

CAMAPPLE: CAMAC INTERFACE TO THE APPLE COMPUTER*

Stephen H. Williams, Gerard J. Oxoby, and Quang H. Trang
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

The advent of the "personal" microcomputer provides a new tool for the debugging, calibration and monitoring of small scale physics apparatus; e.g., a single detector being developed for a larger physics apparatus. With an appropriate interface these micro-computer systems provide a low cost (1/3 the cost of a comparable minicomputer system), convenient, dedicated, portable system which can be used in a fashion similar to that of portable oscilloscopes.

Here we describe an interface between the Apple computer and CAMAC which is now being used to study the detector for a Cerenkov ring-imaging device. The Apple is particularly well-suited to this application because of its ease of use, hi-resolution graphics, peripheral buss and documentation support.

1. Introduction

An early use of dedicated computers in high-energy physics experiments was that of an IBM 1800 "minicomputer" in experiment E-11 at SLAC in 1967/68. This mini (meaning 16 bit) occupied four six-foot racks plus a card reader and printer (which used up the equivalent of three more rack spaces on the floor). Power consumption was measured in multiple KVA and was sufficient to heat the room in the mild California winters and cause air-conditioning problems in summer. The cost of several hundred thousand dollars was considered a bargain compared to central computing systems. Nevertheless this system was spoken of as a portable system.

Fortunately the machines available for dedicated use have evolved to the present range of minicomputers from the powerful VAX's to the truly mini LSI's. These machines are wide-spread in use at SLAC and consequently are well supported by a software staff and shared facilities.

However even the smallest systems, like the LSI with a terminal, dual floppy disks and printer, can be considered portable only because they reside in a single rack and can be moved with a single small forklift. Furthermore while excellent BASIC and FORTRAN language support exists, the documentation, while high in quality and detail, is somewhat intimidating by its thoroughness to the occasional computer user.

The emergence of the "personal computer" has introduced a new level of portability and applicability to small scale physics experiments and the support associated with larger experiments. These systems are low in cost, suitcase-portable, and have simple documentation for highly developed languages that require no software support group. Peripherals such as serial, parallel, ADC and DAC interfaces are available at a fraction of the cost of similar devices for the big brother mini's. While there is no doubt that the ultimate power of the mini's is greater than the micro-computer-based personal computers, the impact of the latter will be felt where that power is unused. In those cases the power of the personal computer is in its portability and ease of use for the unsophisticated computer user.

The cost of a personal computer system is quite low compared to the current minicomputers. The Apple system used in this laboratory includes dual 5 1/4 inch floppy disks, 80 column thermal printer, and the Pascal Language system. Its cost was \$4,200. The cost of an LSI system with terminal, printer, and dual 8 inch floppy disks is currently about \$12,000. This system does not have graphics capabilities, which can be obtained at additional cost.

Since CAMAC is the current standard instrumentation buss for high-energy physics experiments, the only peripheral not commercially available for these types of computers had to be developed. This missing link would allow access to the CAMAC based data which is generated in a test laboratory or test bench.

2. The Computer

The Apple Computer has most of the features common to the spectrum of personal computers: a well-developed BASIC called Applesoft, low resolution (40 by 48) color graphics, high resolution (280 by 192) graphics, analog input devices (game paddles), simple disk operating system, Pascal and FORTRAN with graphics implementation. In addition to these built-in features, many intelligent peripherals are available including: 5 1/4 inch floppy disks, a small thermal printer with graphics capabilities, RS232 serial interface, and analog to digital and digital to analog interfaces. These features are summarized in Table I.

TABLE I. MICROCOMPUTER FEATURES

Mainframe Hardware & Software Specifications

6502 microprocessor
64K bytes address space
ROM Monitor with Disassembler

Color Graphics

Low resolution: 40H x 48V, 15 colors
High resolution: 280H x 192V, black & white
140H x 192V, 6 colors

Sound generation

Cassette interface: 1500 bits per second

Game I/O interface

Two 150K OHM resistive ADC's
Three TTL inputs, four TTL outputs

Integral keyboard

Memory mapped video text page:
40H x 24V characters/lines

Integer BASIC

Applesoft BASIC: Floating point,
trig functions, graphics

Pascal: UCSD version with Turtlegraphics

FORTRAN: compiles Pascal p-code

Peripheral slots: memory mapped
with 2K PROM capability

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.

