EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN-EP/88-67
June 22, 1988

MACINTOSH IN THE LABORATORY

B.G. Taylor

CERN, Geneva, Switzerland

presented at the
Apple European University Consortium Meeting
University of Heidelberg
6 - 9 April 1988

MACINTOSH IN THE LABORATORY

B.G. Taylor

EP Division, CERN, 1211 Geneva 23, Switzerland

Abstract

The marriage of a mass-produced personal computer with versatile international-standard industrial instrumentation systems creates a cost-effective solution to many laboratory small-system requirements. This paper describes how the 68000/68020-based Apple Macintosh family of personal computers can be provided with direct access to VMEbus and CAMAC systems for data acquisition, experiment control and monitoring, and equipment test and development.

1. Introduction

Since Henry Ford, the gulf between the "bargains" available in mass-produced products and the cost of things that cannot be produced in volume has grown steadily. Since the introduction of digital integrated circuits, cost-effective systems design has been based on a philosophy of "adapting bargain components to specific needs" [1]. Now that popular third-generation personal computers are manufactured by mass-production methods in highly automated plant, they have become one of the "bargain components" of today's electronic system designer. By adapting these inexpensive, but powerful, machines to the computing tasks they meet in laboratory instrumentation environments, engineers can realise substantial savings in a variety of control, monitoring, data-acquisition and equipment-development applications.

At the Apple Computer facility in Fremont, California, one of the world's most automated factories turns out several 68000-based Macintosh computers per minute, whereas the production of the 68020-based Macintosh II is currently reported to be some tens of thousands per month. Production on this scale permits a manufacturing and testing efficiency which cannot be approached in the production of microcomputer systems specifically for the much more limited professional instrumentation market.

Mass-produced hardware is only one aspect of the personal computer "bargain component". It is widely recognised that the development of efficient, reliable and well-documented software requires an investment which may substantially exceed that of designing the hardware. When this investment is spread over the vast user-base of a popular personal computer like the Macintosh, the software can be made available at a very attractive cost per system, and it receives the large exposure necessary for thorough debugging.

On the other hand, the introduction of international standards for professional instrumentation, such as CAMAC [2] and VMEbus [3], has brought numerous benefits to online data-handling and computing at scientific research laboratories such as CERN. These nonproprietary standards have made it possible to interconnect electronic equipment of varied origin, and have greatly facilitated the work of integrating large multiprocessor data-acquisition systems whose elements are developed by numerous research collaborators in dispersed institutions.

The high degree of parallelism possible in a distributed configuration of VME multiprocessors minimizes dead-time in the read-out of a large detector facility and allows sophisticated triggering and filtering systems to be implemented. Experience has also shown that systems based on standard instrumentation can be readily re-configured as needs change, and enhanced as technology evolves. Apple Macintosh computers can be used as cost-effective software development workstations for such systems and their graphics-oriented user-interface has proved well-suited to control and monitoring tasks during data-taking.

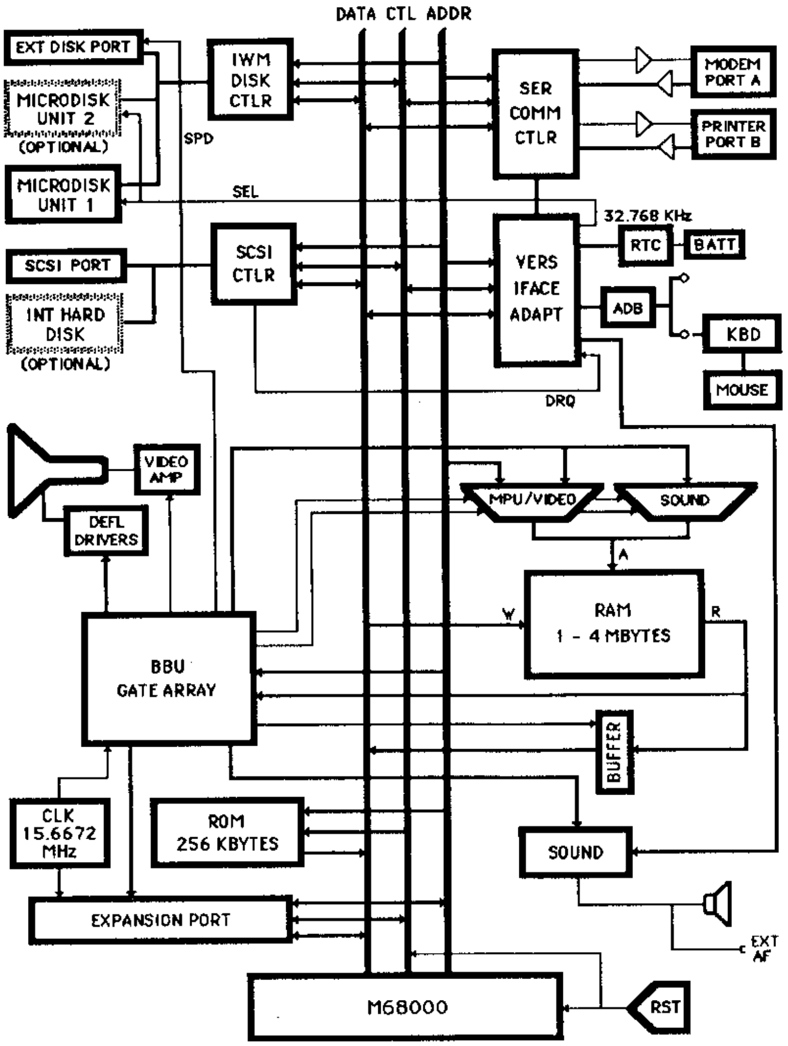


Figure 1 Macintosh SE Block Diagram

This paper describes how the entire Apple Macintosh family of personal computers has been provided with compatible direct access to VMEbus and CAMAC systems, thus marrying the features of these popular mass-produced computers with the versatility of international standard instrumentation systems. The system is called MacVEE (Microcomputer Applied to the Control of VME Electronic Equipment). Several hundred MacVEE systems are currently in use at research laboratories world-wide.

2. MacPlinth

The Macintosh SE block diagram shown in Figure 1 is illustrative of the architecture of the earlier 68000-based Macintosh computers, which are highly integrated machines having no backplane bus structure.

To implement a high-performance interface to these computers, MacVEE makes direct connection to the internal microcomputer bus. This allows selected VME and CAMAC crates to appear within the 68000 microprocessor address space, so that no special software drivers are required to access them.

