ISTITUTO NAZIONALE DI FISICA NUCLEARE
Laboratori Nazionali di Frascati

LNF-85/9

O. Ciaffoni, M.L. Ferrer, L. Trasatti, M. Coli:
CANDI2, A CAMAC MICROCOMPUTER SYSTEM FOR A PHYSICS LABORATORY

Estratto da:
N.I.M. A234, p. 288 (1985)

Servizio Documentazione
dei Laboratori Nazionali di Frascati
Cas. Postale 13 - Frascati (Roma)

Nuclear Instruments and Methods in Physics Research A234 (1985) 288–293
North-Holland, Amsterdam

CANDI2, A CAMAC MICROCOMPUTER SYSTEM FOR A PHYSICS LABORATORY

Orlando CIAFFONI, Maria Lorenza FERRER, Luciano TRASATTI
INFN, Laboratori Nazionali di Frascati, CP 13, 00044, Frascati, Rome, Italy

Moreno COLI
ENEA, Laboratori Nazionali di Frascati, CP 70, 00044, Frascati, Rome, Italy

Received 17 May 1984

CANDI2 is an intelligent CAMAC Crate Controller which implements high resolution color graphics and is intended to be easy to use in a laboratory environment for both data acquisition and process control.

1. Introduction

The need for data acquisition and process control systems based on microcomputers has been developing at an incredible rate in the last few years. At the same time the amount of equipment commercially available has been increasing very rapidly. Nevertheless, in a specialized field like that of physics, a large amount of work is usually necessary to adapt a commercial system to the needs of the laboratory.

We have tried to overcome this difficulty designing a microcomputer system (see refs. [1–15]) which is reasonably general in purpose, but has built in facilities for the more common requirements of our environment.

The system CANDI2 (CAMAC Acquisition Node for Distributed Intelligence) has been designed at the Laboratori Nazionali di Frascati (INFN) with financial support from INFN and with the support of the Research Division and of the Computer Center.

The system is now being assembled and distributed by a commercial firm.

The facilities implemented in CANDI2 are:

1) CAMAC interface: CAMAC is an accepted and widespread data acquisition system. Therefore CANDI implements the functions of a Crate Controller, and the whole computer is built on CAMAC boards. CANDI2 is a 4-slot CAMAC module, which occupies the rightmost positions in the crate, substituting the normal controller.
2) PDP-11 and VAX interface: The use of computers of the Digital family as host computers has been decided considering their large and increasing number. Memory image interchange, ASCII file transfer and Transparent terminal utilities are available for the CANDI system.
3) Color graphics. The use of high resolution graphics for data display, event presentation and in general for a better man–machine interface is rapidly developing. A high resolution color display unit is built into CANDI2.

The main characteristics of the CANDI2 system are:

1) CPU: TMS 9995, 16-bit word, 3 MHz clock.
2) RAM: 128K resident on board; can be expanded off-board up to 16 MB using a memory mapping technique implemented on the CPU board.
3) EPROM: 32K maximum on board containing system firmware. Can be expanded off-board.
4) I/O ports: 2 RS-232C or 20 mA current loop serial ports. Two parallel PIOs on front panel.
5) Printer port: serial, RS-232C.
6) CAMAC interface: replaces the normal Crate Controller.
7) Graphics interface: 8 colors, 2 memory pages, 512 $\times$ 512 pixels, TTL-RGB output with separate synchronism.
8) Capable of color hard-copy.
9) Optional alphanumeric keyboard with alphanumeric display superimposed on the graphic screen.
10) VAX–PDP11 Interface: RS232C or 20 mA current loop, up to 9600 baud.
11) BASIC Interpreter in firmware, 16K.
12) Standard NIM-ESONE CAMAC subroutines implemented in firmware.
13) High level graphic subroutines, PLOT-10 compatible in firmware.
14) Tektronik 4006 emulation in firmware.
15) Several software utilities in firmware (assembler, disassembler, memory inspect/change/dump, memory mapping, etc).

0168-9002/85/$03.30 © Elsevier Science Publishers B.V.
(North-Holland Physics Publishing Division)

2. Hardware configuration

The system consists of three CAMAC boards: CPU, CAMAC Controller and Graphics interface, plus the optional alphanumeric terminal interface.

2.1. CPU system board

The CPU board of CANDI2 is a complex master processing unit incorporating various I/O communication ports. The system can be divided in the following blocks:

(1) CPU: Texas Instruments TMS 9995, 16-bit processor with related logic circuitry to generate and handle the system control signals (memory driving, interrupt handling, I/O signals including Reset, etc.).

(2) EPROM memory: 32 Kbytes containing the system firmware, on four 8 Kbytes EPROMs.

(3) DYNAMIC RAM memory and driving circuitry, consisting of 128 Kbytes (16 dynamic 64 Kbit RAM chips, featuring auto-refresh capability). Pulses for auto-refreshing are stolen during non memory addressing intervals. Access to the whole 160 Kbytes is obtained by changing blocks or pages through appropriate settings of the MEMORY MAPPER registers.

(4) The MEMORY MAPPER and its service latches are accessed directly as a memory position. The MAPPER contains 16 registes, 12 bits wide, containing the extended address (8 + 4 bits) selected by the most significant nibble of the CPU address-bus. The least significant nibble of the 12-bit address contained in the mapper registers replaces the CPU most significant nibble at the mapper output address-bus.

The resulting address for the system becomes:

(a) 12 bits coming directly from the 12 l.s. bits of the CPU address-bus.

(b) 12 bits coming from the registers of the mapper that can be changed via software, giving the possibility to access a memory space of 16 Mbytes (4K times 4 Kbytes memory blocks).

(5) The I/O communication system consists of two serial ports and one parallel port (22 I/O bits + 16 prioritized interrupt inputs).

(a) SERIAL port B is an EIA Standard RS 232/C or TTY current-loop port defined as a DCE communication unit to be connected to the system I/O terminal. This can be any standard terminal or the I/O from the optional internal alphanumeric display board. In parallel to serial port B an output is provided on the front panel (PRINTER PORT) for a printer supporting EIA RS 232/C standard signals.

(b) Serial port A is again a standard RS232/C port defined as a DTE communication unit. If differs from Port B also in that the current-loop signals are optically coupled. Various functions on the I/O serial ports can be altered changing the position of appropriate jumpers. This port has been implemented as I/O port for the HOST COMPUTER (PDP-11, VAX, IBM or HP).

(c) The PARALLEL PORT has been implemented with a TMS 9901 parallel port of the TMS 9900 family. It can be read/written to in Communication Register Unit space and can provide:
  (i) 16 bits of I/O addressed one or more bits at a time.
  (ii) 6 more bits on separate pins which can be used only as inputs.
  (iii) some of the same pins which are used for I/O can also be used to input 16 interrupt lines. These interrupts can be individually enabled by software, and they are automatically prioritized and encoded on four output wires.

2.2. CAMAC interface

The CAMAC interface is built on a board which sits in slot 25 of the CAMAC crate. The interface to slot 24 is supplied by the CPU board through flat cables. The system uses five PIOs (TMS 9901 Parallel Input Output Interfaces) to interface to the CAMAC highway, with the possibility to access the full 24-bit R/W bus.

The system only controls one crate, since we felt that intercrate communication and multicrate data acquisition would be much better performed by various CANDI units loosely interconnected by a Local Area Network. An HDLC (High level Data Link Control) link and an Ethernet interface are being developed to communicate between CANDI2 units.

The fact that CANDI2 does not require a standard Crate Controller makes the system more compact and efficient and much less expensive.

The CAMAC Interface board also contains two PIOs and the associated logic to implement the parallel connection to a host computer or to a floppy disk system (under development).

2.2.1. LAM handling

On top of the standard LAM handling, CANDI2 has a built in facility to ease the use of LAMs in a simple experiment. The crate is divided in three sections (N1 to N8, N9 to N16, N17 to N24), and the LAMs from these sections are directly routed to three different interrupt levels of the CPU. Moreover, any LAM from the crate will trigger interrupt level 12.

2.3. Graphic system

Graphic display units are growing very fast in popularity. The immediate reason is the rapidly decreasing cost of hardware in general (and of memories in particular), while the real need they serve is to answer the classic question: *Who will ever read all of those numbers?*, as referred to multi-kilogram computer printouts.

CANDI Graphics is a high resolution (512 × 512 pixel), 8 color, 2 page raster scan interlaced display unit. The system uses a high resolution raster-scan video terminal and supports a hard-copy unit.

Pixel-by-pixel reads and writes are possible, both for the hard copy unit and for cursor display. Furthermore, the unit is capable of understanding high level commands, that is, point draw/erase, vector draw/erase and alphanumeric character display.

Two separate pages of memory are available, allowing independent preparation of two separate displays. Alternate presentation is also possible, with some amount of "flickering"; this could still be useful, for example for comparison of two different curves.