Peripherals Available

- 5 1/4 inch floppy disk with auto start DOS
- RS232 serial interface
- Parallel interface
- Thermal 80 column printer with graphics
- Graphics Tablet input device
- Multichannel ADC's and DAC's
- 8 inch floppy disk controller
- 3 3/4 digital voltmeter input device
- Real-time clock

Documentation is simple as it is written for a nontechnical market. A second level of hardware and software documentation exists for the more sophisticated user.

3. CAMAPPLE; The Interface

A simple CAMAC Crate Controller¹ is used at SLAC. It permits the computer interface (Branch Driver) to be adapted regardless of speed of execution and word length. The main objective was to have access to one or more CAMAC crates of information with a simple interface. However, with very little added circuitry this branch driver is very powerful.

The interface uses two modules. One housed within the Apple computer is a general purpose I/O subsystem. The other one is the Branch Driver, called CAMAPPLE. For convenience and economy it is plugged into one of the CAMAC crates from which its power is supplied. Figure 1 is a photograph of the system.

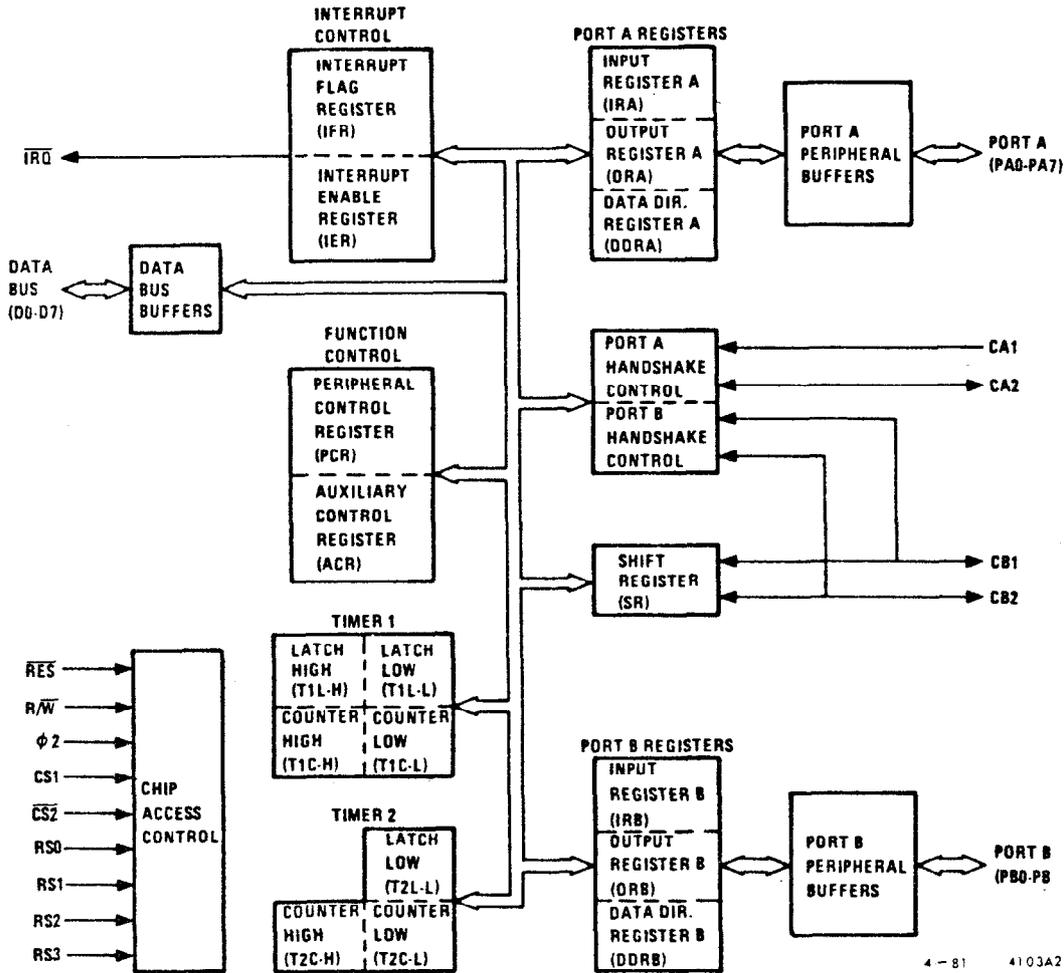
The I/O subsystem is based on the 6522 Versatile Interface Adapter (VIA) Integrated Circuit.² Its block diagram is shown in Fig. 2. The output of the VIA



10-81

4103A1

Figure 1. Photo of System Components



4-81 4103A2

Figure 2. 6522 Block Diagram

drives a flat ribbon cable connected to the CAMAPPLE which in turn, after some decoding, drives the SLAC CAMAC crate controllers.

4. System Implementation

As in any computer I/O system, CAMAC modules require the specification of an address and function. The CAMAC address is the crate number (C), the module station (N) and a four-bit subaddress (A). The function (F) is itself made of five bits, a total of seventeen bits. The scheme chosen decodes the two least significant bits of the VIA Output Register B (ORB) and latches the F, C, N and A information before it is transmitted to the CAMAC branch. The most significant bit of ORB is used to start the CAMAC cycle when set to a logical one and end it when reset to a logical zero. See Table II.

TABLE II. 6522 Registers Used

(1) Register 0: "ORB" Output Register B	
OFFF00:	Latch CAMAC function F
00ICCC01:	Latch CAMAC crate C
	I=0 → BNC interrupt
	I=1 → LAM interrupt
ONNNNN10:	Latch station N
OMAAAA11:	Latch subaddress A
	M=1 → mask interrupt
1XXXXXXX:	CAMAC cycle start
0XXXXXXX:	CAMAC cycle stop
(2) Register 1: "ORA/IRA" Output/Input Register A	
	Data to CAMAC if F is F16-F13 (POKE)
	Data from CAMAC if F is F0-F7 (PEEK)
(3) Register 2: "DDRB" Data Direction Register B	
	Always FF(HEX)
(4) Register 3: "DDRA" Data Direction Register A	
	0: Read from CAMAC
	FF: Write to CAMAC
(5) Register C: "PCR" Peripheral Control Register	
10H11011:	CA1 positive edge
	CA2 Pulse output
	CB1 Positive edge interrupt
	H=1 → normally
	H=0 → if handshake set with ORB
(6) Register D: "IFR" Interrupt Flag Register	
TXXX0XX0:	Test high bit for event
	Or generate 6502 interrupt
(7) Register E: "IER" Interrupt Enable Register	
EXXX0XX0:	E enables interrupt

Since only five bits are available to specify F, C, N and A, this information is passed to the latches with four transfers. The three remaining bits provide the following: (1) two possible interrupting sources to the Apple computer (front panel connector or the CAMAC line Look-At-Me (LAM)); (2) a general interrupt mask;

and (3) a handshake mode between the computer and the CAMAC system.

Data transfers to or from CAMAC are done via the VIA Input/Output Register A (IRA/ORA). The IRA is used if the function is F0-F7 involving the read lines within the crate dataway. ORA is used when the function is F16-F23 for a write to CAMAC command. Under software control the data Direction Register A (DDRA) must be set properly so that the VIA transceivers are pointed in the right direction according to the CAMAC function, inwards for input (F0-F7) and outwards for the data output (F16-F23).

5. Software Specifications

A. CAMAC Function and Address

A block diagram of the CAMAPPLE interface is shown in figure 3.

The FCNA information must be prepared in advance before a function can be successfully performed. The Data Direction Register B (DDRB) of the VIA, address COX2(HEX) where X corresponds to the location of the Apple based I/O board, must always be written with "FF"(HEX) for this application. The set-up is done with four transfers to the VIA ORB. This data is latched within the CAMAPPLE module when writing to the I/O address COX0(16). Bits 1 and 0 specify which CAMAPPLE register is being loaded. This is summarized in Table III.

TABLE III. CAMAC Function Encoding Scheme

BITS		COMMAND
1	0	
0	0	Load bits 6-2 to function register (F)
0	1	Load bits 4-2 to crate register (C)
1	0	Load bits 6-2 to station register (N)
1	1	Load bits 5-2 to address register (A)

B. CAMAC Cycle

A CAMAC cycle is started when the most significant bit of ORB(ORB7) is changed from a zero to one and is ended when ORB7 changes from one to zero. Setting or resetting ORB7 does not affect the content of CAMAPPLE F, C, N and A registers regardless of ORB1 and ORB0. Their content is gated to the branch Highway when ORB7 is set.

C. CAMAC Data Transfers

Data between the CAMAC system and the Apple computer is transferred via the VIA Input/Output Register A (IRA/ORA) at the address COX1(a6). Therefore the Data Direction Register A (DDRA) must be properly programmed. Its content must be 00 if the function set in CAMAPPLE is F0-F7, and to FF(16) if the function is F16-F23(10). Up to three data byte transfers can take place in either direction.

For a read command the CAMAC cycle is started by setting ORB7, then up to three "PEEK" instructions to IRA may be done to transfer a 24 bit CAMAC word in three successive bytes before resetting ORB7. Similarly only one or two bytes can be accessed if desired. The first byte transferred is always the least significant one of the CAMAC word after ORB7 has been set to a one.

In the case of a write command the process is reversed. Up to three "POKE" instructions to ORA may be done. Once the data has been transferred to CAMAPPLE, ORB7 is set to start the CAMAC cycle and reset to end it.

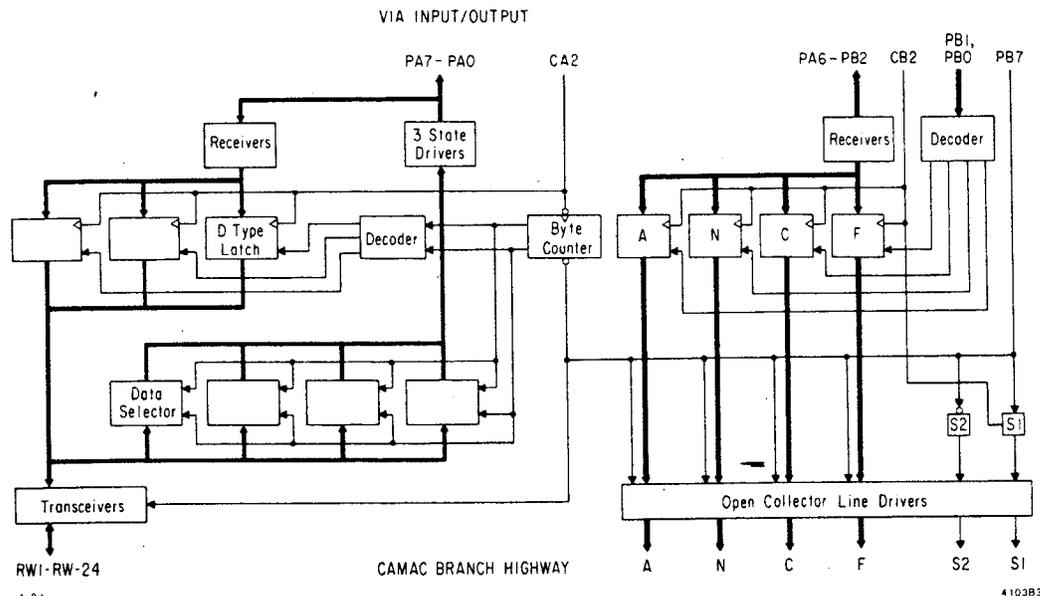


Figure 3. CAMAPPLE Interface Block Diagram

D. External Interrupts

The interface can handle one of two interrupts. The software selects the interrupting source with bit 5 of the CAMAPPLE C register. When bit 5 is zero the interrupting source is from a LEMO connector located on the front panel, when set to a one the CAMAC Look-At-Me (LAM) signal may generate an interrupt. Bit 6 of the A register within the CAMAPPLE module is set to one. This bit acts as an overall interrupt mask. When set to a one the interrupt request is sent to the VIA.

E. Handshake Mode

Several CAMAC functions require a handshake mode. Such as a Test LAM (F8) or Test Status (F27) command. When these commands are presented to a CAMAC system, the addressed module generates the appropriate Q response. It is then necessary that the host computer recognize the response since no data transfer occurs. We have implemented this with the handshake feature of the VIA.

F. Interface Intelligence

A Programmable Read Only Memory (PROM) is located on the I/O board for users defined applications; 256 bytes are accessible with an address ranging from CX00 to CXFF where X is the slot number in which the I/O card is plugged-in.

A 1K-byte PROM was used because of its availability at SLAC. This permits four different routines of 256 bytes each. They are selected by the Dual-In-Line switch pack mounted on the I/O board.

The interface has been made intelligent by the addition of a PROM software which makes it unnecessary for the user to know anything about the interface except the calling address. The Apple peripheral slots each have a command associated with them which calls the lowest address on the PROM. This command is "PR#n" where "n" is the peripheral slot number 0-7.

The CAMAPPLE PROM assumes the existence of seven BASIC variables: F, C, N, A, D, IR and OR. F, C, N, A, constitute the CAMAC function; D is the 16-bit datum read to or from CAMAC depending on the function f;

and IR is a pointer set by the PROM to point to the interrupt flag register (for test/handshake); OR is set nonzero if CAMAC Q signal was present. These mnemonic names allow the user to interact with the CAMAC buss directly or through a BASIC program. For example, after power-on the only commands necessary to execute a CAMAC instruction are:

```
F=0
C=2
N=15
A=4
PRINT D
```

which prints the contents of subaddress 4 at station 15 in crate 2. Repeated readings of the same location can be accomplished with PR#3:PRINT D, since the CAMAC instruction has been latched into the interface.

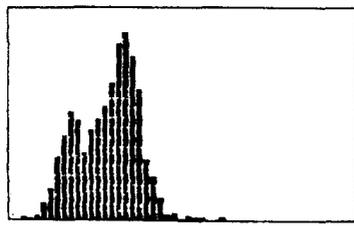
6. APPLICATIONS

The following is a list of applications of CAMAPPLE at SLAC:

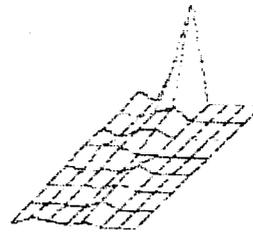
A. The CAMAPPLE system was used to replace the readout function of a NOVA 1200 computer which was used to monitor and control a 200 kilowatt beam line magnet system while the NOVA was out for repairs.

B. The CAMAPPLE system was used to provide local intelligence at a remote site to evaluate new beam position monitors being installed in the SLAC linac as a part of a computerization of the accelerator control.

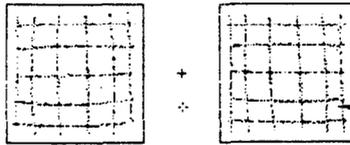
C. The primary application of the system has been in a test lab for the development of a Cerenkov ring-imaging detector.³ This detector must detect single photons and record their position in x and y coordinates. The high-resolution graphics and its ease of application combined with the CAMAPPLE interface to the CAMAC data base make a versatile system. Some typical displays from this work are shown in figure 4.



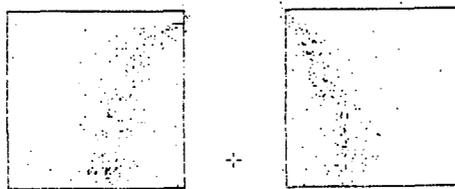
Iron-55 pulse height spectrum



Induced charge on 8x8 pad array



x-y Charge division readout on 2 resistive anodes



Čerenkov ring image

9-81

4103A4

Figure 4. Use of CAMAPPLE and APPLE in Detector Development

7. Conclusion

The CAMAPPLE interface provides access to the CAMAC buss from the Apple computer. This adds a new tool to the spectrum of dedicated computers available for high-energy physics hardware development.

References

1. R. Larsen and D. Horelick, Stanford Linear Accelerator Center preprint SLAC-PUB-1653.
2. SY6522, Versatile Interface Adapter, Synertek, Inc., P. O. Box 552, Santa Clara, California 95052.
3. R. S. Gilmore. W. M. Lavender, D. Leith, S. H. Williams, IEEE, NSS, November 5-7, 1980.