ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Bologna

INFN/TC-82/6
26 Aprile 1982

I. D'Antone : CMI, CAMAC-MULTIBUS INTERFACE.

Servizio Documentazione
dei Laboratori Nazionali di Frascati

Istituto Nazionale di Fisica Nucleare
Sezione di Bologna

INFN/TC-82/6
26 Aprile 1982

I. D'Antone: CMI, CAMAC-MULTIBUS INTERFACE.

ABSTRACT.

A CAMAC data acquisition system based on the INTEL 8085 microprocessor is described. It includes a special crate controller (CMI, CAMAC-MULTIBUS Interface) which adapts directly the Dataway to the bus computer, i.e. MULTIBUS. This interface is connected to the computer as an I/0 device.

1. - INTRODUCTION.

It is well known that a CAMAC crate can be controlled by means of a computer, using a special module in station 25 (or station 24+25) which performs the double function of a crate controller and of a Computer-CAMAC interface. In such a scheme, a CAMAC crate can be treated as if it were a peripheral device attached to the I/0 channel of the computer.

A CAMAC data acquisition system based on the INTEL 8085 microprocessor was developed at the Bologna INFN laboratory where a special crate controller (CMI, CAMAC-MULTIBUS Interface), which adapts directly the Dataway[1] to the bus computer, i.e. MULTIBUS/IEEE 796, has been built.

The CMI module can be employed as an adapter for microcomputers using a MULTIBUS. On the other hand, with SBC standard boards, the CMI can act both as a stand-alone system for data acquisition in small systems and as an acquisition node if it is coupled to a minicomputer. In this case data acquisition from CAMAC is completely controlled by the microprocessor which can eventually execute also a preliminary elaboration, with a floating point processing unit.

2. - CMI HARDWARE DESCRIPTION.

CMI board is housed in a double-width CAMAC module to be installed in station 24 and 25 as a normal crate controller.

The block diagram of the interface between the microprocessor and the CAMAC module is shown in Fig. 1, where the most relevant elements indicated are:

a) MULTIBUS interface ;

b) N. A. F. interface ;

c) CAMAC timing and control ;

d) R/W interface ;

e) LAM interface.

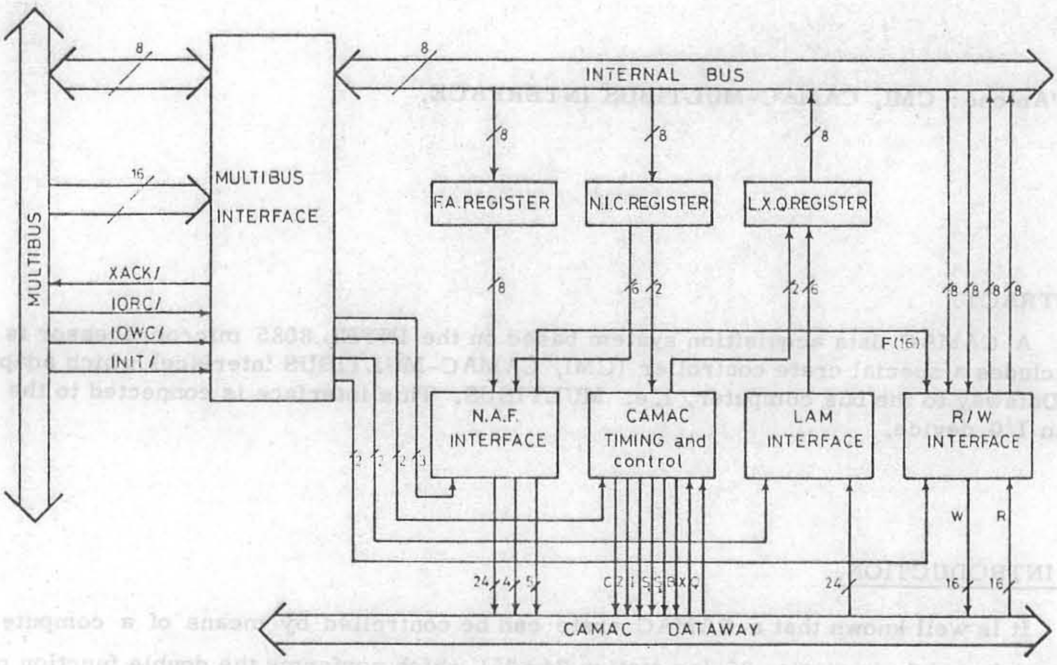


FIG. 1

Since we employ a 8-bit microprocessor we use two 8-bit registers to write Station number N, Subaddress A, function F, and one 8-bit register to read LAM. X. Q. on the CMI board. They are organized as follows :

Bit	7	6	5	4	3	2	1	0
F, A register	F_1	F_2	F_4	F_8	F_{16}	A_1	A_2	A_4
N, I, C register	I	C	N_1	N_2	N_4	N_8	N_{16}	A_8
LAM, X, Q register	ΣL	L_{16}	L_8	L_4	L_2	L_1	X	Q

A CMI board has its own registers connected as I/0 devices, i.e. they are I/0 MAPPED, then they can be accessed only by using the INPUT/OUTPUT instructions.