We have connected the unit to an optional alphanumeric display processor and to a keyboard. This allows presentation of both graphic displays (for example drawings of apparatuses, event display, plus miscellaneous histograms and curves relating to experimental results) and alphanumeric information (program listings, etc.). It is important that the two things are handled by separate processors, to allow scrolling, erasing and editing of alphanumeric data without disturbing the somewhat more complicated graphics underlying it. The alpha display color is automatically adjusted for maximum readability over differently colored areas of the graphic display.

The unit has its own 192 Kbyte dynamic memory, which can also be accessed directly by the CPU.

A graphic color printer can be connected to the system to implement a pixel by pixel hard copy unit.

2.4. Memory configuration

The CANDI2 memory configuration is shown in fig. 1. Two and one half 64K pages are implemented on the CPU board. The Power On configuration is shown as page 0 (32K EPROM + 32K RAM).

Note that, since the Memory Mapper is accessed via the memory I/O space, it is essential that this space be accessible even if the computer is not configured in page 0, otherwise a "no return" situation would occur. On the other hand, for some applications it could be useful to have an uninterrupted memory configuration (e.g. 64K continguous RAM). This is possible in the CANDI2 system using a dedicated CRU location. Executing a CALL "WNDIS" will make the I/O window disappear and be replaced by continuous memory, while executing a CALL "WNENA" will make it reappear always at locations 7F00H to 7FFFH, independently of the Memory Mapper configuration.

3. Software configuration

The CANDI2 system is based on a 16 KByte BASIC interpreter (see manual), plus 16 more KBytes of utilities (CANDI firmware). CANDI firmware is burned into 2 64 Kbit EPROMs. It is organized as a set of machine language subroutines which can be recalled from BASIC.

Table 1 is a list of the available subroutines. For more details refer to the CANDI2 User's manual (ref. [13]).

3.1. Subroutine description

Available routines can be divided in five groups (see table 1 for complete list):
1) CAMAC interface,
2) Communication with the host computer,
3) Graphics,
4) Assembly language programming routines,
5) Memory mapping.

3.2. CAMAC subroutines

The NIM-ESONE set of CAMAC interaction subroutines is becoming an accepted standard (see ref. [15]). The whole set is implemented in firmware, with the addition of a few more routines specifically written to take advantage of the peculiar features of CANDI. A list of the subroutines is shown in table 1.

3.3. Communication with the host computer

Three routines are available for communication with a PDP-11 or with a VAX computer:

(1) Transparent Terminal. This routine makes the microcomputer transparent, thus allowing use of the keyboard to talk directly to the host. The routine also performs a complete Tektronix 4006 emulation, translating Tektronix graphics to CANDI graphics, and thus

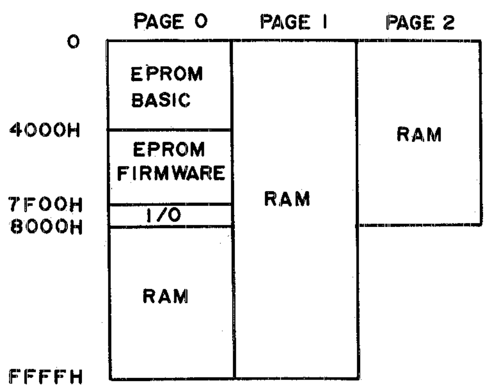

Fig. 1. CANDI2 memory configuration.

Table 1
CANDI2 utilities

NIM-ESONE CAMAC subroutines	
CALL "CDREG"	Compute CAMAC address
CALL "CFSA"	Perform single CAMAC action (24-bit)
CALL "CSSA"	Perform single CAMAC action (16-bit)
CALL "CFGA"	Perform a series of CAMAC actions
CALL "CARPT"	Repeat n times the last CAMAC action
CALL "CDZER"	Clear CAMAC address buffer
CALL "CCCZ"	Perform a CAMAC Z function (initialize)
CALL "CCCC"	Perform a CAMAC C function (clear)
CALL "CCCI"	Set or reset CAMAC I (inhibit)
CALL "CTCI"	Test CAMAC I
CALL "CCCD"	Enable or disable Crate Demand
CALL "CTGL"	Test Crate Demand
CALL "CLDEC"	Decode Interrupt LAM's
CALL "CCLWT"	Wait for LAM
CALL "CDLAM"	Encode a LAM identifier
CALL "CDZEL"	Clear LAM identifier
CALL "CCLM"	Enable LAM
CALL "CCLC"	Clear LAM
CALL "CTLM"	Test LAM
CALL "CGREG"	Decode address buffer
CALL "CGLAM"	Decode LAM identifier
CALL "CFUBC"	Perform multiple CAMAC action (Q stop, 24-bit)
CALL "CSUBC"	Perform multiple CAMAC action (Q stop, 16-bit)
CALL "CFUBR"	Perform multiple CAMAC action (Q repeat, 24-bit)
CALL "CSUBR"	Perform multiple CAMAC action (Q repeat, 16-bit)
CALL "CFUBL"	Perform multiple CAMAC action (L sync, 24-bit)
CALL "CSUBL"	Perform multiple CAMAC action (L sync, 16-bit)
CALL "CFMAD"	Perform multiple CAMAC action (address scan mode, 24-bit)
CALL "CSMAD"	Perform multiple CAMAC action (address scan mode, 16-bit)
CALL "CENAB"	Enable part of a crate for LAM
CALL "CDISAB"	Disable part of a crate for LAM
Host computer interaction subroutines	
CALL "SER"	Memory image transfer (RS/232 or 20 mA current loop line)
LOAD2	Ascii Basic source transfer (RS/232 or 20 mA)
SAVE2	Ascii Basic source transfer (RS/232 or 20 mA)
CALL "TRASP"	Transparent terminal emulation (including Tektronix 4006)
Graphic subroutines	
CALL "INIT"	Graphic system initialization
CALL "PAGE", P	Change page
CALL "BKG", BK, COL	Change background color
CALL "COL", COL	Change writing color
CALL "CUR"	Display graphic cursor
CALL "VECA", X, Y	Draw absolute vector
CALL "VECR", DX, DY	Draw relative vector
CALL "VECDA", X, Y, AT	Draw dotted absolute vector
CALL "VECDR", DX, DY, AT	Draw dotted relative vector
CALL "MOVA", X, Y	Move position absolute
CALL "MOVR" DX, DY	Move position relative
CALL "PNTA", X, Y	Plot point absolute
CALL "PNTR", DX, DY	Plot point relative
CALL "RECT", X, Y	Draw rectangle
CALL "FRECT", X, Y	Draw full rectangle
CALL "CRCMF", R, X, Y	Draw circumference
CALL "CIRCLE", R, X, Y	Draw circle
CALL "CSIZ", SX, SY	Set size for hardware character generator

Table 1 (continued)

CALL "STRING", HV, AT, (STRING)	Draw alpha string
CALL "RGMEM", (RED), (GREEN), (BLU)	Read graphic memory
CALL "WGMEM", RED, GREEN, BLU	Write graphic memory
CALL "HARD"	Print hardcopy
CALL "OUT", DISP, PRINT	CRT display and Printer on or off
CALL "ACRT", EN	CRT alpha enable
CALL "CPOS", X, Y	Return current graphic coordinates
Assembly language programming subroutines	
CALL "ASSEMB"	TM9995 assembler
CALL "DISAS", STADD, EADD	TM9995 disassembler
CALL "MEMY"	Memory inspect/change/dump
CALL "TRUNC", IN, OUT	Floating point to integer conversion
Memory mapping	
CALL "WRMA", STADD0, NBYTES, PAGE, STADDN	Write from page 0 to page *N*
CALL "RDMA", STADD0, NBYTES, PAGE, STADDN	Read from page *N* to page 0
CALL "MAPPIN", PAGE NBLOCKST, NBLOCKE	Allows to use two pages contemporarily
CALL "MAPONE", NRG, PAGEN, NBLOCK	Write one register of the mapper
CALL "WNENA"	Allow addressing the I/O window (7F00H to 7FFH).
CALL "WNDIS"	Disable the I/O window replacing it with continuous memory
CALL "FILMA", DATA, NBYTES, PAGEN, STADDN	Fill NBYTES of page PAGEN starting from address STADDN with the number DATA.

making available all the graphics implemented on big host computers.

(2) Serial link for memory image file exchange through an RS/232 link.

(3) Serial link for source BASIC file exchange and BASIC printout into the host computer. This is particularly useful for source BASIC programs, where the ASCII code is generally much shorter than the corresponding memory image. Two more advantages are gained by using this type of storage: (a) the BASIC code can be edited directly on the host computer; (b) the code is relocatable, i.e. it can be saved and reloaded in different memory area.

It is also possible to produce output from a BASIC program onto a file on the host.