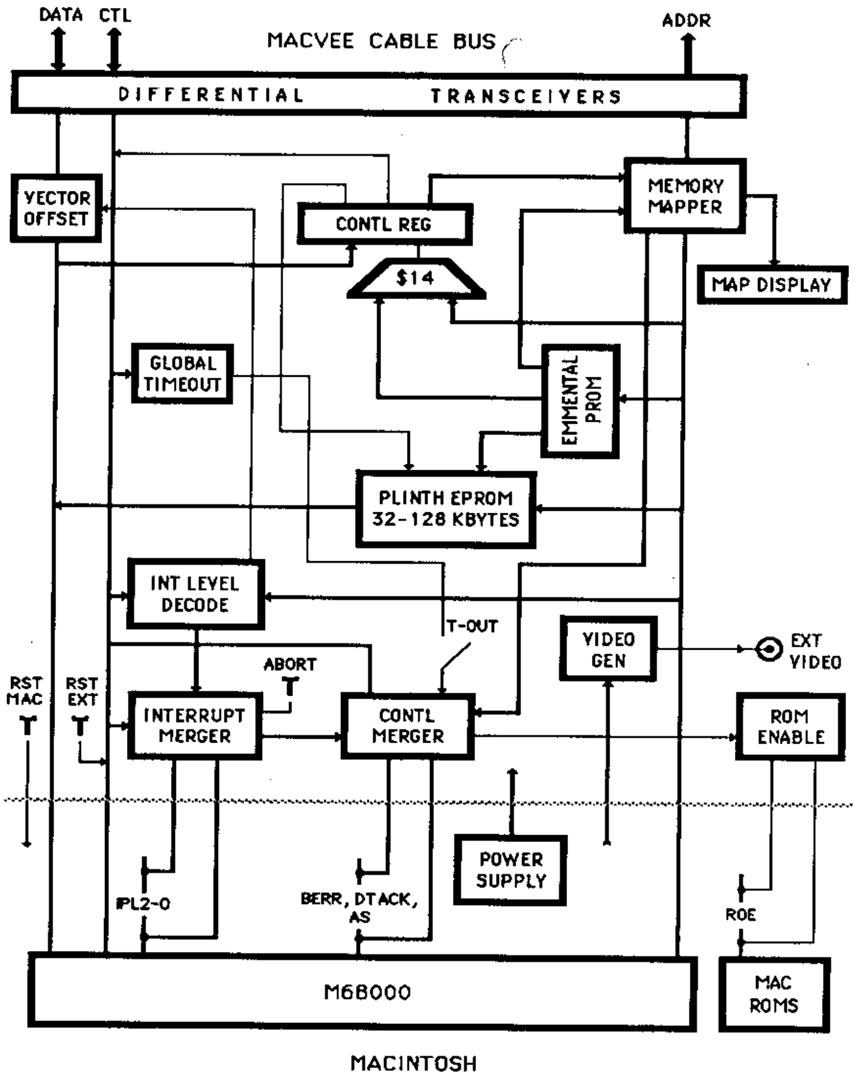


Figure 2 MacPlinth Block Diagram

The connection is made by an electronics plinth called MacPlinth (Figure 2), which attaches to the computer and becomes an integral part of it. MacPlinth incorporates a memory mapper which allows it to access an external address space exceeding 100 Mbytes, in up to 8 VME crates or up to 7 VME crates and up to 8 CAMAC crates. The mapping is controlled by a Schottky 'emmental' PROM, which stores a 4-bit descriptor for each of the 256 64Kbyte segments of the 16 Mbyte address space of the microprocessor, and a 3-bit map selector.

The emmental PROM allocates the free addresses, and those occupied by the incomplete address decoding of Macintosh internal hardware facilities, to VMEbus 'cheese', while assigning 'holes' to the addresses actually referenced by the system software. Any group of segments can be mapped as common area, and code in such an area has access to resources in any of the VMEbus crates. The emmental PROM pattern can be readily changed to adapt to computers with different memory capacities, or to address changes in different versions of the System file or Macintosh hardware. A typical MacVEE SE address map is shown in Figure 3.

MacPlinth has a control register at address $14, the unused most-significant byte of the zero-divide exception vector. As secondary features, it generates a composite video signal for external remote monitors and accommodates up to 128 Kbytes of EPROM for permanent library enhancements. It provides a single-level abort, internal/external reset switches, a watchdog timer and a LED array which indicates the VME crates being accessed.

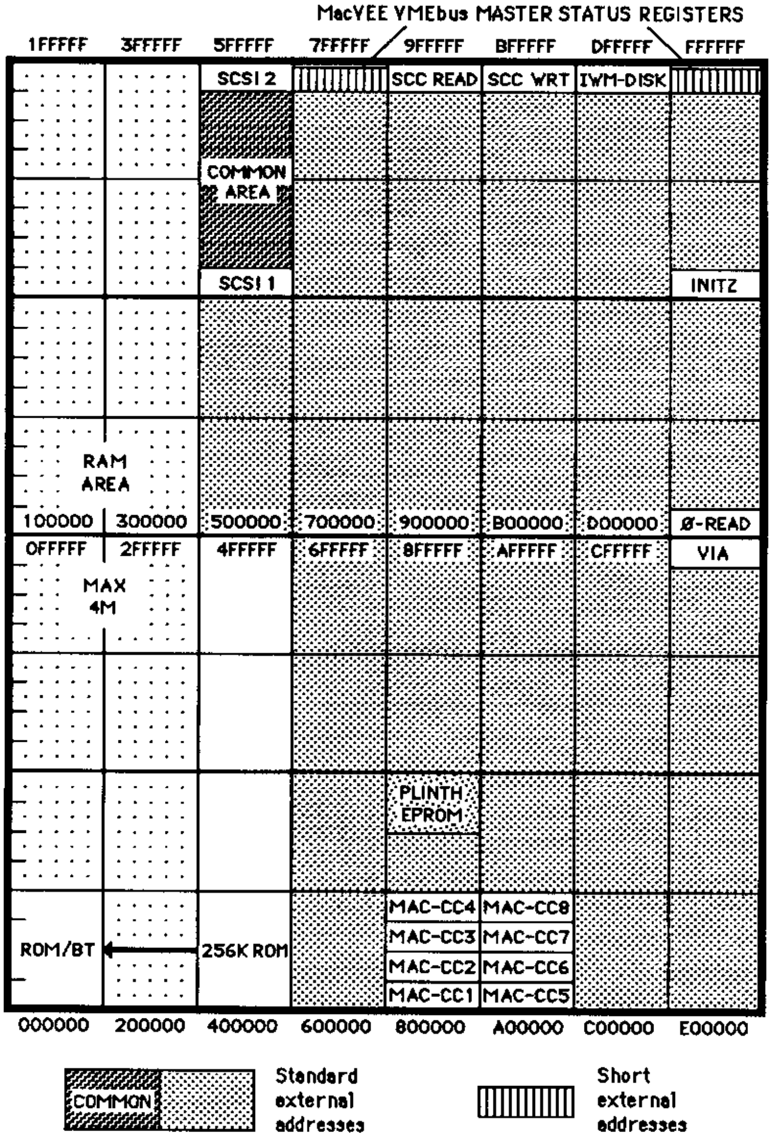


Figure 3 MacVEE SE Address Map

Macintosh internal interrupt codes are decoded on MacPlinth, merged with external interrupt sources, and re-encoded before application to the 68000 microprocessor. MacPlinth assigns three interrupt levels to Macintosh internal auto-vectored (AV) interrupts, three levels to external user-vectored (UV) interrupts, and one non-maskable level to external AV interrupts for fail conditions and CAMAC demands.

During a Macintosh IACK bus cycle, MacPlinth decodes the interrupt level being acknowledged from the LS bits of the address bus. If the level is one which has been assigned to external system UV interrupts, the assertion of VPA by the Macintosh interrupt circuitry is inhibited, and a vector number is fetched from the VMEbus interrupter. In a multi-crate system, MacPlinth automatically accesses only the crate containing the source of the interrupt, independent of the map no. currently in its control register.

68000 microprocessor UV interrupts would normally be vectored through RAM locations \$100 - \$3FF, which are used by Macintosh system software for global variables. On MacPlinth, the 4 MS bits of the vector number are masked, and an offset is added to translate the 16 possible vector numbers to the range \$30 - \$3F. These vector numbers cause the 68000 to fetch its vectors from RAM addresses \$C0 - \$FF, which are reserved by Motorola and not used by Macintosh software.

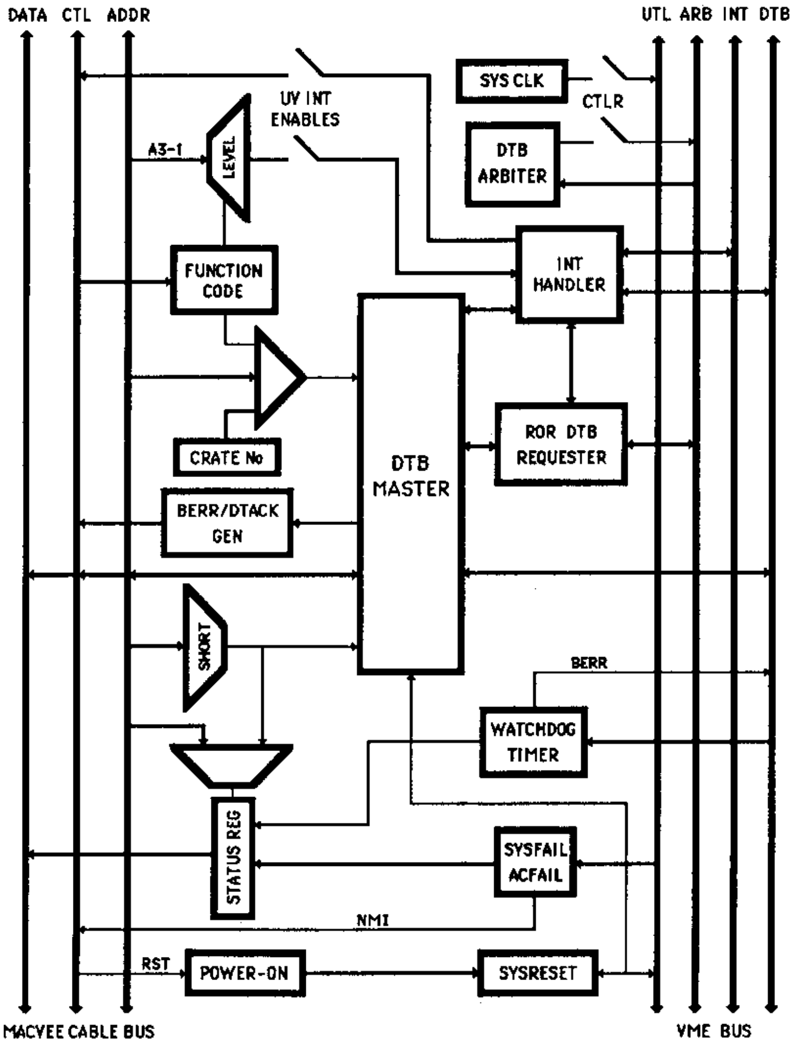


Figure 4 VMEbus Master Block Diagram

3. MacVEE VMEbus Master

VME crate access in all MacVEE systems is provided by a VMEbus master module (Figure 4) incorporating a release-on-request (ROR) data transfer bus (DTB) requester and a 3-level VMEbus interrupt handler which operates in conjunction with the processor in the Macintosh. The module includes slot-1 functions (system clock, SGL bus arbiter and global crate data-transfer time-out). It can be employed as a system controller, or as a normal DTB master in a multi-processor system.