In a given system, each CMI board will have its own base address to communicate with the main processor. This base address is programmable with four switches mounted on the CMI board. The addressing of each CMI board is done by using the MULTIBUS address lines A_4, A_5, A_6, A_7 with a program address that matches the manually programmed switch address. Therefore, up to 16, CMI boards can be connected separately to the computer as I/0 devices. All CMI board interact with the microcomputer via a single path, the MULTIBUS.

2.1. - MULTIBUS Interface.

This interface, shown in Fig. 2, connects the MULTIBUS data bus with the on-board data bus which is connected to the LAM and N.A.F. registers; these are written or read by means of the board I/0 commands. These registers are selected by SELi signals, where i is a value added to a base address of the CMI board.

Furthermore there is a MULTIBUS control logic that accepts the IORC/, IOWC/, INIT/, signals of the MULTIBUS and generates the acknowledge XACK/ to the CPU[2].

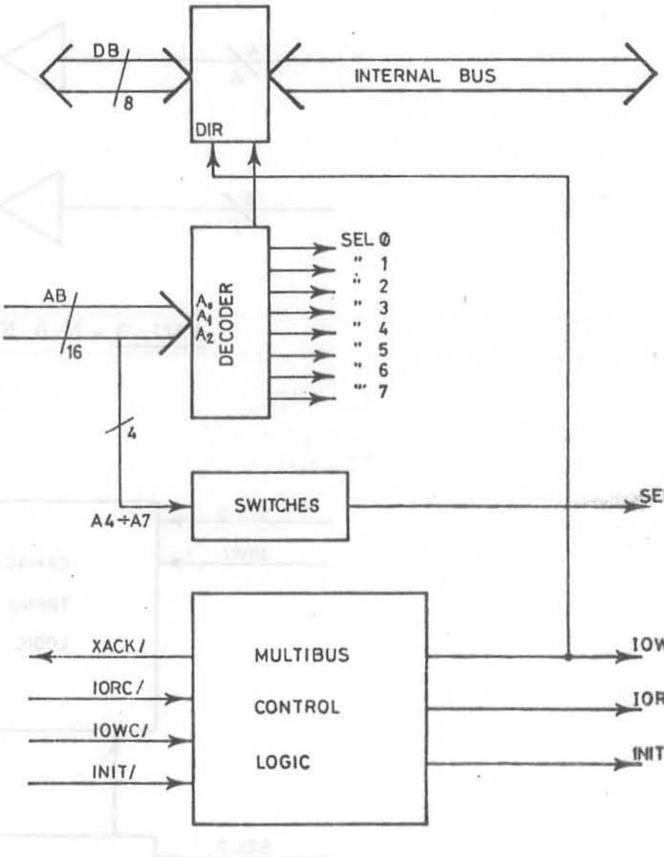


FIG. 2 - MULTIBUS Interface.

2.2. - N.A.F. Interface.

The N.A.F. interface, shown in Fig. 3, is used to select in the CAMAC crate:

a) station number N (N_1-N_{24} : 24 lines);

b) sub address A ($A_1A_2A_4A_8$: 4 lines);

c) function F ($F_1F_2F_4F_8F_{16}$: 5 lines).

This interface employs the bits in the F, A and N, I, C registers to generate with decoders, inverters and buffers the N.A.F. signals on the Dataway.

The F, A register is selected by IOWC/ and SEL3, while the N, I, C register is enabled by IOWC/ and SEL4.

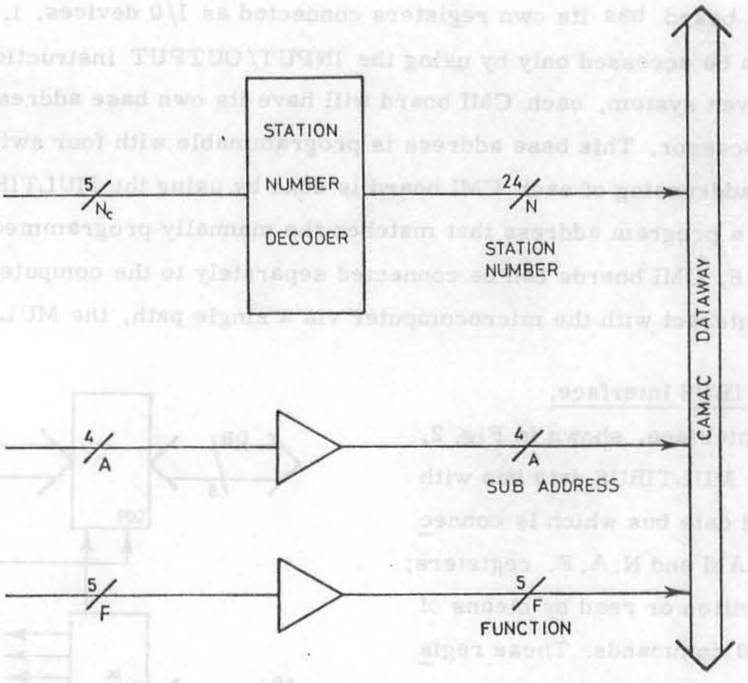


FIG. 3 - N.A.F. Interface.

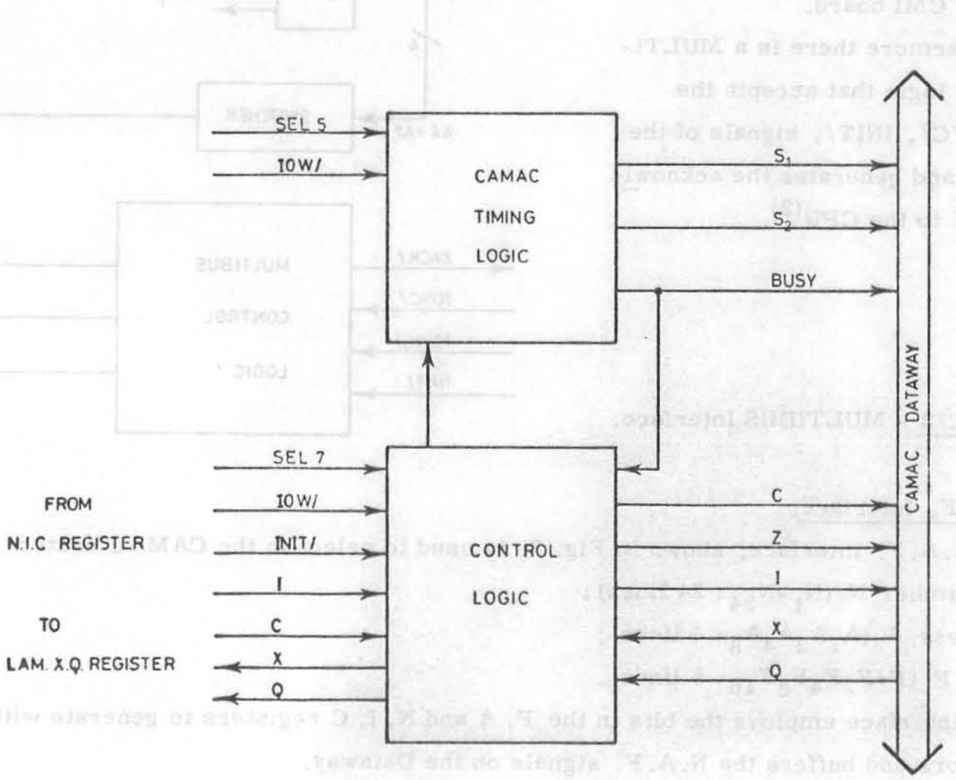


FIG. 4 - CAMAC timing and Control Interface.

Before executing a CAMAC operation, slot number N, function number F and subaddress A must be set; this is obtained by two write operations on the F, A and N, I, C registers. A new set of these registers is not necessary if a previous CAMAC operation used the same values. For example, if only one CAMAC module is being used, its slot number can be set at the beginning of the program, and need not to be set for each operation.

2. 3. - CAMAC timing and control.

This block, which is shown in Fig. 4, generates the S_1, S_2 and Busy signals, when IOWC/ and SEL5 are activated and reads the X and Q CAMAC lines which are memorized as bit 0 and bit 1 of the LAM, Q, X register.

Furthemore it is possible to write bit 7 and 6 of the N, I, C register and to generate the I, C signals in the initial phase according to these bits.

Z signal can be activated by program with IOWC/ and SEL7 and is transferred by control logic on the Dataway.

2. 4. - R/W Interface.

This interface, which is shown in Fig. 5, contains two 8-bit registers for R-lines and two 8-bit registers for W-lines. We have chosen 16 bit because they fit the majority of CAMAC modules.

For example, a read operation from CAMAC is carried out in two steps:

a) the 16 bits of the CAMAC R lines are memorized on a latch, with S_1 signal;

b) the data are transferred to the CPU when IORC/ and SEL0, SEL1 are activated.

R/W control logic allows reading from these registers only if F function is decoded as F(0) to F(15).

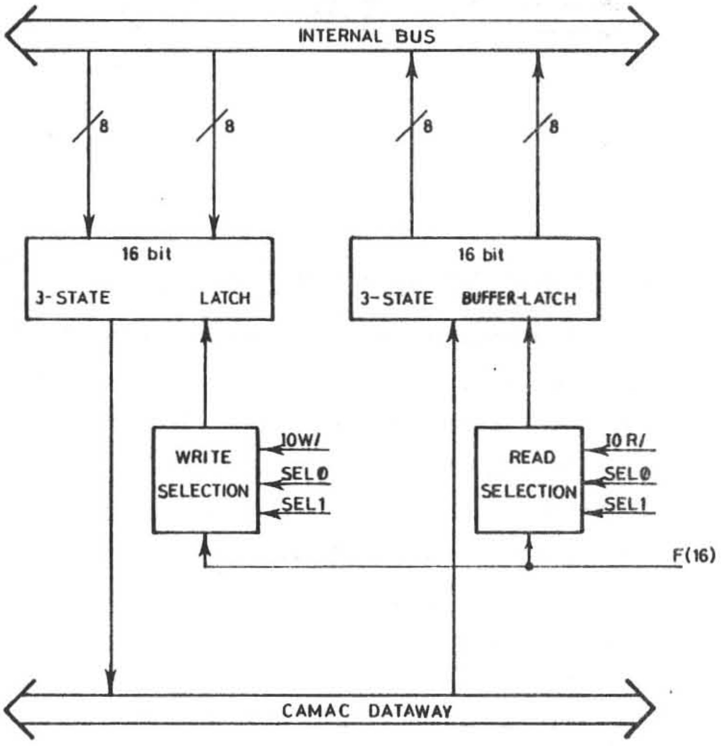
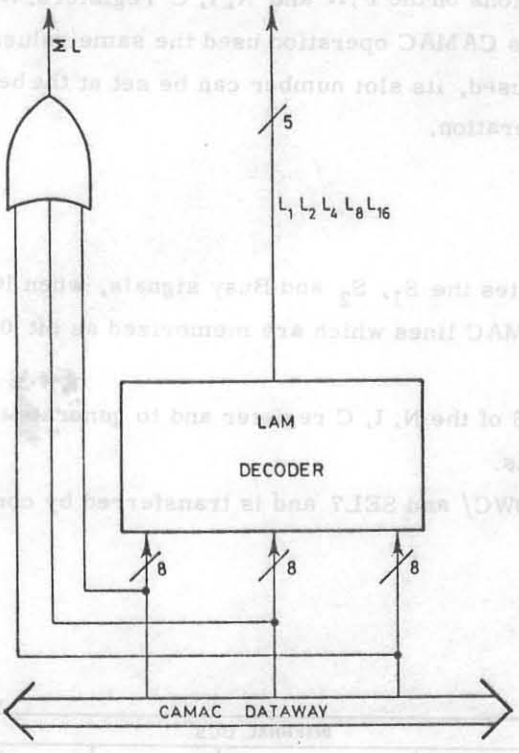


FIG. 5 - R/W Interface.

2.5. - LAM Interface.



This interface, which is shown in Fig. 6, codes the L lines and memorizes this codification in the LAM register. There exists also the ΣL line, i.e. the ORing of all the L lines, that can be used to issue an interrupt to the CPU when a CAMAC module generates a LAM request.

When the CPU reads the LAM register, with IORC/ and SEL2, it also identifies the module interrupting.

FIG. 6 - LAM Interface.

3. - CAMAC INTERFACE SOFTWARE.

A number of home made routines has been implemented to handle the CMI interface. These routines are written in PLM[3] language, as follows:

- CAMAZ: it generates a Z signal, thus allowing CAMAC modules to be cleared;
- CAMCZI (I, C): routine to generate I and C CAMAC control signals;
- CAMINW (N, F, A, ADD, NUM, X, Q): it executes NUM CAMAC cycles and stores NUM words (16 bits) in the array D addressed by ADD, further it returns the status of X, Q lines in X, Q;
- CAMOUW (N, F, A, ADW, NUM): it executes NUM CAMAC cycles, writing NUM words from array W, addressed by ADW;
- LAM: it returns the station number that has generated the LAM request.

These routines can be used by a PLM main program with the

CALL name (variables)

instructions.

4. - CMI APPLICATION.

CMI board has been used tobuild a stand-alone system for data acquisition(*) (Fig. 7).

This system is built by using standard boards based on the microprocessor INTEL 8085, and it contains the following parts :

a) CMI board ;

b) SBC 80/24, containing the 8-bit CPU 8085, RAM and ROM memory, 2 USART (one USART is on the iSBX 351 Serial I/0 board) ;

c) hardware and software interface for video terminal, cassette magnetic type unit.

PLM language test program (~1.5 Kbytes) has been written on the EPROM 2716 to handle various Kinds of CAMAC modules : ADC's, TDC's, pattern units, etc.

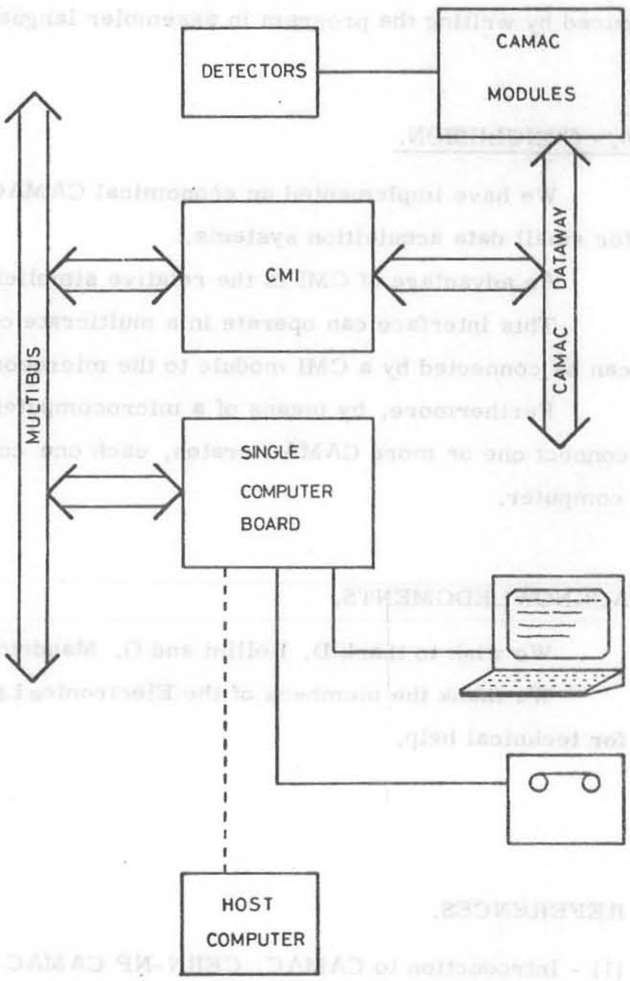


FIG. 7

5. - CONSIDERATIONS ON THE CMI BOARD.

To execute for example a read CAMAC operation in PLM language one must set slot number F, subaddress A and call CAMINW routine by the

CALL CAMINW (N, F, A, ADD, NUM, X, Q)

instructions.

(*) - CMI module has been used as a part of a data acquisition system for the INFN Bubble Chamber Group of Bologna University.

As we can read 1 word every ~150 microseconds it is obvious that this solution with CMI is applicable for data acquisition systems with suitable data rate.

This execution time depends on the CPU clock and on the program structure ; it can be reduced by writing the program in assembler language or by increasing the CPU clock frequency.

6. - CONCLUSION.

We have implemented an economical CAMAC-MULTIBUS Interface that is easily-handled for small data acquisition systems.

An advantage of CMI is the relative simplicity and low cost.

This interface can operate in a multicrate configuration. In this case each CAMAC crate can be connected by a CMI module to the microcomputer as an I/0 device.

Furthermore, by means of a microcomputer operating as a front-end, it is possible to connect one or more CAMAC crates, each one controlled by a CMI module, to a single host computer.

ACKNOWLEDGMENTS.

We wish to thank D. Bollini and G. Mandrioli for discussions and clarifications.

We thank the members of the Electronics Laboratory, in particular W. Lelli and G. Sola, for technical help.

REFERENCES.

(1) - Introduction to CAMAC, CERN-NP CAMAC Note 25-00 (1971).

(2) - Intel Corporation: Intel Multibus Specifications, Santa Clara, California. MULTIBUS is a patented Intel bus.

(3) - PLM is a high-level language designed for system and applications programming for the Intel 8080/8085 microprocessors.