EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

DEVELOPING FOR THE MACINTOSH NUBUS

B.G. Taylor

CERN, Geneva, Switzerland

invited paper presented at the
EuroBus 89 Conference, London
4 - 6 September 1989

DEVELOPING FOR THE MACINTOSH NUBUS

B.G. Taylor

EP Division, CERN, 1211 Geneva 23, Switzerland

Abstract

This paper presents introductory guidelines for European developers of NuBus electronics modules for the Apple Macintosh II family of personal computers. It is based on experience at CERN in developing MacVEE, which interfaces the Macintosh to multi-crate VMEbus and CAMAC systems for data acquisition in large high-energy physics experiments.

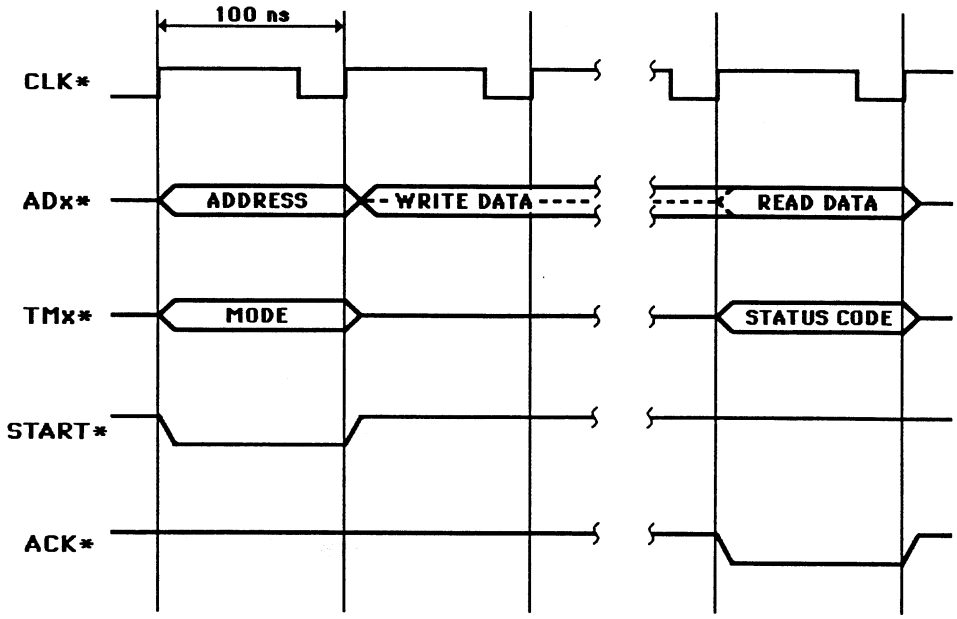
1. Introduction

While a collaborative truce is being declared by the major open systems bus-war participants, the personal computer market continues to be stimulated by a spirit of lively competition. After navigating quieter waters for over seven years, NuBus emerged into the turbulent mainstream of personal computing with the launch of the Macintosh II in March 1987. Prior to its adoption by Apple Computer, the bus was little used outside artificial intelligence workstations by Texas Instruments and Lisp Machine. But as the central architecture of Apple's growing family of 'Modular Macintosh' computers, over 250,000 NuBus systems have now been installed worldwide. Apple Europe revenue for 1988 exceeded $1000M, and Europe is now a substantial market for third-party developers of NuBus products.

The Apple NuBus is based on minor adaptations of ANSI/IEEE Standard 1196 [1], a concise 70-page specification published in August 1988. This standard is in turn a development of the Texas Instruments NuBus specification [2] first published in 1983, and 'NuBus' is a trademark of Texas Instruments Incorporated. The original NuBus was conceived by C. Terman and S. Ward at the Massachusetts Institute of Technology, and was developed by MIT and Western Digital Corporation between 1979 and 1983.

NuBus entered the controversial personal computing world just after full 32-bit microprocessors moved into high-volume production, and as the established industry standard architecture based on the PC/AT bus was showing signs of serious fragmentation. Because of Micro Channel licensing restrictions, board size limitations and compatibility issues, different manufacturers chose to follow the patented MCA and open EISA routes in their PS/2 clones. To add to the diversity, IBM themselves introduced an 80286 version of the PS/2 Model 30 based on the older AT-style bus, but for PC-height cards, while confirming that additional MCA enhancements will be introduced in the new 33 MHz Model 75, presumably using undefined reserved lines in the original specification.

Since the formation in April 1988 of NuGroup, the association of NuBus manufacturers and users, many organizations have gathered forces around the alternative NuBus architecture



Transfer Mode in START cycle

TM1	TM0	
1	1	Write Byte
1	0	Write Word/LongWord
0	1	Read Byte
0	0	Read Word/LongWord

Status Code in ACK cycle

TM1	TM0	
1	1	Transfer Complete
1	0	Bus Error
0	1	NuBus Timeout Bus Error
0	0	Try-Again-Later

Attention cycle (START.ACK)

TM1	TM0	
1	1	Attention-Null
1	0	Reserved
0	1	Attention-Resource-Lock
0	0	Reserved

Figure 1 Basic NuBus Transaction

and over 60 companies have already introduced compatible NuBus products. Other major computer and workstation manufacturers, while not yet participating actively in NuGroup, are reported to be preparing powerful NuBus designs. NuGroup membership is open to European organizations, and enquiries should be directed to:

NuGroup
12747 Barrett Lane
Santa Ana, CA 92705, USA
Attn: Anne H. Weber, Secretary

This paper reviews aspects of NuBus interfacing relevant to European developers of new products for Macintosh II computers. It draws attention to some possible hurdles and minor pitfalls, and includes initial guidance and pointers to sources of necessary information and useful tools. Some currently unresolved issues are also mentioned.

The discussion is largely based on experience at CERN in developing MacVEE [3] (Microcomputer Applied to the Control of VME Electronic Equipment), which interfaces the Macintosh family to multi-crate CAMAC (IEC 516) and VMEbus (IEC 821) systems. This has allowed large multi-processor data acquisition systems [4] using these international instrumentation standards to be controlled and monitored through the friendly graphics-oriented Macintosh user interface. MacVEE is now manufactured in France, Germany and Italy and over 500 systems are currently in service, mainly in European high-energy physics laboratories.

2. NuBus features

NuBus supports the most essential requirements of a modern 32-bit modular multiprocessor system backplane on a single 603-2-IEC-C096 connector carrying only 51 active signals (see Table 1). The NuBus signals can be classified as follows:

Utility signals	4
Transaction control signals	4
Slot identification	4
Address/data/parity	34
Arbitration signals	5
Power	23
Ground/reserved	22

The basic bus protocol for an elementary NuBus transaction is indicated in Figure 1. The bus is generally described as a synchronous one because the signal timing is phased with a fixed system clock, of duty cycle 25% and frequency 10 MHz in the current standard implementation. It might be more accurately defined as a time-quantized asynchronous bus, since the time which can elapse between the generation of a *start cycle* by a master and the *ack cycle* response from a slave can be any integer multiple of the clock period. The protocol is quite independent of any particular processor architecture.

The NuBus ID<3..0>* lines are not bussed, but binary coded at each slot to support the geographical addressing of up to 16 modules. As a result, NuBus boards generally do not require address configuration switches or jumpers. The address and data lines AD<31..0>* are multiplexed to convey address information during the *start cycle* at the beginning of a transaction and 8-bit byte, 16-bit halfword or 32-bit word data later in the transaction. The data size, transfer direction and byte lanes are indicated during the *start cycle* by the transfer mode lines TM<1,0>* in conjunction with AD<1,0>*. During the *ack cycle* which terminates a transaction, the transfer mode lines are coded to signal normal completion or temporary or permanent fail conditions. In the case of temporary failure, which may be due to local locking of the accessed resource, the master is encouraged to *try-again-later.*

NuBus supports multiple masters with distributed arbitration, using a mechanism based on that first described by Taub [5] implemented on the bussed lines RQST* and ARB<3..0>* in conjunction with ID<3..0>*. This eliminates the jumpering of empty slots required in daisy chain systems and allows a fair arbitration scheme in which all masters are peers, but any master may continue its tenure indefinitely by a bus lock. Within a bus lock, a master may also lock shared resources such as multiport memories. Such resource locking and unlocking is controlled by issuing special *attention cycles* during which both START* and

ACK* are asserted together. Higher level multiprocessing conventions such as message-passing protocols are not specified.

All addressable NuBus resources are in a single 4 Gbyte space, of which the lower 15/16 are uncommitted and the top 256 Mbytes are divided into 16 slot spaces of 16 Mbytes each, such that the slot space of slot S is referenced by addresses $F(S)XXXXXX. Each NuBus module requires a ROM at the top of its slot space to define slot occupancy, and this can accommodate auto-configuration code.

	A	B	C
1	-12	-12	RESET*
2	RSVD	GND	RSVD
3	SPV*	GND	+5
4	SP*	+5	+5
5	TM1*	+5	TM0*
6	AD1*	+5	AD0*
7	AD3*	+5	AD2*
8	AD5*	-5.2	AD4*
9	AD7*	-5.2	AD6*
10	AD9*	-5.2	AD8*
11	AD11*	-5.2	AD10*
12	AD13*	GND	AD12*
13	AD15*	GND	AD14*
14	AD17*	GND	AD16*
15	AD19*	GND	AD18*
16	AD21*	GND	AD20*
17	AD23*	GND	AD22*
18	AD25*	GND	AD24*
19	AD27*	GND	AD26*
20	AD29*	GND	AD28*
21	AD31*	GND	AD30*
22	GND	GND	GND
23	GND	GND	PFW*
24	ARB1*	-5.2	ARB0*
25	ARB3*	-5.2	ARB2*
26	ID1*	-5.2	ID0*
27	ID3*	-5.2	ID2*
28	ACK*	+5	START*
29	+5	+5	+5
30	RQST*	GND	+5
31	NMRQ*	GND	GND
32	+12	+12	CLK*

Table 1 NuBus Pin Assignments

The NuBus electrical specifications and timing parameters facilitate low-cost interface design with standard 74AS TTL and CMOS PALs, while LSI chip support is also becoming available. 74F TTL logic may also be used for backplane drivers, but it should be noted that the input loading of some 74F buffers exceeds NuBus receiver specifications. Allowing a maximum driver turn-on time of 35 ns and a backplane propagation delay of 17 ns, receiver signal setup time should exceed 21 ns at the NuBus connector.

Metastability issues [6] are avoided in NuBus by phasing most bus signal transitions with the system clock, although due attention must still be paid to the conditioning of external asynchronous inputs. Figure 2 shows a block diagram of the CERN module which interfaces the NuBus to remote VMEbus and CAMAC crate controllers in a MacVEE system. The state diagram for the interface is shown in Figure 3.

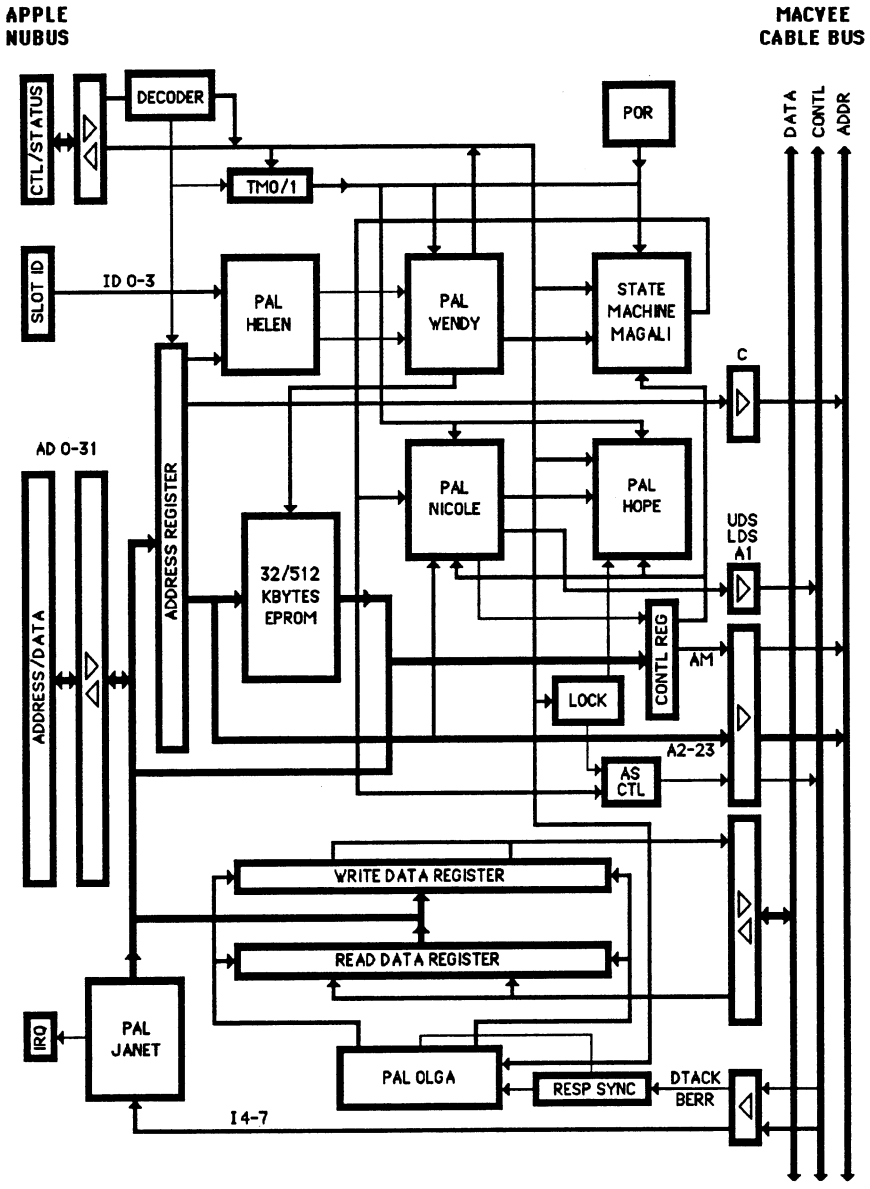


Figure 2 MacVEE NuBus Interface Block Diagram

With its accessible specification, switch and jumper-free conception, reliable single indirect connector and practical board area (about 44% greater than a Micro Channel card) NuBus offers an attractive alternative architecture for 32-bit desktop computing. The bus structure has survived well its first major exposure in a mass-produced personal computer, and appears favourably placed for even wider acceptance in the future [7].

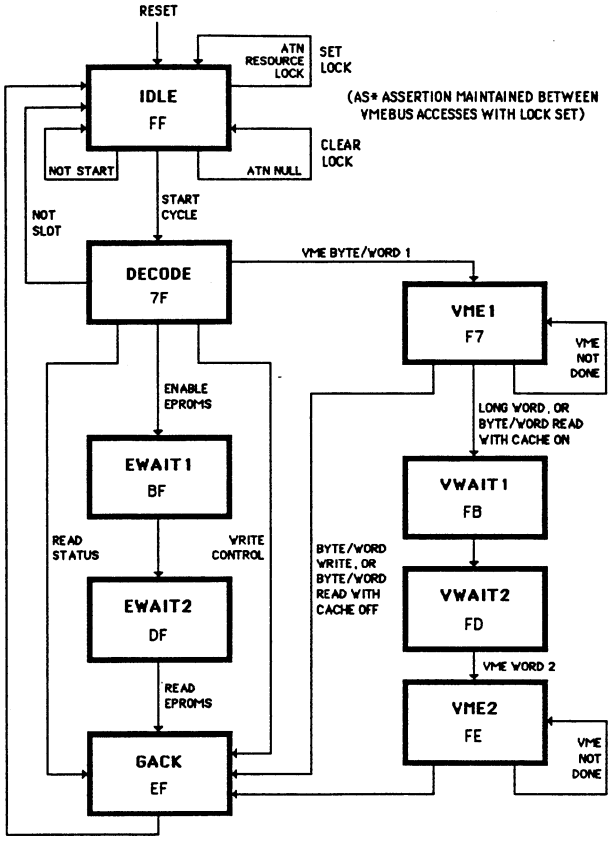


Figure 3 NuBus/VMEbus Interface State Diagram

3. NuBus standards and licensing

The primary reference for NuBus developers is the 'IEEE Standard for a Simple 32-Bit Backplane Bus: NuBus', ANSI/IEEE Std 1196-1987. The document also includes ANSI/IEEE Std 1101-1987, 'IEEE Standard for Mechanical Core Specifications for Microcomputers', which describes mechanical and environmental specifications for Euroboard assemblies. The NuBus 96-pin DIN connectors themselves are described in IEC 603-2-1980, 'Two-Part Connectors for Printed Boards, for Basic Grid of 2.54 mm (0.1 in), With Common Mounting Features'.

3.1 Origin and licensing

Std 1196 is based on an extension of the Texas Instruments NuBus Specification, document number TI-2242825-0001. While Texas Instruments owns issued and pending patents that relate to the ANSI/IEEE Standard, TI grants royalty-free licences to implement NuBus on nondiscriminatory terms and conditions for a one-time fee of $2000. So far such licence agreements have been available only from TI in the USA, but a European version of the licence is currently being formulated. For up-to-date information, potential licensees should contact the Apple Europe Third-Party Licensing Coordinator, Valérie Soupre, or:

Texas Instruments, Inc
12501 Research Boulevard
Austin, TX 78759, USA
Attn: NuBus Licensing, M/S 2151

A number of technical changes to the TI NuBus specification were made for Std 1196. A resource locking mechanism has been defined. Single-ended termination of the Address/Data and Control lines has been permitted, and the driver requirement for these lines has been reduced from 48 mA to 24 mA. The reserved line C23 has been allocated a Power Fail Warning function and may also be used by a NuBus card to turn the main power supply on or off. This could be used by a module which monitors an auto-answer modem for incoming calls. The former grounds A2 and C2 have been reserved for future use, while the former ground A31 has been assigned a Non-Master Request function. The NuBus timeout logic has been simplified to handle the case of no slave response, but not that of a master which acquires the bus but fails to generate a *start cycle.*

3.2 Updating Std 1196

Some minor errors in the TI specification have been carried through to the ANSI/IEEE Standard. In the write transaction timing diagram (Fig 4 of Std 1196) address, transfer mode and start lines should be driven until time D(2) and not as shown. (The block write diagram is correct). Std 1196 assumes incorrectly that worst case arbitration timing occurs with competing masters in slots 5 and A. More recent simulation studies by Texas Instruments show that the specified maximum ARB turn-on time of 83 ns (Fig 15 of the Standard) must be reduced to 40 ns to allow for 4 transitions of ARB0* in an arbitration among modules in slots 4-7, 9 and A.

The NuBus arbitration logic equations which are printed correctly in the TI specification appear to have been erroneously transcribed to Std 1196. (ARB<3..0>* and ID<3..0>* should read ARB<3..0> and ID<3..0> in the equations). In implementing this bus arbitration function it should be noted that the inverters shown in Fig 11 of the Standard should be open collector devices.

The IEEE Microcomputer Standards Committee has re-opened Std 1196 for the correction of errors and the integration of the P1394 high speed serial bus which has been under development for the past two years. Other compatible changes to the standard, including the introduction of optional 20 MHz block transfer operation, are also being considered. The serial bus could be allocated to the reserved pins A2 and C2, while double-rate block transfers could be implemented by re-assignment of the NuBus -5.2v lines.

It is hoped to complete a draft of the revised standard before the end of 1989. Organizations wishing to participate in this activity should contact the secretary of the P1196-R NuBus Revision Working Group:

Michael D. Teener
Peripheral Systems & Products
Apple Computer, Inc
3535 Monroe Street, M/S 69-I
Santa Clara, CA 95051, USA

Std 1196 describes the NuBus concepts and specifies the protocol and physical parameters in a most readable and practical manner. This encourages straightforward implementations and compatibility between modules developed independently by different vendors.

However, the development of Macintosh II NuBus products requires considerably more than compliance with Std 1196, since many details of the memory mapping, byte lane usage, interrupt handling, utility functions, error control, and configuration and driver firmware are determined by the Macintosh hardware and system software rather than the NuBus specification. Since up-to-date information on these aspects has proved somewhat less accessible to European developers, they will be emphasized in this paper.

4. Apple NuBus

The term 'Apple NuBus' is used to refer to the subset of the Std 1196 NuBus features which are implemented in the Macintosh II computer, a block diagram of which is shown in Figure 4.

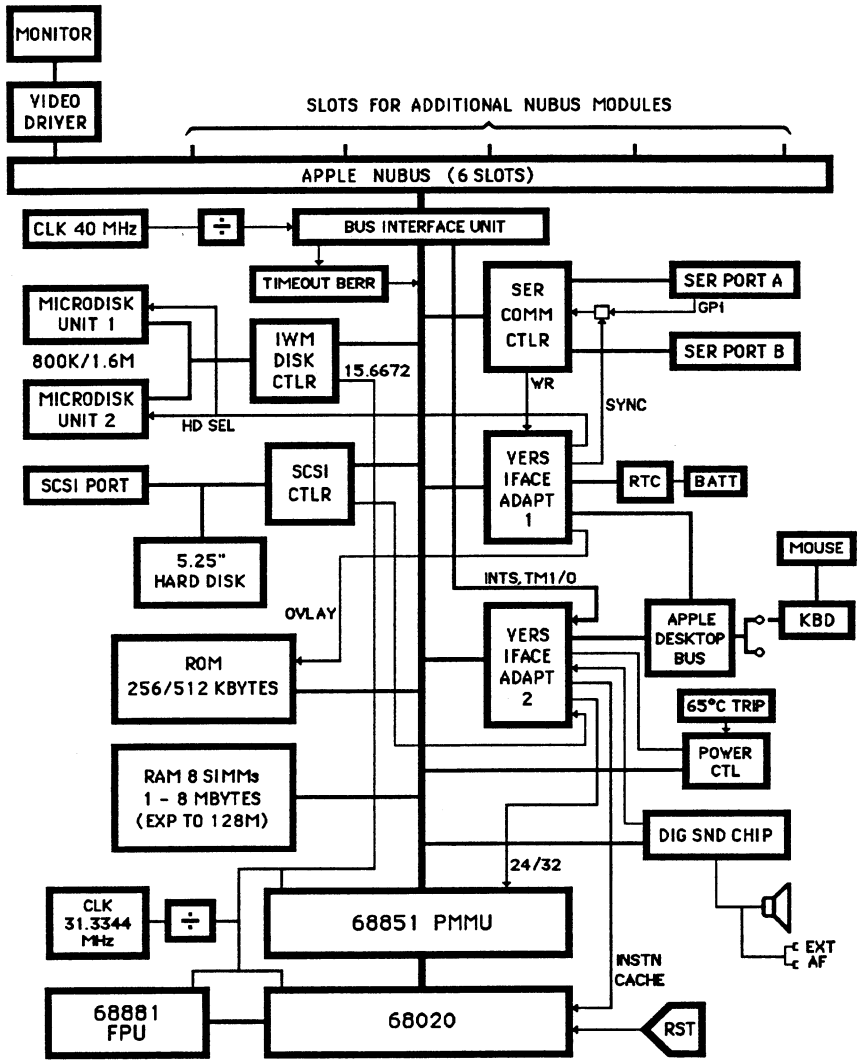


Figure 4 Macintosh II Block Diagram

The principal special characteristics of the Apple NuBus subset are currently as follows:

4.1 Board configuration

Std 1196 specifies alternative triple height and PC-style form factors for NuBus modules. The triple height format was used in TI's original Explorer Lisp computer, and is appropriate for rack-mounted crates of electronics with a high functionality per card or large number of channels per module. The Apple NuBus implements only the more compact PC-style format, which is more convenient for desk-top systems but may require greater use of SMDs or ASICs. The introduction of an intermediate-size double height Eurocard format is currently the subject of discussion within NuGroup.