A Macintosh bus error is generated if a VMEbus operation attempted by any master module fails to complete within 8 μs. The timeout starts to run when the module has been granted the bus and has asserted at least one data strobe. A status register allows various other fail conditions to be identified, such as the VMEbus SYSFAIL signal which is activated by modules having a self-test feature.

The ACFAIL signal, which is associated with a non-maskable auto-vectored interrupt to the Macintosh, gives advance warning of power failure in any VME crate. From the time of the interrupt, Macintosh has a minimum of 4 ms in which to poll the status registers to determine which VME crates are affected by the power fail, and to take appropriate action. After this time, DC power in the affected crate may have fallen too low for the status register to be read reliably.

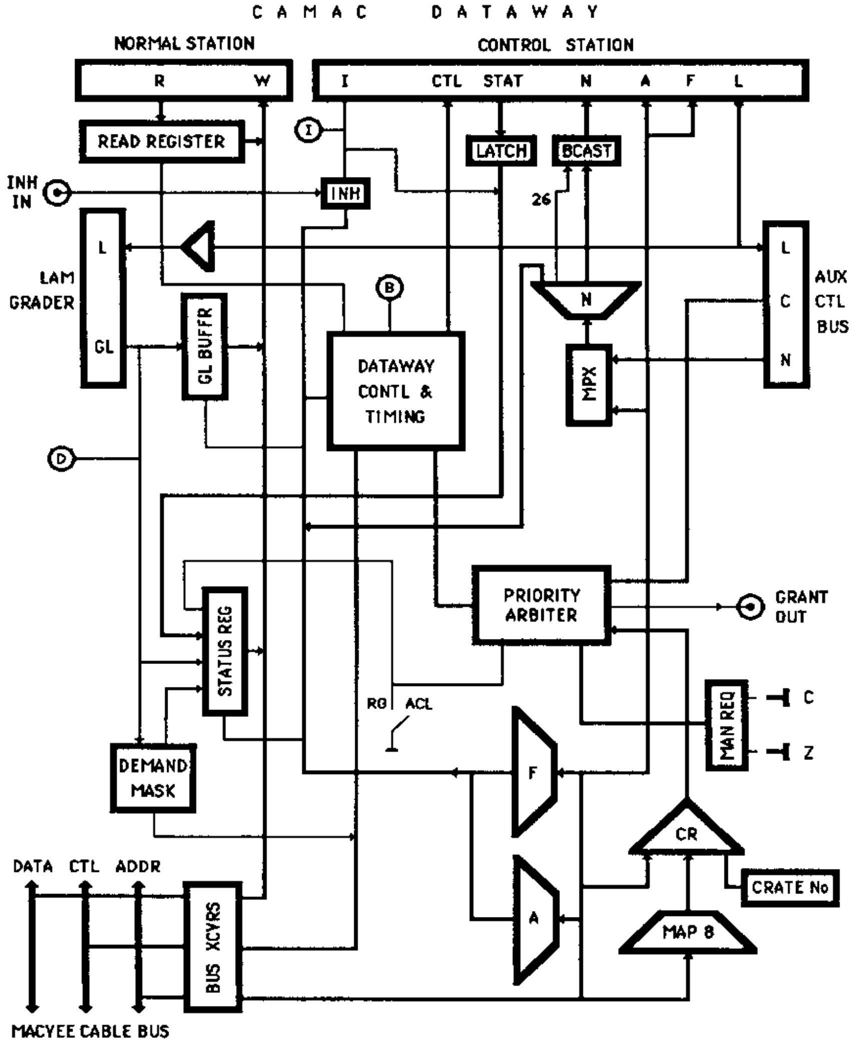


Figure 5 Mac-CC CAMAC Crate Controller

4. Mac-CC

Mac-CC (see Figure 5) is a memory-mapped dedicated CAMAC crate controller for the Macintosh family. It is designed in accordance with CERN recommendations [4] for M68000-based CAMAC port controllers, and additional functions have been chosen to be as compatible as possible with VME CAMAC branch drivers and Type A2 crate controllers, as appropriate.

Each CAMAC crate occupies only 64 Kbytes of the Macintosh address space, and all Mac-CCs are accommodated within map 8. The addresses which they occupy remain free for VMEbus use in maps 1 - 7. Mac-CCs allow CAMAC crates to be accessed without an intermediate VME crate and branch driver in pure CAMAC environments. In this case the CAMAC library subroutines can be accommodated in MacPlinth EPROM. The MacVEE bus uses RS485 differential transmission, and permits greater ranges than an EUR 4600 branch.

Mac-CC is equipped with an EUR 6500 auxiliary controller bus, supporting optional multiple controllers in a CAMAC crate with either R/G (request/grant) arbitration or ACL (auxiliary controller lockout). It is compatible with optional standard LAM graders. ESONE standard CAMAC subroutine libraries have been written for Fortran-77 and various Basics.

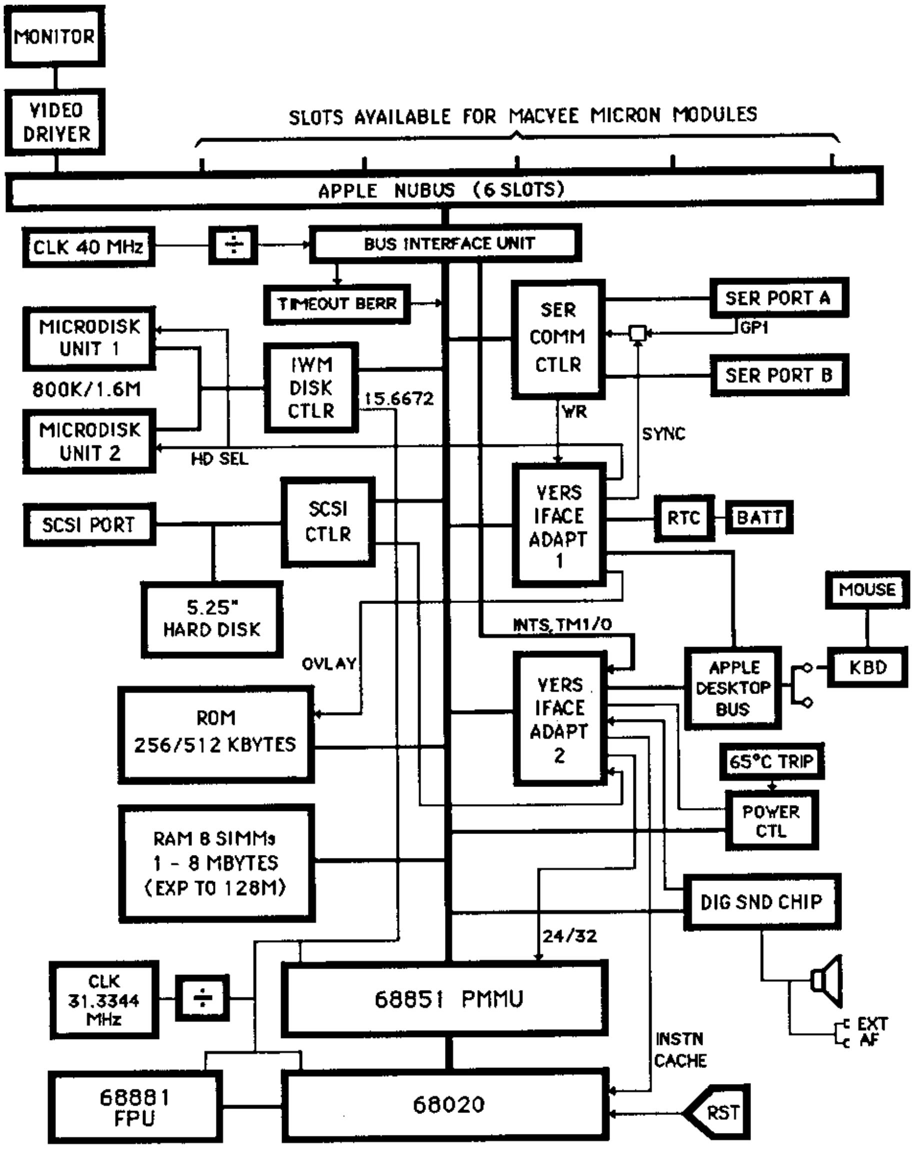


Figure 6 Macintosh II Block Diagram

5. Macintosh II

With its floating-point coprocessor, internal hard disk, high-resolution colour graphics, enhanced toolbox ROM, optional paged MMU, networking and UNIX capability the 68020-based Macintosh II is an attractive personal computer for instrumentation applications in scientific research.

