ISTITUTO NAZIONALE DI FISICA NUCLEARE
Laboratori Nazionali di Frascati

LNF-80/27(R)
14 Maggio 1980

O. Ciaffoni, M. Coli, M. L. Ferrer, S. Li Causi and A. Villalba:

A CAMAC SYSTEM CONTROLLER USING THE TEXAS TMS 9900 MICROPROCESSOR, AS STANDING-ALONE AND PDP 11 CONNECTED UNIT.

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - Laboratori Nazionali di Frascati
Servizio Documentazione

LNF-80/27(R)
14 Maggio 1980

O. Ciaffoni, M. Coli[x], M. L. Ferrer, S. Li Causi[o] and A. Villalba[x]:

A CAMAC SYSTEM CONTROLLER USING THE TEXAS TMS 9900 MICROPROCESSOR, AS STANDING-ALONE AND PDP 11 CONNECTED UNIT.

SUMMARY.

The CCTM 990 controller was designed and developped to handle, in a completely autonomous way, data acquisition and control of process using a standard CAMAC system. Therefore it can take the place of the minicomputer in all the applications, where a lower speed or a lower mass memory capabilities does not forbid them. Connected with a PDP 11, this controller could strongly reduce the engagement requested by the minicomputer to CAMAC instrumentation.

The CCTM 990 standalone system has been completely tested with satisfactory results, although implementations have been designed expecially in order to reduce the CAMAC cycles timing.

Tests on CCTM 990 controller connected with a PDP 11/34 are in progress and they will be presented in a following work.

1. - INTRODUCTION.

Microprocessors have recently proved to be an unexpensive and powerful system to control data acquisition in sophisticated experimental apparatus, expecially in connection with a minicomputer.

In this work an intelligent crate CAMAC controller is presented which is able to control, in an autonomous way, one or more (≤ 4) CAMAC crates and some peripheral devices such as memory and terminals. Moreover, the system can be connected to a PDP 11 minicomputer in order to increase software facilities and mass memory management.

This controller, which will be referred as CCTM 990 (CAMAC Controller TEXAS Microprocessor) is built utilizing a development system based on the microprocessor Texas TMS 9900[1] (μP), and it contains the following parts (see also Fig. 1):

- Board TM 990/100 containing the 16 bits CPU TMS 9900 (see Section 2);
- Board TM 990/302[2] providing hardware and software interfaces for two audiocassettes;

(x) - CNEN, Frascati
(o) - Actually associated to the INFN.

- Board TM 990/206[3], RAM and ROM expansion memory;
- Board BUD 9901 (Board Unibus Dataway). This is a special unit realized by the authors in order to interface the TM 990/100 both with CAMAC and PDP family's unibus;
- TTY, CRT and audiocassettes as pheripheral devices.

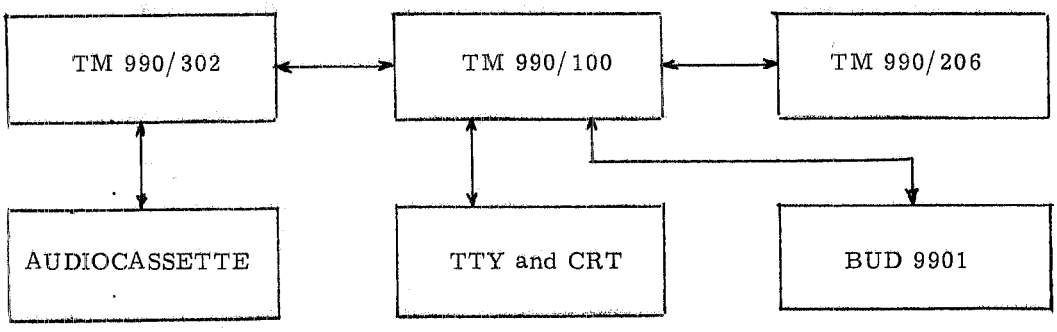

FIG. 1 - CCTM 990 system block diagram.

Although this controller can work in standalone mode for data acquisition from CAMAC, expecially if used with a conveniently large memory, it can be profitably coupled with a PDP 11 minicomputer resulting in a powerful device, with the following characteristics:

a) The data acquisition from CAMAC is completely controlled by the microprocessor which can eventually develop also a preliminary elaboration. No intervention is required from the minicomputer which remains available for doing parallel jobs.

b) The PDP mass storage largerly extends the memory of the CCTM 990. Consequently only a reduced hardware configuration for the controller is required.

c) Software for TMS 9900 can be produced by the PDP itself using cross-compilers writen in a high level language, such as BCPL. This compilers translate the mnemonic into machine Texas language, which can be sent "on line" to the microprocessor.

In Sections 2 and 3 some informations about Texas systems used in the CCTM 990 controller are given in order to facilitate reading of Sections 4 and 5 where hardware for BUD 9901 and software to control CAMAC respectively are presented.

2. - GENERAL FEATURES OF THE BOARD TM 990/100.

The board TM 990/100[4] contains the following parts in addition to the CPU:

a) A 16-bit bus addresses (A00 - A15);
b) A 16-bit bus data (D00 - D15);
c) A 4 Kbytes ROM memory (0000_{16} - $0FFE_{16}$) containing the system monitor TIBUG in the upperst 2 Kbytes;
d) A TMS 9902[5] asynchronous communications controller (ACC) to interface with μP a serial peripheral such as TTY;
e) A TMS 9901[6] programmable system interface (PSI) providing interrupt and I/O parallel ports (see Section 3):
f) Signals encoder to select an ACC and 6 PSI off board which will be used as 16-bit I/O ports.

3. - TMS 9901 (PSI) DESCRIPTION.

This unit (see block diagram in Fig. 2) provides:

a) I/O port with a maximum of 16 bits for output and 22 for input;

b) Interrupt processing with 15 different priority levels ;

c) Real-time clock.

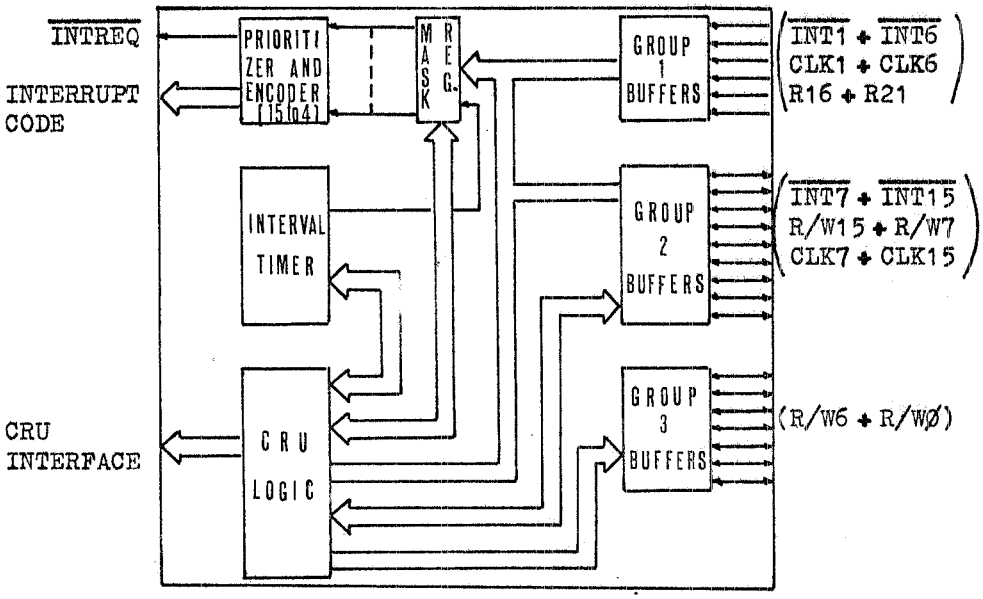

FIG. 2 - TMS 9901 block diagram.

The TMS 9901 access 32 addressables bits. Only 22 of which are users available. These bits are shared in 3 groups of 6, 9 and 7 bits respectively. Each group can accomplishes different functions generally unoverlapped, which are selected by a control bit (CB) corresponding to the bit Ø (see Table I).

TABLE I - Control bit values for each group and operation mode.

Mode \ Group	1 6 bits	2 9 bits	3 7 bits
Interrupt $\overline{INT}$	CB = Ø	CB = Ø	not allowed
Input R	CB = Ø	CB = Ø	CB = X
Output W	not allowed	CB = Ø	CB = X
Clock CLK	CB = 1	CB = 1	not allowed

The CPU uses a Communication Register Unit (CRU) to communicate with the PSI interfaces. The CRU instructions refer to the transfert of many (from one to sixteen) consecutive bits. Bit Ø of any PSI is equivalent to a memory position of the CRU logic (see Table II) and it can be accessed by the CPU loading the relative address - base address - in the register 12 of the work space area. Later on, other bits to sixteen, for any PSI, can be referenced giving in the CRU instruction the displacement respect to this address without any modification of the base address content.

TABLE II - CRU address and signals relatives to each PSI.

Hexadecimal address	Virtual bit	PSI	Active signal
0000 - 003E	0 - 31	PSI 1 off board	$\overline{\text{SEL 1}}$
0040 - 007E	0 - 31	PSI 2 off board	$\overline{\text{SEL 2}}$
00C0 - 00FE	0 - 31	PSI 3 off board	$\overline{\text{SEL 3}}$
0100 - 013E	0 - 31	PSI on board	$\overline{\text{9901 SEL}}$
0140 - 017E	0 - 31	PSI 4 off board	$\overline{\text{SEL 4}}$
0180 - 01BE	0 - 31	PSI 5 off board	$\overline{\text{SEL 5}}$

3.1. - I/O Port.

Input/output functions relatives to one or more bits are programmed using the following CRU instructions :

LDCR R_X, < n. bits > for output ,

STCR R_X, < n. bits > for input ,

where R_X is the workspace register (different from 12) containing (o receiving) data, and < n. bits > are the number of bits (≤ 16) to be read or writed, starting from that contained in register 12.

CRU instructions on single bit are the following :

SBO < n > }
SBZ < n > } for output ,

TB < n > for input ,

when the bit, obtained adding < n > ($-127 \leq <n> \leq 128$) to the address on register 12 as before, must be set to "one", "zero" or must be tested.

Communication between PSI and CPU is only possible in a serial way through the following lines, at a maximum rate of 3.3 Mbits/sec :

CRUOUT for output ,

CRUIN for input ,

CRUCLK for timing .

3.2.- Interrupt processing.

Every time an interrupt from an external device is received by the PSI, it performs the following functions:

a) The interrupt code lines $I_{C0} - I_{C3}$ are loaded with the INTR level ;

b) The interrupt request signal is put on the $\overline{\text{INTREQ}}$ line.

In the CCTM configuration only the PSI on board issues interrupt request to the CPU.

To process an interrupt from this port, the following software operations must be executed:

a) The control bit must be cleared to set the port in interrupt mode ;

b) The bit representing the relative interrupt level must be set ;

c) The CPU status register must be charged with a mask (0 - 15) indicating the maximum interrupt level to be serviced ;

d) The interrupt vectors for each level to be processed must be charged with a branch instruction to the interrupt service routine.

Every time an interrupt occurs, at least one instruction of the service routine is executed before a higher priority interrupt could be processed. During execution of the service routine the mask is loaded with one less than the level value of the interrupt being serviced, so that further interruptions at the same priority level be not allowed. After return from the service routine the interrupt level already serviced remains disabled until the relative bit on PSI is set again.

4. - BUD 9901. HARDWARE DESCRIPTION.

The block diagram of this new interface between microprocessor Texas 9900 and both CAMAC and PDP systems is given in Fig. 3, where the most important parts are indicated:

a) CNAF interface (block n. 3)
b) R/W interface (block n. 4)
c) LAM interface (block n. 5)
d) CAMAC timing and control (block n. 6)
e) PDP interface (block n. 2)

} CAMAC interface

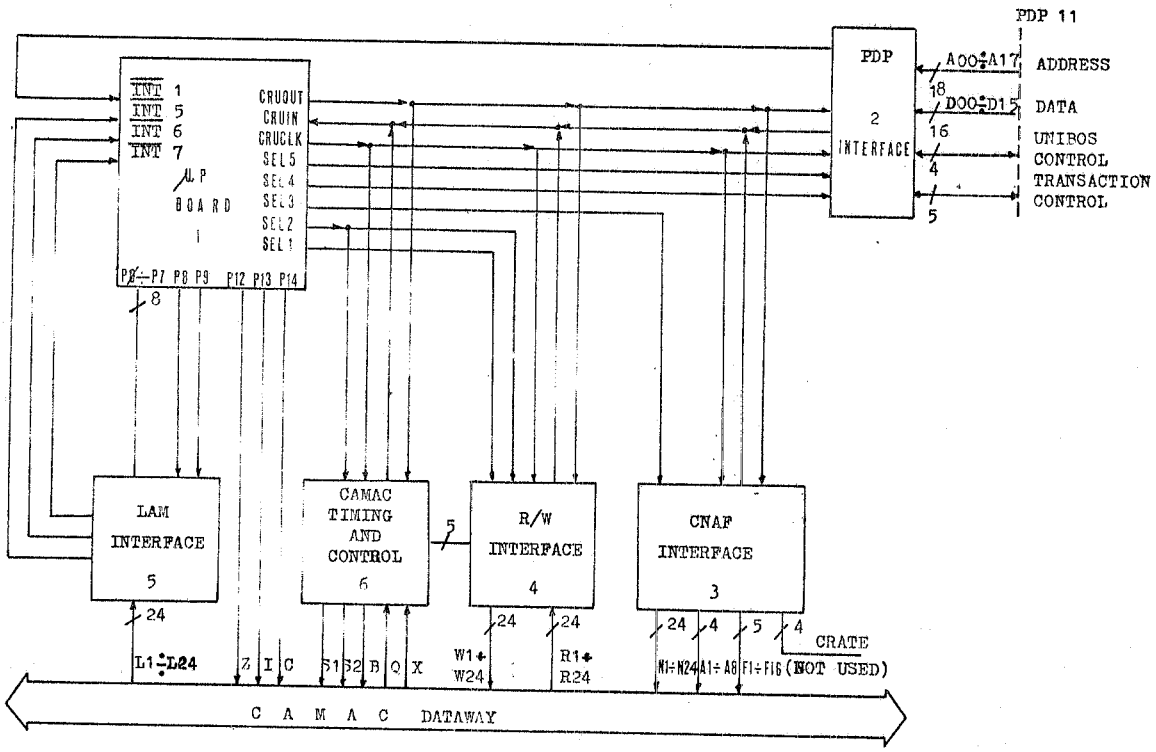

FIG. 3 - BUD 9901 block diagram.

Fig. 3 includes also the μP board (block n. 1) containing the CPU of the CCTM 990. All the blocks communicate inside the board only with the CPU and outside there with the PDP or CAMAC. Communication between CPU and CAMAC or PDP happens through the CRU lines. Other five lines SEL 1 - SEL 5 are used to enable the singles off board PSI providing the 16 bits I/O ports. Communication between CPU and CAMAC uses also 10 lines (P0 - P9) of the PSI on board, 2 lines of which (P8 - P9) are used for control. Pins P12, P13 and P14 on μP board supplie the control data way CAMAC signals Z, I and C respectively.

The interrupt levels used are :

a) Interrupt 1 : coming from the PDP interface when the PDP address the CCTM 990 control status register (CSR) ;

b) Interrupts 5, 6 and 7 coming from the LAM interface when a CAMAC module issues a LAM request[7, 8].

4.1. - CNAF interface.

The CNAF interface, show in Fig. 4, is used to select on the CAMAC system[7] :

a) a station number (24 lines : N1 - N24) ;
b) a subaddress (4 lines : A1, A2, A4, A8) ;
c) a function (5 lines : F1, F2, F4, F8, F16) ;
d) a crate number (4 lines : Cϕ - C3).

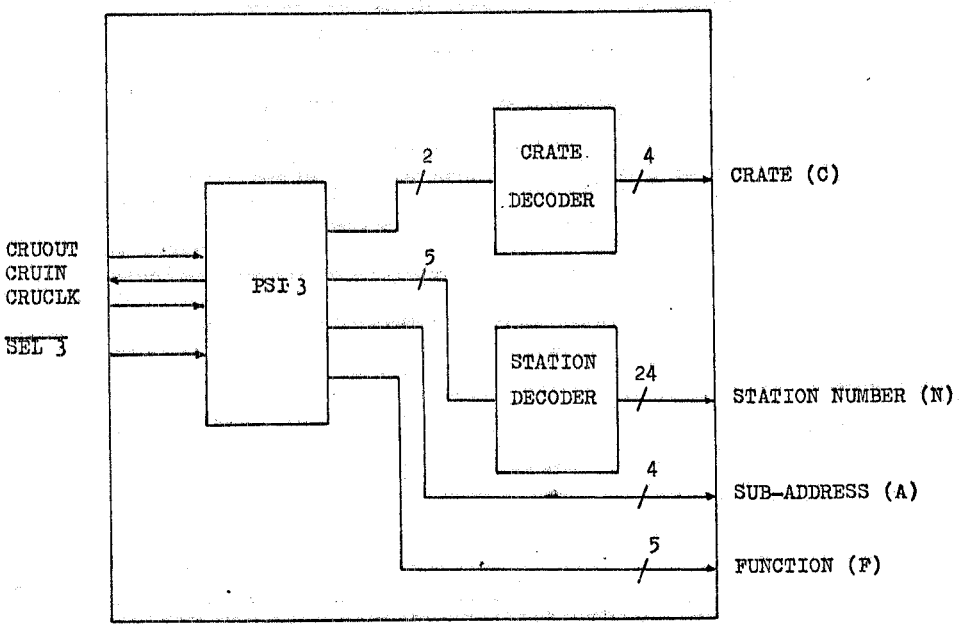

FIG. 4 - CNAF interface block diagram.

This interface, containing essentially some decoders loading the C and N lines, uses a 16 bit I/O port enabled by the $\overline{SEL3}$ signal to send data to the CAMAC system.

4.2. - R/W interface.

This interface, which is represented in Fig. 5, contains two 16-bit PSI port enabled by $\overline{SEL1}$ and $\overline{SEL2}$ signals, whose bits provide an internal 24 bits data bus (16+8). A read operation from CAMAC is accomplished within two phases : The 24 bits of the R lines of CAMAC are memorized on a latch during the first phase, while the data transfer to the CPU is actuated during the second phase.

A write operation into CAMAC is accomplished also in two phases : during the first one, 24 bits data are presented on PSI ports and during the second phase data are loaded into the W CAMAC lines[7].

All signals able to produce input or output within the latches are generated by the "R/W decoder", controlling 2 bits (P8, P9) of the PSI 2.

Residual bits (P11 - P15) of this part have been used within CAMAC timing and control block interface to be described later in Section 4.4.

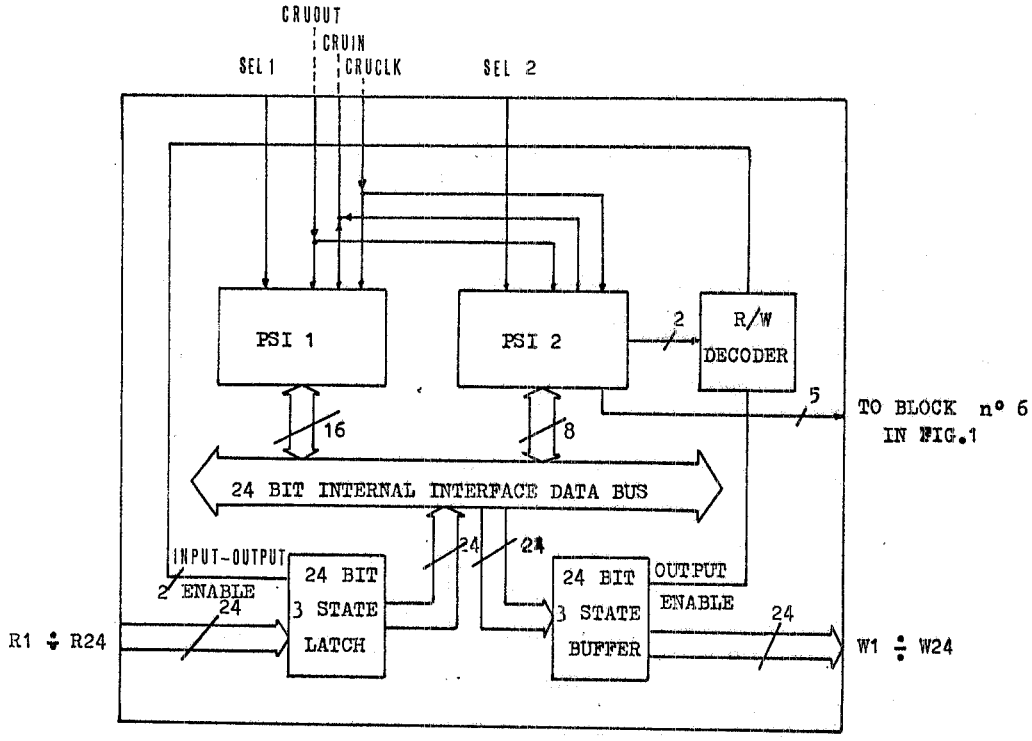

FIG. 5 - R/W interface block diagram.

4. 3. - LAM interface.

This interface, whose block diagram is represented in Fig. 6, issues an interrupt to the μP when a CAMAC module generates a LAM request. The LAM lines are grouped in three different

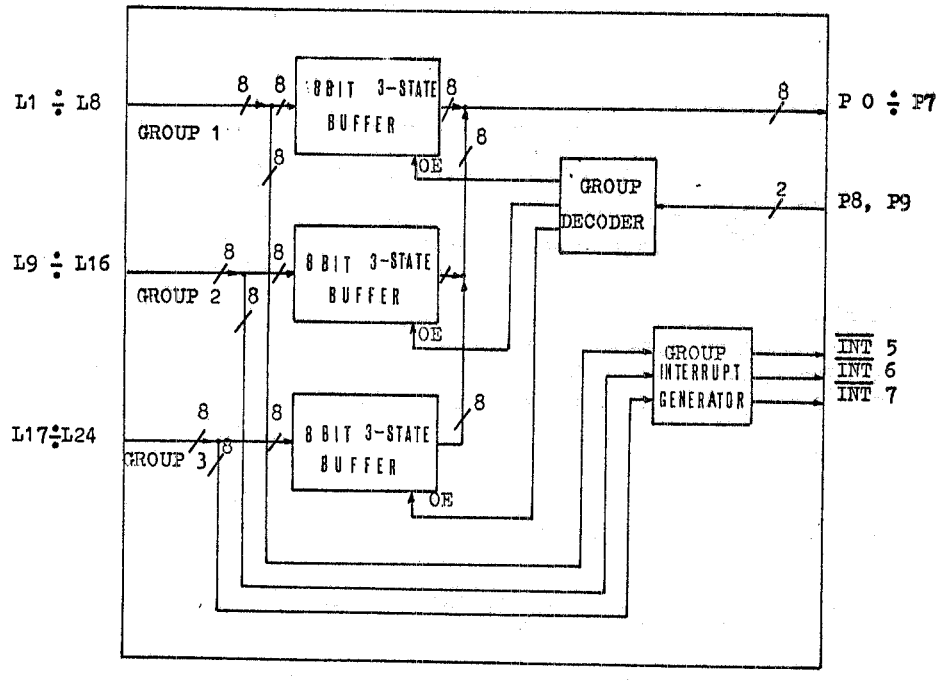

FIG. 6 - LAM interface block diagram.

interrupt levels (see Table III).

The group number is identified according with the interrupt priority level reaching the CPU. After that, the relative service routine must be able to command the "Group decoder" to give to the PSI on board the word rappresentation of the 8 LAM lines status, in order to achieve the station number sending the LAM request.

TABLE III - LAM groups, relative to the station numbers and priority of the interrupt to μP.

Group	Station numbers	Priority
1	$1 \leq N \leq 8$	5
2	$9 \leq N \leq 16$	6
3	$17 \leq N \leq 24$	7

4.4. - CAMAC timing and control interface.

This block reads the Q and X CAMAC lines and generates on the CAMAC dataway:

a) the temporization signals S1 and S2;
b) the bus busy B signal.

The S1, S2 and B signals must be generated by the program according to the timing diagram shown in Fig. 7 (see also Fig. 5).

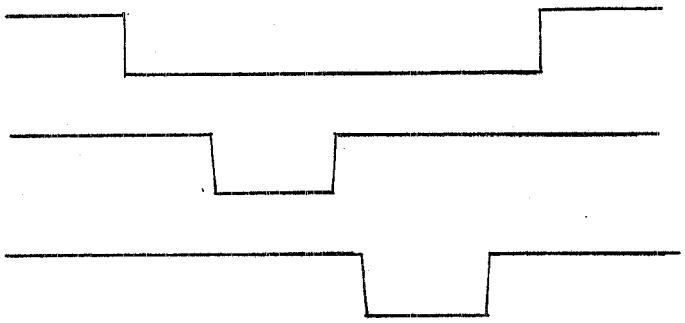

FIG. 7 - Timing diagram.

4.5. - PDP interface.

In order to allow:

a) Typical master-slave transactions between PDP and μP ,

b) Interrupt to the PDP from the microprocessor,

the following lines of the PDP 11 unibus are linked to this interface[9, 10] (see Fig. 8):

a) Unibus control BR4, BG4, SACK, BBSY ;

b) Bus address : A00 - A15 ;

c) Bus data : D00 - D15 ;

d) Transaction control: CØ, C1, MSYN, SSYN, INTR.

The PDP interface contains an "address decoder block" to recognize when the PDP unibus address the CSR (actually 766000_8). In this case an interrupt to the microprocessor is issued at the highest priority level (Level 1).

Data transfer between PDP and CPU (and viceversa) uses the 16-bits PSI port enabled by the $\overline{\text{SEL4}}$ signal, while the relative control functions are accomplished by a "control block" through the PSI port enabled by the $\overline{\text{SEL5}}$ signal.

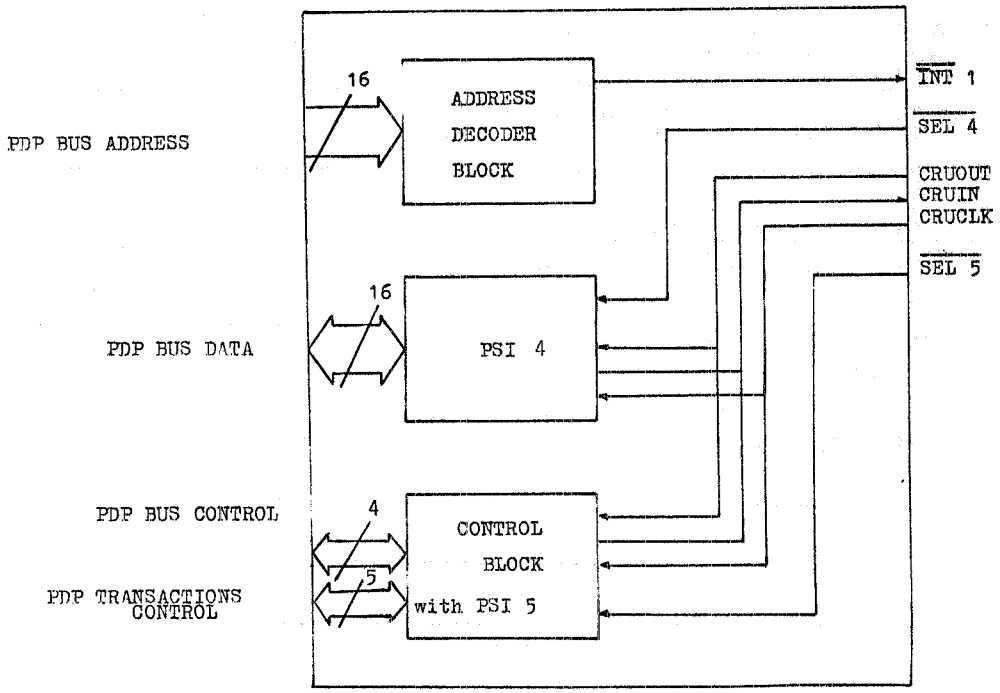

FIG. 8 - PDP interface block diagram.

5. - BUD 9901. Software description.

During the test, a software package to control the hardware performance was writen, which executes CAMAC cycles, data transfers and LAM request acknowledgements, allowing to evaluate the actual efficiency of the CCTM 990 CAMAC controller.

The translation into Texas machine language of this programs, writen in mnemonic Texas language[11, 12], was obtained using a compiler which runs on the CDC 6600 machine of CINECA (Centro di Calcolo Interuniversitario dell'Italia Nord-Orientale). This compiler, writen in BCPL language, analises the sintax errors found in the text and generates an output file with the following informations (see listings):

col. 1 instruction number;
col. 2 absolute hexadecimal address in RAM memory;
col. 3 machine Texas language. One or two words for instruction;
col. 4 mnemonic Texas language;
col. 5 comment.

Using this software package it is possible to produce files containing only the machine language after new relocation. During test a PDP 11/34 both connected with the CDC machine and the CAMAC controller, is used. This configuration will enable in the final configuration, to sent data on-line from PDP to the microspocessor. For this purpose in the next future the compiler will be runned directly on the PDP.

In the following, same examples of the CAMAC routines assembled by a MAIN program, are given. For each routine the minimum execution time required is shown.

```
  MFA    SCOPE 3.4.4 UBOL66 RELEASE420 R.B 10.22
14.57.38.MFA. B4     SCOPE 3.4.4 UBOL720RELEASE420 28/01/80
14.56.56.M7CMCC7   FROM MFA/B4
14.56.56.IP   00000896 WORDS - FILE INPUT   , DC 00
14.56.56.M7CMC,STMFB,T20,DMS00.
14.57.01.ACCOUNT DONE.
14.57.04.COPY,INPUT,INPUTT.
14.57.06.REWIND,INPUTT.
14.57.06.ATTACH,TMSASM,TMSASM,CY=1,ID=M7CXX.
14.57.07.ATTACH,ZZZOP,TMSOP,ID=M7CXX.
14.57.07.PF CYCLE NO. = 001
14.57.07.ATTACH,ZZZPRLG,TMSPRLG,ID=M7CXX.
14.57.07.PF CYCLE NO. = 001
14.57.07.TMSASM,I=INPUTT,B=OBJ.
14.57.08.//LOADER  420        .125 CP       .643 RT
14.57.08.//LOADER  041130/030000-044000 CM    2 TM
14.57.16.OP   00001664 WORDS - FILE OUTPUT , DC 40
14.57.16.MS     3584  WORDS  (    10752 MAX USED)
14.57.16.CPA       3.116 SEC.       654.404 ADJ.
14.57.16.IO         .518 SEC.        25.409 ADJ.
14.57.16.CM       70.852 KWS.       148.790 ADJ.
14.57.16.SS                         828.603
14.57.16.PP        9.289 SEC.      DATE 29/04/80
14.57.16.EJ   END OF JOB, B4
```

col.1 col.2 col.3 col.4 col.5 col.6

```
PAGE 1            29/04/80   14.57.12.              TMS9900 CROSS-ASSEMBLER (1.1)

     1                    *      TMS 9900 ROUTINES FOR CAMAC. LAM REQUEST SERVICE,READ AND WRITE OPERAT.
     2                    *
     3          FF28      R4G1   EQU >FF28        REGISTER 4. LAM GROUP 1
     4          FF04      R4G2   EQU >FF04                              2
     5          FEE0      R4G3   EQU >FEE0                              3
     6                    *
     7                    *                      INTERRUPT VECTORS FOR 5,6 AND 7 LEVELS,  CORRESPONDING
     8                    *                      TO 1,2 AND 3 GROUP LAM RESPECTIVELY.
     9                    *
    10                    * 5  FF20 .........    WORKSPACE
    11                    *    FF40 .........    BRANCH INSTRUCTION
    12                    *
    13                    * 6  FEFC
    14                    *    FF1C
    15                    *
    16                    * 7  FED8
    17                    *    FEF8
    18                    *
    19  FF40              START AORG >FF40
    20  FF40  0460              B @GRUP1
        FF42  2062
    21                    *
    22  FF1C                    AORG >FF1C
    23  FF1C  0460              B @GRUP2
        FF1E  2068
    24                    *
    25  FEF8                    AORG >FEF8
    26  FEF8  0460              B @GRUP3
        FEFA  206E
    27                    *
    28                    *      INITIAL DATA IN WORKSPACE REGISTERS FOR EACH GROUP
    29                    *
    30  FF20                    AORG >FF20      REGISTER 0....7 GROUP 1
    31  FF20  0000              DATA 0,0,0,0,0,0,>1E08,>1E09
        FF22  0000
        FF24  0000
        FF26  0000
        FF28  0000
        FF2A  0000
        FF2C  1E08
        FF2E  1E09
    32  FEFC                    AORG >FEFC                        GROUP 2
    33  FEFC  0000              DATA 0,0,8,0,0,0,>1D08,>1E09
        FEFE  0000
        FF00  0008
        FF02  0000
        FF04  0000
        FF06  0000
        FF08  1D08
        FF0A  1E09
```

```
PAGE 2              29/04/80   14:57:12.                TMS9900 CROSS-ASSEMBLER (1.1)
                               AORG >FED8                        GROUP 3
  34  FED8
  35  FED8  0000               DATA 0,0,16,0,0,0,>1E08,>1D09
      FEDA  0000
      FEDC  0010
      FEDE  0000
      FEE0  0000
      FEE2  0000
      FEE4  1E08
      FEE6  1D09

  36                     *     CAMAC CYCLE, BUSY SIGNAL AND S2 STROBE
  37                     *     MINIMUM EXECUTION TIME- 48 MICROSECONDS
  38                     *
  39  2000               *     AORG >2000
  40  2000  020C         ZI    LI 12,>120        9901/SEL0   P0 BIT 16
      2002  0120
  41  2004  1E0C               SBZ 12            BIT Z       P12 SEL0
  42  2006  1E0E               SBZ 14            BIT C       P14 SEL0
  43                     *
  44  2008  020C               LI 12,>60         9901/SEL2   P0 BIT 16
      200A  0060
  45  200C  1E0D               SBZ 13            BIT BUSY    P13 SEL2
  46  200E  1E0C               SBZ 12            BIT S2      P12 SEL2
  47  2010  1D0C               SBO 12            BIT S2 CLEAR
  48  2012  1D0D               SBO 13            BIT BUSY CLEAR
  49                     *
  50  2014  020C               LI 12,>120        9901/SEL0   P0 BIT 16
      2016  0120
  51  2018  1D0C               SBO 12            BIT Z CLEAR
  52  201A  1D0E               SBO 14            BIT C CLEAR
  53                     *
  54  201C  045B               RT
  55                     *
  56                     *     CAMAC CYCLE SIGNALS- BUSY,S1 AND S2.
  57                     *     MINIMUM EXECUTION TIME-  32 MICROSECONDS
  58                     *
  59  201E  020C         CYCLE LI 12,>60         9901/SEL2   P0 BIT
      2020  0060
  60  2022  1E0D               SBZ 13            BUSY SET
  61  2024  1E0B               SBZ 11            S1 SET
  62  2026  1D0B               SBO 11            S1 CLEAR
  63  2028  1E0C               SBZ 12            S2 SET
  64  202A  1D0C               SBO 12            S2 CLEAR
  65  202C  1D0D               SBO 13            BUSY CLEAR
  66  202E  045B               RT
  67                     *
  68                     *     ROUTINE TO SET CNAF INFORMATION IN CAMAC DATAWAY
  69                     *     DATA IN INPUT- STATION NUMBER N IN REGISTER 2
  70                     *                    SUBADDRESS     A             3
  71                     *                    FUNCTION CODE  F             4
  72                     *     DATA IN OUTPUT- CNAF              IN REGISTER 7
  73                     *     ERROR CONDITION FOR N>24
  74                     *     REGISTER USED-6
  75                     *     MINIMUM EXECUTION TIME-  71 MICROSECONDS
  76                     *
  77  2030  0282         CNAF  CI 2,24
      2032  0018
  78  2034  1209               JLE OK
  79  2036  2FA0               XOP @MESS,14
      2038  203C
  80  203A  045B               RT
  81  203C    45         MESS  TEXT 'ERROR IN N'
      203D    52
      203E    52
      203F    4F
      2040    52
      2041    20
      2042    49
      2043    4E

      2044    20
      2045    4E
  82  2046  0700               DATA >0700
  83                     *
  84  2048  020C         OK    LI 12,>E0         9901/SEL3   P0 BIT 16
      204A  00E0
  85  204C  C183               MOV 3,6
  86  204E  C1C4               MOV 4,7
  87  2050  0A56               SLA 6,5
  88  2052  A1C6               A   6,7
  89  2054  C182               MOV 2,6
  90  2056  0A96               SLA 6,9
```

```
PAGE 3              29/04/80   14.57.13.              TMS9900 CROSS-ASSEMBLER (1.1)
   91  2058  A1C6          A   6,7          10000000*N+10000*A+F
   92  205A  0547          INV 7
   93  205C  3007          LDCR 7,0         CNAF INVERTED IN REGISTER 7
   94  205E  0460          B @CYCLE         BRANCH TO CYCLE ROUTINE
       2060  201E
   95                    *
   96                    *     SERVICE ROUTINES FOR LAM REQUESTS
   97                    *
   98  2062  0584      GRUP1 INC 4
   99  2064  0460          B @NSTAT
       2066  2074
  100                    *
  101  2068  0584      GRUP2 INC 4
  102  206A  0460          B @NSTAT
       206C  2074
  103                    *
  104  206E  0584      GRUP3 INC 4
  105  2070  0460          B @NSTAT
       2072  2074
  106                    *
  107                    *
  108  2074  020C      NSTAT LI 12,>120       9901/SELO   P0 BIT 16
       2076  0120
  109  2078  0486          X 6              EXECUTE INSTRUCTIONS TO LOAD INTO P8 AND P9 BITS
  110  207A  0487          X 7              THE VALUES CORRESPONDING TO THE GROUP REQUESTING INT.
  111  207C  3603          STCR 3,8         READ THE LAM LINES STATUS INTO REGISTER 3
  112  207E  0380          RTWP
  113                    *
  114                    *
  115                    *     MAIN PROGRAM TO ENABLE AND WAIT INTERRUPT FOR LAM REQUEST. AFTER INT.
  116                    *     THE STATION NUMBER SENDING LAM IS FOUND AND A CLEAR LAM FUNCTION IS
  117                    *     SENT AT THE SAME STATION. ATTENDING INTERRUPT, A BIG LOOP IN MAIN PRO-
  118                    *     GRAM IS EXECUTED. IF MORE THAN A LAM REQUEST REACH THE CPU,
  119                    *     SERVICES WILL GIVEN IN THE ORDER 1-2-3 GROUPS.
  120                    *
  121  2080  02E0      MAIN  LWPI >FD00
       2082  FD00
  122  2084  06A0          BL @ZI           INITIALIZATION CYCLE
       2086  2000
  123  2088  020C          LI 12,>100       9901/SELO    (CB)
       208A  0100
  124  208C  1E00          SBZ 0            INTERRUPT MODE
  125  208E  1D05          SBO 5            ENABLE INTERRUPT 5
  126  2090  1D06          SBO 6                             6
  127  2092  1D07          SBO 7                             7
  128  2094  0300          LIMI 7
       2096  0007
  129                    *
  130  2098  04C3          CLR 3            SUBADDRESS  0
  131  209A  0204          LI 4,10          F10 CLEAR LAM REQUEST
       209C  000A
  132                    *
  133  209E  C020      LAM1  MOV @R4G1,0     GROUP 1 INTERRUPT WAS RECEIVED ?
       20A0  FF28
  134  20A2  1305          JEQ LAM2         NO-BRANCH
  135  20A4  04E0          CLR @R4G1
       20A6  FF28
  136  20A8  0201          LI 1,>1D05       REGISTER 1 CONTAINS INSTRUCTION TO ENABLE GROUP 1 INT.
       20AA  1D05
  137  20AC  100F          JMP LAM1
  138                    *
  139  20AE  C020      LAM2  MOV @R4G2,0
       20B0  FF04
  140  20B2  1305          JEQ LAM3
  141  20B4  04E0          CLR @R4G2
       20B6  FF04
  142  20B8  0201          LI 1,>1D06
       20BA  1D06
  143  20BC  1007          JMP LAM1
  144                    *
  145  20BE  C020      LAM3  MOV @R4G3,0
       20C0  FEE0
  146  20C2  13ED          JEQ LAM1
  147  20C4  04E0          CLR @R4G3
       20C6  FEE0
  148  20C8  0201          LI 1,>1D07
       20CA  1D07
  149                    *
  150  20CC  C171      LAMI  MOV *1+,5       R1 REGISTER FROM ACTUAL INTERRUPT WORKSPACE
  151  20CE  C0B1          MOV *1+,2        INITIAL STATION NUMBER FOR THIS GROUP (0,8,16)
  152  20D0  D231          MOVB *1+,8       LINES STATUS  P0....P7
```

```
PAGE 4              29/04/80   14.57.14.              TMS9900 CROSS-ASSEMBLER (1.1)
  153                       *
  154  20D2  0209           LI 9,>100        LOOP TO KNOW THE STATION NUMBERS (1 OR MORE) SENDING
       20D4  0100
  155                       *                LAM REQUEST (BIT 9) EQUAL ZERO)
  156  20D6  0582    LOOP   INC 2
  157  20D8  2609           CZC 9,8
  158  20DA  1604           JNE NOCNA        BRANCH IF BIT=1
  159                       *
  160  20DC  C28B           MOV 11,10        AFTER SAVING REGISTER 11, USED FOR BRANCH, CALL CNAF
  161  20DE  06A0           BL @CNAF
       20E0  2030
  162  20E2  C2CA           MOV 10,11
  163                       *
  164  20E4  0A19    NOCNA  SLA 9,1
  165  20E6  16F7           JNE LOOP
  166                       *
  167  20E8  020C    END    LI 12,>100       9901/SEL0   (CB)
       20EA  0100
  168  20EC  0485           X 5              TO ENABLE OTHER INTERRUPT FROM THE SAME GROUP
  169  20EE  0460           B @LAM1          WAIT AGAIN
       20F0  209E
  170                       *
  171                       *     ROUTINE TO BE CALLED BY - BL @READ - INSTRUCTION
  172                       *     READ FROM CAMAC IN REGISTERS 8 AND 9 (16 AND 8 BITS)
  173                       *     R2 AND R3 MUST CONTAIN N AND A VALUES
  174                       *     MINIMUM EXECUTION TIME -  105 MICROSECONDS
  175                       *
  176                       *
  177  20F2  020C    READ   LI 12,>60        9901/SEL2   P0 BIT 16
       20F4  0060
  178  20F6  1E08           SBZ 8            FROM CAMAC INTO LATCH
  179  20F8  1D09           SBO 9
  180  20FA  04C4           CLR 4            F0 READ FUNCTION
  181  20FC  C28B           MOV 11,10
  182  20FE  06A0           BL @CNAF
       2100  2030
  183  2102  C2CA           MOV 10,11
  184  2104  020C           LI 12,>60        9901/SEL2   P0 BIT 16
       2106  0060
  185  2108  1D08           SBO 8            FROM LATCH INTO CPU
  186  210A  1D09           SBO 9
  187  210C  3648           STCR 8,9
  188  210E  0548           INV 8
  189  2110  020C           LI 12,>20        9901/SEL1   P0 BIT 16
       2112  0020
  190  2114  3409           STCR 9,0
  191  2116  0549           INV 9
  192  2118  045B           RT
  193                       *
  194                       *     WRITE INTO CAMAC
  195                       *     R8 AND R9 MUST CONTAIN DATA TO BE WRITEN
  196                       *     MINIMUM EXECUTION TIME -  72 MICROSECONDS
  197                       *
  198                       *
  199  211A  0201    WRITE  LI 1,>300        P8 & P9 MUST BE CLEAR
       211C  0300
  200  211E  0549           INV 9
  201  2120  0548           INV 8
  202  2122  020C           LI 12,>20        9901/SEL1   P0 BIT 16
       2124  0020
  203  2126  3009           LDCR 9,0
  204  2128  020C           LI 12,>60        9901/SEL2   P0 BIT 16
       212A  0060
  205  212C  5201           SZCB 1,8         CLEAR P8 AND P9 TO WRITE INTO CAMAC
  206  212E  32C8           LDCR 8,11
  207  2130  0204           LI 4,16          F16
       2132  0010
  208  2134  C28B           MOV 11,10
  209  2136  06A0           BL @CNAF
       2138  2030
  210  213A  C2CA           MOV 10,11
  211  213C  045B           RT
  212                       *
  213  213E  FF40           END START

>>>>> 0 ERROR  DETECTED IN THE SOURCE FILE <<<<<<<
```

ACKNOWLEDGEMENTS.

One of the authors (M. L. Ferrer) thanks Mr. J. Montuelle (from Mr. I. