will likely be used in Blue, Pink, and AU/X software environments. On Eclipse, two sets of priorities are provided: Blue and AU/X, as noted in the table above.

Note: To handle some real-time events, interrupts below a certain level are temporarily disabled. There can be strong interactions between real-time events due to this, and with the traditional Mac I/O architecture (without PICs or equivalent) multiple real-time events are difficult to handle. Time-critical events include:

- Incoming and outgoing Appletalk packets to the SCC
- Formatting a disk with the IWM or SWIM
- Generating sound output
- Buffering sound input
- Updating screen position of pointer

76

Macintosh Technical Note #221: *NuBus Interrupt Latency (I was a teenage DMA Junkie)* summarizes some of the non-real-time aspects of the Macintosh, from the point of view of NuBus card developers.

In system 7.0 running virtual memory, page fault interrupts can stack up, causing indeterminate periods where lower-level interrupts are masked. Interrupt handling is not completed until after the Completion Routine, if present, is executed. The interrupt routines generally use only locked-down memory, but the Completion Routines can access any part of memory thus page faults could result.

Macintosh	680XX Interrupt Level							
	0	1	2	3	4	5	6	7
Mac Plus	None	VIA	SCC	SCC+ VIA	Sw	Sw+VIA	Sw+SCC	Sw+VIA+SCC
Mac SE, Classic	↓	VIA/SCSI				Sw+ VIA/SCSI		Sw+SCC+ VIA/SCSI
Mac II, IIx, IIcx, SE/30	↓	VIA1	VIA2	None	SCC	None	Pwr Sw	Sw
Mac IIci, Tim	↓	↓	VIA 2 in RBV	↓	↓	↓	↓	NMI+ Parity
Mac IIsi	↓	↓		↓	↓	↓	None	68HC05 NMI
Portable, Aruba, Ashai	↓	VIA+ASC	SCC	↓	← Diagnostics →			NMI
Mac IIfx	↓	← Programmable →						
Mac LC	↓	VIA 1 in VISA	VIA 2 in VISA	None	SCC	None	None	68HC05 NMI
Eclipse, Spike Blue	↓	VIA1	VIA2	Software	↓	None	None	NMI, Parity, Yancc
Eclipse, Spike AU/X	↓	None		Ethernet	↓	Sound	VIA1	

clock

Sw = Programmer's Switch

Figure 2-1 Macintosh Interrupt Priorities

76

Cache

All Macintosh microprocessors from the 68020 onwards have some form of cache memory to speed up execution time:

68020 (Mac II, LC) - 256 byte direct-mapped instruction cache, organized as 64 long-word entries.

68030 (Mac IIx, IIcx, IIci, SE/30, Mac IIfx, IIsi, ...) - 256 byte direct-mapped instruction cache, 256 byte direct-mapped data cache, both organized as 16 lines with 4 long-word entries per line. The data cache is a write-through cache. The Mac II, IIx, and IIcx use the single entry mode to fill the caches. The Mac IIci and Mac IIfx allow the use of burst reads to fill the cache.

68040 (Eclipse, Spike, Cyclone) - 4 Kbyte instruction cache, 4 Kbyte data cache. Both are organized as four-way set associative caches with 64 sets of four, 16 byte lines. Eclipse supports both write through and copyback modes of updating write and also supports the bus snooping mode to insure cache coherency when other bus masters write to memory.

In addition to the 68030's built-in cache, Mac IIfx has an on-board direct mapped 32 Kbyte cache for both data and instructions. It is a write-through cache, with no wait states for reads or writes. This cache is closely associated with the DRAM array and provides no opportunities for cache coherency problems.

The Mac IIci has provision for a plug-in cache card. The amount of cache and tag memory is limited by how much can reasonably fit on the card. As with any cache external to the MDU, it is direct-mapped and caches only physical memory. Normally the cache is enabled by having

76

the CACHE signal active until a miss or write takes place, at which time, normal memory cycles are run.

Usage Note: A IIci's cache card's organization of data and tag memory is determined only by the card. System software will not make any assumptions about the cache card's organization, and only the card's diagnostic software should directly access cache card RAM.

Programmer's Note: Caches are organized on long word boundaries. On machines with burst reads or writes, programs will run most efficiently if data structures are organized so that items are on long-word or 4 long-word boundaries, where possible.

Usage Note: On 68030-based machines, it is possible for other Bus-Masters to write into system memory that is already cached on the processor chip. Macintosh Technical Note #261: *Cache as Cache Can*, explains ways of insuring cache coherency on these machines.

76

3. MEMORY

All read/write memory, read-only memory, and I/O device registers are mapped into the 680XX memory space. With the exception of memory on NuBus cards, parameter RAM, and specialized buffers, the Macintosh read/write memory is generally contained in DRAM or SRAM modules, which are either socketed or soldered to the mother board. The Macintosh Toolbox routines and various system traps are contained within Read-Only Memory (ROM) or in patches to the ROM, which exist in main memory. The Macintosh RAM and ROM capabilities are described in the next two sections, followed by sections outlining memory mapping.

RAM

The RAM in Macintoshes is implemented by either soldered-in chips, plug-in SIMMs, or through plug-in RAM expansion cards. Tables 1.1 through 1.5 in the Introduction give the following information on RAM for each Macintosh:

1. The minimum installed memory size
2. The maximum allowable memory chip access time
3. The size of RAM chips supported

Note: The Mac II, IIx, IIcx, and SE/30 do not support 4 Mbit DRAMs, since refresh cycles on these machines can put these DRAMs into a test mode.

On the Mac IIci, Mac IIfx and Eclipse, parity is an option available to provide a data integrity check over the DRAM. On these machines, installation of 9-bit SIMMs is required. On the Mac IIci and Mac IIfx, a PGC (Parity Generator Chip) must also be installed at the factory. When a parity error is detected, an interrupt is generated and the processor then polls a VIA bit to find out the cause of the interrupt. On the Mac IIci and Mac IIfx, there is no time penalty when parity is used. In Eclipse, 0, 1, or 2 clock wait states are inserted when parity is enabled, depending on the type of memory cycle run.

On machines with DRAM, refresh cycles are automatically generated by the memory controller chip. The refresh cycles are arranged so that they do not delay video access cycles (if applicable), and have minimum interference with normal CPU accesses.

All Macintosh RAM subsystems are capable of supporting single transfer read and writes of the sizes supported by the processor (8 and 16-bit transfers for the 68000, and 8, 16, and 32-bit transfers for the 68020, 68030, and 68040). 68030 processors are capable of doing 4-long-word bursts on instruction cache fills, however, computers based on the the GLUE chip (Mac II, IIx, IIcx, SE/30), are incapable of running these burst transfers. 68040 processors can run 4-long-word burst reads and writes for cache line reads/pushes and during the MOVE16 instruction.

RAM can also be expanded on NuBus cards, in systems with the NuBus. However, this expansion memory is handled differently: the memory is mapped into NuBus space, and thus is not contiguous with the main on-board memory. **Software Designer's Note:** Page tables and DMA-accessible RAM must be part of the on-board main memory, so that bus errors are handled correctly.

ROM

ROM in a personal computer usually provides enough software to boot up the system by pulling a more complete version of the operating system off of a disk. In the Macintosh, significantly more is included in the ROM, which assists in Apple maintaining copyright control over the Macintosh software. In cost-sensitive systems, putting more system functions in ROM frees up the more-expensive RAM. The Macintosh ROM sizes and types are summarized in Tables 1.1 through 1.5.

Most new machines have provision for ROM expansion or replacement. The Mac LC has a 42-pin ROM socket available. The portable provides a 50-pin connector for a ROM expansion board. The Mac SE/30, IIx, IIcx, IIci, and Mac IIfx have 64-pin ROM SIMM modules, that can handle ROMs up to 8 Mbytes.

The ROM initially is mapped into low memory upon power up or restart, starting at address $00000, in order to allow the processor to access the cold start power-up vectors. Depending on the machine, clearing of the ROM overlay bit in a VIA, or simply accessing the ROM at its normal high address changes the ROM addressing back to its normal range in high memory.

The Portable's ROM expansion board is explained in Macintosh Technical Note #255: *Macintosh Portable ROM Expansion*.

Any writes to ROM must not produce a bus error. Writing to the ROM doesn't make sense, but it sometimes happens.

76

Architectural Memory Maps

(Note: most of the next three sections is derived from the PAR Technical Note #5, *Virtual Memory Implementations at Apple*, by Henry Kannapell).

The 68000-based Macintoshes, (Classic Mac, Mac SE, Portable, Aruba, Asahi) have 24 address lines, giving an address space of 16 Mbytes. The 68020, 68030, and 68040 have 32 address lines, allowing 4 gigabytes total address space. The memory space in both cases is split into regions covering RAM, ROM, I/O Devices, and NuBus slots, if used. On the 32-bit address machines, a mapping from 24 bit address space to 32-bit address space is available, in order to support programs written for the older Macs that are not "32-bit clean", i.e. use the upper address bits for purposes other than addressing.

The "Architectural Address Space" is the space seen by the microprocessor executing code. (In the previous version of this document, this was referred to as "logical" address space. The name has been changed to avoid confusion with virtual memory nomenclature.) In most Macintoshes expansion RAM is made contiguous with the installed RAM by direct hardware mapping. Address space allocated to RAM or ROM that is not completely filled by the ROM or RAM installed actually installed contains aliased images of the installed RAM or ROM. In these cases, the physical address space (as seen from the point of view of the address bus on the mother board) is the same as the architectural address space. However, on the Mac IIci, IIsi, and other machines that use the MMU to make expansion memory contiguous, the architectural address space is different from the physical space. Since this translation is actually a simple form of virtual memory, the MMU executes page walks through the page tables, and may occasionally have to bring in page table entries from memory, even if virtual memory is not running.

Note to NuBus and PDS (Processor Direct Slot) designers: In Macintoshes where the architectural addresses are different from the physical addresses, will result in bus masters other than the microprocessor, such as NuBus masters, seeing the raw physical address space, with discontiguous memory. Thus all references to memory by these add-on cards must be with physical addresses. Use Memory Manager calls for translation.

Figure 3.1 shows the generic architectural memory maps for the Macintosh in 24-bit mode and 32-bit mode. These are not the final hardware mapping seen by virtual memory systems, rather, this is the "Macintosh Programmer's Physical Model". Detailed memory maps are given in each processor's ERS.

The 24 bit mode mapping is the typical address space of the 68000-based Macs, as well as Mac II-class Macs set to the 24-bit mode. The bottom part of memory is RAM, the ROM starts at $80 0000, the Nubus Slots are allocated 1 Megabyte each from $9X XXXX (Slot 9) to $EX XXXX(Slot E). The final addresses at $FX XXXX reference I/O devices on the Macintosh motherboard. Note that in some machines, notably the Mac LC-style machines, the ROMs have been moved to the higher address of $A0 0000.

Since not all machines have six NuBus slots, and even if they did, not all six cards might be populated, the free 1 Megabyte pieces of address space are taken over by some of the virtual memory schemes and by video RAM.

Hardware Designer's Note: The 68020 bit manipulation instructions can read bytes before the video RAM space, so valid memory space must be provided before the video space. This problem has been solved in the past by either having the software set the pointer to the base of screen memory to the first place where video address space wraps, or by moving the base pointer 32 or 64 bytes up from the start of video space. Hardware must insure that this situation can be handled without errors.

The ROMs cause a special problem. As the ROMs exist in the middle of the architectural address space, they limit the amount of contiguous memory that the system can access. This is because the ROMs must exist at the same address in the logical address space as they do in the architectural address space. Since the ROMs are accessed in both User mode and Supervisor mode in Virtual memory systems, they must be visible to both address maps. Older ROMs have internal references to the architectural addresses; thus they cannot be moved. On the Mac LC and newer CPUs, the ROMs are position-independent.

In 32 bit mode, the address space between machines capable of virtual memory are fairly consistent. The ROMs always exist at $4XXX XXXX, and are conveniently aliased for several addresses to access the ROM.

Figure 3.2 shows how the Mac IIci and IIsi-class machines map their physical address space to both 24-bit and 32-bit architectural space. Note that the video buffer areas, which are actually resident in main RAM, is mapped up into the NuBus super slot $B so it can be treated like a NuBus Card.

Virtual Memory Address Mappings

The Supervisor Logical Address Space for the Macintosh is shown in figures 3.3, 3.5, and 3.6. In all virtual memory systems under consideration (A/UX, Pink, System 7.0 VM), there is only one Supervisor space; it is entered into through traps, interrupts, and other exceptions. This is in contrast to User space, in which there could be multiple User address spaces. The Supervisor space has access to all of the machine's I/O areas, and all of the physical RAM in the system.

The User Logical Address Space is significantly different for each of the different virtual memory systems. In the case of System 7.0 VM, the User and Supervisor address spaces are exactly the same (figure 3.6). There is currently no protection between the different modes, so it is possible for the User mode program to write over the supervisor space (similar to the operation of a non-VM Macintosh). Nothing in the user space is required to be locked into memory, unless it is being accessed by I/O devices. There is only one process in system 7.0 VM; it may be either supervisor or user space. The single user logical address space must be allocated to all user mode programs that are running. No two user programs can own the same logical address. Each program has its own stack, code, and heap in the machine. Since the memory is fragmented, however, the stack may be unable to grow downward even if there is memory available between the top of the heap and the bottom of the stack. This condition may happen when the stack is above the ROM ($90 0000 and greater) and the heap is below the ROM ($7F FFFF and less).

In the case of Pink, the User space is always from 16 Megabytes up to 4 Gigabytes (figure 3.5). Any write to the lower 16 Megabytes will result in an access violation. There can be any number of different User mode processes running, all of which can own any Logical address in their Logical address space. Pink goes one step further, and allows multiple tasks to run in a single virtual address space. This means that switching from tasks within a virtual address space does not require establishment of a new page table set nor flushing of the address translation cache. This can result in much faster task switching than is possible with a system such as Unix, that must establish a new virtual address space for each process switch.

Since Logical address 0 is not owned by the User process, any read or write access to this location because of a nil pointer will result in an access violation, which will help identify these errors. Pink allows a user to install a handler to process this type of exception.

In the case of A/UX 1.1, the user space was limited to 256 Megabytes starting at location $0000 0000 and going to location $0FFF FFFF (figure 3.4). In the standard Unix model, User processes do not access I/O at all, but only access memory. When it is necessary to change an I/O device, a system call is made, which changes the address space into the kernel. The kernel then affects the I/O system on behalf of the User process. The User stack starts at the top of memory, and the code (Text in Unix parlance) starts at the bottom.

In the case of A/UX 2.0, the user space is the entire 4 Gigabytes available (figure 3.4). The stack starts at the top, and the code starts at the bottom. Any number of User mode processes can run, all of which can own any logical address in its User logical address space.

A/UX has a special User mode process called StartMac that supports the Macintosh architectural address space in A/UX. The StartMac process memory is limited to 16 Megabytes in 32 bit mode, because some parts of the system software are not 32 bit clean. The StartMac process in

76

24 bit mode has the usual 8 Megabytes of memory in a 16 Megabyte total address space. The bottom addresses are mapped to RAM, the ROM is mapped in at the usual location, and the Nubus slots are mapped in at their expected locations. The Macintosh I/O ports are not mapped in, as they are owned by the Supervisor address space. StartMac starts as a normal Unix process, with its own Stack, Data, and BSS (uninitialized data area), and Text regions. It then creates a memory area to support the Macintosh memory model (which includes the stack and heap) and converts to it. Subsequent memory references are within the Macintosh architectural address space.

76

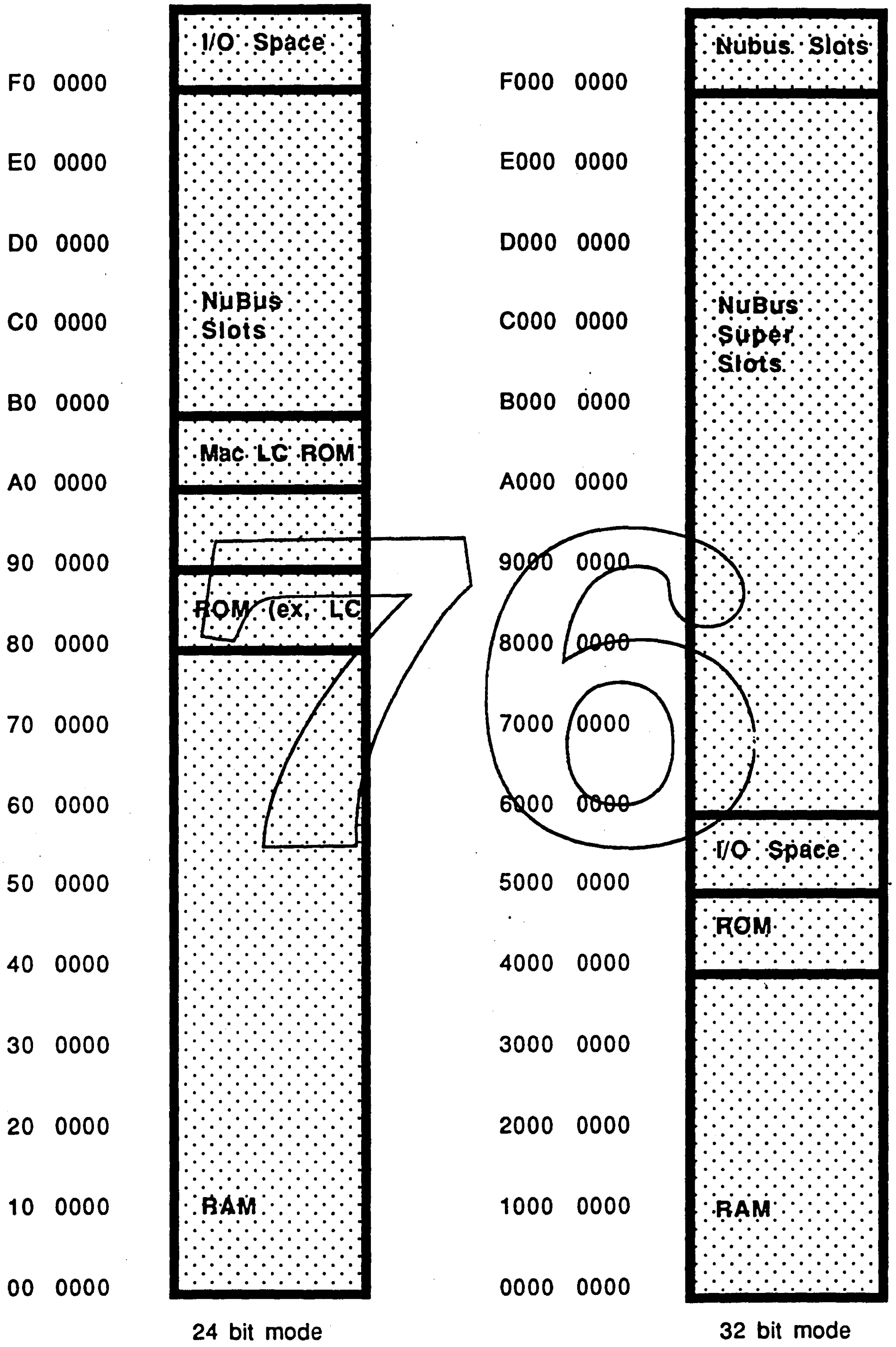


Figure 3.1 – Macintosh Architectural Address Space

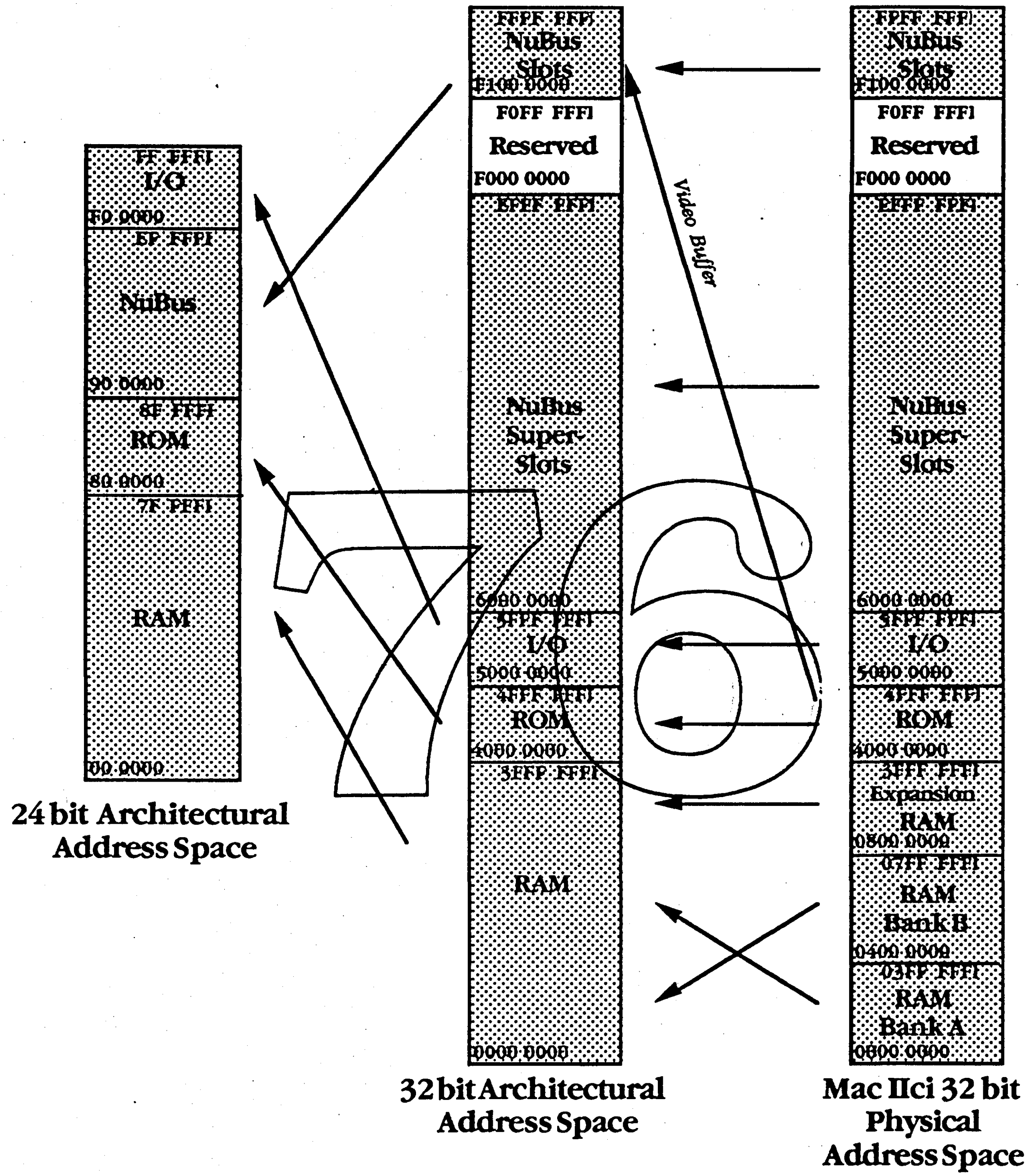


Figure 3.2 – Mac IIci, IIsi Physical to Architectural Address Mapping

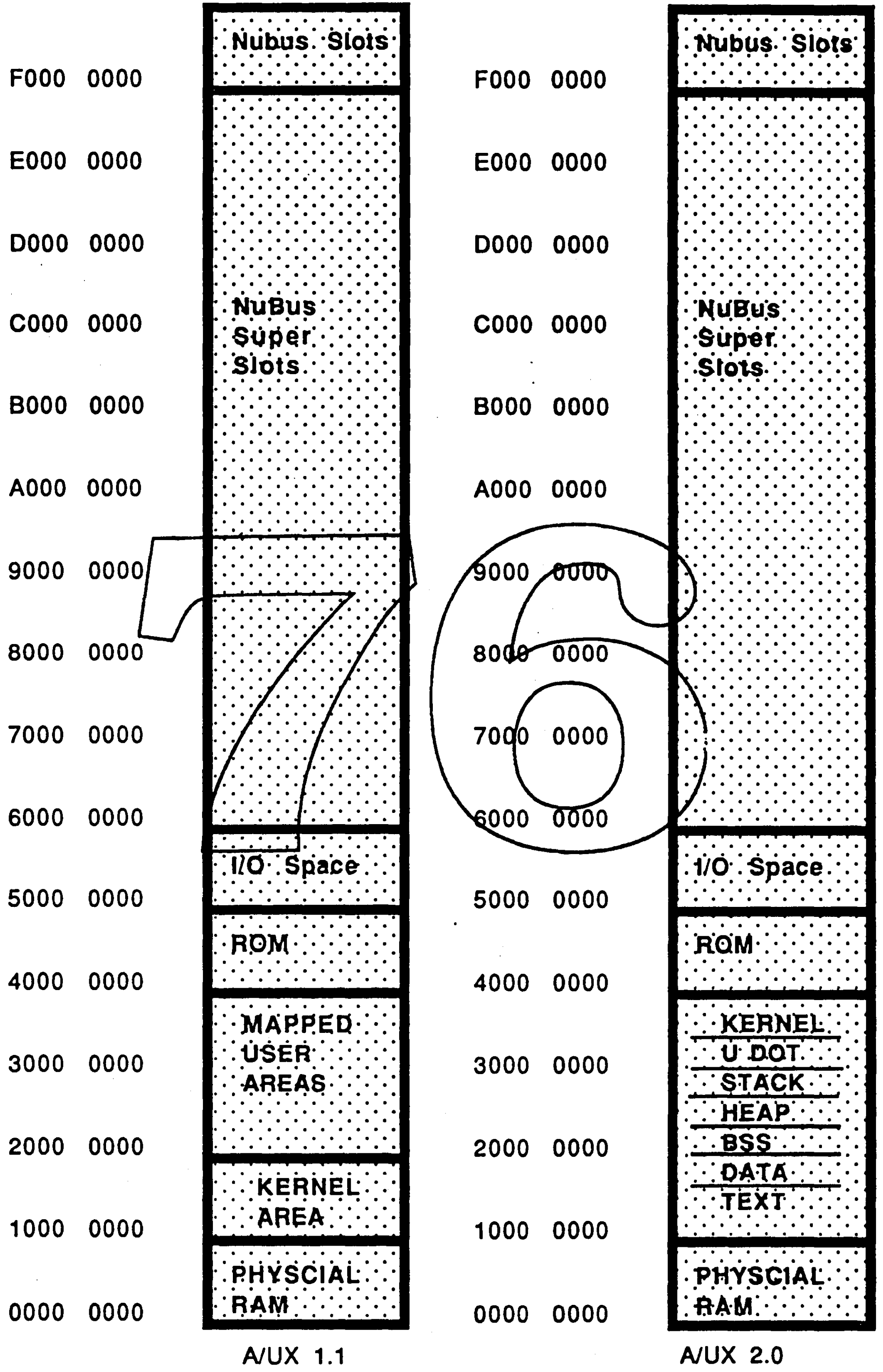


Figure 3.3 – A/UX Kernel (Supervisor) Address Space

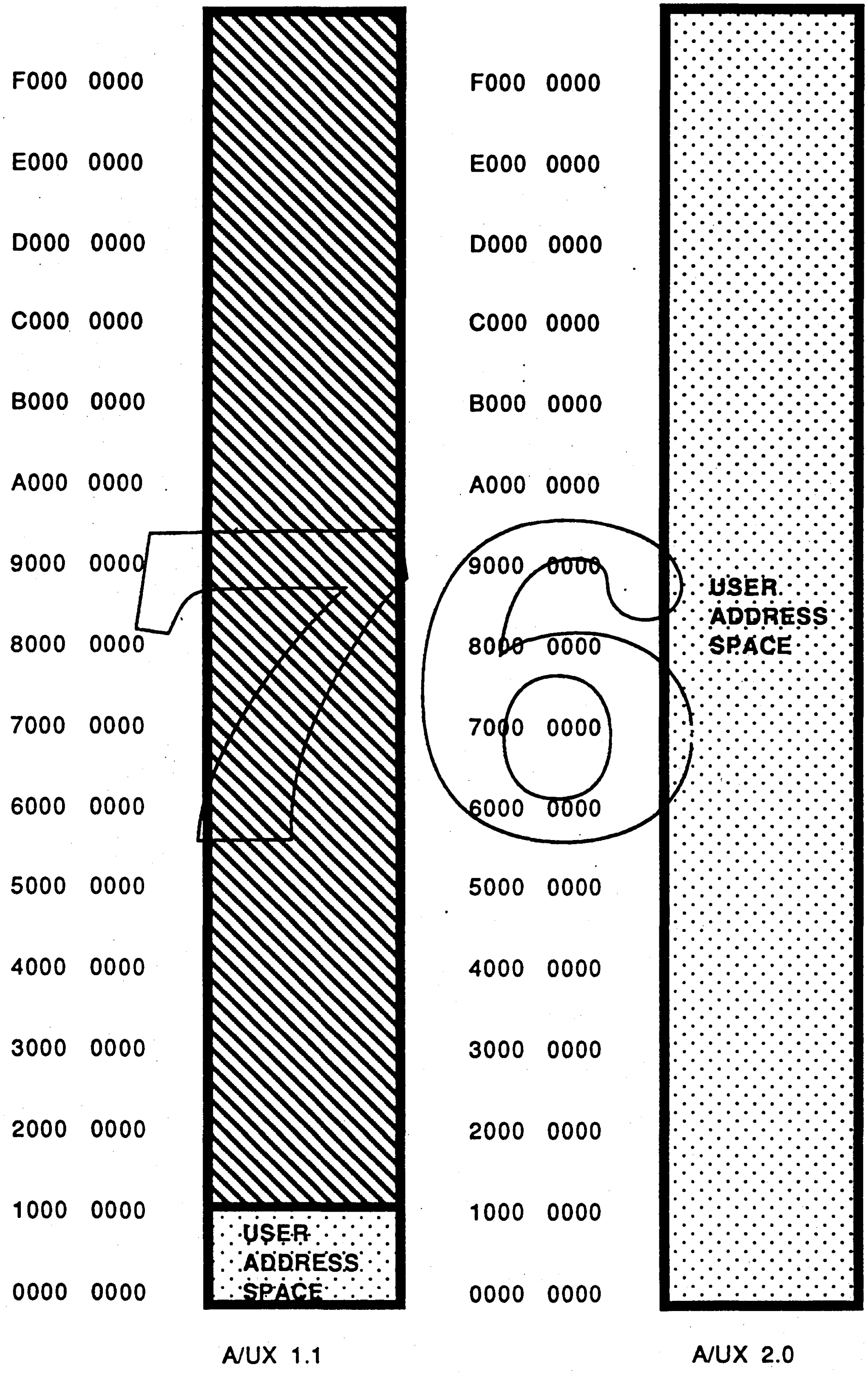
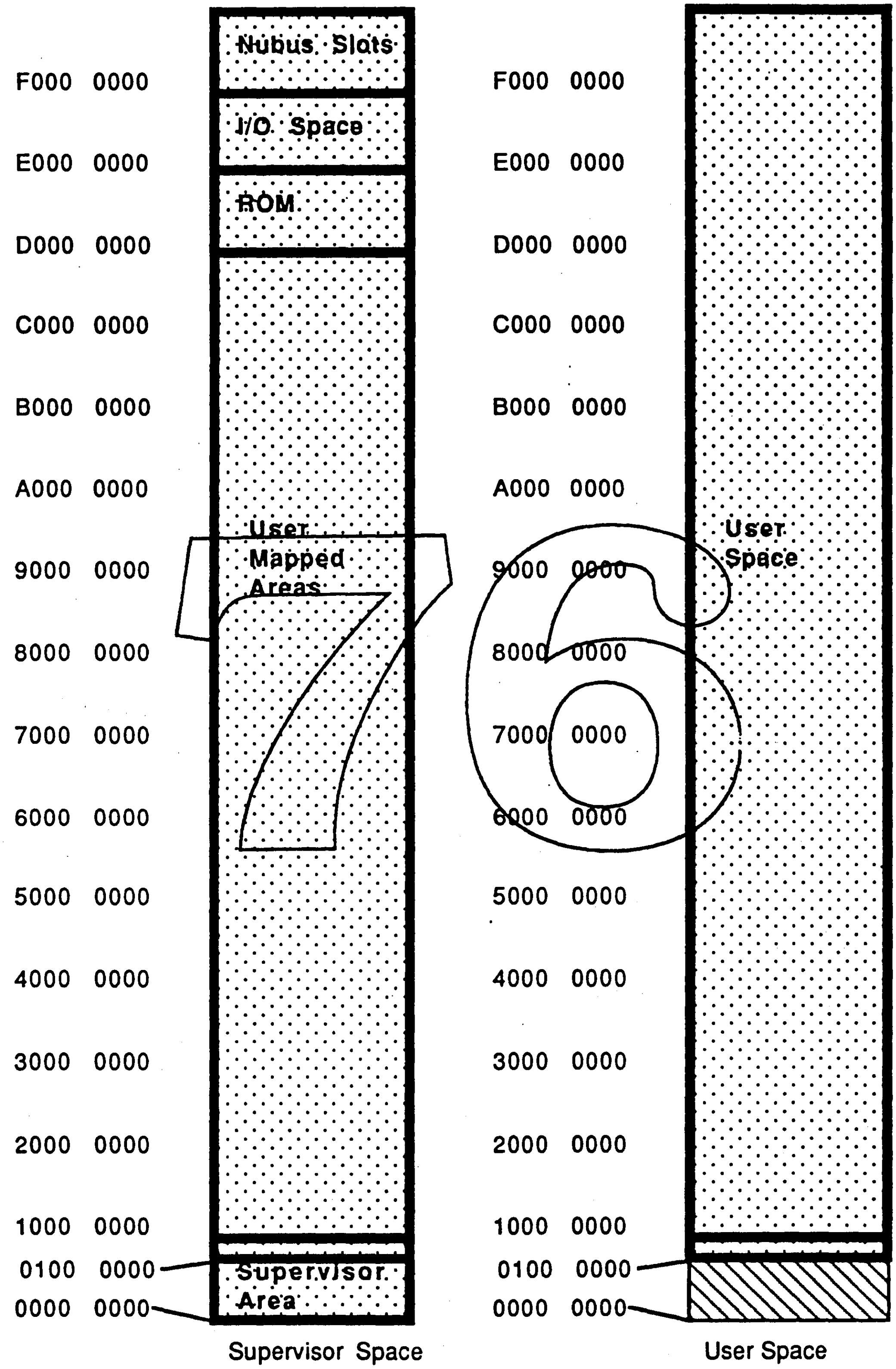


Figure 3.4 – A/UX User Address Spaces



76

Figure 3.5 – Pink Supervisor and User Address Space

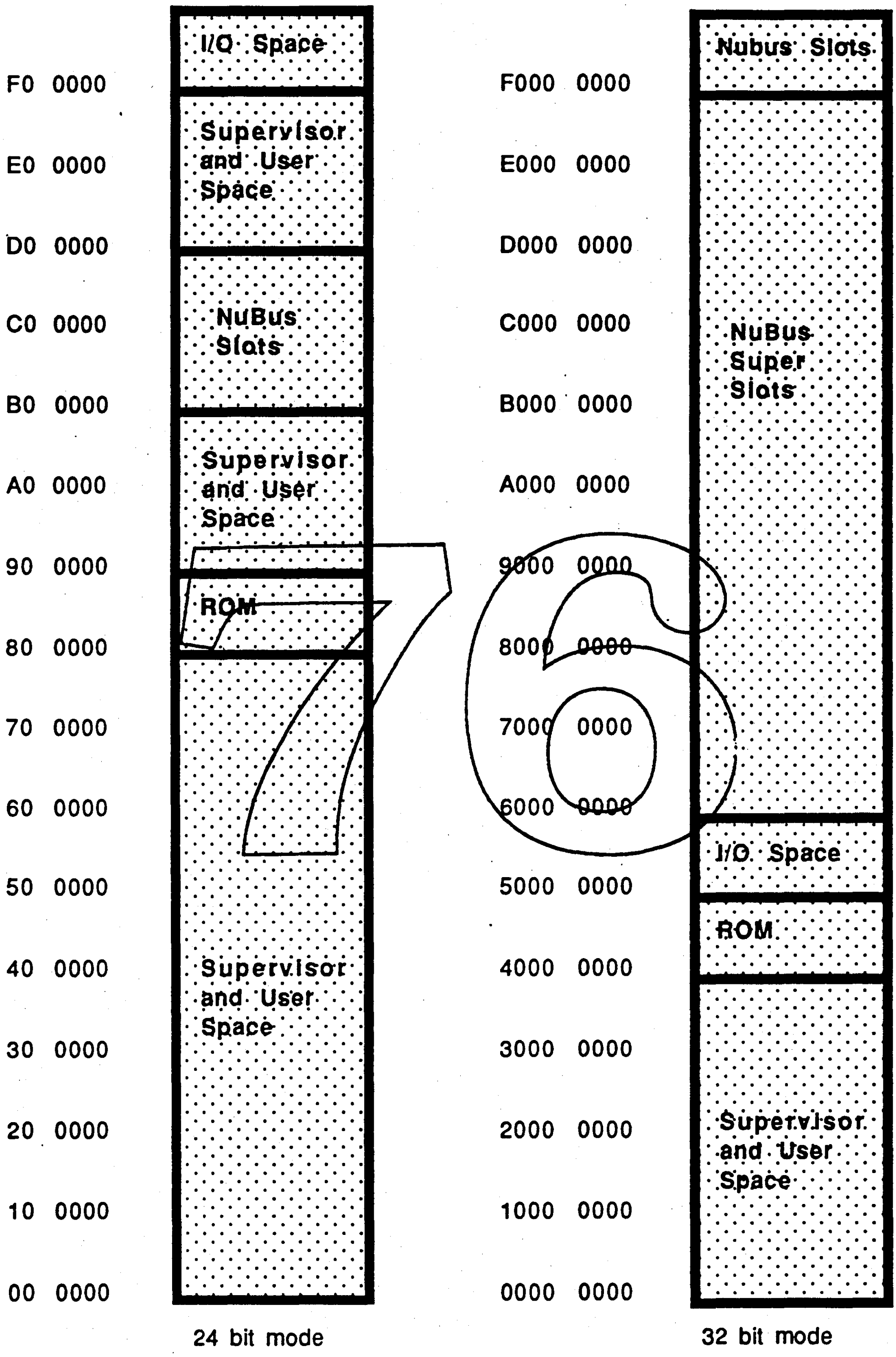


Figure 3.6 – System 7.0 VM Supervisor and User Address Space

4. SYSTEM EXPANSION

All Macintoshes, except the Classic Mac form factor machines (Mac Plus, Classic, Apollo), and most portables provide some form of system expansion by plug-in cards. Most Macintoshes provide a form of processor-direct expansion and the Mac II product line supports the NuBus bus architecture. Tables 1.1 through 1.5 summarize the PDS and NuBus slots available in the Macintoshes.

Processor Direct Slots

The direct processor expansion connectors (or Processor Direct Slots, PDS) bring out power, the microprocessor bus signals, and arbitration signals to a connector. Logic on the add-on board can take control of the bus (be a "bus-master"), and get access to RAM, ROM, and I/O devices. In the past, differences in the CPU, memory subsystem, and physical enclosure resulted in most direct expansion boards being unique to their respective CPUs. Where possible, PDS card form factors and interfaces are being standardized. Figure 4-1 shows the types of access cards have to the microprocessor, the overall card size, and power available to the card. **Hardware Designer's Note:** PDS cards can have adverse affects on EMI, signal quality, speed, etc. of the CPU. Macintosh motherboard designers need to carefully spec the allowable electrical loading of PDS cards.

The Mac IIci expansion slot was designed just to support an add-on cache, and as such, does not directly support a bus master on the card. However, it is possible to "trick" the bus into getting bus mastership, with a loss of performance, as was done on the Holy Grail demonstration card as part of the Cathedral DSP effort.

76

Details on PDS cards can be found in the ERS's of their respective machines. Additional information on the Portable PDS card is given in Macintosh Technical Note #254: *Macintosh Portable PDS Development.*

NuBus Interface

NuBus is a 32-bit bus that originated at MIT, then was subsequently developed at Western Digital Corp. and Texas Instruments. It allows multiple I/O or processor boards to be connected to the Mac II family of products. Figure 4-1 shows the number of NuBus slots available per Macintosh CPU. Tables 1.1 through 1.5 show the number of slots as well as the NuBus interface ASIC used.

All Macintoshes up through the Mac IIfx and IIsi conform to the original NuBus standard, as detailed by ANSI/IEEE standard 1196–1987. Details of the Apple implementation of the NuBus are given in *Designing Cards and Drivers for Macintosh II and Macintosh SE*, Addison-Wesley, 1987. Apple deviates from the ANSI/IEEE 1196-1987 standard in the following ways:

- - 5.2 V is not provided. All the -5.2V pins are connected together.
- Pins A2 and C2 are reserved in the 1196-1987 spec; they are grounded in the Mac II, IIx, and IIcx. They are open in the Mac IIci, IIsi and IIfx.

The dimensions in the original Mac II box provided more space around the NuBus cards and connector opening than stated in the NuBus specs. Hardware designers should be careful to not assume that the Mac II dimensions will be standardized. Macintosh Technical Note #234: *NuBus Physical Designs – Beware* discusses details of this.

76

Macintosh	NuBus Slots	Direct Expansion		
		Access	Board Size	Power
Mac Plus, Classic, Apollo	0	N/A	N/A	N/A
Mac SE	0	68000 direct	8.4"x4.7"	+5V@1.5A -5V@.1A +12V@.15A -12V@.1A 7 W max.
Mac II	6	N/A	N/A	N/A
Mac IIx	6	N/A	N/A	N/A
Mac SE/30	0	68030 direct	8.4"x4.7"	Same as Mac SE
Mac IIcx	3	N/A	N/A	N/A
Mac IIci	3	68030 cache	3"x6.1"	+5V@1A
Portable, Aruba	0	68000 direct	4.2"x2.8"	+5V@50ma +12V@25ma see Mac Tech Note #254
Mac IIfx	6	68030 direct	12.9" x 4.0"	+5V@2A -12V@.1A +12V@.175A
Mac IIsi, Oceanic, Buccaneer	1 w/adapter (with NuBus power,size)	68030 direct w/adapter	8.4"x4.7"	Same as Mac SE
Mac LC	0	68020 direct	approx. 3" x 7"	+5V@.8A -5V@.02A +12V@.2A
Columbia, Vail	0	68030 direct	approx. 3" x 7"	+5V@.8A -5V@.02A +12V@.2A
Eclipse	5	68040 direct	12.9" x 4.0"	TBD
Spike	2	68040 direct	12.9" x 4.0"	TBD

Figure 4-1 Macintosh Processor Expansion

Not all of the available features of the NuBus are supported by current Macintosh CPUs. Unsupported features include:

- Bus Parity
- Block Move operations to/from CPU and RAM. These will be partially supported by Eclipse and Spike (using yancc), and will be fully supported in Cyclone (using MUNI).
- Maximal transfer rates are not attained, due to insufficient buffering between the

processor and NuBus. This will be improved on machines using the yancc and MUNI controllers.

It should be noted that 68020 and 68030 microprocessors do not generate block move transactions over NuBus. However, block moves between NuBus cards can be supported by any NuBus Macintosh. **Hardware Designer's Note:** Any bus master in a system that expects to access NuBus memory space must be able to handle bus retry cycles.

Details on the NuBus Controllers can be found in the specifications for the following chips:

Mac CPU	Controller	Apple Part No.
Mac II, Mac IIx, IIcx	NUCHIP	344S0606
IIci, IIsi, Oceanic	NUCHIP30	344S1020
Mac IIfx	BIU2	344S0075
	BIU30	344S0074
Eclipse, Spike	yancc	TBD
Cyclone	MUNI	TBD

Details on NuBus power capabilities are given in *Designing Cards and Drivers for Macintosh II and Macintosh SE*, and Macintosh Technical Note 260: *NuBus Power Applications.*

NuBus'90

A higher performance version of NuBus (Nubus '90) is being defined by Apple, NuBus users and IEEE, and is currently in public review, until July 31, 1991, at which time it is expected to go to IEEE sponsor ballot. In addition to correcting minor typos and a timing error, it adds the following changes:

- The mechanical specifications in the 1987 document did not reflect industry standards for the PC-style board, and were updated to reflect current standards.
- The -5.2V lines are deleted from the PC-format cards. This freed 8 signal lines which are now used for the following optional features:
  a) A 2X block transfer protocol, allowing burst transfers approaching 80 Mbytes/sec. (new signals are: CLK2X*, CLK2XEN*, TM2*)
  b) A cache coherency protocol. (new signals are: CM<2...0>*, CBUSY*)
  c) Standby power. (STDBYPWR)
- The maximum number of boards in PC-style systems was reduced to 8, with corresponding changes to the terminator values.

* – The NuBus'90 draft specification (P1196R/D1.3) is available from:

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1264
800/272-6657

Copies of the specification will be available between December 31, 1990, and July 1, 1991. The IEEE/CS price is $20 and the non-member price is $25.

- Defined the formerly reserved lines A2 and C2 as a serial bus using the protocol defined by the proposed IEEE P1394 specification. (See the Chefcat section for more details).

Computers using the yancc and MUNI NuBus controllers do not support the 2X block transfers between the motherboard and cards, but they do supply a CLK2X* signal to allow 2X block transfers between cards. In order to protect the 2X clock driver from old cards that inadvertently shorted the -5.2V lines together, the presence of shorted lines is sensed by the CLK2XEN* line. If shorted, the CLK2X* driver is disabled.

The Nubus '90 cache coherence protocol is not planned to be used by any Macintoshes in the near future.

The optional standby power supplies a small amount of +5 volt power when the main power supplies are turned off (but still connected to the AC line). This allows a low power circuit on a NuBus card to power on the system using the PFW* line. Eclipse plans to provide standby power. On machines such as Spike that do not plan to provide standby power, the recommended implementation is to connect the STDBYPWR pins on the NuBus connector together, but not connect them to anything else.

No currently planned Macintoshes will have P1394 (Chefcat) interfaces on the motherboard. However, new NuBus implementation should bus the P1394 lines (A2 and C2) and terminate them per the NuBus'90 spec (different than the standard NuBus termination).

For more information on NuBus'90, refer to the latest specification. The current revision under public review is *Standard for a Simple 32-Bit Backplane Bus: NuBus*, P1196R/D1.3 (11/21/90), available from the IEEE Computer Society*.

76

Future Directions:

All new processors, except the lowest entry-level systems and most of the portables, will provide a Processor Direct Slot. An effort is being made to offer a standardized '040 direct slot, which would take the address space of one of the NuBus slots.

Several new NuBus controllers are in development (yancc, MUNI) that improve performance by allowing pended writes ("dump and run"), block moves, and improved synchronization. These chips will improve performance for block-oriented applications, such as disk I/O, advanced data communications protocols, data acquisition, and video applications. They will not, however, do much to improve the general Quickdraw "bitblt" operation, since these are characterized by random-access read-modify-write operations. For this high-speed graphics application, a direct processor slot will provide highest performance.

As video and other high-bandwidth data transfers become more common, the need for busses of higher performance than NuBus is increasing. Several high speed bus alternatives under consideration at Apple include the BLT (for Hurricane, formerly Jaguar), the SCI bus (see David James in ATG), and Red Sky (see Paul Sweazey, ATG).

5. ADVANCED I/O ARCHITECTURE

Peripheral Interface Controller (PIC)

The PIC chip, used in the Mac IIfx, Eclipse and some Laserwriters, off-loads the main processor of the floppy-disk, the SCC, and ADB tasks by providing an isolated 6502 processor that can communicate to the main (host) processor by shared memory or DMA transfers. One PIC is used to control the SCC, and another controls the SWIM (floppy-disk) and ADB. The main components of the PIC are:

- A 65CX02 core processor running at 2 MHz
- Two DMA controllers
- A 16-bit timer
- A RAM expansion bus allowing an external RAM to be shared by both the PIC processor and the host processor
- Two Digital Phase-Locked Loops (DPLLs) to support high-speed serial communications

For more details, see the PIC spec, Apple number 343S1021. The PIC RAM is dual-ported between the PIC processor and the host system. It can be up to 60 Kbytes, but a typical size (in Mac IIfx) is 32 Kbytes.

Each of the PIC's DMA controllers can service one port of the SCC. Normally, the SCC is set-up and run by the PIC, with the processor getting an interrupt from the PIC when the data transfer is done and the data is resident in the shared PIC memory. Note: There is a "Bypass" mode that lets the processor control the SCC directly, but a command has to be sent to the PIC to enable this bypass mode. Software that controls the SCC directly will not work unless this bypass mode has been set, an example being the original MIDI driver.

In an effort to support a high-speed Localtalk network, the PIC generates a high-speed clock (C16M + 10 in Mac IIfx), and provides two DPLLs to convert the incoming FM0 data format to NRZ, which can be handled better by the SCC. In addition, there is a carrier sense detect circuit, which works better at detecting the presence of a data packet than the traditional use of the hunt bit and missing clock bit within the SCC. These enhanced Localtalk features are not currently used by the system software.

The SWIM floppy disk chip is supported by the PIC, using the same DMA controller hardware as used with the SCC. However, since there are minor timing differences, a SCC•/SWIM mode bit is used to change the chip to the SWIM mode.

The ADB (Apple Desktop Bus) is supported on machines with the SWIM controlled by a PIC by using the 65CX02 processor to implement a UART in software, communicating with the ADB via the GPIN and GPOUT pins. The standard ADB protocol is supported by 6502 in the PIC.

SCSI DMA

The SCSI DMA chip, used in the Mac IIfx, combines the 53C80 SCSI controller, used on earlier Macintoshes, with a 68030 bus-compatible DMA controller and a watchdog timer. Its key features are:

- 53C80 SCSI host adapter cell with enhancements
- DMA bypass mode (for current software compatibility)
- Hardware handshake mode
- 32 bit DMA transfers with 32-bit addressing
- Supports misaligned (non-modulo 4) data buffer addresses into "normal" (non-NuBus) memory
- Supports non-modulo 4 DMA byte counts
- Programmable 32-bit watchdog timer detects absence of DMA activity
- 3 Mbytes/sec asynchronous transfer rate
- Automatic SCSI bus arbitration ("old" arbitration scheme supported, too)

The SCSI DMA chip is more fully described in its spec, Apple number 343S0064. It requires two external 8-bit latches (typically 74F373s) to multiplex the address bus. Three data transfer modes are supported: polled mode (used in the original Mac), hardware handshake mode (used in subsequent Macs), and DMA mode. **S/W Compatibility Note:** The SCSI DMA chip comes out of reset in the polled mode, however most SCSI software is written for hardware handshake mode. The SCSI DMA chip must be set to hardware handshake mode before this software can be run.

Eclipse I/O Subsystem

The 68040 CPU used in the Eclipse system has a bus interface that is incompatible with I/O devices designed for 68000 through 68030 CPUs. In order to handle the I/O devices and controllers developed for Mac IIfx and previous machines, an I/O bus Adapter (IOA) was developed. This consists of two ASICs: the Relayer (formerly called the MIPS (Microprocessor Interface to Peripheral Subsystem)) and JDB (Junction Data Bus). The IOA converts memory cycle requests from the 68040 bus to cycles on the 68030 bus. Bus masters on the 68030 bus are permitted (such as SCSI DMA, SONIC, etc.), but there is no provision for a 68030 processor on the 68030 I/O bus. Burst transfers longer than 4 bytes from the 68040 bus to the 68030 bus are prohibited, but full burst transfers are allowed on the 68040 CPU bus.

The JDB chip manipulates the data path between the 68040 System bus and the 68030 I/O Bus, as well as generating parity on the data and supplying the system reset functions. It has the following features:

- Implements dynamic bus sizing for data from System to I/O Bus
- Adjusts data from I/O Bus to be correctly aligned for 68040 bus protocol
- Supports non-aligned transfer requests from bus masters on the I/O bus
- Supports 8, 16, and 32 bit slave devices on the I/O Bus
- Performs parity generation and checking for main memory data

- Provides reset signals to the CPU, memory controller, I/O devices and NuBus devices.

The Relayer chip manipulates the bus control lines between the System Bus and I/O bus, as well as controls the data flow in the JDB chip. Its features are:

- Supports Read-Modify-Write cycles to the I/O Bus (for NuBus)
- Supports retries from NuBus to System Bus
- Generates correct bus snooping signals for 68040 from I/O Bus masters
- Does bus arbitration for both I/O and System busses
- Decodes I/O address space and provides chip selects for some I/O chips and the yancc NuBus controller
- Has timer for System bus timeout (15.5 µs @ 33 MHz, 20.5 µs @ 25 MHz)
- Generates TA (Transfer Acknowledge) on System Bus for all breakpoint and interrupt vector cycles

For more information on the Eclipse busses and I/O Subsystem, see: *Relayer Chip Specification* (by Ken Karakotsios) and *Junction Data Bus (JDB) Chip Specification* (by Jano Banks).

PSC-Based I/O Subsystem

The Cyclone and Oceanic computers have a separate I/O bus controlled by a combination data-path/DMA-controller chip. This is called PSC in Cyclone, and JISC in Oceanic. Both are based on the DMA architecture developed for the PSC. The chip connects to the 68030 or 68040 bus on one side, and the I/O bus on the other. They provide decoders and other specific interfaces to the I/O chips. such as SCSI, ISDN, SCC, Floppy and Ethernet controllers, and the Singer sound codec. The PSC also handles interrupts and contains the VIA1 and VIA2 functions.

76

The heart of the PSC and JISC chips is the DMA engine. The DMA architecture is theoretically expandable to 255 channels, with up to 127 "register sets" per channel. Each channel contains a 32-bit configuration register, a 16-bit control register, and one or more register sets. Each register set contains a starting address, a 32-bit count, and a 16-bit command/status register. Multiple register sets per channel allow one or more register set to be reloaded while a DMA transfer is going on using a different register set. If more than one register set is used, the DMA transfer can automatically transfer from one register set to another when one set is finished. This scheme allows the CPU interrupt handling latency to be put in parallel, rather than in series with active transfers.

The PSC and JISC include data FIFOs for each channel, sized appropriately for each channel's needs. These allow data to continue to be transferred under conditions where the bus is unavailable.

To give an example of the capabilities of the PSC, here are the characteristics for each DMA channel:

Channel Number	Name	I/O bus Width	# of Register Sets	Internal FIFO size	Device to be driven
0	SCSI1	16 bits	2	16	53C94 SCSI
1	SCSI2	16 bits	2	16	53C94 SCSI
2	ENetRd	16 bits	3	16	AMD MACE Ethernet Read
3	ENetWr	16 bits	3	16	AMD MACE Ethernet Write
4	FDD	8 bits	2	5	Floppy Disk Controller
5	SCC1A	8 bits	2	5	85C30 SCC Channel A
6	SCC1B	8 bits	2	5	85C30 SCC Channel B
7	ISDN1	8 bits	1	5	ISDN B1
8	ISDN2	8 bits	1	5	ISDN B1
9	ISDN3	8 bits	1	5	ISDN B2
10	ISDN4	8 bits	1	5	ISDN B2
11	SndIn	8 bits	1	16	Singer Sound Input Serial DMA
12	SndOut	8 bits	1	16	Singer Sound Output Serial DMA

Table 5.1 PSC DMA Channel Features

For more information on the PSC, see *PSC ERS* by Mike Bowes, and the *JISC ERS* by Jim Cape.

76

6. GRAPHICS

Macintosh graphics uses a bitmapped technique to allow arbitrary images to be displayed. The bitmap RAM (also known as the frame buffer or Video RAM) itself may be on the motherboard or on one or more plugin cards. The cards may be direct slot or NuBus cards. By using plug-in cards, it is possible to connect and use multiple displays on a single Mac.

The pixels on a Mac display are square and are nominally 72 per inch (about 28 per cm.). In generating display pixels from the corresponding bitmap, one of four techniques is used:

Monochrome — There is a one-to-one correspondence between monochrome display pixels and bits in the map. The bit value in the map determines whether the pixel is on (white) or off (black). This always provides just two colors.

Grayscale — The bits in the bitmap are taken in groups and used to drive a grayscale display. Each group of bits is treated as an unsigned, binary number and fed to a DAC. The pixel luminosity is proportional to the analog output of the DAC. This typically provides from 4 to 256 gray levels.

Color Lookup Table — For most color (and some grayscale) displays, a Color LookUp Table (CLUT) is used. Each entry in the color lookup table has three eight-bit values: one each for red, green, and blue. The CLUT entries are set by the system software as appropriate. To generate display pixels, the bits in the bitmap are taken as groups. Each group of bits is used as an index into the CLUT. The values of the corresponding CLUT entry are fed to three DACs and the output of each DAC generates the corresponding color component in the pixel: red, green, or blue. This typically provides 256 colors out of a palette of 16 million colors.

76

24-bit Color — With this technique, bits in the bitmap are taken as three 8-bit groups for each pixel. These eight bit groups are used to directly drive the color component DACs. This provides 16 million possible colors.

Graphics Hardware Implementation

Built-in graphics hardware is very dependent on the Macintosh system architecture and desired performance. The graphics video subsystems for the Macintoshes with built-in video will be summarized in this section.

Mac Plus/SE Video — The early Macintosh architecture (Mac Plus, SE, Classic) put the video memory in the main RAMs and interleaved video memory requests with CPU requests. In these machines, the display is 512x342x1 pixels, and the active display is requesting data from memory about 67% of the time. In the Mac Plus, during the active display time, one word is fetched for the display for every word fetched for the CPU. The SE and Classic have slightly lower video refresh overhead by fetching one long word (32-bits) of display data for every four words of CPU data.

SE/30 Video — The Mac SE/30 drives a 512x342x1 display from a 64Kbyte bank of VRAM,

76

controlled by state machines in PALs. The VRAM is broken into two 21888-byte video buffers, one of which can be filled while the other is being displayed. This scheme does not slow the main processor bus during screen refresh periods.

Mac LC-style Video — The Mac LC and subsequent low-end machines and some portables use a more sophisticated built-in video scheme that handles multiple monitor types and has less overhead than the original Macintosh. The VISA chip was originally planned to be used for the LC, but was shelved in favor of the V8 chip. However, a variation on the VISA chip, the Eagle, will be used on the Apollo. VISA fetches its video data from main memory in 64-bit (2 long-word) bursts. Running the Rubik monitor at 4-bits/pixel (512x384x4) uses 38% of available CPU cycles, while running a Mac II monochrome monitor (640x480x1) uses 9.4% of available CPU cycles. Other monitors supported by VISA are Rubik in monochrome (512x384x1), and Apple // color mode (560x384x4).

The V8 chip uses VRAM to isolate video buffer refreshes from the CPU. Depending on the amount of VRAM installed, it can support the following pixel depths:

Mode	No VRAM	256K VRAM	512K VRAM
512x384	-	1,2,4,8	1,2,4,8,16
560x384	-	1,2,4	1,2,4,8
640x480 (Mac II)	1	1,2,4	1,2,4,8
640x480 (VGA)	-	1,2,4	1,2,4,8

76

Note that a VGA mode is supported, as well as a monochrome 640x480 mode with no VRAMs installed. The 256K VRAMs are the same as used in the 8.24 Graphics Card. The 640x480 mode differs from the Mac II video standard in that its timing is derived from the 31.3344 MHz CPU clock, rather than a 30.24 MHz clock. This results in 16 extra black pixels on each side of the displayed pixels. The scan rates are the same, within .1%, but the pixels are out of square by 3% (too narrow). As mentioned in the monitor ID section, one of the monitor ID lines is used to signal the Jersey Monitor to compensate for this, if connected to an LC.

Color lookup tables built into the CLUT/DAC are used to take the multiple bits per pixel and map them into the desired colors or gray scales.

Portable Video — The Esprit-Class and Tim-Class portables (Portable, Aruba, Asahi, Tim LC, and Tim) use a display scheme that uses part of the main memory for the video buffer, but avoids contention with the CPU.

All the portables shipping or in design, except Companion, use a 640x400 pixel display. The Omaha controller is designed to handle the active matrix display, and has a 68000 bus interface. The Omaha II adds the ability to drive STN (Super Twist ???) displays. The DDC (Display Driver Chip) can support both types of displays, and is designed to connect to a 68020/30 bus. Companion plans to use a 640x480 display, and will likely have grey-scale capability.

DBlite and Companion support video to an external monitor, through the use of an optional adapter card. The video circuitry is built-in to their V8-style memory/video controller chips, MSC and BMC.

RBV-style Video — The Mac IIci and IIsi use main memory for the video buffer, but have a more sophisticated buffering and output processing scheme than the monochrome Macintoshes. The IIci uses Bank A of the motherboard DRAM as a framebuffer. The necessary video signals (e.g. horizontal blanking) are generated by the RBV IC. These are driven through the Brooktree Bt478 which contains the CLUT and the three video DACs. The Monitor ID bits (see below) establish the correct horizontal and vertical scan rates and sync pulses.

The RBV and Bank A of DRAM share a separate data bus which can be disconnected from the CPU data bus. The RBV asks the Memory Decode Unit (MDU) for data as needed. The MDU disconnects the RAM data bus from the CPU and performs a burst read; the data is then clocked into the 16 long-word RBV FIFO. In the RBV, the data is arranged and moved into a shift register and shifted out 1, 2, 4 , or 8 bits at a time, depending on what type of monitor is connected, or what bit mode is selected by the control panel. The shift register data from the RBV goes into the Bt478 along with the dot clock and sync and blanking signals. The Bt478 uses the input data as a index (0-255) into its CLUT. The three color components from the corresponding CLUT entry are passed to their video DACs and the resulting analog video is sent to the monitor. The Bt478 also contains a register that allow entries of the CLUT to be modified.

Because the RBV needs data at a fairly high bandwidth from Bank A DRAM, there is contention for the bank between the processor and the RBV with the RBV taking precedence. There is no contention for Bank B DRAM nor for ROM and the I/O devices; the processor runs at full speed with these. Approximate percentage of Bank A memory bandwidth used by the RBV on the 25 MHz IIci is as follows:

Video Depth (bits)	13-inch Mac II Color Monitor (640 x 480 pixels)	15-inch Portrait Monitor (640 x 780 pixels)
1	6 %	13 %
2	13 %	26 %
4	26 %	65 %
8	64 %	Can't

DAFB-Based Video — The DAFB (Direct Access Frame Buffer) ASIC is the heart of on-board video for the Eclipse and Spike computers. It provides a 68040 bus interface, control of up to four 512Kbyte banks of VRAMs, and a flexible video refresh and timing generator. The use of VRAM isolates the CPU from video buffer refreshes. Burst-mode accesses are permitted to non-word-interleaved VRAMs. The DAFB also holds the "Turbo" SCSI controller. Full information on the DAFB is given in *DAFB, Eclipse Direct Access Frame Buffer Controller Preliminary Specification,* by Dale Adams.

Video timing for DAFB is through the "Swatch" video timing generator, developed by the Video Products group (formerly SEG). Some modification and simplifications were made to the Swatch circuit used within the DAFB, such as eliminating the ability to lock to an external sync signal. For more information on Swatch, see the *Swatch Specification,* ver 2.01, Feb. 15, 1990, by Beth Clough. The video timings available are very flexible, so, in addition to the standard Macintosh configurations listed below, it can support VGA monitors and NTSC and PAL video timings. An external RGB-to-Composite converter is required for NTSC or PAL.

Unlike other mother-board-based video systems, DAFB allows convolution, which permits the use of interlaced monitors without flickering. For convolution, VRAM memory accesses are arranged so that three scan lines of video can be sent to the CLUT/DAC, where each pixel is the weighted-average of the 24-bit values of the pixel above and below the current pixel.

The following table shows the allowable bit depths vs amount of VRAM for Macintosh-compatible monitors:

Screen Size	1 bank (512Kbytes)	2 banks (1Mbyte)	4 banks (2 Mbytes)
512x384	1,2,4,8	1,2,4,8,32	1,2,4,8,32
640x480	1,2,4,8	1,2,4,8	1,2,4,8,32
640x480(interlaced, w/convolution)	–	1,2,4,8	1,2,4,8,32*
832x624	1,2,4,8	1,2,4,8	1,2,4,8,32
640x870	1,2,4	1,2,4,8	1,2,4,8
1152x870	1,2,4	1,2,4,8	1,2,4,8

* – no convolution

Monitor Identification

Starting with the Mac IIci computer and the 8.24 video card, Macintosh video circuits have been able to identify (to various degrees) the monitors plugged into them, in order to make the computers auto-configure when a new monitor is installed. The monitor can signal the CPU as to how to configure the video timing circuits. Not all systems support all available monitors; the following table show who supports what:

	Sense ID Bits			aa	bb	cc						
Monitor	0	1	2	bc	ac	ab	Size	RBV	V8	JISC	DAFB	8.24
Vesuvio	0	0	0	--	--	--	1152x870	–	†			
Portrait Monochrome	0	0	1	--	--	--	640x870		†			
Rubik	0	1	0	--	--	--	512x384					
2-page Mono.	0	1	1	--	--	--	1152x870	–	‡			
NTSC Monitor	1	0	0	--	--	--	var	–	†			–
Portrait Color	1	0	1	--	--	--	640x870		†			
12"/13" Mono, Color	1	1	0	--	--	--	640x480					
PAL Encoder	1	1	x	00	00	00	var	–	–			–
NTSC Encoder	1	1	x	01	01	00	var	–	–			–
VGA	1	1	x	01	01	11	640x480	–	–			–
PAL Monitor	1	1	x	11	00	00	var	–	–			–
Goldfish	1	1	x	11	11	10	832x624	–	–			–

‡ – The V8 interprets monitor ID = 011 as a VGA monitor.

† – The V8 ignores monitor ID bit 1, so must not be used with these monitors.

Table 6.1 Video Monitor ID Capabilities

There are two ways of using the monitor sense lines: looking at the sense lines directly, and selectively strobing each line and observing the others. The original Apple monitors simply grounded one or more of the sense lines. This obviously limited the number of monitors to less than 8. As new monitors and TV adapters have been added to the product line, another scheme was needed to expand the number of codes, yet stay within the pin limitations of the 15-pin video connector. By selectively connecting diodes or wires between the pins, 28 additional ID codes are available.

The algorithm for reading the ID code is:

1. Read ID0..2.
2. If any bit is zero, this is a conventional ID code. Stop.
3. If all bits are 1, then, for each ID line, pull down the ID line and record the value on the other ID lines.
4. If all bits are 1, then there is no monitor connected. Stop.
5. Otherwise, use the extended ID code. Stop.

The extended ID codes are shown in the table above by the nomenclature: ab, ac, ba, etc, with the letters read vertically. Line "a" corresponds with ID line 0, "b" with line 1, and "c" with line 2. Thus the first column "ac" is the value of line "c" when line "a" is pulled down.

The list of sense line code assignments is kept by Mark Gonzales.

76

7. VIDEO OVERLAY

A new feature that is beginning to be added to Macintosh motherboards, either directly or through plug-in cards, is video overlay. This permits a standard television image, either NTSC (used in the Western Hemisphere and Japan, generally where 60 Hz power is used) or PAL (used in the rest of the world, with 50 Hz power) to be displayed on a Macintosh graphics display. At the current time, video overlay is only being supported on plug-in cards, but in several cases, the motherboard video circuits must be aware of the video overlay card, and have a special interface connector. As the cost of the video overlay hardware drops and the feature becomes more integrated into the Macintosh operating environment, the overlay circuitry will likely move onto the motherboard.

Note: in this chapter, the word "video" refers to the television signal or image, and "graphics" refers to the Macintosh display system.

There are three video overlay systems in design for Macintoshes: Touchstone, EVOC and Oceanic. Touchstone is a NuBus-based overlay card, and is included here for reference, since the software dealing with overlay will have to be aware of Touchstone. EVOC is an optional card that plugs into the Mac LC. It will likely be used on future High-Volume Macintoshes. Oceanic is designing a video overlay system that that uses the Sebastian video overlay controller. All of these support NTSC and PAL video input. They also support arbitrary size and position on the video image on the graphics screen. The features of each system are listed in table 7.1:

	Touchstone	EVOC	Oceanic
Output format	Mac II displays, NTSC, PAL, SECAM	512x384 only	Mac II displays, NTSC, PAL
Video Resolution, depth	640x480x24	640x480x15	640x480x16,32, 768x532x16(?)
Overlay keying, depth	color, 8 bits	color & chroma, 1 bit	color (3 bits?), chroma (1 bit)
Other features	Genlock, capture	Capture	Capture

Table 7.1 Video Overlay Features

Explanation of Features

Video Overlay Keying – Once a video overlay function is available, there must be some means of determining, for any given pixel position on the screen, whether a graphics pixel shows, a video pixel shows, or some combination of the two shows. There are two basic means for doing this: color keying and chroma keying.

Color keying is used when graphics is the controlling element. An application using color keying picks a single QuickDraw color to be the key color. Wherever that color would otherwise appear on the screen, video appears instead. For example, let's say you want to show an Okay button over a live, rectangular video image. You would first draw a rectangle using your chosen key color. The overlay circuitry would then show video in that area of the screen. Then, you would draw your button and text in any colors except the key color. In these drawn areas, graphics (e.g. the text "Okay"and the button boundary) would show rather than the video.

Chroma keying is used when video is the controlling element. An application using chroma keying picks a video color (or range of colors as video is not as precise as graphics) as the key color. Wherever that color or range of colors would otherwise appear on the screen, graphics appears instead. For example, let's say you want to have a live, video person move around on your screen. You would chose your chroma key color to be whatever color makes up the video background, e.g. if your live person was standing in front of a blue curtain, you would choose blue (or a range of blues) as the chroma key color. On your screen, you would see only the person from the video; you will see graphics (e.g. windows, icons, etc.) everywhere where the curtain would otherwise appear.

By having more than one bit of overlay plane depth, each pixel may be a mix of video and graphics. With two bits of depth, for example, you could have the following possibilities for each pixel:

1. 100% graphics
2. 75% graphics, 25% video
3. 50% graphics, 50% video
4. 100% video

76

Genlock – Genlock allows the video and synchronizing timing in a Macintosh controller to be synchronized to an external video system.

Capture – Capture allows a digital image of a video overlay frame to be stored in the Macintosh system.

8. SOUND

All Macintoshes have a speaker, an audio output jack, and hardware to produce sound. New Macintoshes, starting with the Mac IIsi and LC, will also have sound input capability. Future machines will link a telephone interface into the sound subsystem.

The original Macintoshes, including the Mac SE and Classic, used the "Mac Plus Sound" (described below) which is simple, but limited in quality and capabilities. Starting with the Mac II, the Apple Sound Chip (ASC) began to be used, which provided more capabilities (primarily stereo sound) and better buffering. The product line is currently in transition towards higher sound quality and significantly higher sound processing power, exemplified by the Cathedral DSP and Singer codec. In the interim time, two other sound subsystems have been developed: Batman, which adds sound input, sample rate conversion and CD-XA decompression to the ASC, and the V8 sound system, (as used in the Mac LC) which provides sound input and monophonic ASC emulation to high-volume systems.

Tables 1.1 through 1.5 in the Introduction list the sound subsystems used in current and future Macintoshes. The category "Sound Out – Digital" lists the digital processing section of the sound output hardware. "Sound Out – Analog" lists the D/A conversion device (if any) and analog filtering and amplification hardware. "Sound Input" lists both the digital and analog devices used to implement sound input. The following sections describe in detail the various circuits and devices used in the Macintosh sound subsystem.

Appendix 2 has block diagrams of representative types of Macintosh sound hardware subsystems.

76

Sound Output Hardware (Digital)

"Mac Plus Sound" – the sound output circuits used in the early Macs, Mac Plus, Mac SE, and the Mac Classic. A 370-byte buffer is kept in main memory, and either a PAL or part of the BBU generates addresses to pull data from RAM during the vertical retrace period, and put it out as a PWM signal. Details on the implementations in the early Macs and Mac SE are given in the *Macintosh Family Hardware Reference*, Addison-Wesley, 1988. The Mac Classic uses the same sound circuit as the SE, but changes the output jack to a stereo connector with both channels fed from the mono output. The maximum output level for the Classic with 32Ω headphones is 1 volt p-p on each channel.

ASC – the "Apple Sound Chip", also know as the "Foley Sound Chip", introduced with the Mac II, enhances the Classic Mac sound by implementing in hardware much of what was done in ROM code on the earlier machines and adds a four-voice synthesizer (Wavetable) mode. In addition, stereo operation is provided by doubling up the buffers and sound processing hardware. Getting sound data out is no longer coupled to the vertical retrace period, since data is written from the microprocessor to 1 Kbyte FIFOs, one per channel. The hardware and software interface to the ASC is described in its spec (Apple part number 344S0053 for the original version and 344S0063 for the cost reduced version). Despite the added features of the

ASC, nearly all sound software makes use of the subset that is runnable on Mac Plus Sound.

V8 – The Mac LC uses an emulation of a subset of the ASC that is built into its V8 chip. The 1 Kbyte FIFO resides in memory, and is accessed automatically by DMA cycles to produce one channel of PWM sound. **Hardware/Software Debug Note:** A pseudo-random counter is used to address the 1Kbyte block of memory. Tracing the filling or emptying of the "FIFO" in memory is not at all obvious. See the V8 spec (Apple part no. 343S0116) for information. Subsequent computers based on the V8 or VISA architecture (Apollo, Columbia, Vail, and DBlite) use basically an identical sound circuit.

Batman – Batman is an enhanced version of the ASC that is pin-compatible to the ASC. Batman removes the wavetable synthesizer modes of the ASC, but adds:

- Improved interrupts
- One-channel 8-bit sound input capability
- Full stereo mixing and panning capability
- 10 bit PWM resolution on output
- Support of 16 bit sound output data through a serial interface
- Hardware sample rate conversion
- Hardware CD-XA decompression

Batman comes up in a mode that is software compatible with the ASC, except that the wavetable modes are gone. The sample rate conversion and CD-XA decompression may be used by future releases of blue and pink software. See the *Batman Enhanced Sound Chip* spec (by Dick Craddock) for details. **Programmer's Note:** The wavetable mode is not supported in Batman, Elsie, and future sound hardware on the Macintosh. The ASC wavetable mode is rarely used, since it doesn't allow synchronized amplitude changes.

76

Cathedral – Cathedral is the code-name for the hardware and software effort to get a DSP incorporated onto the Macintosh mother board. The chip used will be the AT&T 3210, a version of their standard 32C modified per Apple's requirements. The 3210 has a generic bus-master interface that can connect to a 68030 or 68040 bus, as well as the 386 bus. See Eric Anderson's *Cathedral Overview* for a general description of Cathedral, AT&T's *WE DSP3210 Digital Signal Processor Information Manual* for the Cathedral DSP chip spec, Hugh Svendsen's *Bass-O-Matic II Digital Signal Processing Manager Specification*, and Eric Anderson's *Bass-O-Matic II Programming Model* for the software description. Benefits of the Cathedral DSP include:

- Substantial MIPs available for:
  - CD-XA or other compression and decompression
  - High quality sample rate conversion
  - Multi-channel support
  - Enhanced Text-to-Speech
  - Modem functions, through V.32 (full-duplex 9600 baud
  - Graphics, Number Crunching, and other non-real-time tasks.
- Programmability permits future use of updated or unanticipated functions.

- Time-sliced multi-tasking operating system (called Bass-O-Matic II) permits clean allocation of resources to real-time and non-real-time tasks.

In the Cathedral architecture, the DSP communicates to the rest of the system as a bus master to memory and I/O devices. It has a built-in serial port which is intended as a telecommunications (modem) port. It will handle stereo 16-bit sound input and output by the use of the Singer D/A-A/D chip. See fig. 14-6 in the Appendix for a typical Cathedral implementation.

Enhanced V8 – It is expected that in future high-volume machines that an improved sound subsystem will be installed in the 2nd generation V8 chips, such as Sonora and Everest. Potential improvements include: stereo output channels and decompression.

Sound Output-only Hardware (Analog)

SONY – The "Sony Sound Chip" (Apple part number 343-0045), originally designed for the first Macintosh, provides one channel of filtering, volume control, and speaker amplification. The basic characteristics of the Sony Sound chip are described in its spec (343-0045-01), although more details are given in the Sony spec (Sony's part number is CX1063AP). In addition to its sound functions, the Sony Sound Chip also provides a power-on detection circuit.

Sporty – Sporty combines most of the functions of the two Sony Sound chips, along with a separate speaker amplifier (as used in the SE/30) into one chip. The only missing function is power-on voltage detection. See the *Sporty Chip Spec* by Steve Austin for details.

Sound Input Hardware (Digital)

V8 – The V8 ASIC is used to transfer digitized sound into the Mac LC system. The serialized sound input data from the DFAC is transferred into the CPU through a 1 Kbyte FIFO implemented in main RAM using DMA. The V8 input logic is designed to appear identical to software as the Batman hardware.

Discrete – The Mac IIsi had to be finalized before Batman was ready, so the sound input function is emulated using PALs and a commercial FIFO part. This circuit is functionally equivalent to the Batman sound input logic. The Mac IIsi may be revised to use Batman in the future.

Batman – Batman has a single-channel sound input circuit that takes a serial bit stream from the DFAC (or equivalent 8-bit A/D converter) and buffers it in one of the on-chip 1Kbyte FIFOs. This limits Batman to one output channel while recording. See the *Batman Enhanced Sound Chip* spec (by Dick Craddock) for details.

Sound Input/Output Hardware (Analog)

DFAC – The DFAC (Digital Filter And Converter) is an analog ASIC chip which provides line or microphone input amplification, anti-aliasing filters, automatic level control, and an 8-bit A/D converter with an asynchronous serial data output. See *Specifications for Digitally Filtered Audio Chip (DFAC)*, by Doug Farrar for details. "Playthrough" (playing the sound input through the sound output for monitoring purposes) is available, but is only mono when used in stereo output systems.

76

The DFAC takes its input at microphone levels (2 mv = full signal). On machines with a single mic input jack, if mono or stereo line level signals are to be used, an adapter cable with a pair of RCA jacks is used to merge the stereo signal to mono, and reduce the signal level by a factor of 200.

Discrete – The Mac IIsi had to be finalized before DFAC was ready, so the sound input function is emulated using discrete op-amps, filters, and converter. It has the same functionality as the DFAC circuit. The Mac IIsi may be revised to use DFAC in the future.

Ex-Lax – Ex-Lax, designed for the Hurricane project, and used by Cyclone, integrates stereo microphone preamps, line-level buffers, headphone amplifiers, and speaker amplifiers into one ASIC. Multiplexers allow two stereo external inputs and the buffer outputs to be selected to drive the input preamps. Analog voltage regulation and a digital control block are also included in Ex-Lax. See the *ExLax Chip Final Functional Specification*, by Wil Oxford, Steve Austin, and Bill Aspromonte.

Singer – This is the code name for a very-low cost stereo 16-bit codec (A/D and D/A converter) made by IT&T. It will have a combined signal-to-noise ratio of -80 db, and will not require complex anti-aliasing filters, due to its sigma-delta conversion technology. **Hardware Developer's Note:** This chip communicates with the system through a full-duplex serial port, which needs to be supplied by part of the host system, generally as part of a general-purpose ASIC. See *Singer Stereo Codec Specification*, by Eric Anderson for details.

SAC – SAC is an analog input/output interface chip intended to work with Singer in non-Cathedral systems. It will have input AGC (Automatic Gain Control) and built-in headphone amplifiers. See *SAC Chip Spec* by Steve Austin.

76

Phone Line Interface:

The telephone interface to the Mac sound system is not well defined at this time. An ISDN NuBus card is currently available to developers. The Ex-Lax chip, designed for Hurricane, will provide multiplexers to bring an analog telephone signal into the sound channel, and can be used in Macintosh systems. Cathedral-based systems will have a separate modem codec connected to the DSP3210's serial port. The SAC does not have an analog phone-to-sound input connection, since it relies on the digitized phone sound to be transferred to the Macintosh Sound Manager in software.

Sound Connectors:

Output: All Macintoshes use mini-phone jacks for sound output. The Mac 128 through Mac SE used a mono jack that switched out the built-in loudspeaker. The entire Mac II family through Mac IIfx (not including the SE/30) use a stereo connector that both disconnects the speaker and signals the processor when a plug is inserted. The speaker is connected to the left channel, so software needs to be aware when the speaker is enabled, so it can combine left and right channels. The SE/30, Mac IIsi, and any machines using Sporty, Ex-Lax, or SAC have a separate speaker amplifier that electronically mixes the left and right headphone channels. For more details on the output connectors, see *Save $$ on audio output connectors* by Dave Wilson, June 30, 1989.

Input: A stereo mini-phone jack is used for the sound input connector. On machines currently in design, this connector is mic-level only (approx. 2 mv for full output, without AGC). An external adapter for line-level input is supplied that takes two RCA jacks and reduces the level by a factor of 200 and merges them into the left channel (tip). It is desirable that the RCA jacks and level-reducer be built into future Macintosh computers. A software-selectable mic/line level selector for the mini-phone jack input is also desirable.

Phantom power for an electret microphone is present on the right channel (ring), although a signal could be present here also. It will be mixed with the left channel internally. When true stereo input is available, it is expected that a stereo mini-phone jack will still be used, but no internal mixing of the channels will take place before the A/D converters.

Software Sensing: On Mac II-class machines that drive their speakers from the left channel only of the ASC, software can sense whether a headphone jack is plugged-in, so that the Sound Manager can mix the two channels. In future designs, sensing insertion of both the headphone or microphone jacks is desirable, to allow for future options, such as custom equalizers or other functions that are dependent on what is plugged-in.

Future Interfaces: There are no plans to change the mini-phone jack interface for the analog sound interface, other than building in the line-level RCA input jacks. When digital audio interfaces are designed-in, industry standards, such as the AES/EBU interface used on professional audio equipment and CD players, will likely be used.

76

9. MASS STORAGE

Macintosh mass storage is divided into two categories: floppy drives and everything else. Everything else is: magnetic hard disks (internal or external), CD-ROM, WORM, magneto-optical disks, tape, and no doubt others. Floppy drives are connected by dedicated floppy ports; everything else, including some other peripherals like scanners, is connected by the Small Computer System Interface (SCSI).

FLOPPY DRIVES

There are currently two variations in floppy disk controller interface that have been used in Macintoshes: the Integrated Woz Machine (IWM chip) which supports 400K and 800K disks using GCR (Group Code Recorded) encoding, and the Sander, Woz Integrated Machine (SWIM) chip which supports 400K, 800K (GCR) and 1.4M disks (using Modified Frequency Modulation (MFM) encoding). Both variations contain registers used to manipulate control lines leading out to the connected drive(s). Neither are "smart"; they contain just the minimal hardware to transform the coded serial bit stream into parallel data or vice versa. A cost-reduced version of the SWIM chip, SWIM-2, is currently under development. To address the real-time performance limitations that the IWM or SWIM place on the system, a true smart controller chip is being developed, code named New Age.

All Macintosh floppy disks (both GCR and MFM encoded) are formatted with their sectors interleaved 2:1. For example, on a track with 8 sectors, the sectors are numbered: 1, 5, 2, 6, 3, 7, 4, 8, 1, 5, etc. This allows one sector time for the processor to prepare to read or write the next sector.

76

The IWM Chip

The IWM is an ASIC implementation of Steve Wozniak's GCR floppy interface in the Apple II. The Macintosh uses the same GCR encoding, although with a different file system implementation. In order to get the best bit density for a given disk technology, the speed of the drive is varied, depending on what track is being accessed. There are five speed zones, with the fastest being at the outer tracks (12 sectors/track) and the slowest on the inner tracks (8 sectors/track). On early Macs using 400K drives, the speed control was derived from buffers in memory that paralleled the sound buffers. On all newer Macs with 800K or 1.4M drives, the speed control is automatically handled by the drive. The nominal data rate of GCR floppies is 489.6 Kbit/sec.

GCR encoding uses NRZI coding, which produces a transition every time a "1" is encountered. In order to maintain a constant stream of transitions and to improve the error-detection capability, a process called "nibblizing" is done on writes and "denibblizing" is done on reads. This process (US patent #4,564,941, Wooley et al, Apple Computer Inc.) takes three data bytes, does longitudinal check sums over them, shifts the resulting data, then puts the result through a lookup table that outputs bytes with a maximum run-length of 2 (no more than two non-transitions in a row). This process guarantees the correct run-length requirement and provides good error-detection capabilities. The address fields use a different, somewhat simplified nibblizing/denibblizing process.

Hardware Designer's Note: The IWM does not have any internal state machine or controller; it relies on the main processor to supply it with data at the appropriate times, or to closely poll it for incoming data or status. As a result, it imposes severe availability requirements on the main processor, making it difficult to add other devices to the system that may "hog" the bus, even briefly. Because handling the SCC also has similar tight timing requirements, the floppy driver code checks for SCC data while doing floppy operations. A detailed description of this problem, as well as a good description of general floppy operation is given in PAR Technical Report #4, *Effects of Real Time Devices on Floppy Disk Performance*, by Scott Sarnikowski.

The IWM also handles motor control and head select. See the IWM Spec (Apple # 343S0041-A) for information. Since all new Macintoshes use SuperDrives (GCR & MFM), the IWM chip is no longer used in new designs.

The SWIM Chip

The SWIM chip is a superset of the IWM, incorporating all the functionality needed to read and write GCR data. In addition, the ISM mode of the SWIM chip can read and write MFM data to achieve greater data densities and to read and write formats used by other microcomputers, notably those using 765 or equivalent floppy disk controllers, such as IBM-PCs.

The Manchester or Modified Frequency Modulation (MFM) code used by SWIM is related to GCR but with significant differences. Like GCR, a transition occurs whenever a 1 bit is encountered. In addition, a transition occurs between any two adjacent zeros. See Figure 8.1, "MFM Encoding".

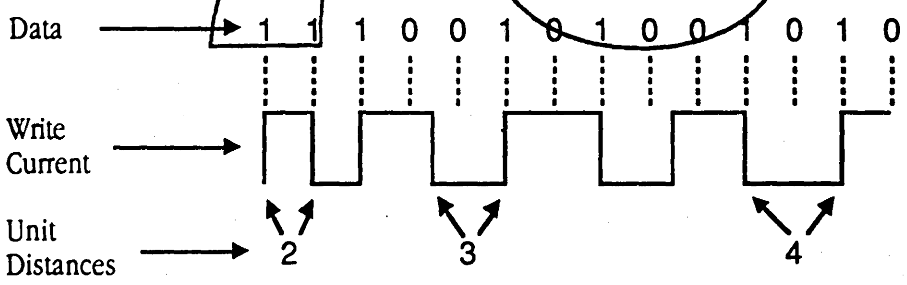


Fig. 8.1 MFM Encoding

The MFM code produces 2, 3, and 4 unit distances between transitions. These are resolved during reading into the original data. A result of this scheme is that no nibblizing is required. In order to assure good data integrity, separate two-byte CRCs (Cyclical Redundancy Checks) are done across the address and data fields.

The SWIM chip has the same real-time processor availability requirements as the IWM, as mentioned in the IWM write-up, above, so similarly requires special caution when used in a system that has other real-time bus masters. The SWIM's nominal data rate during reads or writes is 500 Kbit/sec.

For more information, see the SWIM Spec, Apple # 343S0061-A.

SWIM2

The ISM part of the SWIM chip contains extra circuitry intended for clock and data extraction and write precompensation, which is not used by Macintosh software drivers. In addition, it is possible to use the ISM circuit to handle GCR floppies, making the IWM part of the SWIM redundant. The SWIM2 ASIC cell eliminates the unused and redundant circuitry and adds the capability of working with 1 Mbit/sec drives (the 2.8 MByte Vulcan drive). SWIM2 does not eliminate the real-time requirements of the IWM and SWIM. Engineer: Steve Smith.

New Age

In an effort to overcome the severe real-time constraints that the SWIM and IWM present to the system bus, a floppy disk controller chip, based on industry-standard designs, is being developed that includes both GCR and MFM capabilities. The New Age chip is basically a NEC 765-type controller (as used in IBM-PCs) with the GCR capability added and the disk drive interface changed to the Apple IWM interface standard. Since the chip has all internal state machines needed to seek, do sector reads and writes, and format, the processor gets interrupted only at the completion of these large operations. It can support the planned 1 Mbit/sec data rate used in the 2.8 MByte Vulcan drive. For more information, see *Specification for Apple Floppy Disk Controller (New Age)*, by Scott Sarnikowski.

Software Developer's Note: Since the New Age controller hides the raw data from the processor, traditional Macintosh copy protection schemes will not work. An alternative is to use schemes that have been used in the IBM-PC world, such as writing a hidden sector or by padding incomplete sectors with non-zero values. A "raw dump" command is being proposed for the New Age command set, which would allow the processor to access the raw data, allowing Macintosh protection schemes to work.

Apple Floppy Drive Interface

The electrical interface to the Apple floppy disk drives dates from the Apple II era, and is often called the "IWM" interface, even though it is used on all current and planned Apple floppy drives. For a description of this interface, see one of the drive specifications, such as for the Superdrive, Apple # 699-0477-A.

SMALL COMPUTER SYSTEM INTERFACE (SCSI)

Mass storage that is not a floppy disk, most often magnetic hard disks, is connected via SCSI. Macintoshes implement all of the features, including protocols and arbitration, of the IEEE SCSI standard. For more information on the SCSI standard, see IEEE Section D, ANSI IX3T9.2 (Version 17B). Like the floppy interfaces, control of the SCSI interface is accomplished through a set of memory-mapped read/write registers.

Devices attached to SCSI have an ID from 0 through 7; the processor is always ID 7. An internal magnetic hard disk (a very common SCSI device) always has ID 0. The SCSI chain must always have one terminator, either separately or as part of a peripheral device. SCSI chains with multiple peripherals or long connecting cables must have two terminators, one at each end of the chain.

The Macintosh SCSI implementation differs from IEEE standard in two ways:

1. The IEEE standard uses a 50-pin flat ribbon connector. Macs use this for the internal connector only; the external CPU connector is a DB-25. Connectors for external devices is the IEEE standard.

2. Macs provide power for terminators but the terminators themselves are not provided. They are either included in internal SCSI devices or provided by a separate terminator.

The low-end machine has a complete SCSI implementation. It supports programmed I/O as well as pseudo-DMA modes using an AMD 85C80 combination SCC/SCSI chip. For more information, see AMD Publication # 12582.

A pseudo-DMA mode lets the 53C80 perform SCSI perform SCSI bus protocol while the CPU polls it for completion. A handshake DMA mode allows the CPU to access the 53C80 without polling. In this case the processor is held off by a signal until the 53C80 has successfully transmitted or received a byte of data. This can only be used with high speed peripherals because the handshake must be completed within 16 µs or a bus error will occur. Different address space areas are used when the 53C80 is used in its different modes. For more information, see 53C80 chip spec.

Cyclone will use the NCR 53C94 Advanced SCSI Controller. The primary differences between this and the 53C80 are an 18 bit DMA interface (16 data bits and 2 parity bits) and higher performance (20 Mbyte/sec DMA, 5 Mbyte/sec SCSI). For more information, see the NCR 53C94, 53C95, and 53C96 Advanced SCSI Controller Data Sheet.

10. Networking & Communications

SCC Serial Ports

The SCC (Serial Communications Controller) contains two independent ports (A & B) that can support either synchronous or asynchronous serial data protocols. The SCC chip is fully described in the Zilog *Z8030/Z8530 SCC Serial Communications Controller Technical Manual.* The SCC is also available as a cell which is used in the Zilog "Combo" chip (85C80), and the AMD Curio chip, also known as the "Super Combo" (AMD number 79C950).

The external interface to the SCC is well documented in the *Macintosh Family Hardware Reference.* The important points are repeated here.

The following pins are supported by both ports:

76

Name	Meaning	RS-422 equiv.
TxD+	Transmit Data (active high)	TXD+
TxD-	Transmit Data (active low)	TXD-
RxD+	Receive Data (differential + input)	RxD+
RxD-	Receive Data (differential - input)	RxD-
HSKo	Output Handshake	/DTR
HSKi	Input Handshake or ext CLK	/CTS, /RTxC
GPi	General Purpose Input	DCD, RTxC (A only)*

* - Note that the HSKi inputs are active low, whereas the GPi inputs are active high.

The transmit data pins are tristated until RTS for the port is set. **Developer's Note:** The GPi lines are not available on the Mac Plus, LC, or Classic. The Mac Plus's DCD handshake line is used as part of it's mouse interface.

RS-422 drivers and receivers are used, permitting balanced differential signals to be used on the data lines. One side of the differential receivers for the HSKi and GPi lines are grounded, so bipolar (+ and - voltage) signals must be used on these lines. On the RxD and TxD lines, grounding the positive side of the differential data receiver, and using only the negative side of the differential data transmitter allows the use of the EIA RS-423 standard, which is compatible with RS-232 for up to about 50 feet of cable. For longer cables or high data rates, unipolar differential (RS-422) connections should be used.

The internal logic of the SCC chip runs off of the PCLK, which has the following frequencies:

Mac Plus, Mac SE, Mac II, IIx, IIcx, Mac SE/30, Portable, Aruba, Asahi	C3.7M (3.672 MHz)
Mac IIci, IIsi, IIfx, LC, and all newer Macintoshes	C8M (7.8336 MHz)

Port B's transmit and receive clock (RTxCB) is always connected to C3.7M. Port A's clock

(RTxCA) is normally connected to C3.7M, except when the VIA bit "vSync" is active, which runs port A's clock from the GPi pin of port A's connector. An exception is for the Mac Plus, LC, and Classic, which have port A always connected to C3.7M. Within the SCC chip, port A has higher interrupt priority. These two differences make port A more useful for synchronous data transfers. Port A is marked on the cabinet as the "Modem" port, and port B is marked as the "printer" port. The ability to run the SCC off of external clocks is used by several Apple and third-party products, such as the MIDI interface.

Programmer's Note: The SCC has a special requirement than no accesses to the chip be made at least 4 PCLKs from a previous read or write to the chip. In all but the Mac Plus, this hold-off is handled automatically in hardware. The programmer must obey this timing restriction in the Mac Plus. In addition, the Mac Plus requires that even-addressed byte-wide transfers be done for reads to the SCC and odd-addressed byte-wide transfers be done for writes, in order not to upset the processor clock phasing.

Apple's local area network system, Appletalk, connects Macintoshes and other computers together using Localtalk (using the SCC), Ethertalk (using an Ethernet interface), or other LAN technologies. The Localtalk Link Access Protocol (LLAP) configures the SCC to transfer data using the SDLC protocol using the FM-0 self-clocking encoding technique, running at 230.4 Kbaud. The Appletalk network protocols and a more detailed description of the Localtalk electrical and mechanical specifications are in *Inside Appletalk* by Gursharan Sidhu, Richard Andrews, and Alan Oppenheimer, published by Addison-Wesley, 1989. A more general overview of Appletalk is given in the *Appletalk Network System Overview* by Apple Computer, Addison-Wesley, 1989.

76

The LLAP utilizes several special features of the SCC to simplify its software implementation:

- CRC Generator
- Missing Clock Detector
- Address Detector
- Configurable DMA and INT outputs

The use of these functions is closely tied to the Localtalk timing, and is the main reason the SCC architecture is an integral part of the LLAP implementation.

Localtalk has several critical timing requirements that potentially could be disrupted by other activities on the processor bus. The most obvious one is the fact that the CPU is used to poll the SCC to transfer Localtalk data. With Localtalk running at 230.4 Kbaud, one byte is transferred every 35 µs. Since the receive buffer in the SCC is only 3 bytes deep, the interrupt from an incoming Localtalk packet must be serviced within 100 µs. The transmit buffer is only 1 byte deep, reducing the maximum service time to 35 µs to avoid premature termination of the outgoing packet. Less obvious time critical functions include: establishing the interpacket spacing through the use of the CTS line, and the generation of abort bits at the end of a packet. All these timing constraints make it difficult to connect devices such as Cathedral, to the system bus that can "hog" the bus, even if only briefly. The use of the SCC by Localtalk is described in detail in PAR Technical Report #2, *Effects of Reduced Processor Response on Localtalk*, by Scott Sarnikowski.

The PIC, used in the Mac IIfx and Eclipse, isolates the SCC from the system bus, running the LLAP in the 6502 processor cell within the PIC. Additionally, there is provision to support a high speed (1.6 Mbit) Localtalk protocol through special provisions on the PIC. See the "Advanced I/O Architecture" chapter of this document or the PIC chip spec (Apple number 343S1021) for more details. The PIC provides a "pass-through" mode that allows the CPU to control the SCC directly. This is important in applications, such as the MIDI driver, written to run on any Macintosh, that use the SCC in special modes.

The cost and high part count for using the PIC has made it unattractive for low-cost CPU designs. An alternative is to improve the SCC so that it can handle the inter-packet spacing and abort-bit transmission automatically. The following "Localtalk Enhancements" are being built into the ESCC module of the Curio chip (also known as Super Combo):

- Generation of a SYNC pulse before the SDLC opening flag
- Generate two SDLC opening flags
- Generate an abort sequence at the end of the Localtalk packet
- 8 byte depp transmit and receive buffers
- Disable receive during transmit
- Supply a separate DMA request pin

These changes may also be applied to other SCC or ESCC chips or cells in the future. See *Revised Summary of requirements for Localtalk SCC*, a memo by Scott Sarnikowski (8/24/90), for Apple's definition of the SCC with Localtalk enhancements, and the *Curio (79C950) Objective Specification*, by AMD for the Curio spec.

ADB

The Apple Desktop Bus (ADB) is a serial bus used to connect keyboards, mice, graphics tablets, etc. to the CPU. It is a single master, multiple slave bus running on a party-line, open-collector, asynchronous bus. Plus 5 volts is provided to run the various ADB devices, at up to total maximum of 500 ma (except for the portable). A power-on signal is provided to turn on CPUs with the soft-start feature. Detailed timing and protocol information is provided in the *Apple Desktop Bus Specification (revision D)*. The interface description and a good overview of the ADB is given in the *Macintosh Family Hardware Reference.*

Tables 1.1 through 1.5 in the Introduction lists the ADB chip used in the different Macintoshes. The "ADB" chip is a customized 4-bit microprocessor, Apple No. 342S0440. The Egret (also known as the ADBμ) is an 68HC05 8-bit single-chip microprocessor. Cuda is an improved version of Egret with higher ADB performance. Egret and Cuda are described in the next section, System Support Functions. Both implementations use a bit-serial interface to VIA1, composed of a clock (generated by the ADB controller), bidirectional data, and two mode lines, ST0,1. The ADB unit can generate an interrupt, which is processed by VIA1. On Mac IIfx and Eclipse, the ADB is implemented in 6502 code running on the SWIM PIC chip. On the portables, the ADB is implemented as part of the power manager chip.

Ethernet

Apple software can support Appletalk and TCP/IP protocols running on Ethernet. On CPUs with

NuBus, Ethernet cards provides Ethernet connectivity. There are two types of built-in Ethernet being designed into systems: SONIC and MACE.

SONIC is the National DP83932 network controller chip. The SONIC chip contains a 10 Mb/s Manchester encoder/decoder unit, serializer and deserializer, receive and transmit FIFOs (32 bytes each), and a 32-bit DMA (bus master) bus interface. More information can be found in the DP83932 Data Sheet, by National Semiconductor. The SONIC is also used in Apple's plug-in Ethernet controllers, John Galt (NuBus) and Elan (Mac LC).

MACE is an Ethernet interface chip and ASIC cell designed by AMD. It includes the Manchester encoder/decoder unit, serializer and deserializer, receive and transmit FIFOs (128 bytes each), and a 16-bit passive (non-bus-master) bus interface. It has a 16-bit passive (non-bus-master) bus interface. More information can be found in the AMD 79C9416 Data Sheet, or in AMD's *Objective Specification - Advanced Combo (Curio)*, product number 79C950.

Macintoshes with built-in Ethernet, as well as Apple's newer Etherrnet cards (ELAN and John Galt) use a modification of the ISO 8802.3-specified AUI (Attachment Unit Interface), called the Apple Attachment Unit Interface (AAUI). The AAUI provides all of the functions of the ISO-standard 15-pin AUI, but eliminates the 500 ma capability at 12V, permitting it to be used on equipment that cannot supply that much power. Additionally, the unique connector eliminates confusion with the 15-pin video connector that exists on most Macintoshes. A full description of the AAUI can be found in the John Galt ERS, by Suzy Brown.

Adapters allow the AAUI to connect to three types of 10 MB/s Ethernet interconnection schemes:

- Cheapernet (Thin - 10BASE2) - small coax cable
- Original thick coax Ethernet cable (10BASE5)
- Twisted-Pair (10BASET)

For more information on the Ethernet local area networking standard, see the ISO standard 8802-3 : 1989 (ANSI/IEEE Std 802.3-1988), 1989.

ChefCat

A new desktop LAN technology called ChefCat is being developed within Apple, aimed at replacing both the ADB and SCSI. It uses a copper wire twisted pair running the IEEE P1394 protocol. With a standard burst data rate of 39 Mbits/sec, it can handle existing ADB and SCSI devices, as well as new desktop applications such as digitized sound and compressed video. The proposed IEEE P1394 bus has the following features:

- a physical layer supporting both low cost cable media and many ANSI/IEEE 32-bit busses, including NuBus'90.
- variable speed transmission with a standard speed of 49.152 Mbaud/sec and an enhanced speed of 196.608 Mbaud.
- both fair and priority arbitration mechanisms with all nodes guaranteed at least partial access to the bus.
- block, single quadlet, and "isochronous" transfer modes.
- dynamic address assignment that does not require switches or a physical slot number.

- consistency with the proposed IEEE P1212 Control and Status Register Architecture Specification.

Felix is an ASIC designed to be the interface between Chefcat and a 68040 system bus. It is described in the *Felix ERS*, by Dan O'Connor and Dan Hillman.

More information on the P1394 standard can be found in its draft IEEE specification, *High Speed Serial Bus, P1394/Draft 3.2v2*, available from Mike Teener. Our specific implementation of the bus at Apple is described in the *ChefCat Architecture* by Mike Teener. The *P1212 Control and Status Register Architecture Specification* is currently in IEEE sponsor review, and is available within Apple from David James.

FDDI

FDDI is a high-speed (100 Mbit/sec) LAN using an improved token ring protocol. <more information forthcoming>

76

11. System Support Functions

REAL TIME CLOCK, PARAMETER RAM

All Macintoshes keep a valid time and date as well as various control-panel and other parameters while the machine is turned-off. The clock and non-volatile RAM are kept on the RTC chip Egret or Cuda chip, or in the power manager chip (for portables).

RTC (early CPUs)

The Real-Time Clock chip (RTC) generates an interrupt (through VIA1) to the system every second. The absolute time and date are calculated from the 4 byte seconds register within the RTC. The RTC has its own 32.768KHz oscillator, and is powered by an on-board alkaline or lithium battery, so that correct time is kept under power-down conditions.

A 256-byte battery-backed-up RAM on the RTC holds system configuration information and control panel settings. It is organized as 8 sectors of 32 bytes each, accessed by the "extended address" mode. Twenty bytes of RAM can also be directly accessed in order to provide compatibility with an older version of the RTC (Apple part number 343-0040), which had only 20 bytes of storage.

A full technical specification and programming description is contained within the RTC spec, Apple part number 343-0042-B.

Portables

The real-time clock, parameter RAM, and ADB are handled in the power manager chip in the portable Macintoshes. See chapter 12 (Power Supply and Control) for more information.

Clock, Parameter RAM Functions in Egret and Cuda (new CPUs)

The Mac LC, IIsi, Eclipse, Companion and subsequent CPUs will use a new design to handle their real-time clock, parameter RAM, ADB, and other functions. It is based on the Motorola 68HC05EG 8-bit microprocessor, and currently exists in two implementations: Egret and Cuda. The Egret is also know as the µADB. These designs emulate the ADB functions handled by the earlier ADB chip, as well as the following functions:

- Soft Power-on control and power-on RESET
- 256 byte parameter RAM
- Flash ROM Write Enable (if used)
- Keyboard-invoked and programmer's switch NMI
- Keyboard-invoked and programmer's switch RESET
- 1-second resolution system clock
- 1-second interrupt
- Soft power control and auto power-up (if used)

- Monitor Connected Detection (if used)
- Fast Reset
- Control of DFAC sound input functions
- Fileserver auto-power up

Commands use a packet structure protocol with extended pseudo-device commands. ADB interrupts are passed to the VIA1 (or equivalent). Interrupts from the pseudo functions are passed to the 680XX processor by three tri-state lines that connect to the processors interrupt vector pins, IPL0..2 (Mac LC). A full description of the Egret interactions with the 680XX, ADB state diagrams, and signal timings is given in the *Elsie/Catalina ADBµ ERS* (distributed by Robin Coles).

Egret waits the worst-case time before completing an ADB transfer. Cuda implemented a change in the 68HC05 code which improves the ADB transfer performance by more efficiently handshaking the ADB transfers. Cuda is described in the *Cuda ERS* by Ray Montagne.

VIA Functions

The VIA (Versatile Interface Adapter) is a multipurpose chip that has a general-purpose parallel port, a serial interface to the RTC and ADB, and as a pair of counter-timers. The original Macintosh used the 6522, but was later superseded by the CMOS version, 65C22. Starting with the Mac II, an Apple custom version of the VIA, known as the 6523, began to be used. This chip synchronized the serial port inputs and fixed a few minor logic bugs. In the Mac II, IIx, IIcx, and SE/30, this chip was used for both the VIA1 and VIA2 functions. In newer machines with higher levels of integration the VIA2 and sometime the VIA1 functions are incorporated into other ASIC devices. See tables 1.1 through 1.5 in the Introduction for the locations of the VIA functions.

76

For more information on the different VIA implementations, see:

6522	6522 Spec, Apple No. 337-6522
6523	6523 Spec, Apple No. 338S6523
6525	
RBV	RBV Spec, Apple No. 343S1019
OSS	OSS Spec, Apple No. 344S0076
V8, JISC, PSC	*VIA Cell Preliminary Spec.* (by Mike Bowes)

The bit assignments of the VIA registers varies from processor to processor. See the bit assignments in Appendix 2, Tables 16-1 through 16-4 or refer to the computer ERSs for ones not listed. Note that the V8 chip in the Mac LC also has several other registers in addition to the VIA registers. See the V8 chip spec, Apple part number 343S0116 for details. **Note:** In all machines, VIA bits PA0,1,2 and PB7 are reserved for sound. Although these may be used for burn-in and CPU ID, they can be driven by software, so should have soft pull-ups or pull-downs.

The timers in the VIA2 chip are not used, and are not even implemented in the newer ASIC implementations, so they should never be assumed to be available.

Programmer's Note: The VIA has many quirky aspects, particularly regarding the timers.

Some code has either intentionally or inadvertently taken advantage of these quirks, so that designers replicating the VIA in ASICs have tried retain some of these. However, not all quirks may be supported on these or future chip designs. See comments in Mike Bowes' *VIA Cell Preliminary Spec.* for examples of these changes.

An important function of the VIA is to allow the machine configuration to be read by the ROM, system software and diagnostics. VIA bits PA0, 1, 2, 4, and 6 alert the ROM at boot time of whether the machine is in burn-in (in manufacturing) and what type of board is being brought up. Bits PA0, 1,2 have to be set with soft pull-ups or pull-downs, since these bits are programmed as the sound volume level for machines with "Mac Plus Sound".

The VIA CA1 input bit is the VBL interrupt. VBL stands for Vertical BLanking, and is 60.15 Hz, as used in the original Macintosh scanning circuits. On newer machines with either plug-n video cards or different scan frequencies, this 60.15 Hz rate must be maintained for software timing compatibility.

76

12. POWER SUPPLY & CONTROL

POWER SUPPLY

Macintoshes use three different types of power supplies. These are described separately.

Combination Power Supplies

Macintoshes up to and including the Mac Plus, as well as the Mac Classic, use a combination power supply. In this case, the power supply is combined with the analog circuit for the CRT. This included vertical sweep, horizontal sweep, and CRT high voltage. Voltages supplied were +12, +5, and -12. Changing the input from 120 volts AC to 240 volts AC required moving a jumper and changing a fuse. See Drawing Number 050-0097-U.

Modular Power Supplies

From the Mac SE on, modular supplies are used that accept input voltages as wide as from 85 to 270 volts AC. A typical supply is that used in the Mac IIcx (Spec. Number 699-0392-A). Its specifications are:

Input voltage	85-270 VAC
Input line frequency	47-63 Hz, single phase
Line dropout immunity	20 ms. minimum

Output currents and Power:

Load Condition	+5 V	+12 V	-12 V	Total Power
Minimum Load	2.0 A	20 mA	20 mA	10.5 W
Maximum Load	12.0 A	1.5 A	1.0 A	90 W
Peak Load	12.0 A	3.0 A	1.0 A	108 W

(15 seconds, maximum, 10% duty cycle)

76

The IIcx power supply also provides a soft power on feature. The supply continuously provides a low, +5 volt control voltage at a minimum of 1 ma. During soft powerup, this voltage is applied to a power supply line and the supply begins functioning normally.

Protection is included for short circuits, over-voltage, over-temperature, and over-current.

Portable Power

The primary power source for portables is their built-in rechargeable batteries. The type of batteries used, their amp-hour capacities, and their expected usage times are listed in Table 1.3 in the Introduction.

The portables are supplied with plug in adapters which can both recharge the built-in battery and run the portable indefinite off the AC power lines. A typical adapter is the one for the Portable: it runs off of 85-270 volts AC and provides 7.5 volts at up to 2.0 Amps. A small back-up battery is also provided to keep the RAM alive while the main battery is either completely dead, or while it is being changed.

POWER CONTROL

Soft Power-On

All Mac II-class computers have a soft power on/off, which permits the AC power to be turned on from either an ADB device (typically the keyboard) or a NuBus card. Power off is done through the power-off interrupt, supplied by either the Shut-Down menu or by the rear panel power switch. This initiates a clean shut-down sequence and ends with the power supply shutting off.

Portable Power Management

In order to maximize use time on the battery, the portable Macintoshes use a dedicated Power Manager Processor. The specific power manager used is listed in Table 2 in the Introduction. This processor reduces power consumption by disabling clocks to peripheral chips like SWIM and disabling various power planes during sleep or idle periods.

To further reduce power consumption, the processor activity is reduced as much as possible during periods of low activity. Techniques used include:

- Injecting 64 wait states into the main processor RAM/ROM cycle during idle.
- Saving the processor state, and completely powering off the processor.
- Significantly slowing the processors clock.

13. APPENDIX 1 - VIA Bits

One VIA: Mac SE, Classic		Two VIAs: Mac II, IIx, IIcx		VIA/OSS - Mac IIfx	
VIA		*VIA1*		*VIA1*	
PA0	SND.VOL0	PA0	Burnin	PA0	BURNIN~
PA1	SND.VOL1	PA1	Burnin	PA1	CPU.ID0
PA2	SND.VOL2	PA2	Burnin	PA2	CPU.ID1
PA3	MODEM	PA3 O	MODEM (ext clk)	PA3	MODEM
PA4	HI.DRIVE~	PA4	OVERLAY	PA4	CPU.ID2
PA5	SEL	PA5	SEL	PA5	SEL
PA6	VID.PG2~	PA6	CPU.ID1	PA6	CPU.ID3
PA7	SCC.WREQ~	PA7 I	SCC.WREQ~	PA7	SCC.WREQ~
PB0	RTC.DATA	PB0	RTC.DATA	PB0	RTC.DATA
PB1	RTC.CLK	PB1	RTC.CLK	PB1	RTC.CLK
PB2	RTC~	PB2	RTC~	PB2	RTC~
PB3	ADB.INT~	PB3	ADB.INT~	PB3	ADB.INT~
PB4	ADB.ST0	PB4	ADB.ST0	PB4	ADB.ST0
PB5	ADB.ST1	PB5	ADB.ST1	PB5	ADB.ST1
PB6	SCIRQ.MSK	PB6	n.c.	PB6	PAR.EN~
PB7	SND.RES	PB7	n.c.	PB7	PAR.ERR~
CA1	60.15HZ (VBLK~)	CA1	60.15HZ (VBLK~)	CA1	60.15HZ (VBLK~)
CA2	1SEC~	CA2	1SEC~	CA2	1SEC~
CB1	ADB.CLK	CB1	ADB.CLK	CB1	ADB.CLK
CB2	ADB.DATA	CB2	ADB.DATA	CB2	ADB.DATA
		VIA2		*RBV*	? - OSS
				Slot Interrupts Register	
		PA0 I	SLOT.IRQ1~	0	SLOT1.IRQ~
		PA1 I	SLOT.IRQ2~	1	SLOT2.IRQ~
		PA2 I	SLOT.IRQ3~	2	SLOT3.IRQ~
		PA3 I	SLOT.IRQ4~	3	SLOT4.IRQ~
		PA4 I	SLOT.IRQ5~	4	SLOT5.IRQ~
		PA5 I	SLOT.IRQ6~	5	SLOT6.IRQ~
		PA6 O	RAM.SIZ0	6	SLOT0.IRQ~
		PA7 O	RAM.SIZ1	7	Reserved
				VIA2 Data Register	
		PB0 O	CDIS~	0	CENABLE~
		PB1 O	BUSLOCK~	1	BUS.LOCK~
		PB2 O	PWR.OFF~	2	PWR.OFF~
		PB3 I/O	FC3/CPU.ID0	3	CFLUSH~
		PB4 I	TM1A~	4	TM1A~
		PB5 I	TM0A~	5	TM0A~
		PB6 I	SND.EXT~	6	SND.EXT~
		PB7 O	60.15HZ~ SQ WV OUTPUT	7	PAR.TEST~
				Interrupt Flags Register	
		CA2	SCSI.DRQ	0	SCSI.DRQ
		CA1	ANY.SLOT~	1	ANY.SLOT~
		Shift Register Interrupt		2	EXP.IRQ~
		CB2 I	SCSI.IRQ	3	SCSI.IRQ
		CB1 I	SND.IRQ~	4	SND.IRQ~
		Timer 2 Interrupt		5	Reserved
		Timer 1 Interrupt		6	Reserved
		VIA2.IRQ7		VIA2.IRQ	

76

RAM SIZE
0-256K
1-1M
2-4M
3-16M

020 I CACHE DISABLE

32/24

INT 6 WAS SOFT POWER

Table 13.1 Macintosh VIA History (part 1) (thanks to Noah Price for these tables)

VIA/RBV: Mac IIci			VIA/RBV: Mac IIsi	
VIA1			*VIA1*	
PA0	BURNIN~		PA0	BURNIN~
PA1	CPU.ID0 PA1		CPU.ID0	
PA2	CPU.ID1 PA2		CPU.ID1	
PA3	MODEM PA3		MODEM	
PA4	CPU.ID2 PA4		CPU.ID2	
PA5	SEL PA5		SEL	
PA6	CPU.ID3 PA6		CPU.ID3	
PA7	SCC.WREQ~		PA7	SCC.WREQ~
PB0	RTC.DATA		PB0	n.c. EGRET NC.
PB1	RTC.CLK		PB1	n.c.
PB2	RTC~ PB2		n.c.	
PB3	ADB.INT~		PB3	CVR_SESSION
PB4	ADB.ST0PB4		VIA_FULL	
PB5	ADB.ST1PB5		SYS_SESSION	
PB6	PAR.EN~		PB6	n.c.
PB7	PAR.ERR~		PB7	n.c.
CA1	60.15HZ (VBLK~)	FROM RBY	CA1	60.15HZ (VBLK~)
CA2	1SEC~ CA2	FROM RTC	~~1SEC~~	
CB1	ADB.CLK		CB1	ADB.CLK
CB2	ADB.DATA		CB2	ADB.DATA
RBV	*RBV*			
	Slot Interrupts Register			
PA0	SLOT.IRQ1~		0	SLOT1.IRQ~
PA1	SLOT.IRQ2~		1	SLOT2.IRQ~
PA2	SLOT.IRQ3~		2	SLOT3.IRQ~
PA3	SLOT.IRQ4~		3	SLOT4.IRQ~
PA4	SLOT.IRQ5~		4	SLOT5.IRQ~
PA5	SLOT.IRQ6~		5	SLOT6.IRQ~
PA6	RAM.SIZ0	RAMSIZE	6	SLOT0.IRQ~
PA7	RAM.SIZ1		7	Reserved
	VIA2 Data Register			
PB0	CENABLE~	CACHE ENABLE	0	n.c.
PB1	BUS.LOCK~		1	BUS.LOCK~
PB2	PWR.OFF~	SOFT Power off	2	PWR.OFF~
PB3	FCFLUSH~	CACHE FLUSH	3	n.c.
PB4	TM1A~ 4	I NUBUS ERRCODES	TM1A~	
PB5	TM0A~ 5	I	TM0A~	
PB6	SND.EXT~	I INT/EXTSPKR	6	n.c.
PB7	6PAR.TEST~		7	n.c.
	Interrupt Flags Register			
0	SCSI.DRQ		0	SCSI.DRQ
1	ANY.SLOT~		1	ANY.SLOT~
2	EXP.IRQ~		2	EXP.IRQ~
3	SCSI.IRQ		3	SCSI.IRQ
4	SND.IRQ~		4	SND.IRQ~
5	Reserved		5	Reserved
6	Reserved		6	Reserved
7	VIA2.IRQ		7	VIA2.IRQ

RBY Interrupt Inputs

7- NMI
6- PWRFAIL (NOT USED) 5- EXPIRQ (NOT USED)
4- SCC
3- LVL3 IRQ (NOT USED)
2- VIA2 (INTERNAL)
1- VIA1

Table 13.2 Macintosh VIA History (part 2)

76

76

Portable		Mac LC		Tim, TimLC	
VIA		*VIA Cell inside V8* & VISA		*VIA1*	
PA0	PM D0	PA0	BURNIN*	PA0	BURNIN*
PA1	PM D1	PA1	ID0 = "0"	PA1	CPU.ID0
PA2	PM D2	PA2	ID1 = "1"	PA2	CPU.ID1
PA3	PM D3	PA3	"0"	PA3	Modem*.
PA4	PM D4	PA4	ID2 = "1"	PA4	CPU.ID2I
PA5	PM D5	PA5	HDSEL	PA5	Sel
PA6	PM D6	PA6	ID3 = "1"	PA6	CPU.ID3
PA7	PM D7	PA7	SCC.WREQ/	PA7	SCC.WREQ*
PB0	/PMREQ	PB0	"0"	PB0	RTC.DATA
PB1	/PMACK	PB1	"0"	PB1	RTC.CLK
PB2	VIA_TEST	PB2	"0"	PB2	RTC~
PB3	/MODEM	PB3	ADB.INT/	PB3	n.c.
PB4	/DRIVE1B	PB4	ADB.ST0	PB4	n.c.
PB5	HDSEL	PB5	ADB.ST1	PB5	n.c.
PB6	/STEREO	PB6	"0"	PB6	n.c.
PB7	/SCCW.REQ	PB7	SND RES	PB7	n.c.
CA1	60.15HZ (VBLK~)	CA1	60.15HZ (VBLK~)	CA1	60.15HZ (VBLK~)
CA2	1SEC~	CA2	n.c.	CA2	1SEC~
CB1	/PMINT	CB1	ADB.CLK	CB1	n.c.
CB2	SCSI_IRQ	CB2	ADB.DATA	CB2	n.c.
		VIA Cell inside V8		*VIA2 inside ORCA*	
		PA0	n.c.	PA0	PMgr Data 0
		PA1	n.c.	PA1	PMgr Data 1
		PA2	n.c.	PA2	PMgr Data 2
		PA3	n.c.	PA3	PMgr Data 3
		PA4	n.c.	PA4	PMgr Data 4
		PA5	n.c.	PA5	PMgr Data 5
		PA6	n.c.	PA6	PMgr Data 6
		PA7	n.c.	PA7	PMgr Data 7
		PB0	"1"	PB0	CDIS
		PB1	"1"	PB1	PMGR.ACK
		PB2	"1"	PB2	PMGR.REQ
		PB3	24/32 Bit Map	PB3	HMMU 24/32 Bit
		PB4	"0"	PB4	n.c.
		PB5	"0"	PB5	n.c.
		PB6	SND EXT = "1"	PB6	SND.EXT*
		PB7	"0"	PB7	MODEM.RESET
		CA1	SLOTIRQ/	CA1	Any SLot*
		CA2	n.c.	CA2	SCSI.DRQ
		CB1	SNDINT/	CB1	Sound IRQ*
		CB2	SCSIIRQ/	CB2	SCSI IRQ*
		Timer 2 Interrupt		Shift Register Interrupt	
		Timer 1 Interrupt		Timer 2 Interrupt	
		VIA2.IRQVIA2.IRQ		Timer 1 Interrupt	
				VIA2.IRQ	

Table 13.3 Macintosh VIA History (part 3)

700/900 USE 6522's

76

Eclipse		Spike		CYCLONE	
VIA 1		~~VIA Cell inside VISA~~ VIA1		VIA CELL IN PSC	
PA0	BURNIN*	PA0	BURNIN*	PA0	—
PA1	CPU.ID0	PA1	CPU.ID0	PA1	CPUIDØEXT
PA2	CPU.ID1	PA2	CPU.ID1	PA2	CPUID1EXT
PA3	n.c.	PA3	Modem*	PA3	MODEM EXT/INT CLK
PA4	CPU.ID2	PA4	CPU.ID2	PA4	CPU ID 2 EXT
PA5	n.c.	PA5	HDSEL	PA5	—
PA6	CPU.ID3	PA6	CPU.ID3	PA6	— CPUID3EXT
PA7	n.c.	PA7	SCC.WREQ*	PA7	—
PB0	n.c.	PB0	RTC.DATA	PBØ	
PB1	n.c.	PB1	RTC.CLK	PB1	
PB2	n.c.	PB2	RTC*	PB2	
PB3	XCVR_SESSION*	PB3	ADB.INT/	PB3	TREQ-
PB4	VIA_FULL	PB4	ADB.ST0	PB4	BYTEACK- } CUDA
PB5	SYS_SESSION	PB5	ADB.ST1	PB5	TIP-
PB6	Software IRQ*	PB6	Software IRQ*	PB6	
PB7	A/UX.IntEn*	PB7	A/UX.IntEn*	PB7	
CA1	60.15HZ (VBLK-)	CA1	60.15HZ (VBLK-)	CA1	60.15Hz — INTERNALLY generated by PSC
CA2	n.c.	CA2	1Sec*	CA2	<NOT USED>
CB1	VIA_CLK	CB1	ADB.CLK	CB1	CUDA CLK
CB2	VIA_DATA	CB2	ADB.DATA	CB2	CUDA DATA
VIA2		~~VIA Cell inside VISA~~ VIA2		RBV THING IN PSC	
PA0	Ethernet IRQ*	PA0	Ethernet IRQ*	PAØ	
PA1	Slot A IRQ*	PA1	n.c.	PA1	SCOTA (NC)
PA2	Slot B IRQ*	PA2	n.c.	PA2	SLOTB (NC)
PA3	Slot C IRQ*	PA3	n.c.	PA3	SLOTC
PA4	Slot D IRQ*	PA4	Slot D IRQ*	PA4	SLOTD
PA5	Slot E IRQ*	PA5	Slot E IRQ*	PA5	SLOTE
PA6	Video IRQ*	PA6	Video IRQ*	PA6	SLOTØ VIDEOVBL
PA7	n.c.	PA7	n.c.	PA7	
PB0	n.c.	PB0	DFAC Latch Enable		
PB1	Bus Lock*	PB1	Bus Lock*		
PB2	n.c.	PB2	Power Off*		
PB3	n.c.	PB3	DFAC Config. Data		
PB4	LED	PB4	DFAC Config. Clk		
PB5	33MHzSysClk	PB5	33MHzSysClk		
PB6	n.c.	PB6	n.c.		
PB7	60.15Hz	PB7	60.15Hz		
CA1	Any Slot*	CA1	Any Slot*	CA1	ANYSLOT IRQ
CA2	SWIM PIC IRQ*	CA2	n.c.	CA2	SCSI DRQ
CB1	Sound IRQ*	CB1	Sound IRQ*	CB1	~~S~~-NC —
CB2	SCSI IRQ*	CB2	SCSI IRQ*	CB2	SCSI IRQ
Shift Register Interrupt		Shift Register Interrupt			
Timer 2 Interrupt		Timer 2 Interrupt			
Timer 1 Interrupt		Timer 1 Interrupt			
VIA2.IRQ		VIA2.IRQ			

Table 13.4 Macintosh VIA History (part 4)

APPENDIX 2 – SOUND HARDWARE BLOCK DIAGRAMS

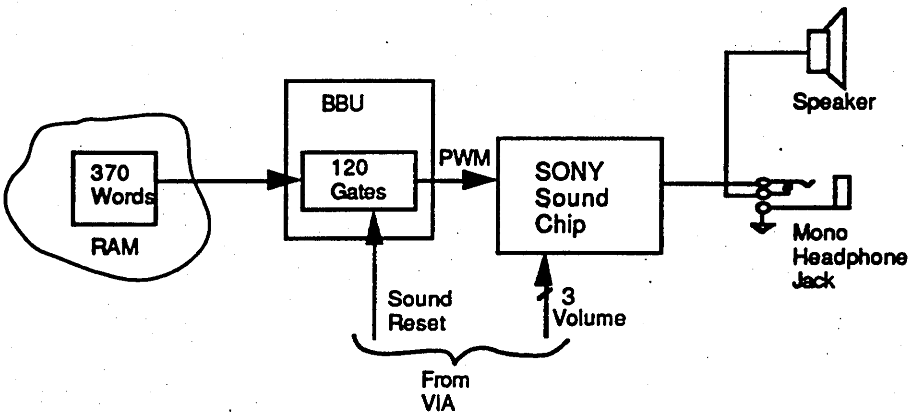


Fig. 17.1 – Sound Hardware for Mac SE, XO

76

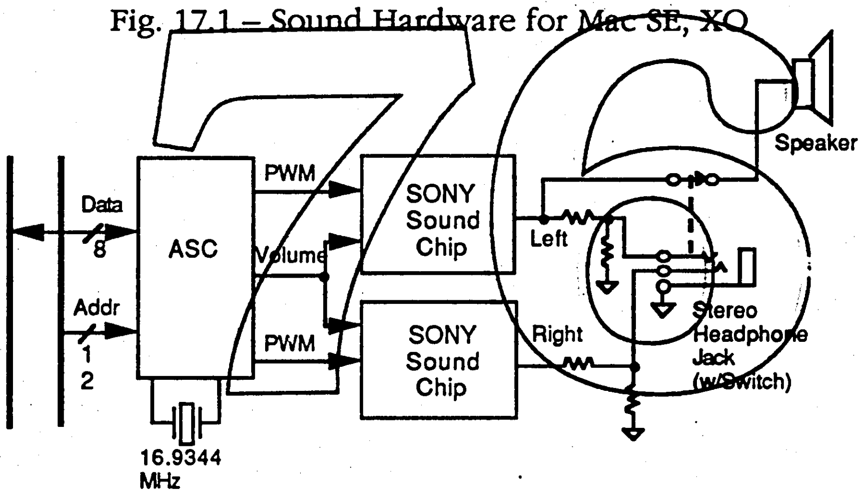


Fig. 17.2 – Sound Hardware for Mac II, IIx, IIcx, IIci, IIfx

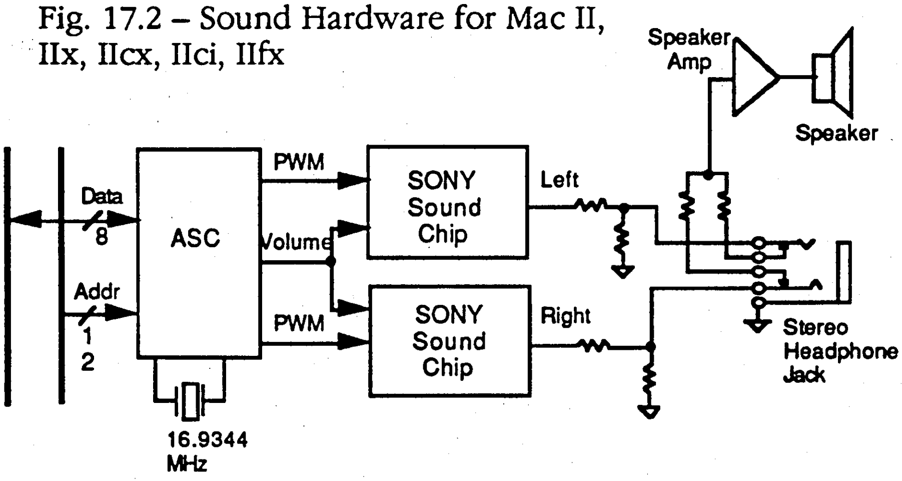
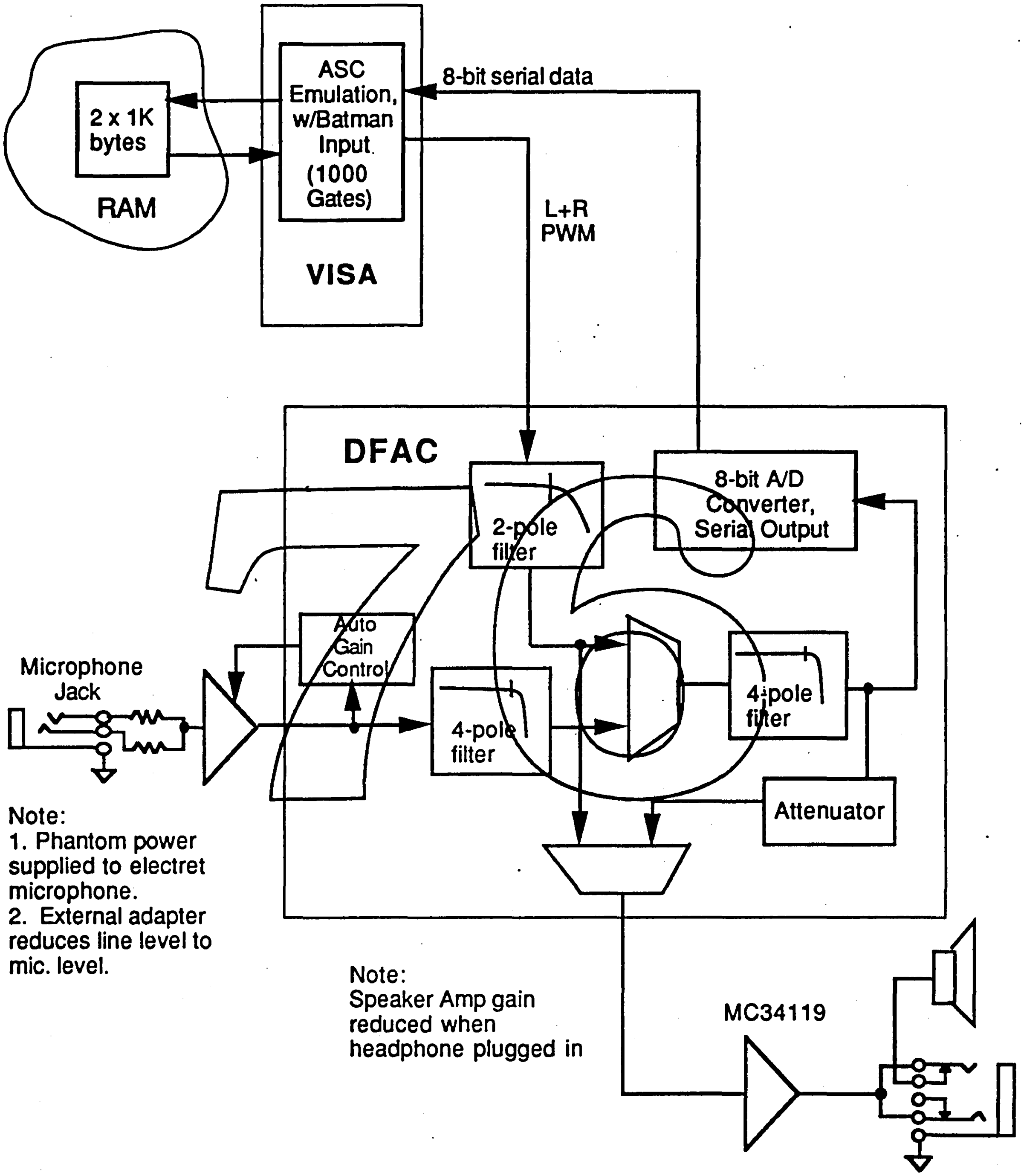


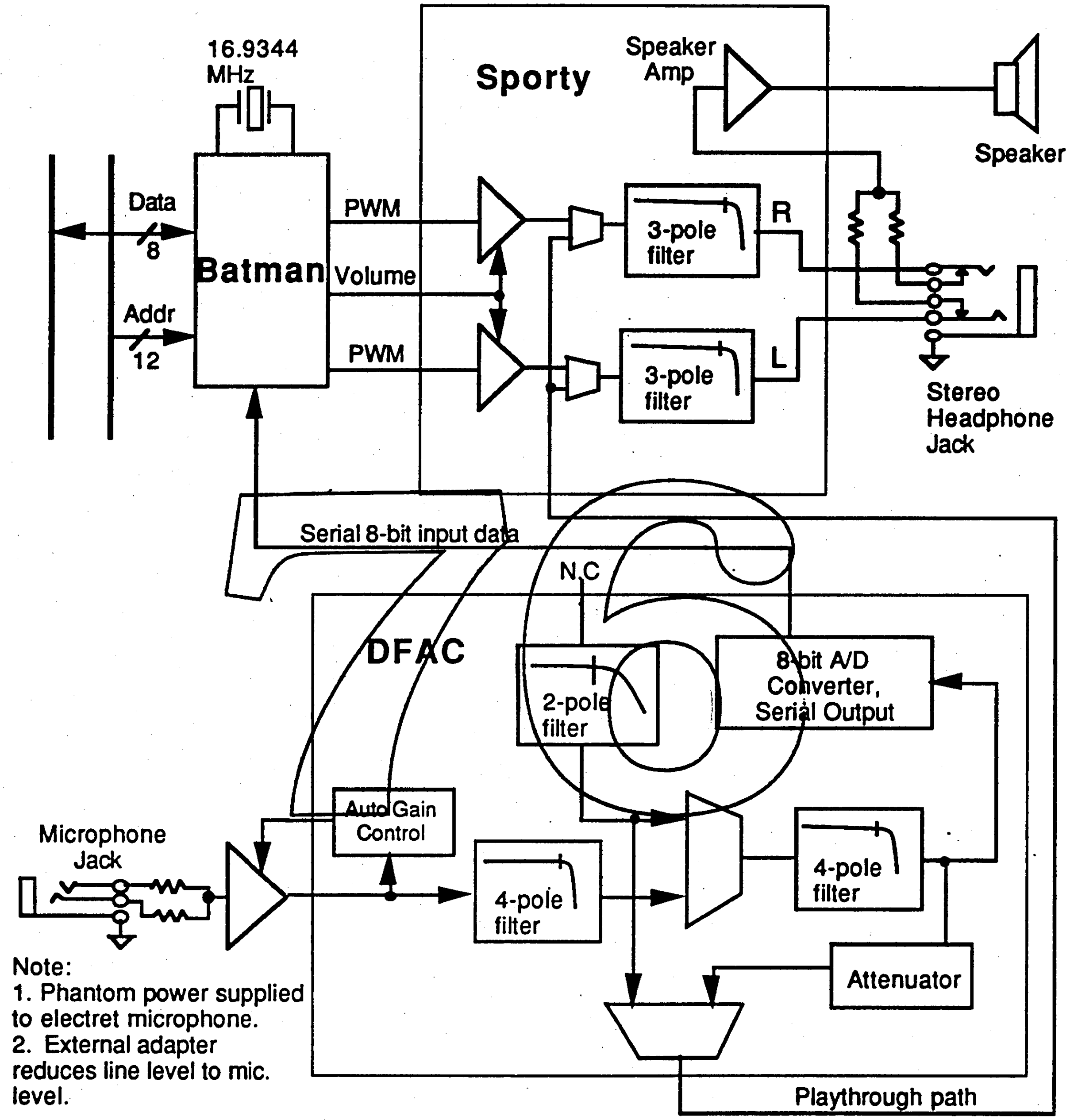
Figure 14.3 – Sound Hardware for Mac SE/30



Note:
1. Phantom power supplied to electret microphone.
2. External adapter reduces line level to mic. level.

Note:
Speaker Amp gain reduced when headphone plugged in

14.4 – Sound Hardware For Elsie



Note:
1. Phantom power supplied to electret microphone.
2. External adapter reduces line level to mic. level.

Fig. 14.5 – Sound Hardware for Mac iisi (ASIC Implementation)

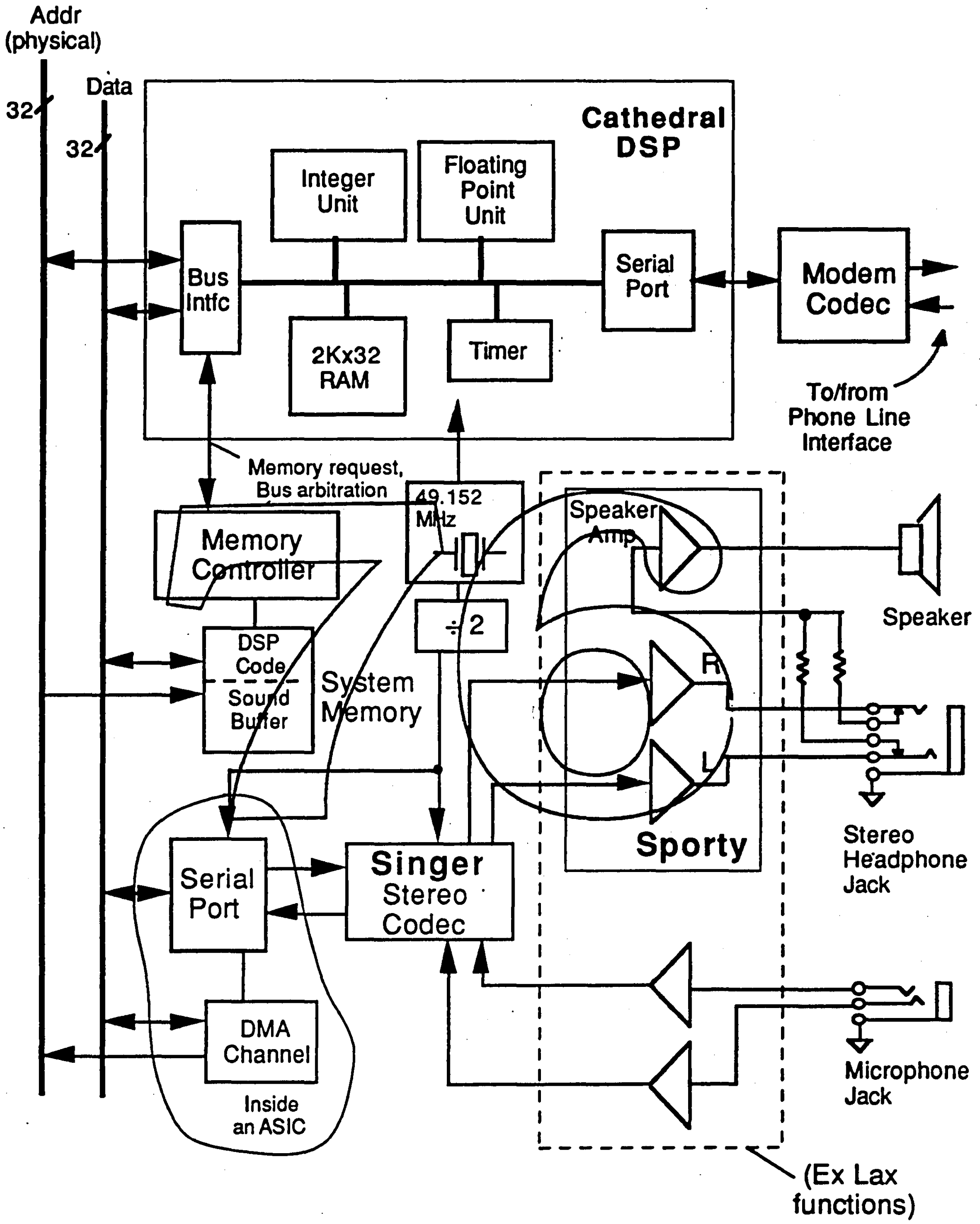


Fig. 14.6 – Sound Hardware for Cathedral-based Machines

15. BIBLIOGRAPHY

There are many detailed sources of information on the Macintosh. The ones mentioned in this document are itemized by chapter here. There are several types of information referenced here:

Motorola Manuals - Available from Motorola: Apple's current Bay Area Motorola sales engineer is Gwen Waddoups, (408) 991-7423.

The Inside Macintosh Library series, published by Addison-Wesley: Available at the Wishing Well or the Company Store. An online version of Inside Macintosh is available on Phil & Dave's Excellent CD: The Release Version, and on the Developer CD volume on the Bloated Wartface (DTS) server in the DTS/Dev Press zone on the Apple engineering network.

Macintosh Technical Notes: Available on the Developer CD Series (i.e. Disky Business), and on both the Developer CD and the DTS Goodies volumes on the Bloated Wartface (DTS) server.

PAR/AIM Technical Reports: These reports, generated by the Architectural Investigations and Modeling group in Functional Technology are available from Muffy Saunders (4-6780) at MS 60-AR.

76

Released Apple technical documents, including chip specs: Documentation with Apple part numbers can be obtained through Document Control at Apple sites. In Cupertino, Document Control is located in Bandley 5.

Unreleased Apple technical documents: Generally available from the author listed. Management approval may be needed for sensitive documents.

PROCESSOR

MC68000M/AD, M86000 Programers Reference Manual

M68000UM/AD MC68000 User's Manual, 1989

MC68020 32-bit Microprocessor User's Manual, 2nd Ed., 1985

MC68030UM/AD MC68030 User's Manual, 1987

MC68040UM/AD, MC68040 User's Manual, 1989

MC68040DH/AD, MC68040 Designer's Handbook, 1989

Inside Macintosh, Vol. V, pp. 2-5, Addison-Wesley

Macintosh Technical Note #117: Compatibility: Why & How

Virtual Memory Management

MC68030 Enhanced 32-bit Microprocessor User's Manual

Atlantic and Pacific Theory of Operation by Noah Price

PAR Tech. Report #5, Virtual Memory Implementations at Apple by Henry Kannapell

Floating Point Assistance

MC68881/MC68882 Floating-Point Coprocessor's User's Manual, Motorola Inc., 1987

Macintosh Technical Note #236: Speedy the Math Coprocessor

Macintosh Technical Note #235: Cooperating with the Coprocessor

Apple Numerics Manual, 2nd Edition, Addison-Wesley

Interrupts

Operating System Support by Steven Ray

Macintosh Technical Note #221: NuBus Interrupt latency (I was a teenage DMA Junkie)

Cache

Macintosh Technical Note #261: Cache as Cache Can

MEMORY

Rex ERS, Paul Baker (Mac LC ERS)

V8 Chip spec, Apple number 343S0116

MDU User Manual, Michael Dhuey,

PGC User Manual, Michael Dhuey

FMC ERS (Apple # 344S1024), John Fitch

Memory Control ASIC (Orwell) Description, Amy Lindburg

Esprit CPU ERS, Rev 3.0, Roger Mohme, section 3.7

ROM

Macintosh Technical Note#225: Macintosh Portable ROM Expansion

Memory Maps

PAR Tech. Report #5, Virtual Memory Implementations at Apple

SYSTEM EXPANSION

Processor Direct Slots

Macintosh Technical Note #254: Macintosh Portable PDS Development

NuBus Interface

IEEE standard for a simple 32-bit backplane Bus: NuBus, IEEE, 1988

Designing Cards and Drivers for Macintosh II and Macintosh SE, Addison-Wesley, 1987

Macintosh Technical Note #234: NuBus Physical Designs

Part number: 344S0606 (Mac II, IIx, IIcx)

Part number: 344S1020 (Mac IIci)

Part number: 344S0075, 344S0074 (Mac IIfx)

Designing Cards and Drivers for Macintosh II and Macintosh SE, Macintosh Technical Note #260: NuBus Power Applications

Standard for a Simple 32-bit Backplane Bus: NuBus, p1196R/D1.3 (11/21/90)

ADVANCED I/O ARCHITECTURE

PIC Chip spec, Apple number 343S1021

SCSI DMA Chip spec, Apple number 343S0064

Relayer Chip Specification by Ken Karakotsios

Junction Data Base (JDB) Chip Specification by Jano Banks

PSC ERS by Mike Bowes

JISC ERS by Jim Cape

GRAPHICS

VISA chip spec, Part number: 343S0101

RBV chip spec, Part number: 343S1019-A

Omaha chip spec, (Dick Craddock 10/87)

DAFB, Eclipse Access Frame Buffer Controller Preliminary Specification by Dale Adams

Swatch Specification, ver 2.01 by Beth Clough

VIDEO OVERLAY

Touchsone ERS, ver 2.4A10 by Ben Yung

SOUND

Sound Input/Output Hardware

ASC Chip spec: Apple number 344S0053 (original version), 344S006 (cost reduced version)

V8 Chip spec, Apple number 343S0116

Macintosh Family Hardware Reference, Addison-Wesley 1988 chapters 6,13

Batman Enhanced Sound Chip spec by Dick Craddock

Cathedral Overview by Eric Anderson

WE DSP3210 Digital Signal Processor Information Manual by AT&T

Bass-O-Matic II Digital Signal Processing Manager Specification by Hugh Svendsen

Bass-O-Matic II Programming Model by Eric Anderson

Singer Stereo Codec Specifications by Eric Anderson

Bass-O-matic II Specifications by Hugh Svendsen

Bass-O-matic II DSP Programming Model by Hugh Svendsen

SONY Sound Chip, Apple number 343-0045,Sony part number: CX1063AP

Sporty Chip Spec by Steve Austin

Ex-Lax Specification by Wil Oxford, Steve Austin, and Bill Aspromonte

Batman Enhanced Sound Chip spec by Dick Craddock

Specifications for Digitally Filtered Audio Chip (DFAC) by Doug Farrar

76

Sound Connectors

Save $$ on audio output connectors by Dave Wilson, June 30, 1989

MASS STORAGE

PAR Tech. Report #4, Effects of Real Time Devices on Floppy Disk Performance by Scott Sarnikowski

IWM Chip spec, Apple number 343S0041-A

SWIM Chip spec, Apple number 343S0061-A

SWIM2 Specification by Steve Smith

Specification for Apple Floppy Disk Controller (New Age) by Scott Sarnikowski

85C80 Chip Spec, Advanced Micro Devices publication number 12582

53C80 Chip spec, Apple number

NCR 53C94, 53C95, and 53C96 Advanced SCSI Controller Data Sheet

NETWORKING & COMMUNICATIONS

SCC Serial Ports

Zilog Z8030/Z8530 SCC Serial Communications Controller Technical Manual

Macintosh Family Hardware Reference (pp 20-8 to 20-13), Addison-Wesley

Inside Appletalk, Sidhu, Andrews, & Oppenheimer, Addison-Wesley, 1989

Appletalk Network System Overview, Apple Computer, Addison-Wesley, 1989

PAR Technical Report #2, Effects of Reduced Processor Response on Localtalk by Scott Sarnikowski

Revised Summary of requirements for Localtalk SCC, a memo by Scott Sarnikowski (8/24/90)

Curio (79C950) Objective Specification, by AMD

ADB

Apple Desktop Bus Specification (revision D)

Macintosh Family Hardware Reference (pp 19-13 to 19-25)

Ethernet

DP83932 Data Sheet, by National Semiconductor

79C9416 Data Sheet, by National Semiconductor

ISO standard 8802-3 : 1989 (ANSI/IEEE Std 802.3-1988), 1989 (available from IEEE)

Chefcat

High Speed Serial Bus, P1394/Draft 3.2v2, by Mike Teener

ChefCat Architecture, Mike Teener

Felix ERS by Dan Hillman