The Macintosh II has an open architecture based on the Apple NuBus, an adaption of the NuBus whose specification is currently IEEE proposed standard P1196. This specification is in turn a development of the Texas Instruments NuBus specification published in 1983. The original NuBus was conceived at the Massachusetts Institute of Technology, and was developed by MIT and Western Digital Corporation between 1979 and 1983.

Prior to its adoption by Apple, NuBus was little used outside AI workstations by Texas Instruments and Lisp Machine. NuBus is a time-quantized asynchronous computer backplane bus structure, with geographical addressing and multiplexed 32-bit address and data lines, which supports multiple masters with distributed arbitration. Its key design goals are claimed to be sparsity of mechanism and system architecture independence. The basic bus protocol for an elementary NuBus transaction is indicated in Figure 7.

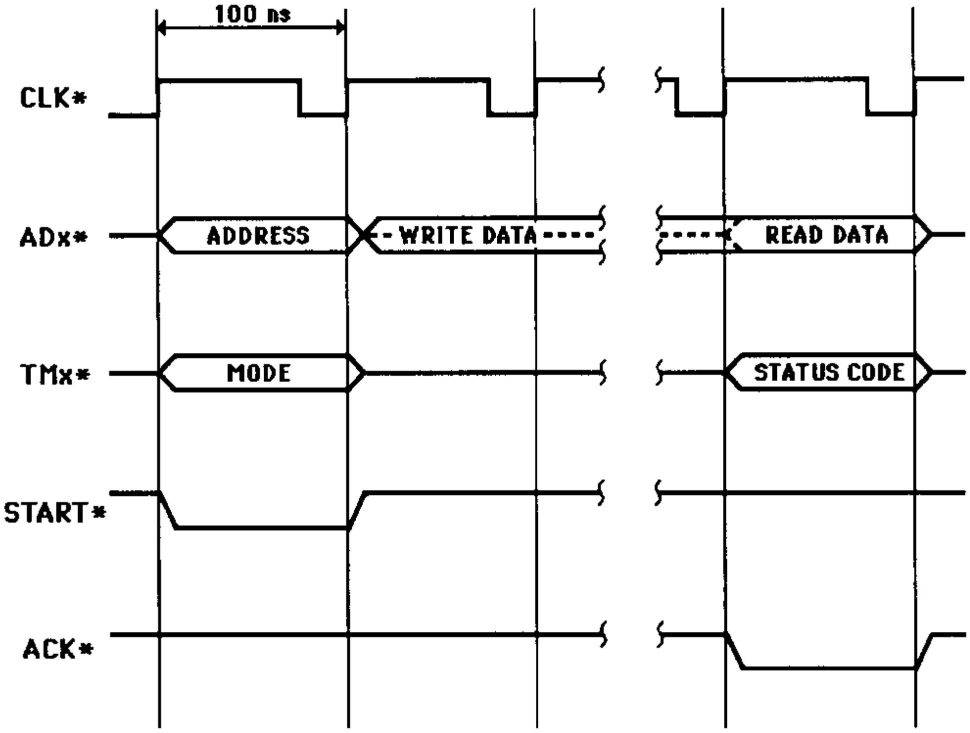


Figure 7 Basic NuBus Transaction

6. MICRON

The Macintosh II is provided with six Apple NuBus slots. It can be interfaced to MacVEE systems by MICRON (MacVEE Interface Card Resident On NuBus), a block diagram of which is shown in Figure 8. MICRON is compatible with the MacVEE VMEbus master modules and Mac-CC memory-mapped CAMAC crate controllers used with the earlier 68000-based Macintoshes.

MICRON allows VMEbus or CAMAC crates simply to appear within the address space of the 68020 microprocessor of the Macintosh II, so that no special software drivers are necessary to access them. In 32-bit addressing mode, the Macintosh II can even execute programs in VMEbus RAM or EPROM. There is no address translation, so that the system is not limited to the execution of position-independent code.

MICRON supports all data transfer operation types on the Apple NuBus, and automatically performs the required VMEbus operations (eg. two demultiplexed 16-bit accesses for one longword NuBus operation) so that the interface is transparent to the user. The Macintosh II/NuBus interface performs byte-swapping to match the big-endian byte ordering of the 68XXX devices to the little-endian NuBus byte ordering. This byte-swapping does not affect the order of bits within bytes. All of the address non-alignment functionality of the Macintosh II 68020 is supported through to VMEbus slaves.

Control and status registers are provided on MICRON to allow VMEbus address modifiers to be specified, interrupt sources to be determined, and for mode selection and software reset. The registers are allocated unique addresses outside those used in the computer or external system. Restricted VMEbus resource-locking is implemented through Apple NuBus attention cycles, which are generated automatically when 68020 RMW instructions are executed.

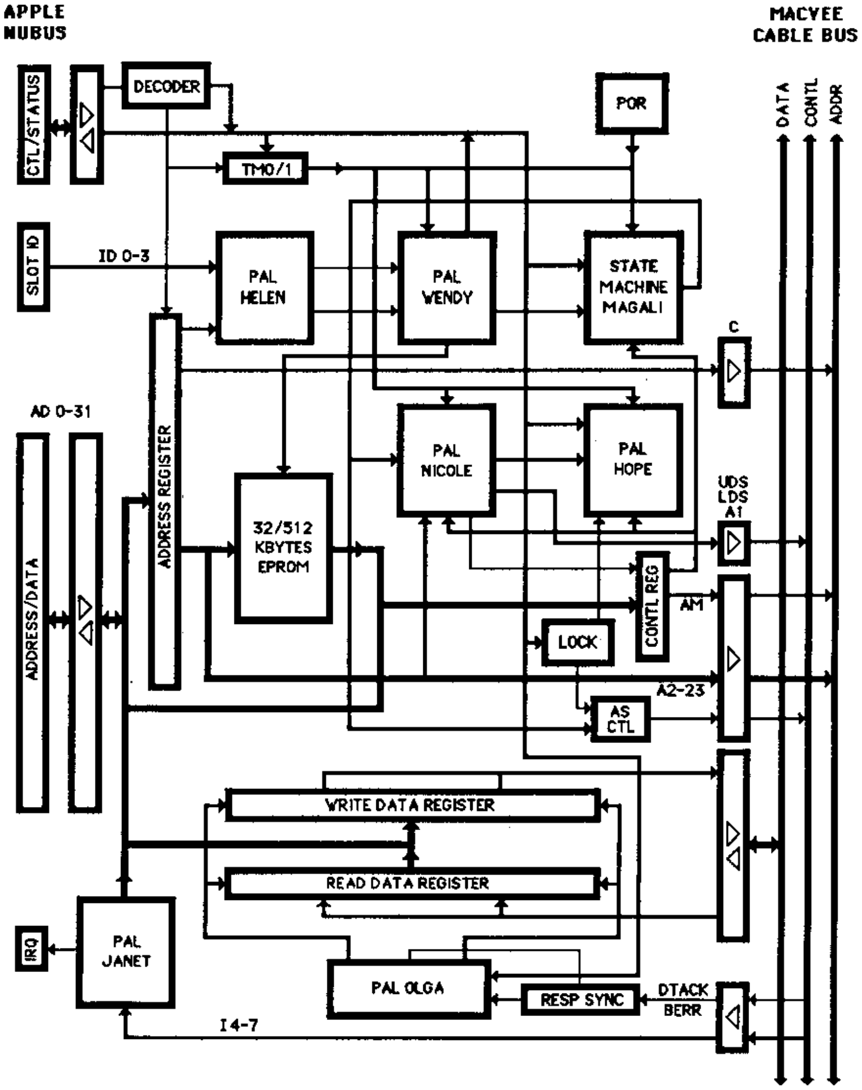


Figure 8 MICRON Block Diagram

Data caching, which may be introduced in future 68030-based Macintosh architectures, is supported as a programmable option. In data cache mode, MICRON always acquires a 32-bit word from VME for any type of NuBus read. The MICRON state diagram shown in Figure 9 indicates the operations performed by the module for the different Macintosh II/VMEbus data transfer types, on-card EPROM and status read, control register write and attention cycles.