3.4. Graphics routines

Twenty six graphic routines have been implemented (see table 1) to enable the user to access easily all the graphics capabilities of the system, up to vector generation, circle drawing, graphic memory read/write, alphanumeric string drawing and Tektronix 4006 emulation.

3.5. Assembly language programming subroutines

It is possible for the CANDI user to write his own assembler language subroutines. This can be very useful when execution speed is important, since assembler language programs execute much faster than the corresponding BASIC programs (but are more difficult to write).

Routines can be inserted into memory in three ways:

1) Using the MEMY CANDI routine and inserting the routines directly in machine language.
2) using a cross-assembler program running on a VAX. This program interprets the Texas assembler language and produces object code, which can be directly down-loaded to CANDI using the SERIAL CANDI routine.
3) Using the CANDI2 resident assembler, which produces object code from source given by terminal. The assembler works in a single pass, allowing the use of labels two characters long.

A disassembler program in firmware allows to restitute standard TM 990 assembler language from object code.

3.5.1. Memory mapping routines

Seven routines are available to use a memory address space larger than allowed by the CPU address bus (64 Kbytes): Using these routines, all of the CANDI2 memory can be used as follows:

1) To memorize data.
2) To have different BASIC programs that the CPU can use alternatively

We give an example of two separate programs that run alternatively, one from page 0 and one from page 1: Assuming we are in page 0 (Mapper in transparent state) we write the program to be transferred to page 1:

```
NEW 0D000H
10 PRINT "PAGE ONE"
20 CALL "WRMA", 0, 0 ! To pass to page 0
30 GOTO10            ! To execute again when
                       coming back from
                       page 1
-------------------------------------
CALL "WRMA", 0D000H, 3000H, 1, 0D000H
                     ! To copy all of the
                       BASIC work area to
                       page 1
-------------------------------------
```

Now we write the program for page 0:

```
NEW 0D000H                       ! To clear page
                                   0
10 PRINT "PAGE ZERO"
20 CALL "MAPPIN", 1, 0DH, 0FH    ! To use BASIC
                                   with page 1
                                   work-area
30 GOTO10                        ! To continue
                                   execution
-------------------------------------
RUN
```

The output obtained will be:

```
PAGE ZERO
PAGE ONE
PAGE ZERO
...
```

4. Conclusions

We have tried to build a system which is easy to use and reasonably general purpose, to satisfy the needs of a Physics Laboratory. Much effort has been spent on making CANDI2 "user friendly". We hope to have saved some time and effort to CAMAC users.

References

[1] O. Ciaffoni, M. Coli, M.L. Ferrer, S. Li Causi and A. Villalba, Frascati Report LNF – 80/27 (1980).
[2] O. Ciaffoni, M. Coli, M.L. Ferrer and L. Trasatti, Frascati Report LNF – 81/24 (1981).
[3] L. Trasatti, Frascati Report LNF – 81/31 (1981).
[4] O. Ciaffoni, M.L. Ferrer and L. Trasatti, Frascati Report LNF – 81/36 (1981).
[5] O. Ciaffoni, M. Coli, M.L. Ferrer and L. Trasatti, Data acquisition in high energy physics, 84th Corso, Soc. Italiana di Fisica, Bologna, Italy (1983).
[6] O. Ciaffoni, M. Coli, M.L. Ferrer and L. Trasatti, Frascati Report LNF – 81/65 (1981).
[7] O. Ciaffoni et al., EURATOM-ENEA Report 81.55 (1981).
[8] L. Trasatti, Frascati Report LNF – 82/15 (1982).
[9] M.L. Ferrer, Frascati Report LNF – 82/28 (NT) (1982).
[10] R. Bertoldi, O. Ciaffoni, M. Coli and L. Trasatti, Frascati Report LNF – 82/48 (NT) (1982).
[11] R. Bertoldi, O. Ciaffoni, M. Coli, M.L. Ferrer, A. Martini and L. Trasatti, Frascati Report LNF – 82/83 (NT) (1982).
[12] M. Pistoni and L. Trasatti, Frascati Report LNF – 82/85 (NT) (1982).
[13] O. Ciaffoni, M. Coli, M.L. Ferrer and L. Trasatti, Frascati Internal Report LNF – 83/67 (R) (1983).
[14] G.M. Bilei, O. Ciaffoni, M.L. Ferrer and L. Trasatti, Frascati Internal Report LNF – 84/3 (R) (1983).
[15] ESONE/NIM Committees, Subroutines for CAMAC (June 1978).