4.2 Power supplies

The -5.2v supply specified in Std 1196 is not provided in Macintosh II family computers. However the 8 pins allocated to this supply are bussed to all Apple NuBus slots, so that if the voltage is generated by a converter on one NuBus module it may be utilized by circuitry on any of the others. Some or all of these pins may be re-assigned by the P1196-R study.

Std 1196 does not specify the maximum power available for each NuBus slot. For Apple NuBus modules, it is recommended that designers observe limits of:

+5v 2A max +12v 175 mA max -12v 100 mA max

Users of the Macintosh II/IIx are advised to respect a total loading from all slots of:

+5v 12.8A max +12v 1.3A max -12v 1.0A max

For the Macintosh IIcx the total loading should not exceed:

+5v 6.0A max +12v 525 mA max -12v 300 mA max

The power requirements of Apple NuBus modules should always be specified by their manufacturer to allow users adding expansion cards to calculate their power budget in accordance with the directions given in the owner's manuals.

Even for NuBus modules which receive power from an external source, the total dissipation by expansion cards in a Macintosh II should be maintained below 90W to prevent undue temperature rise within the cabinet. Protection circuitry shuts down the system if the temperature exceeds 65°C.

4.3 Block transfers

Std 1196 defines a transaction mechanism for NuBus block transfers of length up to 16 words (64 bytes). Since there is no flow control mechanism for a block read initiated by a NuBus master, it must be able to accept data as rapidly as 100 ns/word. As this rate exceeds that possible with the Macintosh II, block transfers are not currently implemented by the computer.

The use of maximum-length block transfers would permit a maximum theoretical NuBus throughput of 37.5 Mbytes/sec. Without the block transfer mode, a minimum of two bus cycles of 100 ns is required to transfer each 32-bit word, reducing the maximum throughput to 20 Mbytes/sec. Because of additional delays in the Macintosh II internal NuBus interface, the rate at which data can be exchanged between a NuBus module and Macintosh II RAM is in practice typically less than 4.5 Mbytes/sec.

4.4 Reset

Std 1196 requires that NuBus Reset should be asserted for a minimum of 1 ms. While the Macintosh II hardware generates a Reset of 200 ms duration, the ROM startup code subsequently asserts the signal for a period of 33 µs. This second Reset starts 3 µs after the line has been released by the hardware reset following power-on or caused by pressing the computer reset switch.

When Restart is selected from the Special menu in the Finder, the code generates two 33 µs Reset signals on the NuBus separated by about 3 µs.

4.5 Bus parity

As the Macintosh II does not currently implement NuBus parity checking, the System Parity and System Parity Valid signals are pulled high. Both signals may be ignored by NuBus modules which do not check parity.

4.6 Slot addresses

Std 1196 permits NuBus implementations with up to 16 slots. The Macintosh II and IIx have six and the Macintosh IIcx has three physical slots, one of which is normally occupied by a video display interface.

The slots have hard-wired IDs S from $9 to $E ($9 to $B for the IIcx), and the main logic board ROM and I/O can be addressed from NuBus as virtual slot 0. However, Apple Computer recommends that the Macintosh I/O devices should not be accessed from NuBus modules. The slot space scheme specified in Std 1196 is extended by the definition of 256 Mbyte superslots referenced by addresses $(S)XXXXXXX.

4.7 Interrupts

Std 1196 describes interrupt generation by Event Transactions or Non-Master Requests. The Macintosh II implements only the Non-Master Request mechanism, which allows the generation of Macintosh interrupts by NuBus slaves without the arbitration logic required to become masters. Event Transactions could be used to post interrupts to other NuBus modules having appropriate address monitors.

Std 1196 permits the Non-Master Request pins to be bussed or not bussed between NuBus slots. In the Macintosh II individual lines are run from each slot to a Glue chip in which they are OR'd. Both the individual NMRQ* signals and their logical OR are input to VIA2, which can generate a level 2 autovectored interrupt to the processor.

4.8 Configuration ROM

While Std 1196 does not specify the content of the configuration ROM, an Apple NuBus module must have a configuration ROM (called the declaration ROM) containing firmware with a particular internal structure. The firmware defines slot resources which allow the module to be uniquely identified by the Macintosh Slot Manager, and may also include an icon, data, initialization or driver code. While Texas Instruments NuBus modules use a Forth-like language which can be interpreted by different types of processor, the firmware of Apple NuBus modules is native 680X0 code.

5. NuBus developer documentation

A range of documents, in addition to ANSI/IEEE Std 1196, is required for Apple NuBus development for the Macintosh II.

5.1 Inside Macintosh Library

The Inside Macintosh Library, produced by Apple Computer and published by Addison-Wesley, includes the following titles. They are readily available from computer bookshops.

Technical Introduction to the Macintosh Family

Macintosh Family Hardware Reference

Designing Cards and Drivers for Macintosh II and Macintosh SE

Inside Macintosh Volumes I-III, Volume IV and Volume V

Inside Macintosh X-Ref

Cards and Drivers and *Inside Macintosh Volume V* are indispensible, and *Inside Macintosh Volumes I-III* and *IV* also contain relevant material. The treatment in *Technical Introduction* is quite superficial and, in spite of the promising title, *Family Hardware Reference* contains very little information about the NuBus implementation with the exception of a description of the VIA which is involved in interrupt handling.

Inside Macintosh Volumes IV and *V* contain descriptions of hundreds of changes and additions to the original material in *Volumes I-III*. *X-Ref*, which provides a 128-page integrated index to material in the other volumes, can also be procured from Apple Computer as a set of MPW text files on disk. A looseleaf edition of *Inside Macintosh* plus *X-Ref* is now available. While no formal updates are planned, this allows revised material from later volumes to be appended to the original chapters. A CD-ROM called *Developer Helper*, which contains a comprehensive selection of resources for developers, has recently been issued by Apple Computer.

Pioneer Apple NuBus developers will have worked with preliminary versions of these publications supplied through APDA, the Apple Programmer's and Developer's Association. Even the Final Draft APDA documents contained major discrepancies and omissions and it is indispensable to procure the Addison-Wesley volumes. In addition the first bound editions are not without errors and a developer should attempt to procure the very latest versions. It should also be noted that changes are sometimes introduced between different printings of the same edition.

The following sections of *Inside Macintosh Volume V* are of particular interest to NuBus product designers:

1 Compatibility Guidelines

19 The Start Manager

23 The Device Manager

24 The Slot Manager

25 The Deferred Task Manager

33 The Operating System Utilities

Appendix C: System Traps

Additional Device Manager information is given in *Volumes II* and *IV*. The Start Manager orchestrates all of the activities related to system testing and startup. The Device Manager provides the routines for NuBus device I/O and maintains an interrupt queue for each slot, while the Slot Manager enables programs to communicate with NuBus cards.

The Slot Manager is very flexible and has numerous calls, only a few of which will be required for a typical NuBus module. Apple recommends that developers use only the *principal* or *specialized* routines, not the *advanced* routines which are used by the operating system. In particular, one should avoid re-initializing the Slot Manager with _InitSDeclMgr. The Deferred Task Manager is useful for handling NuBus card interrupts which initiate lengthy tasks. The Operating System Utilities include routines to control the address mapping, which is described later.

All Macintosh toolbox and operating system routines are implemented as A-line traps. Developers should be aware that the descriptions in *Inside Macintosh* are frequently cryptic and sometimes erroneous. Code examples are rare, the material relevant to a given topic is often somewhat dispersed, and experimentation or disassembly are required to clarify the operation of some traps. The _StripAddress trap, which may be used in NuBus card drivers, is incorrectly documented in the Operating System Utilities section of *Volume V* (see TN#213 below).

5.2 Macintosh Technical Notes

Unfortunately, even the 3,200 pages of documentation cited above do not include all the information the designer requires. Both Macintosh hardware and system software are in continuous and rapid evolution, in keeping with Apple Computer's thrust towards sustained innovation, and developers require to track these changes with minimum delay. For this purpose Apple's Macintosh Developer Technical Support Group publishes *Macintosh Technical Notes* which expand on and clarify the publications and discuss errors found in software, hardware and documentation.

Although over 700 pages of Technical Notes have been published, rather few concern NuBus interfacing issues. The following contain relevant material:

TN#0	About Macintosh Technical Notes
TN#2	Compatibility Guidelines
TN#117	Compatibility: Why and How
TN#129	_SysEnvirons: System 6.0 and Beyond
TN#148	Suppliers for Macintosh II Board Developers
TN#212	The Joy of Being 32-bit Clean
TN#213	_StripAddress: The Untold Story
TN#221	NuBus Interrupt Latency
TN#228	Use Care When Swapping MMU Mode
TN#234	NuBus Physical Designs - Beware

While it is possible to subscribe to *Technical Notes* in printed form or on disk, and a HyperCard stack version has been introduced, they are also distributed over electronic mail networks such as Usenet, as described later.

5.3 NuBus Designers' Workbook

A practical guide for NuBus developers has been published by Eclipse Technology, Inc. The *NuBus Designers' Workbook for the Mac II* is based on the company's experience in designing interface cards for the Macintosh II. It includes details of their SuperSCSI DMA-driven SCSI port, which achieves sustained transfer rates up to 4.4 Mbytes/sec using either asynchronous or synchronous SCSI devices.

The workbook, which is claimed to save NuBus developers several months of research, is available for $175 from:

Eclipse Technology, Inc
6714 Marbo Court
Falls Church, VA 22046, USA
Attn: Joseph B. Wooldridge

6. NuBus developer support

During the early years following the initial introduction of the Macintosh, the European hardware designer was very much the Cinderella of the Apple developer family. Apple Computer's prime support service, the Certified Developer Program, was restricted to the USA. Time-consuming reverse engineering of circuit boards and code was required to clarify details of the computer's hardware and software which were inadequately documented or considered proprietary secrets. During the period of maximum turmoil at Cupertino [8], technical enquiries from overseas developers would typically be ignored or referred to salespersons who were not equipped to respond. Repeated promises of improved information flow and support in Europe were not fulfilled.

From January 20 1989, the Certified Developer Program, which was free of charge, was replaced by new Apple Partners and Associates Programs. The full Partners Program, which is open to developers of NuBus products which will be marketed commercially, requires a sign-up payment of $750 and annual renewal fee of $600. The developer must be based in the USA.

6.1 Apple Europe

Recognizing that almost a quarter of Apple's total revenue comes from Europe, with an installed base of over 500,000 Macintoshes here, a recent reorganization created Apple Europe as one of the company's four main operating units. Apple Europe, which has 12 subsidiaries and 15 independent marketing companies, has a declared commitment [9] to improved developer relations and European project support. *Apple exchange*, a newsletter for European developers and distributors, has been launched with a circulation of over 3000 copies. European commercial and non-commercial developer programs have been established and a developer contact has been appointed in each Apple Europe subsidiary. Details may be obtained from:

Apple Computer Europe
Cedex 51
92065 Paris La Défense, France
Attn: Developer Services - R&D Europe

6.2 APDA

From January 19 1989, the administration of APDA, the Apple Programmer's and Developer's Association, has been taken in-house by Apple Computer and is being entirely reviewed under the guidance of the Developer Channels group within Apple Product Marketing. APDA distributes developer documentation and software from Apple Computer and third parties to some 20,000 members worldwide, and has local chapters in several European countries.

Individual developers or representatives of larger organizations in Europe can join APDA for an annual fee of $35 by contacting:

Apple Computer, Inc
20525 Mariani Avenue
Cupertino, CA 95014-6299, USA
Attn: APDA, M/S 33G

The fee covers the cost of a quarterly information catalogue from which members may order developer documentation and tools. Applicants have to represent to Apple that they are 'a developer of computer software and/or hardware and have substantial experience in computer

programming and in using programmer tools and utilities'. European APDA applicants must also file a letter of assurance restricting re-export to specified countries and defining product end-use.

It is possible to communicate with APDA from Europe by sending an Earnet/Bitnet message to:

xb.das@stanford.bitnet (xb$das for some mailers)

with a subject line:

apda@applelink

An actual subject can be entered by adding an exclamation mark after applelink, as for example:

Subject: apda@applelink!NuBus Interfacing

7. Usenet for NuBus developers

Only US Apple Partners receive official consultative technical support from Macintosh Developer Technical Support (Mac DTS) by e-mail and relatively few European sites currently have access to the AppleLink electronic mail and information service, which is implemented over General Electric Information Services (GEIS) networks from a host computer in Ohio.

However, any NuBus developer can tap a large resource of relevant expertise through Usenet [10], the international network of Unix users which now links over 17,000 sites and 580,000 participants. Average Usenet activity is about 3000 messages per day, totalling some 6.6 Mbytes of traffic. No site in Europe is far from an organization which can provide a Usenet newsfeed and the cost of participation for local sites (but not for backbone sites) is modest. A PC/AT compatible can support a minimum system.

News reading software such as *readnews*, *rn* or *vnews* is in the public domain, and it is only necessary to subscribe to those newsgroups which are of interest to activities at the local site. For NuBus development the following are relevant:

comp.binaries.mac	(moderated by macintosh-request@felix.uucp)
comp.sources.mac	(moderated by macintosh-request@felix.uucp)
comp.sys.mac	
comp.sys.mac.digest	(moderated by info-mac-request@sumex-aim.stanford.edu)
comp.sys.mac.hypercard	
comp.sys.mac.programmer	
comp.unix.aux	

The total traffic level in these Macintosh groups currently exceeds 500 articles per week. July 1989 statistics indicate that comp.sys.mac was taken by 96% of all Usenet sites, where it was read by 40,000 participants at a cost ratio of $0.09/month/reader. Provided the local system has adequate disk storage it is not necessary to scan all articles regularly, as *rn* has powerful facilities for searching by pattern in subjects, headers or whole articles.

The non-moderated newsgroups provide a forum for the discussion of problems and solutions with other developers, and informal exchanges with engineers at Apple Computer in Cupertino. Participants can post technical queries for distribution to the local site, to

Europe only (Eunet), or to the entire Usenet world. They can post follow-up articles on subjects of widespread interest, or exchange detailed technical dialogue with other individuals by e-mail.

The following departments of Apple Computer can be contacted by e-mail to an address of the form:

{*Address*}@applelink.apple.com

Destination	*Address*
APDA	APDA
Bug Report Center	Apple.Bugs
Customer Relations	C.Relations
Developer Programs	DevServices
Developer Technical Support	MacDTS
Licensing	SW.License
Public Relations	Apple.PR
Tools and Compilers	Dev.Tools

Binary files, including executable code and formatted text or graphics, can be posted after processing by the encoding and compression utilities *BinHex* and *StuffIt*. New and revised *Macintosh Technical Notes* are posted in *BinHex* form to comp.binaries.mac, and some 20 - 30 Apple Computer employees regularly participate in the non-moderated Macintosh newsgroups in an unofficial capacity. Apple participants have even assisted by passing technical questions to appropriate specialist colleagues at Cupertino who are not on Usenet, and relaying their responses.

8. Network archives for developers

In July 1989 Apple Computer opened an FTP site which archives *Macintosh Technical Notes*, tools, sample code and documentation. Developers with FTP access to Internet hosts should FTP to apple.apple.com (130.43.2.2) using account:*anonymous* and password:*guest*. Once logged on, change directories to pub/dts/ and get the *readme* file which explains the archive content and structure.

A current list of all the archived files is maintained in the help/ directory together with a list of the most recent additions. The following mac/ subdirectories are available:

docs	Macintosh technical documentation
hacks	Useful, unsupported hacks
mpw	Current MPW interface files
q+a	Macintosh Q & A stack
sc	Macintosh sample code
sys.soft	System software information
tn	Macintosh technical notes

Because of licensing restrictions, the tools distributed by APDA are not available from this archive.

The moderators of the comp.sys.mac.digest Usenet newsgroup maintain a major archive at sumex-aim.stanford.edu (36.44.0.6), which may be accessed using FTP with account *anonymous*, any password. Help files are in /info-mac/help and recent-addition and

complete indexes in /info-mac/help/recent-files.txt and /info-mac/help/all-files.txt respectively. The archive currently contains 1900 files, the majority in *BinHex* form.

For the convenience of European developers a copy of the Stanford info-mac archive is available on the Earnet server IRLEARN. It is updated from sumex every night. From IBM *VM* it suffices to enter *tell macserve at irlearn dir* to retrieve the list of recent additions, or *tell macserve at irlearn dir all* to access the complete file list. To retrieve *Technical Note 234,* for example, one would then enter *tell macserve at irlearn get tn/tn234.hqx.* IRLEARN keeps track of requests, and at present only 10 commands per day per account are accepted.

9. Macintosh II addressing

The 32-bit address space of the Macintosh II is allocated to the computer motherboard and NuBus resources as shown in Figure 5. While the 680X0 microprocessors can address the full 4 Gbyte address space, the current (System 6.0 series) full-release versions of the Macintosh operating system are restricted to 24-bit addressing and the high-order byte of a master pointer contains flags used by the Memory Manager.

It is planned to introduce 32-bit addressing with System 7.0, seed versions of which will be available to developers in autumn 1989, and at some time 32-bit addressing will be made standard on new Macintosh computers. Meanwhile Apple recommends that 32-bit addressing conventions and methods be used whenever possible as they provide the best guarantee of future software compatibility. 32-bit addressing guidelines are given in *Technical Note 212.* The A/UX operating system runs entirely in 32-bit mode.

9.1 24/32-bit addressing

A 24-bit mode of memory mapping has been implemented for the interim situation and to maintain compatibility with applications which were developed for the earlier 68000-based Macintoshes. System trap macro _SwapMMUMode (system trap word $A05D) allows switching between the 24-bit and 32-bit modes of address translation. Enter with D0=1 to change to 32-bit mode and with D0=0 to restore 24-bit mode.

SwapMMUMode supports both the standard Address Mapping Unit (AMU) and the optional 68851 Paged Memory Management Unit (PMMU). It executes in about 60 µs with the former and 70 µs with the latter. In 24-bit mode the NuBus superslot spaces are inaccessible and the lower 1 Mbyte of each of the six slot spaces $F(S)0XXXXX is mapped to $(S)XXXXX.

Inside Macintosh Volume V warns that in 32-bit addressing mode some toolbox routines may not function properly and the Memory Manager cannot be used, although interrupt handlers do work in either 24-bit or 32-bit mode. As the Memory Manager has to be present before any patches can be loaded, a ROM update may be required for 32-bit operation even with System 7.0 but Apple is researching alternative solutions.

Hence when designing a NuBus module which requires a total address space of less than 1 Mbyte it should be located at the bottom of the slot space, and superslot space avoided. By decoding on the module only the 24-bit addresses $(S)XXXXX, it can be accessed by addresses of the form $F(S)(S)XXXXX in both 24-bit and 32-bit modes.

9.2 Mapping problems

The version of the Slot Manager in Macintosh II ROMs shipped up to mid-January 1988 does not run in 32-bit mode, so that only NuBus modules whose declaration ROM aliases

from the upper to the lower 1 Mbyte of the slot space can be recognized. As it may prove troublesome for users to have the early computers updated (a main logic board swap is required for the versions with soldered-in ROMs), it is desirable to provide for such aliasing in designing the address decoding of all NuBus modules.

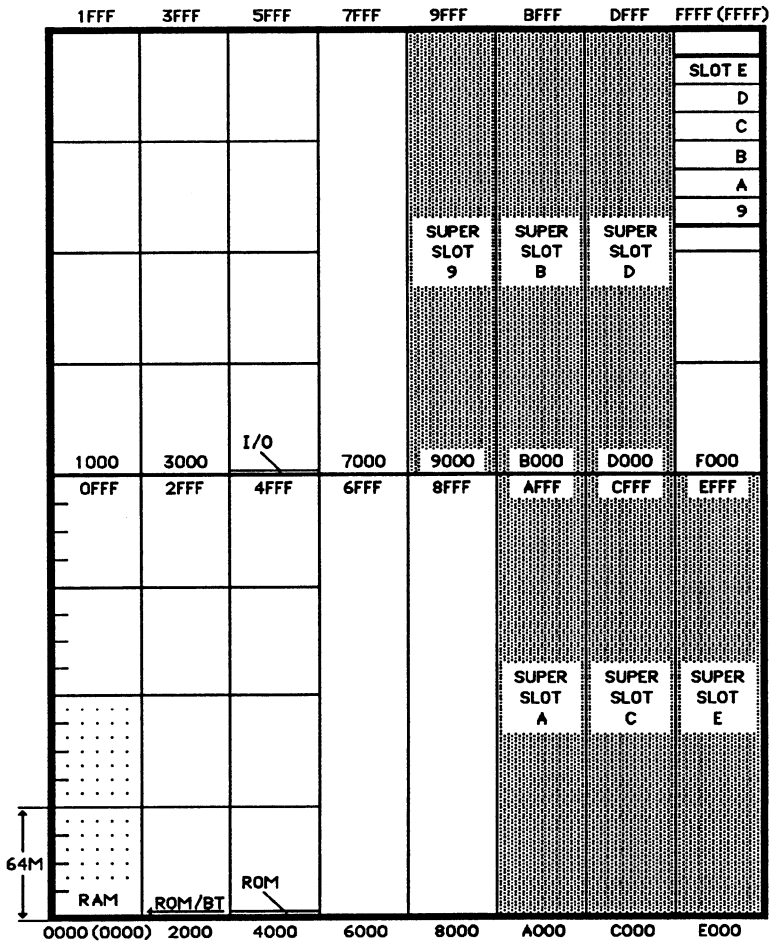


Figure 5 Macintosh II Address Map

Where this is not possible, the modules may be shipped with a 'NuBus Tester' application which is available from Apple Computer. The application alerts the user that an upgrade is required if it detects that the original version ROM is installed. Object files are also available which can be linked to allow a CheckNuBus call from the module's application. The original dialog should be retained, as it was set up by Apple's legal department.

NuBus modules which require the full 16 Mbyte slot space or the 256 Mbyte superslot space can only be accessed in 32-bit mode. An application written to run in this mode should at present switch to 24-bit mode before calling any Macintosh II ROM routine and return to 32-bit mode thereafter. For example, the code for handling a NuBus timeout bus error which occurred in 32-bit mode should switch to 24-bit mode before calling the Dialog Manager to display an alert box.

It is also important to ensure that the high byte of the program counter is clear before switching to 32-bit mode. If code is loaded into memory as a resource (by _GetResource),

the high byte of the master pointer contains memory manager information which would appear in the program counter if a JSR to the code is performed (typically by JSR (A0) with the master pointer in A0). This applies to stand-alone executable code resources such as ADBS, FKEY, INIT and HyperCard XCMD and XFCN, but not to standard CODE resources. The problem may be avoided by cleaning up the program counter from within the resource code before the _SwapMMUMode trap:

```
LEA             *+$000C,A0
MOVE.L          A0,D0
_StripAddress
MOVEA.L         D0,A0
JMP             (A0)
```

10. Apple NuBus data transfers

The Macintosh II NuBus interface comprises a bidirectional Bus Interface Unit (BIU) together with transceivers, encoding and decoding logic. The BIU consists of four state machines which implement the control logic for accesses to and from the NuBus, and for handling resource locked transactions and error conditions.

10.1 Data byte placement

The interface implements byte swapping to preserve byte addressing consistency between the big-endian 680X0 microprocessors and little-endian NuBus ordering. This byte swapping does not affect the ordering of bits within bytes. Apple NuBus documentation, such as Fig 4-2 in *Cards and Drivers*, should not be interpreted as indicating that there are fixed data paths between the NuBus and microprocessor byte lanes. The byte lane mapping shown applies only to 32-bit word transfers. During 680X0 byte reads, for example, NuBus lanes 0, 1, 2 or 3 are routed to D31 - D24 of the microprocessor according to the address being accessed, and D23 - D0 are unused.

However, it is not necessary to implement a corresponding lane demultiplexer on a NuBus module to access an on-card byte-wide device, such as a declaration ROM. The device may be associated with a single NuBus byte lane, and the data read by the Slot Manager at every 4th byte address. Program code such as a device driver will be re-packed in Macintosh RAM before execution. The Slot Manager verifies that the data format corresponds with the ByteLanes value at the head of the format block by checking the test pattern $5A932BC7.

The byte placement is important for any NuBus module that stores data or code which may be accessed by other processors, as when the Macintosh is used as a software development workstation for multiprocessor VMEbus systems [11]. Table 2 shows the translation required in this case between the data byte lanes, lower-order address bits, active VMEbus data strobes and NuBus transfer mode signals for byte, word and longword interchange between the Apple NuBus and VMEbus modules with single even and odd byte (D08(EO)) and double byte (D16) capability.

10.2 Non-aligned transfers

The NuBus interfaces of Macintosh II family computers support all the non-alignment functionality of the 680X0 microprocessors. The interfaces generate multiple standard NuBus transactions with appropriate 680X0 dynamic bus sizing DSACK responses until the non-aligned access is completed. In addition to 680X0 longword accesses at odd addresses, non-aligned accesses are required for certain bit-field instructions which have 3-byte operands.

680X0 operation	NuBus TM1	TM0	A1A0	Data	VMEbus A1	Data strobes	Data	
WRITE								
Byte 0	1	1	0	D31-D24	0	UDS	D15-D8	
1	1	1	1	D23-D16	0	LDS	D7-D0	
2	1	1	2	D15-D8	1	UDS	D15-D8	
3	1	1	3	D7-D0	1	LDS	D7-D0	
Word 0/1	1	0	1	D31-D16	0	UDS.LDS	D15-D0	
2/3	1	0	3	D15-D0	1	UDS.LDS	D15-D0	
Lword	1	0	0	D31-D16	0	UDS.LDS	D15-D0	(1st op)
				D15-D0	1	UDS.LDS	D15-D0	(2nd op)
READ - Data Cache Off								
Byte 0	0	1	0	D31-D24	0	UDS	D15-D8	
1	0	1	1	D23-D16	0	LDS	D7-D0	
2	0	1	2	D15-D8	1	UDS	D15-D8	
3	0	1	3	D7-D0	1	LDS	D7-D0	
Word 0/1	0	0	1	D31-D16	0	UDS.LDS	D15-D0	
2/3	0	0	3	D15-D0	1	UDS.LDS	D15-D0	
Lword	0	0	0	D31-D16	0	UDS.LDS	D15-D0	(1st op)
				D15-D0	1	UDS.LDS	D15-D0	(2nd op)
READ - Data Cache On								
Byte 0	0	1	0					
1	0	1	1					
2	0	1	2	***All cases:***				
3	0	1	3					
				D31-D16	0	UDS.LDS	D15-D0	(1st op)
Word 0/1	0	0	1	D15-D0	1	UDS.LDS	D15-D0	(2nd op)
2/3	0	0	3					
Lword	0	0	0					

Byte numbering per VMEbus specification, byte swapping per Apple NuBus specification

Data:	D31 ---- D24	D23 ---- D16	D15 ---- D8	D7 ---- D0
Apple NuBus Backplane:	AD7 ---- AD0	AD15 ---- AD8	AD23 ---- AD16	AD31 ---- AD24

Table 2 NuBus/VMEbus Data Interchange

All non-aligned reads can be completed by a maximum of two NuBus transactions, since the additional bytes acquired by 32-bit word reads can be ignored by the processor. In the case

of non-aligned writes, however, up to three NuBus transactions may be required to split a 680X0 longword into component byte and halfword writes in the appropriate byte lanes. NuBus cards which can be accessed by non-aligned transfers must be able to support a full 32-bit read. In general, all RAM-like cards should support all NuBus data transfer types.

10.3 Data caching

Apple specifies that, in order to support the data caching implemented by 68030 and later microprocessors, NuBus slave modules should supply all 32 data bits in response to every type of NuBus read request. It is recommended that general memory devices always respond in this way.

For a NuBus interface to an 8-bit or 16-bit data channel, this could result in unnecessary overhead when used in a 68020-based Macintosh II. For such more specialized cards, which are controlled by their own device drivers, the access mode may be software controlled by declaring some address ranges as non-cacheable.

10.4 Resource locking

The Macintosh II NuBus interface implements both bus locking and resource locking mechanisms. NuBus accesses generated by Macintosh processor read-modify-write instructions are automatically framed with *attention-resource-lock* and *attention-null* cycles, so that the mechanism is transparent to the programmer.

The interface also detects resource locking by other masters, and locks out the Macintosh processor if motherboard resources are subsequently addressed by them. If the Macintosh processor cannot complete its current operation because it also requires NuBus access, it relinquishes the local bus and retries the bus cycle after the NuBus master has generated an *attention-null cycle*. In the case where the Macintosh processor was itself initiating a read-modify-write cycle, it is aborted by a bus error and must be retried by software. In time-critical situations bus locking may be implemented by NuBus modules to optimize the data transfer rate. Apple recommends that the bus should not be locked for more than four data transfers before unlocking for rearbitration.

While the bus locking and resource locking mechanisms are specified in Apple documentation such as *Cards and Drivers*, attention is drawn to the fact that the following statement has been added to *Inside Macintosh Volume V* between the APDA Final Draft and the first Addison-Wesley edition (Section V-4, Compatibility): '... the TAS instruction, which uses a special read-modify-write memory cycle, is not supported by the Macintosh SE and Macintosh II hardware.' While no official explanation has been given for this retraction, it has been reported that problems can arise when the Macintosh floating point coprocessor is being accessed. A motherboard HAL change introduced in 1988 is also believed to relate to NuBus multiprocessor support in the computer. Apple has not yet responded to requests to clarify these issues or specify the motherboard upgrade policy in Europe.

The Macintosh II can be programmed to lockout NuBus access to motherboard resources through PB1 of VIA2. This mechanism is used by Macintosh system software to protect time-critical processor activity, such as floppy disk transfers via the IWM controller. The NuBus interface generates a *try-again-later* status response to attempted accesses from other masters while this lock is set.

10.5 Error detection

NuBus slave status responses of TM<1,0>* = X,H (*error* and *try-again-later*) generate a Macintosh II 680X0 microprocessor bus error. One of the BIU state machines generates an

ack cycle with TM<1,0>* = H,L if a NuBus transaction fails to complete within 25.6 µs, and this also causes a bus error.

To identify these errors the status code can be read from data register B of VIA2 (TM1 = PB4*, TM0 = PB5*). Since the VIA always indicates the status code of the latest NuBus transaction it must be read by the error handler before any subsequent NuBus operation, such as video card access.

11. Apple NuBus interrupts

Macintosh II interrupt levels are assigned as follows: level 1 - VIA1, level 2 - VIA2, level 4 - SCC, level 6 - PwrIRQ, level 7 - NMI. Devices in any of the NuBus slots can generate non-master interrupt requests which result in a level 2 autovectored interrupt to the processor through CA1 of VIA2. NuBus interrupts are enabled at startup, but are disabled when using MacsBug. If an interrupt is generated for which no service routine has been installed processing halts and an error dialog box is displayed.

11.1 Interrupt mechanism

NuBus interrupts are enabled by setting bits 7 and 1 in the VIA2 interrupt enable register, and may be disabled by setting bit 7 and clearing bit 0. VIA2 interrupts can originate from other sources, such as internal timers, the ASC and SCSI interfaces. When a NuBus interrupt occurs, bit 1 is set in the VIA2 interrupt flag register.

The Macintosh II ROM Device Manager, or alternative user-written interrupt-handling code, determines the NuBus slot which has generated an interrupt by reading VIA2 data register A. Interrupts from slots 9 - E result in active low data bits PA0 - PA5 respectively. The Device Manager maintains an interrupt queue for each slot interrupt. The slot queue element contains a priority parameter which determines the order in which the slots are polled and the routines called. The Device Manager resets the VIA2 interrupt flag and executes an RTE to the interrupted task when a polling routine indicates that the interrupt is satisfied.

11.2 Interrupt latency

The current Macintosh II system software average overhead time, from the generation of a NuBus interrupt to entry of the interrupt service routine, is about 160 µs. The interrupt handler may then require to arbitrate for NuBus mastership and may be held off by other modules, which could even lock the bus.

The interrupt latency can be considerably greater when certain floppy disk, LocalTalk and other tasks are active, and in view of this Apple has declared that the Macintosh II should not be considered a realtime computer. Adequate data buffering must be provided on NuBus modules used in realtime applications, and particular attention should be given to the manually-initiated operations which can significantly extend the interrupt response time.

Inserting a blank floppy disk disables interrupts for up to 25 ms, and formatting it disables interrupts for up to 300 ms. Some floppy disk accesses (block reads) can disable interrupts for up to 15 ms, while LocalTalk accesses can disable interrupts for up to 22 ms.

All slot interrupts, including slot VBL interrupts, hold off other slot interrupts until their interrupt routines have run to completion. VBL tasks may be of varying length since applications, as well as drivers, can install VBL tasks. Cursor updating is done at slot VBL time and can hold off all other slot interrupts for 700 - 900 µs. In *Technical Note 221* Apple provide code which defers the cursor updating routine to run at a lower priority interrupt level

which will not hold off NuBus interrupts. Some flickering of the cursor is visible when this technique is used.

12. Interface variants

Macintosh NuBus developers should note that there exist systematic differences in some characteristics of the backplane signals between the different models of the Macintosh II family, and even between different versions of the same model. In addition to relative signal timing and termination arrangements, secondary signal characteristics such as negative undershoot and noise immunity vary between the different Macintosh II models. It is therefore advisable to fully test new designs in each of them.

From about mid-1988 an 84-pin PLCC 'Nuchip' custom VLSI circuit has been fitted to Macintosh II motherboards in place of the set of HALs used in the NuBus interface of computers shipped since March 1987. Although the signal timing of both versions is well within Std 1196 specifications, a number of NuBus video cards from different third-party manufacturers initially had compatibility problems with the new logic boards, requiring module design revisions. In the case of the so-called 'cost reduction' logic board using the Nuchip, the NuBus START* and TM<1,0>* signals are asserted 18 ns and released about 7 ns earlier than with the original motherboard.

The Nuchip is also used on the Macintosh IIx and IIcx logic boards. In the case of the Macintosh IIcx the START* signal, which is pulled up to +5V on the Macintosh IIx and both versions of the Macintosh II logic board, has a Std 1196 +3V termination. Depending on the NuBus module loading, ringing can exceed the lower threshold 13 ns after it is first crossed, a characteristic which is quite absent in the original HAL implementation of the interface.

Apple Computer has stated that signal changes caused by the lower capacitance of the three-slot Macintosh IIcx NuBus backplane can result in the malfunctioning of Apple EtherTalk cards in that computer, particularly in a machine used as a fileserver having no other cards installed. Revisions K and later of the EtherTalk card have been modified to accommodate this condition and an update policy is being defined.

13. NuBus LSI support

LSI circuit implementations of the principal NuBus interface functions can save designers substantial time and development effort compared with solutions using PLDs and standard MSI. Both the NuBus board area and the interface power required can be reduced substantially by the use of LSI logic.

13.1 Pinnacle Micro NuChips

Pinnacle Micro has introduced a set of custom HCMOS gate arrays called NuChips. (These are not the same as the 'Nuchip' VLSI circuit used on the Macintosh II 'cost reduction' logic board cited above). Two identical NuChips are interconnected as shown in Figure 6 to implement a 32-bit NuBus slave interface, the HORL input of the high order device being connected to ground. In addition to the NuBus slot ID comparison function, NuChips decode a ROM address space with handshake generation.

While separate address and data registers are provided, they are multiplexed to a common user bus by mode control inputs. A minimum of 4 NuBus clock cycles is required for a read or write transaction. NuChips are packaged in 84-pin PLCCs, characterized for 0°C to 70°C operation, and cost $19.95 per set.

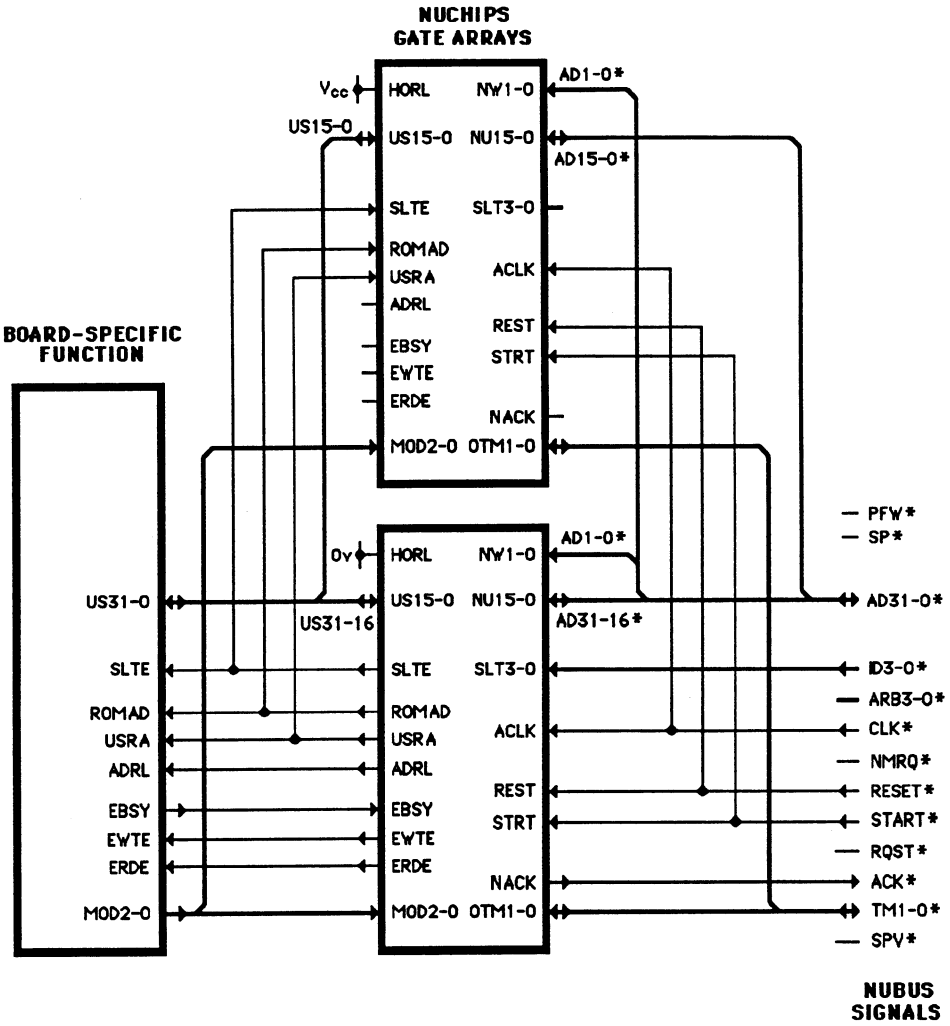


Figure 6 Pinnacle NuChips Interface Block Diagram

13.2 TI interface chipset

Texas Instruments has introduced a NuBus generic interface chipset comprising the SN74ACT2440 controller and the SN74BCT2420 address/data transceiver. The ACT2440, which is built with TI's 1-μm EPIC advanced CMOS process, includes NuBus master, slave and arbitration logic and provides the buffer control signals to drive two BCT2420s.

The interface controller is TTL-compatible and its NuBus outputs are designed to drive the terminated backplane directly. The BCT2420 utilizes an advanced BiCMOS process featuring bipolar drive characteristics but low standby power. It has three 16-bit I/O ports to handle all the multiplexing and demultiplexing of the address/data lines. Both devices are available in 68-pin PLCCs and are characterized for 0°C to 70°C operation.

The block diagram in Figure 7 indicates how the chipset is interconnected to implement a 32-bit master/slave interface for a NuBus module. Texas Instruments claim that the three-chip solution replaces as many as 45 discrete logic devices, and uses 66% less board space and 90% less power.

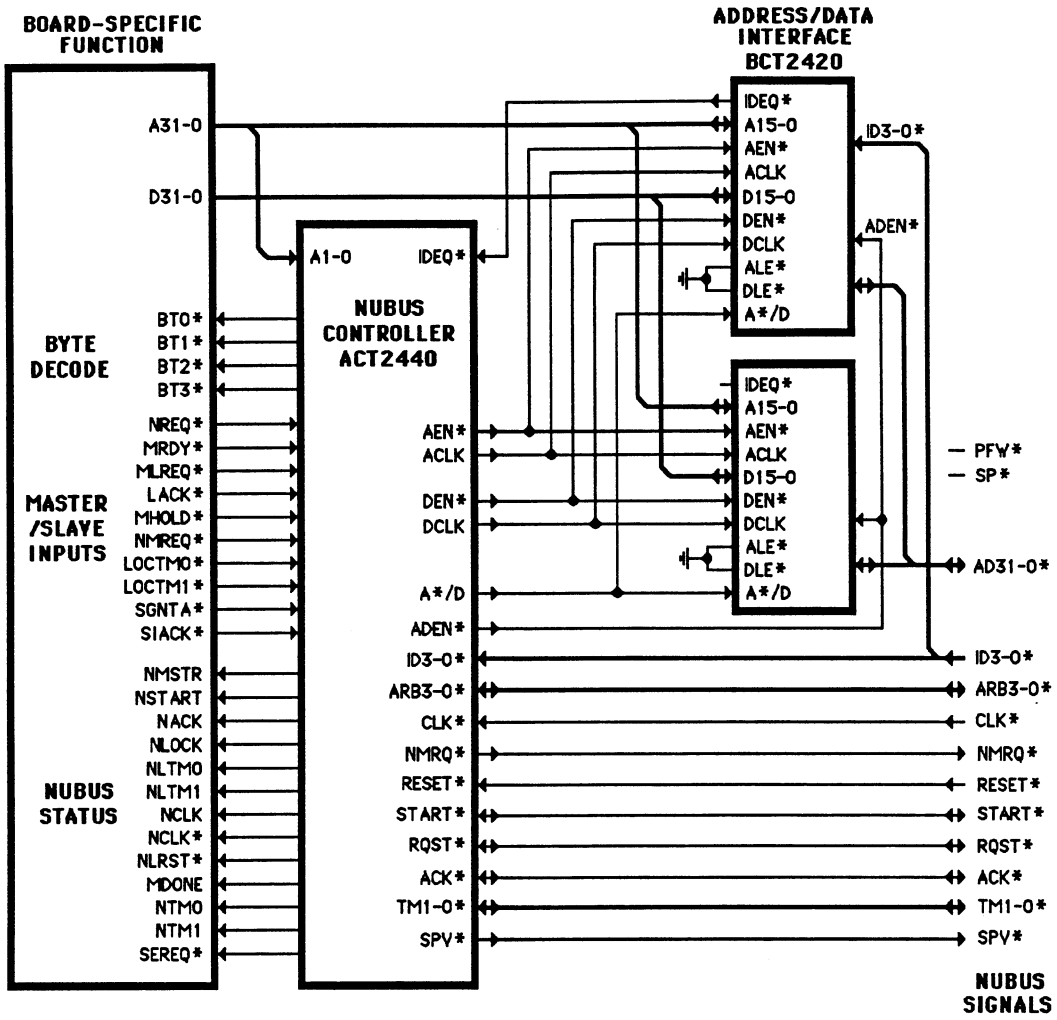


Figure 7 TI Chipset Interface Block Diagram

The NuBus controller incorporates master and slave state machines synchronized to the NuBus clock. The board-specific logic interacts with the controller through a group of control and status signals which allow it to request NuBus master read or write accesses with automatic implementation of bus arbitration when required. The controller generates *attention-resource-lock* and *attention-null* cycles if requested by a master lock request input, and it incorporates a 16-state master timeout.

The NuBus slot ID code comparison function is provided in the address/data transceiver part, while the controller provides the logic for slave read or write accesses from the NuBus to on-board resources. It detects NuBus resource locking and also provides a mechanism to resolve access conflicts for local circuitry which can be both master and slave. The NuBus status response and low-order address signals are decoded to provide four enables for the active data byte lanes.

The TI interface chipset provides a simple connection to NuBus for master-only, slave-only [12] and master/slave modules. Additionally, it provides extra status and control lines to facilitate more sophisticated configurations. In quantities of 1000, the ACT2440 is expected to cost $24 and the BCT2420 $13.33.

13.3 VTC interface controller

Using their VL2000 ECL process, VTC Inc developed a prototype NuBus controller chip designated VS630, and second silicon was scheduled during 1989. However, VTC have indicated that they do not currently plan to bring this device to market.

14. NuBus hardware

NuBus designers may require basic or intelligent prototyping boards, extender cards, module expansion shields, backplanes or complete powered NuBus chassis for their developments. At present the number of sources of such hardware is quite limited.

14.1 Basic prototyping boards

With the increasing availability of CAE systems for schematic capture, logic simulation and pc board layout, interface designers have less need for conventional circuit prototyping on development breadboards. But ready-made NuBus prototyping boards can prove very convenient to users without such facilities who require to assemble one-off interfaces for special applications. Apple NuBus bare prototyping boards are available in Europe from:

ATEC
Applied Telecom Technologies
Homburgstrasse 10
D-1000 Berlin 49, West Germany
Attn: Guido Körber

In single quantity the 'MacBreeder' breadboard, a single-sided FR4 glass fibre prototyping board with 4270 plated solder pads for user circuitry, costs DM 89.

Other vendors of prototyping boards include:

Augat, Inc
33 Perry Avenue
Attleboro, MA 02073, USA

InterLogic Industries
85 Marcus Blvd
Melville, NY 11747, USA

Bare prototyping boards and extender boards are manufactured by:

Diversified I/O, Inc
1008 Stewart Drive
Sunnyvale, CA 94086, USA

Vector Electronic Co
12460 Gladstone Avenue
Sylmar, CA 91342, USA

The following vendors offer prototyping boards which are equipped with NuBus slave interface logic and a generic declaration ROM, with a wire-wrap area for additional user circuitry:

ADEX Corporation
1750 Junction Avenue
San Jose, CA 95112, USA

Creative Solutions, Inc
4701 Randolph Rd, Suite 12
Rockville, MD 20852, USA

The ADEX MacProto board has address and data latches, an 8-bit bidirectional user port, a 27C64 declaration EPROM, and an address decoder and basic slave state machine implemented with four GALs. About 60% of the board area is available for wire-wrapping.

The Hurdler II board from Creative Solutions provides a NuBus slave interface to a Zilog Z8536 CIO counter/timer and parallel I/O unit, implemented with five PALs and a bus

transceiver. The 8536 provides a 12-bit user I/O port and three 16-bit timers, and it supports NuBus interrupt generation. Alternative 50-pin IDC and DB25 Centronics parallel connectors can be mounted on the board, which is equipped with a 2716 declaration EPROM containing a comprehensive device driver. A schematic and driver listing are provided, and about 40% of the board area is available for wire-wrapping.

14.2 Macintosh Coprocessor Platform

The prototyping of intelligent NuBus I/O cards is supported by the Macintosh Coprocessor Platform (MCP) from Apple Computer. The MCP is equipped with a 10 MHz 68000 microprocessor with 64 Kbytes of ROM, and can accommodate from 512 Kbytes to 4 Mbytes of RAM. Different versions of the MCP module have been produced. In one version the on-card 16-bit data/24-bit address bus is interfaced to NuBus by two special integrated circuits developed by Texas Instruments.

The MCP NuBus 16-bit interface controller circuit contains master, slave and arbitration logic together with controls for the on-board 68000 and memory. A companion NuBus transceiver circuit handles the multiplexed transmission of addresses and data. Both devices are packaged in 100-pin quad flat packs which are claimed to replace about 20 programmable and discrete logic chips. About 45% of the MCP board area is available for custom wire-wrapping.

Apple supports third-party development using the MCP with A/ROSE (Apple Real-time Operating System Environment) which provides task scheduling, memory management, integrated inter-process communication (Apple IPC) and dynamic naming. IPC services are currently available under the Macintosh O/S and are under development for A/UX. The Apple IPC driver handles message-passing between Macintosh applications and A/ROSE applications running on the MCP or on another computer. A diagnostic application is available, together with support code and examples to test user-developed MCP hardware.

A/ROSE is a compact (under 15 Kbytes) fully distributed message-based operating system with a fast context switching time of 200-250 µs (for the MCP 68000). It currently supports 256 tasks per card and 32 priority levels. A/ROSE enquiries should be directed to:

Apple Computer, Inc
10500 N. De Anza Blvd
Cupertino, CA 95014, USA
Attn: Susan Grabau, M/S 27C

MCP card and software licensing fees will be set individually in each European country but are expected to be close to $500 for a complete package including documentation and developer support.

14.3 Apple NuBus expansion shields

Apple NuBus modules should be equipped with a metal expansion shield which provides mechanical support for the board, offers a secure mounting for any I/O connectors, and enhances the EMI shielding in the Macintosh II. This part, which is variously called by Apple a connector shield, expansion port cover, or card I/O fence, is available from Vector Electronic Co (see above) and the following suppliers:

Galgon Industries, Inc
37399 Centralmont Place
Fremont, CA 94536, USA

North American Tool and Die
999 Beecher Street
San Leandro, CA 94577, USA

European suppliers of an equivalent standard part, or of customized components, include:

Globe Manufacturing GmbH
Weitzesweg 4
6368 Bad Vilbel 4, West Germany

ORKIS Images Numériques
52 rue du Coq
13001 Marseille, France

14.4 NuBus module dimensions

Std 1196 allows an overall tolerance of 5 mm for the height of the expansion shield and, while not rigorously defining the vertical dimension of the connector aperture in the shield, could be interpreted as indicating (in Fig 20) that it may approach 74.5 mm. Apple has specified that, to ensure compatibility with the rear apertures of all computers, the maximum connector dimension should be 66.81 mm, located from 15.44 mm above the reference plane. Popular 50-pin IDC connectors cannot be used if this restriction is observed.

However, *Technical Note 234* specifies that the rear aperture dimensions, which are 80 x 17 mm for the Macintosh II and IIx and 75.61 x 14 mm for the Macintosh IIcx, will not be less than 74.55 x 11.9 mm in future models.

In addition to NuBus module support fingers similar to those provided in the lid of the Macintosh II and IIx, the Macintosh IIcx has auxiliary card separators located about 10 mm behind the module expansion shields. The separators interfere with components in this area of height greater than 9.9 mm, which is less than the Std 1196 tolerance of 15.24 mm.

14.5 NuBus chassis

Third-party manufacturers have developed over 100 different modules in the PC-style NuBus format, and this range is increasing steadily. Over 20 NuBus CPU cards are already available, incorporating processors ranging from the Intel 80X86 for MS-DOS compatibility, higher speed versions of the Motorola 680X0 family, AT&T's 25 Megaflop DSP32, AMD 29000/29027 and Motorola 88100/88200 RISC processors, to Weitek's XL-8032 three-chip 20 Megaflop array-processor set and Inmos T800 transputers. A 20-transputer configuration (5 Levco TransLink NuBus boards) is claimed to attain Cray ray-tracing performance.

Supporting hardware is becoming available which allows the implementation of such NuBus systems outside the context of Macintosh II configurations. NuGroup has declared its support of the development of alternative workstation platforms and NuBus-based industrial/manufacturing control systems. NuBus backplanes, card cages and chassis are available from:

Advanced Control Technology
34 Steamwhistle Drive
Ivyland, PA 18974, USA

ELMA Electronic, Inc
41440 Christy Street
Fremont, CA 94538, USA

ISIS
2149 O'Tolle Ave., I
San Jose, CA 95131, USA

Standard Logic, Inc
4940 E. La Palma Avenue
Anaheim, CA 92807, USA

Powered standalone chassis in a Macintosh II form-factor enclosure for housing Apple NuBus modules are manufactured by:

Second Wave, Inc
9430 Research Blvd
Echelon II, Suite 260
Austin, TX 78759-6541, USA

The NuBus backplane of this company's Expanse II/SA chassis has eight slots on 1-inch centres (a somewhat wider spacing than in the Macintosh II family) and is equipped with system clock and bus timeout logic. Interface cards and a cable assembly are available which allow the chassis to be used as a Macintosh II expansion card cage. The additional NuBus slots can be accessed by the Macintosh in 32-bit mode, when slot spaces 1-8 and superslot spaces 6-8 can be addressed.

15. Industry consortium

The 32-bit NuBus architecture of the Macintosh II family opens these computers to new applications outside the office environment. Automatix Inc produces industrial versions of the Macintosh II for use on the factory floor. Their AI90 computer has EMI shielding, power conditioning and filtered ventilation and is housed in a NEMA-2 drip-proof rack mount enclosure.

Rack-mounted ruggedized Macintosh II system units for industrial environments are also produced by GreenSpring Computers. Industrial I/O NuBus cards are available as well as 68020-based modules supporting A/ROSE. The systems have a vibration-resistant housing with a heavy steel chassis, cast aluminium front bezel, and additional cooling and filtering.

In June 1989 a Consortium for Laboratory & Industrial Applications of the Macintosh was formed with the aim of working with Apple Computer to establish the Macintosh as a recognized computing platform in scientific laboratory and factory applications. Consortium membership, for which there is a fee of $1000 per corporate site, is open to both third-party vendors and end-users and enquiries should be directed to:

Consortium for Laboratory & Industrial Applications of the Macintosh
c/o Cirrus Technology
49 Midgley Lane
Worcester, MA 01604, USA
Attn: Shari Worthington

16. NuBus test instruments

General-purpose logic analyzers are frequently employed to capture sequences of bus transactions during system development. In the VMEbus world, dedicated bus analyzer modules have also proved popular as they trade versatility for portability, convenience and lower cost. A dedicated NuBus analyzer module priced from $2495 has been introduced by Applied Physics and Nissho Electronics. A BusLink module is also available which allows the NuBus to be interfaced to external test equipment. Enquiries should be directed to:

Applied Physics, Inc
1291-E Cumberland Avenue
Purdue Research Park
West Lafayette, IN 47906, USA

The N9300 Analyzer is a NuBus member of the BusTrak family which can capture up to 32,000 states of width 48 channels (43 on the NuBus, 5 external). The trigger may be positioned anywhere within the trace buffer and associated with NuBus states or external events.

The analyzer is a PC-style format NuBus module which is connected to an RS232C terminal for setup, control and the display of captured data. Slot tenure and bus mastership can be

identified during analysis. The module's firmware includes a menu-driven user shell with search facilities to scan for events within specified fields or conditions. Trigger commands allow specific bus events to be captured and a simple histogram mode permits real-time bus traffic analysis by slot space.

The NuBus analyzer samples bus states synchronously with the 10 MHz system clock and has no provision for higher-resolution timing analysis. Since the NMRQ* pins are not bussed in the Macintosh II, external channels must be used to connect to the appropriate slots to trace NuBus interrupts.

17. Apple NuBus card firmware

The designer of an Apple NuBus module must equip it with a declaration ROM containing slot resources (*sResources*) for each of its functions, an *sResource list* for each *sResource*, a *Board sResource list*, an *sResource directory* and a *format block*. The declaration firmware may be generated with any development system which supports assembler coding.

While Apple's Macintosh Programmer's Workshop (MPW) may be considered somewhat heavy artillery for this task, it is inexpensive and the assembler is supplied with Include files of Macintosh System, Declaration ROM and Slot Manager equates. Several debuggers are available, ranging from the low-level MacsBug (APDA, $35) to more expensive tools like TMON (ICOM Simulations, $125) and MacNosy (Jasik Designs, $350) having higher-level user interfaces.

An example of the structure of the declaration firmware for a NuBus module is indicated in Figure 8. Specifications are given in *Cards and Drivers*, and only additional clarifications are presented here.

17.1 Testing firmware

The declaration firmware should be tested by calling the Slot Manager routines and verifying the status returned in the low-order word of register D0 after execution. The routines are specified by setting D0 to one of 35 long-integer selectors before invoking the _SlotManager trap macro. The errors reported by non-zero status results are tabulated in *Inside Macintosh Volume V* (Section V-456/457). It should be noted that early Macintosh II ROMs do not contain an *sResource* for virtual slot 0.

17.2 The format block

The Length field in the format block should specify the number of bytes actually occupied by the declaration firmware, which is considerably less than the address range if all 4 byte lanes are not used. Similarly all offsets count only the bytes in the NuBus byte lanes being used, not the address difference. The CRC in the format block is calculated over the number of bytes specified in the Length field using the following algorithm:

```
    Start pointer at (Top of ROM - Length)
    Initialize CRC to 0
1   Rotate CRC left by one bit (with ROL.L #1 instruction)
    If pointer is at CRC field, goto 2
    Get a byte
    Add byte to CRC (with ADD.L instruction)
2   Increment pointer (by 1, 3 or 4) to next data byte
    Goto 1 until done
```

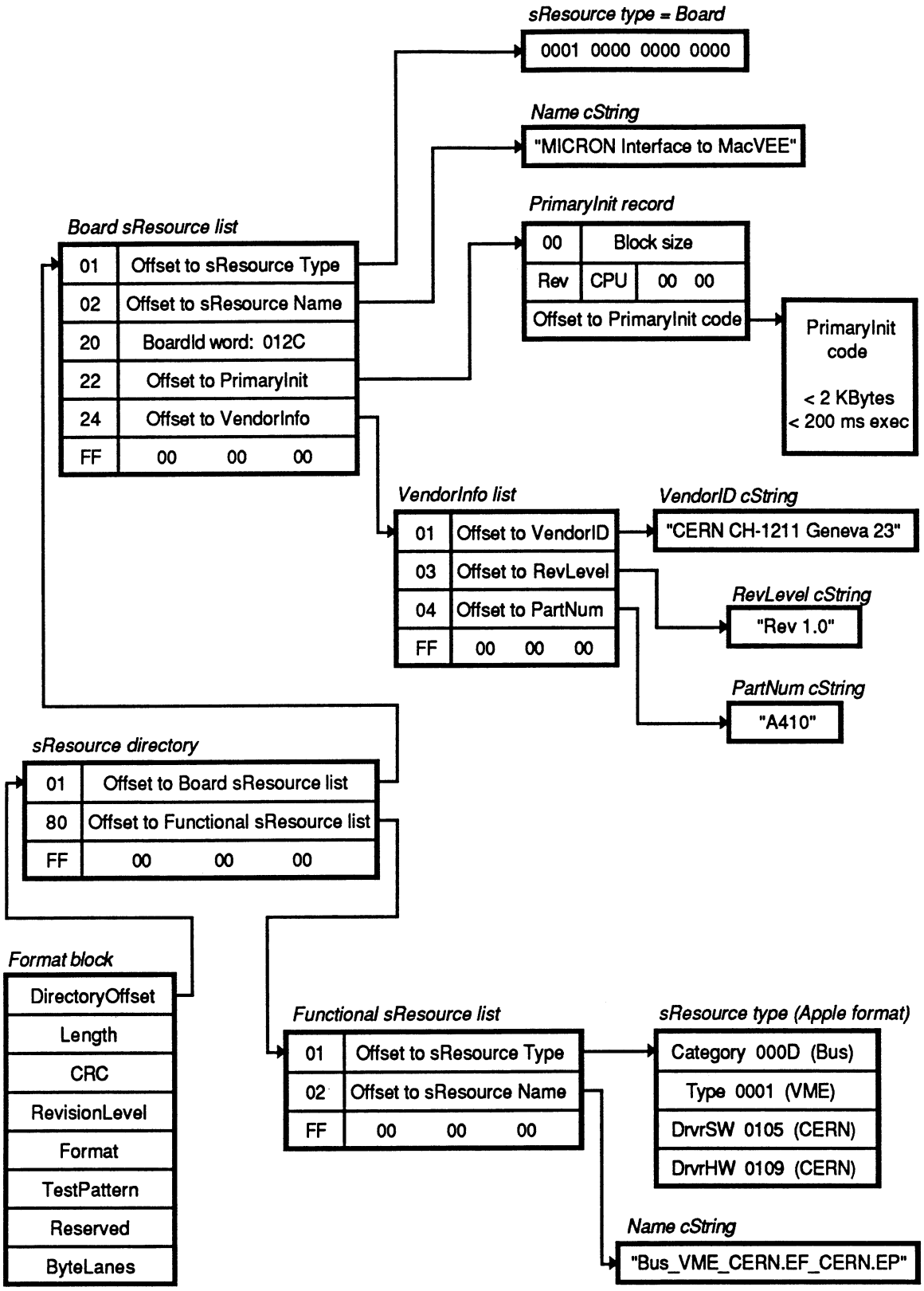


Figure 8 MacVEE NuBus Interface ROM Structure

While *Cards and Drivers* states that a crcPatch tool is supplied with MPW, it has not been included in any release through MPW Version 3.0. It can be procured from Macintosh Developer Technical Support (MacDTS), together with a data tool which strips off the code 0 segment, and puts the code 1 segment in a data file. crcPatch is run after the declaration ROM file has been assembled and linked, and before the code is converted to a data file for downloading to an EPROM programmer.

The ROM RevisionLevel field is for the version number of the declaration ROM, not of the Macintosh ROMs. Only values 1-9 are accepted by the Slot Manager. This revision level may be different from that of the board itself, which is specified by ID3 in the VendorInfo list referenced through ID36 in the Board sResource list.

Through ID34, the Board sResource list may also reference a PrimaryInit record which contains a RevisionLevel field for the board's initialization code. At least one version of the ROM Slot Manager only accepts a value of 02 for this parameter. Note that ID numbers assigned to entries in sResource lists are indicated by Apple in decimal, not hexadecimal, and that the entries must be in ascending numerical order.

The parameter in the ByteLanes field in the format block should indicate the byte lanes used by the declaration ROM itself, not those used for access to the hardware facilities actually supported by the module. For example, the MacVEE interface supports 32-bit NuBus transfers, but has a ByteLanes value of $78 for its byte-wide declaration ROM in lane 3. Although ByteLanes values are quoted in *Cards and Drivers* for four groups of three lanes, a module's declaration ROM area must be implemented with 8, 16 or 32-bit width.

17.3 The Board sResource list

The description of the Board sResource list in *Cards and Drivers* is somewhat misleading, and some important equates are omitted from the sample code. While the Board sResource list requires sRsc_Type and sRsc_Name entries, these are not composed as specified for standard sResources. Instead, the sRsc_Type entry should be an offset to the invariable 8-byte record $0001 0000 0000 0000, which may be interpreted as declaring to the Slot Manager the function of 'being a board'. The Name cString should in this case be the normal text description of the module. For example, Apple's standard video interface has the text 'Toby frame buffer card'.

The BoardId entry is a unique 16-bit word identifier for the module which is assigned to the designer by Apple Computer. It is obtained from MacDTS at Cupertino. Requests to 'Apple Technical Support', cited in *Cards and Drivers*, may be treated by salespersons and fail to reach the appropriate party. A HyperCard stack with integrated tutorial is available to facilitate the preparation of a request including the required data.

The Board sResource list of a NuBus module which can lock out the Macintosh processor may optionally include a TimeOut entry which specifies the number of re-tries to be attempted by the Slot Manager. A default value of 100 is used if this parameter is not specified.

17.4 The PrimaryInit record

The primary initialization code for the NuBus module is executed very early in the startup sequence, immediately after the motherboard diagnostic has terminated and the power-on beep generated, and before the startup screen's driver is opened. It may not call any Macintosh firmware except the Slot Manager. Adventurous developers have been known to access the video card directly from this code, but it should be noted that PrimaryInits are run for each NuBus module in ascending slot order, so that those for modules to the left of the video card will be executed before that for the video card itself.

The code must be less than 2 Kbytes in length and execute in under 200 ms. It should return a negative status value if a fatal error occurs when initializing the module, in which case the Slot Manager will be prevented from accessing it.

17.5 The VendorInfo list

As the Slot Manager does not use the cStrings in this list, the designer is allowed some freedom as to their content. For guidance, the declaration ROM of Apple's standard video interface contains the following:

VendorID: Apple Computer RevLevel: Beta-7.0 PartNum: TFB-1

17.6 The functional sResource lists

One functional sResource list should be included for each separate function which is supported by the NuBus module. At present the Category, cType, DrvrSW and DrvrHW equates are assigned on request by MacDTS. While standard Category and Type IDs will in due course be catalogued in MPW equates, the DrvrHW assignment is unique to each NuBus module design.

A unique DrvrSW equate is required for a module which has a special (non-Apple) driver interface. If this interface is fully defined, independent software developers can create applications which find all compatible NuBus boards knowing that the different types of hardware will be handled appropriately by their own drivers. This is done by calling SNextTypesRsrc with appropriate spCategory, spCType and spDrvrSW fields, and the spDrvrHW field masked off.

The Name cString referenced in the functional sResource list should be formed by stripping the Cat, Typ and DrvrSW/HW prefixes from the sResource type assignments and then linking them by underscores. For example, Apple's standard video card has the Name 'Display_Video_Apple_TFB'.

The modular structure of the declaration firmware permits convenient cross-referencing. For example, a NuBus module which supports only one type of function but has several modes of operation may have separate sResource lists for each mode. The sResource lists may share the same sResource type, name, icon and driver while referencing different mode lists, and these mode lists may share common parameter blocks where appropriate.

18. Conclusion

Technological developments in personal transportation, personal communications and personal computing can play a widespread role in society only when they are readily accessible to the non-specialist. Automobile engineers have eliminated the hassle of manual ignition timing, carburettor chokes and starting cranks. Radio engineers have obsoleted filament rheostats, antenna tuners and regeneration controls; while computer engineers have long discarded the console switch register, the card punch and the manual binary bootstrap.

By exploiting the NuBus architecture designers can now continue this progress by doing away with expansion card software installation procedures, module address switches and backplane daisy-chain jumpers. Complementing the graphical user-interface approach to software, this friendly approach to hardware integration represents a significant step towards allowing the user of a modern personal computer to 'just drive'. This is an important goal if the computer is to continue its evolution from an institutional calculating engine to a mass personalized knowledge tool for the 21st century.

Acknowledgements

This paper is largely based on experience at CERN in developing MacVEE, which was funded and supported by C. Rubbia and members of the UA1, ALEPH and OPAL experiment collaborations and the H1 Group at DESY, Hamburg. The author is particularly indebted to K. Bos, F. Cindolo, S. Cittolin, A. Contin, M. Demoulin, A. van Dijk, J. Dorenbosch, E. Eisenhandler, N. Ellis, J. Feyt, V. Frigo, P. Giacomelli, W. Haynes, W. Jank, V. O'Dell, P. Petta, E. Pietarinen, J.-P. Porte, W. von Rüden, F. Saldana, D. Samyn, H. Verweij, H. von der Schmitt and G. Walzel for their valuable assistance, and to U. Binder and J.W. Stutz of Industrade AG and J. Maurer of Apple Europe for their helpful cooperation.

References

[1] IEEE Standard for a Simple 32-Bit Backplane Bus: NuBus, *ANSI/IEEE Std 1196-1987*, IEEE, New York, August 1988.

[2] NuBus Specification, *Document No. TI-2242825-0001*, Texas Instruments Inc, Irvine, 1983.

[3] B.G. Taylor, Macintosh in the Laboratory, *CERN Report EP/88-67*, Geneva, 1988.

[4] W.J. Haynes, The H1 VME-based Data Acquisition System, *Proc VMEbus in Research Conf*, Zurich, October 1988, pp. 191 - 201.

[5] D.M. Taub, Contention-resolving circuits for computer interrupt systems, *Proc IEE*, Vol 123, September 1976, pp. 845 - 850.

[6] IEEE Standard for a Versatile Backplane Bus: VMEbus, *ANSI/IEEE Std 1014-1987*, IEEE, New York, 1987, Appendix D: Metastability and Synchronization.

[7] W.C. Nowlin, Is the NuBus really a better bus?, *Electronic Engineering Times*, 6 February 1989, pp. 56/73.

[8] J. Sculley with J.A. Byrne, *Odyssey*, Harper & Row, New York, 1987, pp. 227 - 359.

[9] N. Minkley, Editorial, *Apple Exchange*, Developer Services - R&D Europe, Apple Computer Europe, October 1988.

[10] B. Anderson et al, *UNIX Communications*, The Waite Group, Howard W. Sams & Co, Indianapolis, 1988, pp. 135 - 295.

[11] S. Cittolin, The UA1 VME Data Acquisition System, *CERN Report DD/87-7*, Geneva, 1987.

[12] L. Moriarty, R.K. Breuninger and K. Newman, Designing Simple NuBus Slave-Only Applications using Texas Instruments SN74ACT2440 and SN74BCT2420, *Application Report*, Texas Instruments Inc, Dallas, 1988.