7. MICRON Addressing

The large addressing range of the 68020-based computer allows a full 128 Mbytes of contiguous address space to be dedicated to each external system and directly accessed by the microprocessor in 32-bit mode. System calls are available for switching addressing modes, and the address range can be allocated in up to 8 VME crates, or up to 7 VME crates and 8 CAMAC crates, in any mix.

MICRON itself can accommodate up to 512 Kbytes of EPROM in an additional separate address space. The EPROM quad stores 32-bit words and can contain executable code. A small part of this EPROM contains configuration information ('declaration data') which is read by the Macintosh II slot declaration ROM manager at start-up to determine the characteristics of the hardware present in its slots. The remainder is free for any semi-permanent library extensions, test routines or application programs which the user may wish to store there.

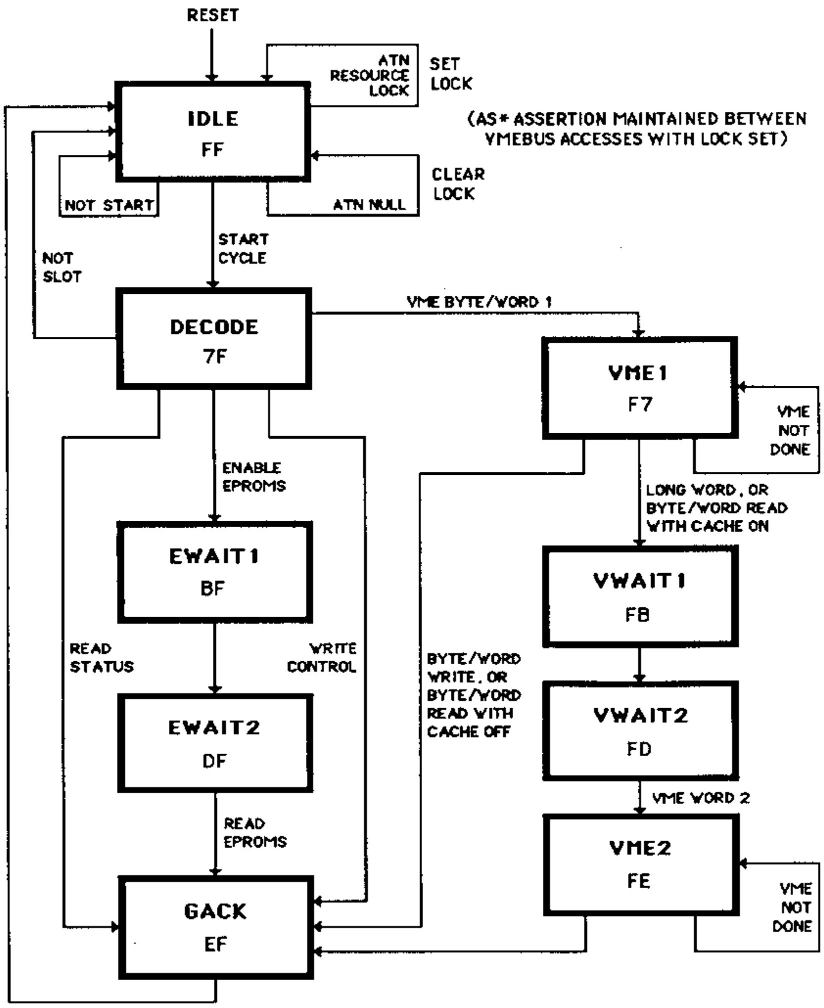


Figure 9 MICRON State Diagram

The six NuBus slots available in Macintosh II are identified by hex ID codes (S) from 9 to E. One slot is normally occupied by a Macintosh II video driver card. Each physical slot is associated with a Superslot address space of 256 Mbytes and a Slot address space of 16 Mbytes (see Figure 10). This association of address allocations with physical slots means that MICRON does not require any jumpers or address configuration switches.

The allocation of VMEbus and CAMAC crates to Superslot space is shown in Figure 11. MICRON on-card EPROM, control and status registers appear in the associated Slot space. VMEbus address modifiers AM0 - AM3 and AM5 are selected by the MICRON control register, while AM4 is controlled automatically by the MacVEE VMEbus master module according to the address area being accessed. AM4 = 0 in the 64 Kbyte address areas \$(S)X7F0000 - \$(S)X7FFFF and \$(S)XFF0000 - \$(S)XFFFFFF (S = 9-E, X = 0-7). With AM3 = AM5 = 1 selected, this results in VMEbus short addressing in these areas and standard addressing elsewhere.

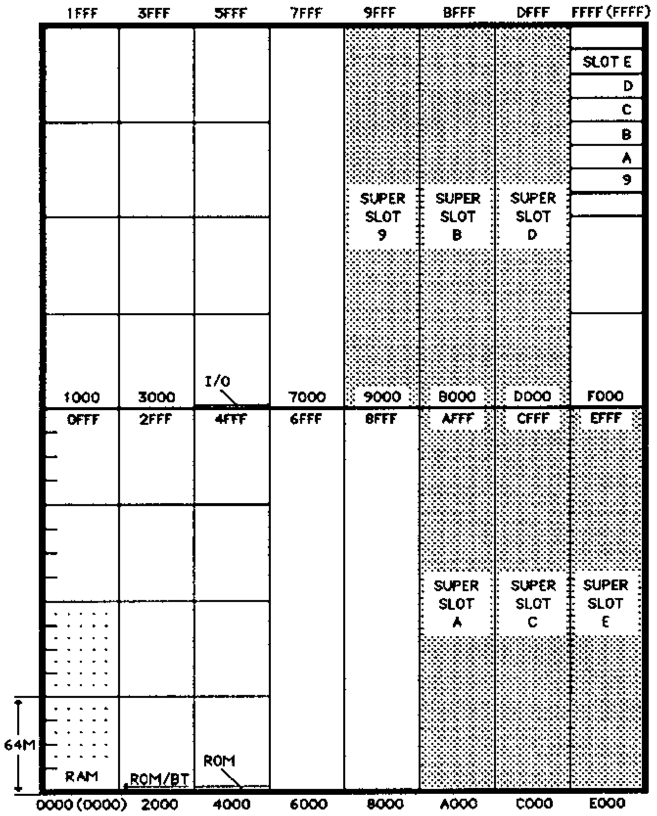


Figure 10 Macintosh II Address Map

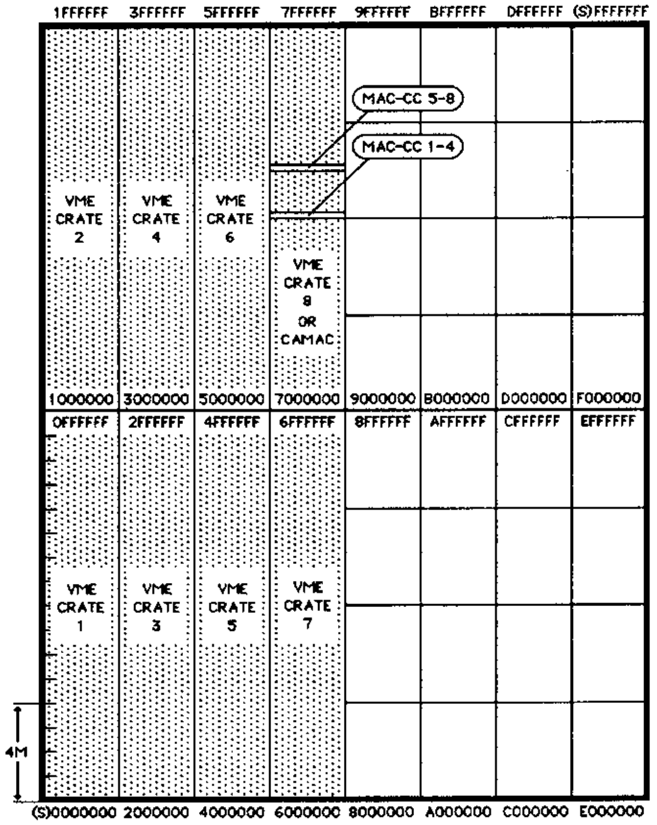


Figure 11 MICRON Address Map

As a result of the linear addressing scheme, even for multiple MICRON modules, no map control register is required as in 68000-based Macintosh systems. A common address area is also unnecessary, since Macintosh II software residing in any VME crate can directly access resources in any other VME crate, as well as those internal to the computer itself. Up to 3 MICRON modules may be installed in a Macintosh II to provide direct access to 24 VMEbus crates per computer.

8. Interrupts

Macintosh II interrupt sources are assigned as follows (see Figure 6): level 1 - VIA1, level 2 - VIA2, level 4 - SCC, level 6 - PwrIRQ, level 7 - NMI. Devices in any of the six Macintosh II NuBus slots can generate non-master interrupt requests which result in a level 2 autovectored interrupt to the 68020 through CA1 of VIA2. The Macintosh II ROM device manager determines the NuBus slot which has generated an interrupt by reading data inputs PA0 - PA5 of VIA2, and maintains an interrupt queue for each slot interrupt. The slot queue element contains a priority parameter which determines the order in which the routines are called and the slots polled.

While the Apple NuBus does not support multi-level user-vectored interrupts, MICRON provides a facility to allow VMEbus interrupters on different levels to be distinguished, and their vector numbers to be acquired by a pseudo-acknowledge cycle.

9. Applications Illustrations

MacVEE systems are used for many different applications at CERN, at other research laboratories, and by industry. Space allows only a few examples to be cited for illustration.

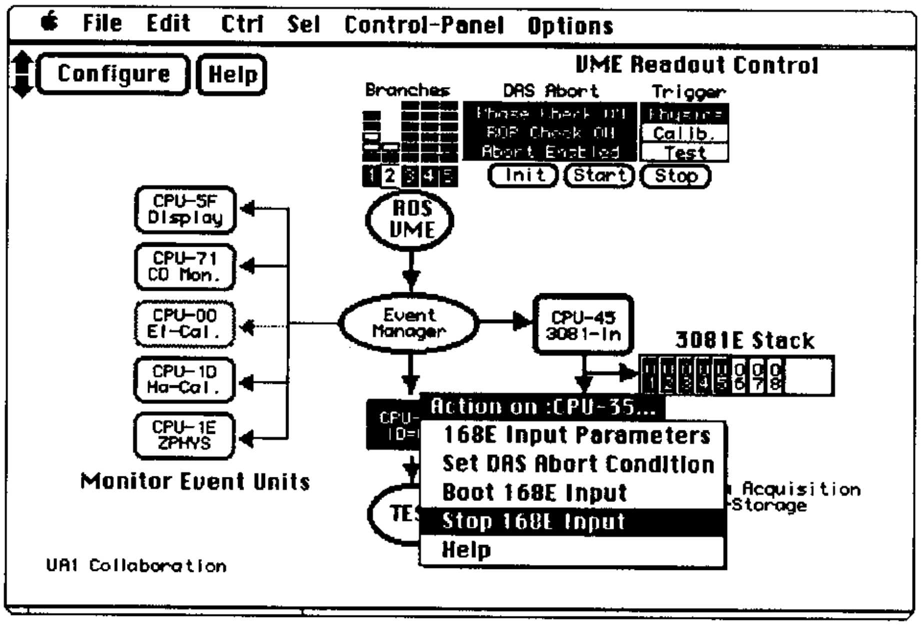


Figure 12 MacVEE Interactive System Control

The direct integration of the Macintosh family with VMEbus and CAMAC systems has allowed the development of several novel approaches to the provision of facilities for user interaction with large experiments, and work in this area is continuing. Figure 12 shows an example from part of an experiment system control application. All the structures represented may be opened with the mouse to access associated menus and dialog boxes, and the selected control inputs are executed in the VMEbus system elements through MICRON.

This application is one of many created on the Macintosh itself using a resident development system [5] implemented by CERN's UA1 software team. The system integrates the real-time Fortran-77 compiler of Heidelberg University [6] with an assembler, editor, linker and data communication utilities. It supports the programming and control of multiprocessors in VME.

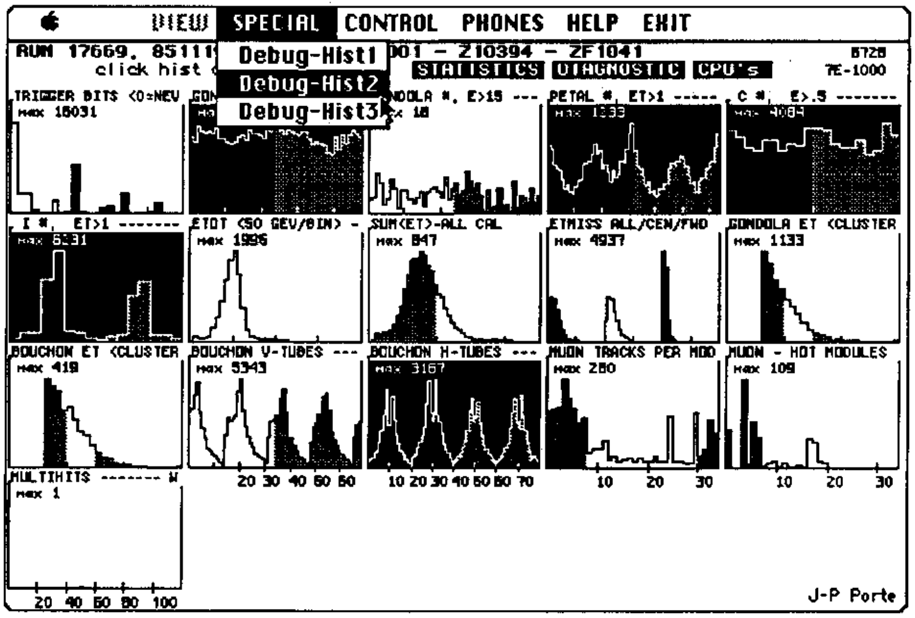


Figure 13 MacVEE Experiment Monitoring

Figure 13 illustrates data acquisition monitoring at this experiment by an application written in Basic. The program has interactive facilities for diagnosing any abnormalities which it detects in the statistics, and even advises the operators by Macintosh speech messages.

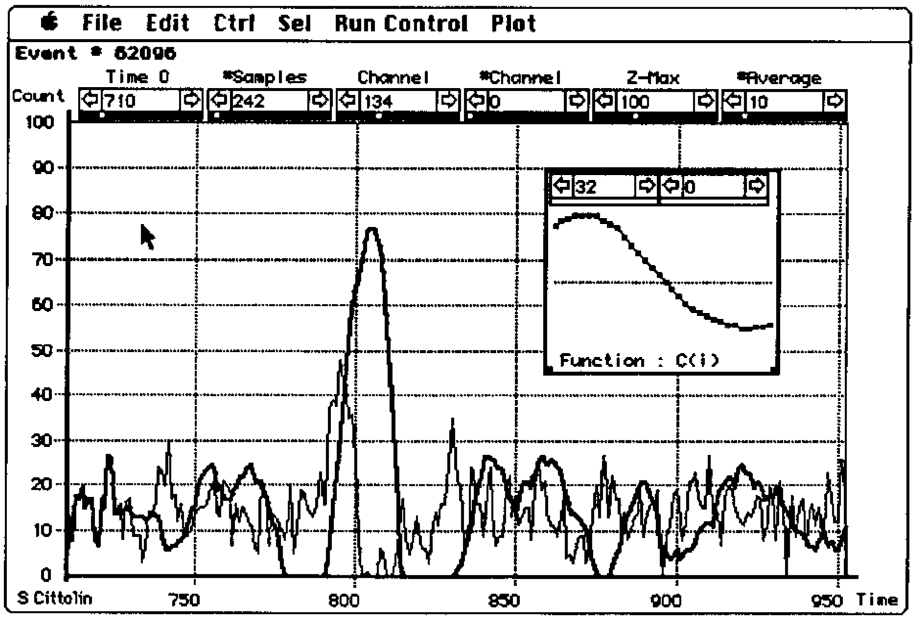


Figure 14 MacVEE Interactive Signal Analysis

At UA1, multiple MacVEEs are used for the control and monitoring of the subsystems of the apparatus [7], as software development workstations for a distributed hierarchy of 60 VMEbus multiprocessors in 20 VME crates, as data-acquisition consoles with an automatic logging function, and as controllers for a farm of emulators of IBM 3081 mainframes.

In the development laboratory, MacVEE systems are used for the testing of new VMEbus or CAMAC-based instrumentation systems. Since the software can be run entirely in the Macintosh, its integrity is safeguarded from problems with modules being debugged on the VMEbus. User-friendly access to Fastbus systems has been provided [8] via a fast sequencer.

The flexibility of the Macintosh user interface encourages the interactive approach which proves fruitful in experimental work. Figure 14 shows an example of detector signal analysis in which the inset function C(i), with which the input signal is cross-correlated, may be modified interactively in MacDraw style. In the multichannel signal display illustrated in Figure 15, the view geometry and other parameters can be varied directly using the Macintosh mouse. It proves considerably easier for physicists to master this type of interface than to remember the cryptic keyboard sequences of a conventional command language.

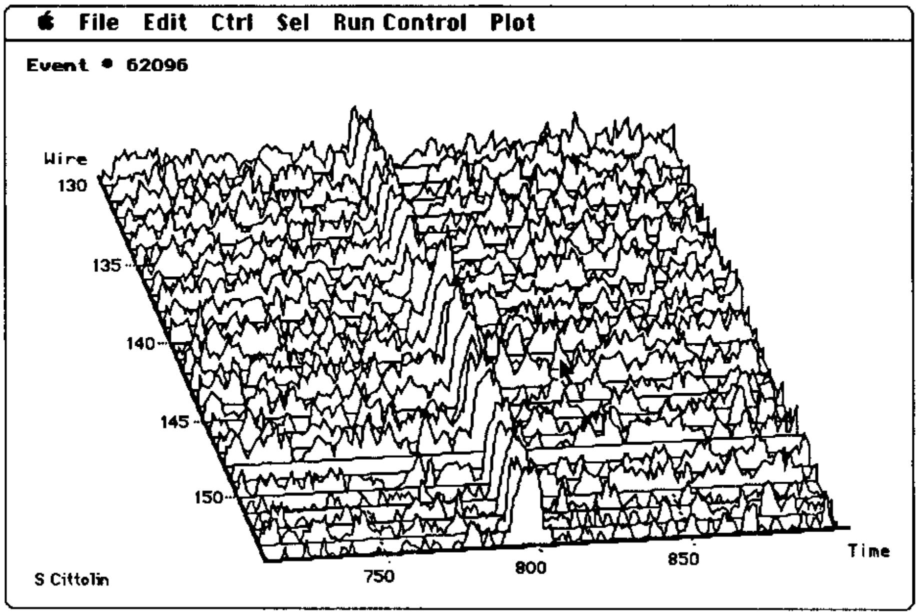


Figure 15 Multichannel MacVEE Signal Display

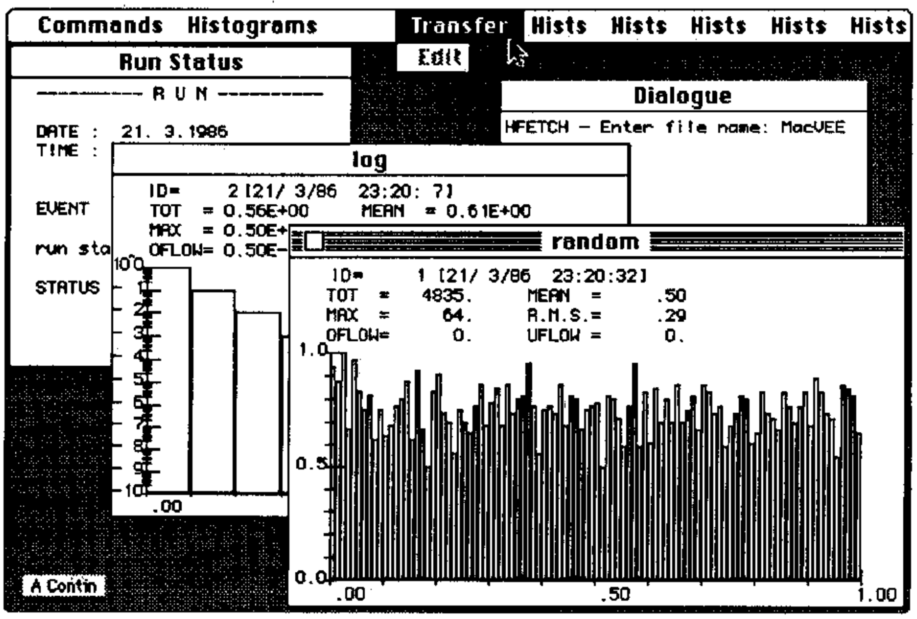
Other MacVEE users have based their work on the Absoft/Microsoft Fortran-77 compiler for the Macintosh. Fortran libraries [9] have been written which include subroutines for such functions as the creation and management of pull-down menus and windows, the handling of mouse inputs, and the dynamic linking of external libraries. A comprehensive histogram package has been produced (see Figure 16) and CAMAC libraries have been written. A general-purpose data acquisition package [10] has been created and employed at several small experiments in which the only computers used are MacVEE systems.

MacVEE provides direct multi-crate VMEbus and CAMAC access to any resident Macintosh language capable of reading and writing at known absolute addresses. Some 50 software houses have now generated resident systems for Ada, APL, Assembler, Basic, BCPL, C, Cobol, Forth, Fortran-77, Lisp, Logo, Modula-2, Neon, Pascal, Prolog, Scheme, Simula-67, Smalltalk-80 and VIP. There are at least 10 compilers for C, and 8 for Pascal, available for the Macintosh today.

One of the most comprehensive development environments is the Macintosh Programmer's Workshop (MPW) from Apple Computer. The MPW incorporates a shell/editor with Unix-like features, a macro-assembler, linker, debugger, resource maker/disassembler/editor, utility package and compilers. It currently supports Ada, Assembler, C, Fortran, Modula II and Pascal, and programs written in MPW C can be ported to A/UX with few modifications. MacVEE tools for VMEbus and CAMAC operations have been integrated directly in the bourne-style MPW shell [11].

MPW has been selected as the programming environment for the multiple MacVEE systems which are being installed for the control and monitoring of the new H1 facility [12] at the HERA electron-proton collider at DESY, Hamburg. The 100 VME crates of multiprocessors required at this experiment for the read-out of over 220,000 channels of data from the 2,500-ton detector will be controlled by a dozen Macintosh II computers with MICRON.

In less than 4 years the Macintosh has evolved from a machine with mediocre language support to the preferred vehicle for some of the latest developments in software engineering. There are currently 7 object-oriented languages available for the Macintosh in full or pre-release state. One of the most interesting initial applications of object-oriented programming is with MacApp, the expandable application framework for the computer. MacApp implements the Macintosh user interface in skeleton form, and provides hooks for custom objects. To create a Macintosh application using MacApp, one designs the objects required for the specific tasks and installs them into the framework. Programming laboratory applications by customizing MacApp can reduce the amount of code to be written by a factor of 4 to 5, with a corresponding decrease in the required development time.



```
---------- R U N ----------

DATE :  21. 3.1986
TIME :

EVENT

run sta

STATUS
```

```
HFETCH - Enter file name: MacVEE
```

```
ID=      2 [21/ 3/86  23:20: 7]
TOT   = 0.56E+00      MEAN   = 0.61E+00
MAX   = 0.50E+
OFLOW= 0.50E-
```

```
ID=      1 [21/ 3/86  23:20:32]
TOT   =   4835.      MEAN   =     .50
MAX   =     64.      R.M.S.=     .29
OFLOW=       0.      UFLOW  =      0.
```

Figure 16 MacVEE Histogramming

10. Utilities

A wide range of Macintosh utilities has been created by software developers in educational and research environments, placed in the public domain, and distributed internationally by electronic mail networks. In addition, a number of CERN-written utilities are of particular value in a MacVEE environment.

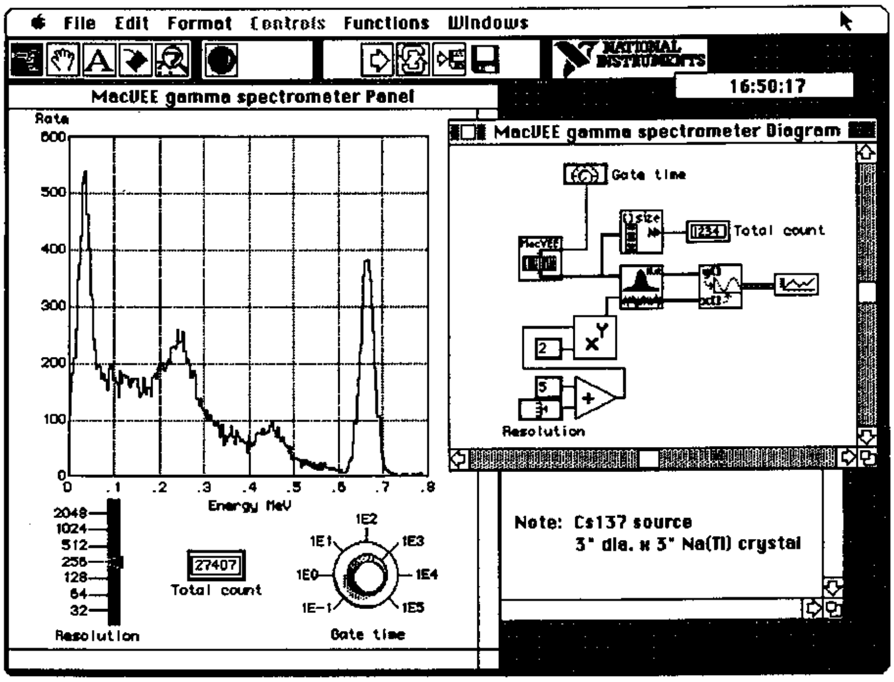
For example, AMERICA (A MacVEE External Ramdisk Install and Configure Application) [13] allows a user to install up to 12 ramdisks simultaneously in any configuration of VME crates, and to work at very much higher speed than from a hard disk. The application has an excellent implementation of the Macintosh user interface for the automatic scanning of VME crates, and the mounting and unmounting of VMEdisks, as they are called.

A set of VME Utilities of general application has been written as a desk accessory, which allows them to be accessed concurrently with a MacVEE application. They provide facilities for common operations such as the initialization and testing of dynamic RAM in VME, the rolling in and out of code, and tools for the editing of data in VME.

11. Graphical Programming

While Fortran-77 application programs and physics libraries remain the mainstay of scientific programming at CERN, some MacVEE users are exploring less conventional approaches to software development. It may be possible to provide HyperCard with direct access to the VMEbus [14] through XCMD and XFCN resources written in a compiled language. XCMDs have the same calling interface as HyperTalk handlers, the same abilities to send messages and evaluate expressions, and exist in the same inheritance hierarchy.

For over 10 years, the dataflow model has been researched in the context of non von Neumann machines having various degrees of parallelism. The model can also be applied to the problem of real-time task definition and intercommunication for the conventional sequential architectures of today's personal computers. LabVIEW [15] is an outstanding example of a software construction environment for the Macintosh which combines dataflow concepts and traditional program-control structures with a graphical-element instruction set.



Note: Cs137 source
3" dia. x 3" Na(Tl) crystal

Figure 17 MacVEE Programming with LabVIEW

An example of a simple MacVEE virtual instrument (VI) using LabVIEW is shown in Figure 17. The functionality of such a VI is determined by a block diagram which is configured by the user and effectively constitutes self-documenting application source-code. The VI comprises a hierarchy of sub-VIs which are created by selecting and arranging graphical objects, wiring them together, and assigning them icons.

Using the graphical interface, the experimenter can readily traverse the hierarchy to observe or modify the VI during its execution. The use of the LabVIEW system in MacVEE signal-processing applications is further enhanced by the provision of an extensive library of functions for waveform manipulations, fast convolution, FFT and power-spectrum analysis.

Such CASE (Computer Aided Software Engineering) systems are potentially of great utility for a laboratory tool such as MacVEE. This is because an interactive approach to experiment development is often most productive, whereas in the classic software development process initial analysis and design is followed by coding and testing phases which typically require at least as much time. Preliminary work [16] using MacVEE systems with Macintosh II at CERN and in the USA suggests that the intermarriage of CASE techniques with international instrumentation standards and conventional scientific libraries holds considerable promise.

12. Past and Future

To conclude this 1988 seminar about laboratory applications of a modern personal computer, it is interesting to glance back a quarter of a century, and also to try to glimpse what the future may bring over the same time span.

In 1963, CERN was operating the Ferranti Mercury computer. This was a magnificent piece of hardware whose numerous equipment cabinets occupied a very large room, but the computing power offered was less than that of today's Macintosh, which is about the same size as its paper tape reader. The CERN computer centre had also installed an IBM 709, the first of a long series of big number-crunchers to follow. Once again, however, the CPU of that machine is outclassed by the present 68020 microprocessor chip.

These examples remind us that one of the most striking advances made in the last 25 years has been in the area of miniaturisation, and there are indications that this may continue during the next 25. One of these indications comes from Japan, where the government has launched a massive R & D program called 'Human Frontiers'.

An important aspect of the Human Frontiers project is the development of electronic circuits based on protein molecules. Molecular electronics researchers are seeking to create molecular components which will participate in their own assembly, rather like the DNA molecule. Initial applications of such self-assembling biochips might be in mass storage devices, because of the very high data density potential - some 4 orders of magnitude greater than with present optical recording techniques.

But the fabrication of a full molecular computer could be envisaged, and such a biochip could be very much smaller and very much more powerful than any silicon chip. Indeed, biochips would be so tiny that one could imagine their being embedded in the human brain, perhaps as a coprocessor. It is an intriguing thought that, 25 years from now, that just might be the ultimate Apple 'personal' computer.

13. Conclusion

The MacVEE system described in this paper provides the Apple Macintosh family with an intimate connection to multi-crate VMEbus and CAMAC systems. This permits direct access to international standard laboratory instrumentation from these popular mass-produced 68000/68020-based personal computers, with their friendly graphics-oriented user interface and wide range of resident programming languages and libraries.

The introduction of the mouse/windows/graphics environment in the research laboratory at an affordable price has spurred the development of new approaches to the control and monitoring of large experiments and new facilities for small data-acquisition and instrumentation-development systems. Further technical details are given in user manuals [17,18] which are available from the author, while the equipment itself is now manufactured industrially in France, Germany and Italy.

Acknowledgements

The development of MacVEE was funded and supported by C. Rubbia and members of the CERN UA1, ALEPH and OPAL experiment collaborations. 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, 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 cooperation. Special thanks are due to J.W. Stutz and M. Widmer of the Apple Computer Division of Industrade AG, Switzerland, for their invitation to present this paper at the European University Consortium Meeting at the University of Heidelberg in April 1988.

References

[1] T.R. Blakeslee, *Digital Design with Standard MSI and LSI - Design Techniques for the Microcomputer Age,* Wiley, New York, 1979, pp. 1-13.

[2] CAMAC, Updated Specifications, Vols. 1 and 2 (Commission of the European Communities, Luxembourg, 1983/84).

[3] VMEbus Specification Manual (VITA, Veldhoven, The Netherlands, 1985).

[4] E.M. Rimmer, CERN Implementation Recommendations for M68000-based CAMAC Port Controllers, *CERN Report, DD Division,* Geneva, 1983.

[5] MacUA1: The M68K-VME UA1 Macintosh-based Development System, *UA1 Technical Note,* CERN, Geneva, 1987.

[6] H. Von der Schmitt, RTF/68K. Real-Time Fortran-77 for 68K Processors - Manual of Compiler and Run-Time Library, *Heidelberg University Report,* 1986.

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

[8] B. Theurant, MacFB User Manual, *LPNHE Report,* Paris, 1987.

[9] A. Contin, Fortran Data Acquisition Software for Macintosh, *CERN Report, EP Division,* Geneva, 1987.

[10] F. Cindolo, MacDAQ - User-programmable Data Acquisition System, *BCFL Collaboration Report,* CERN, Geneva, 1988.

[11] W.J. Haynes, VME Tools, H1 VME Program Library, *DESY Report,* Hamburg, 1988.

[12] W.J. Haynes, The H1 VME-based Data Acquisition System, to be presented at the VMEbus in Research Conference, Zurich, 1988.

[13] P. Petta, AMERICA: A MacVEE External RAMdisk Install and Configure Application, *UA1 Technical Note 86-23,* CERN, Geneva, 1986.

[14] A. Dieckmann, unpublished communication, University of Heidelberg.

[15] LabVIEW Scientific Software for the Macintosh, National Instruments Corp., Austin, Texas, 1988.

[16] D. Samyn, unpublished communication, UA1 collaboration, CERN, Geneva.

[17] B.G. Taylor, The MacVEE Hardware User Manual, *CERN Report, EP Division,* Geneva, 1988.

[18] B.G. Taylor, The MICRON User Manual, *CERN Report, EP Division,* Geneva, 1988.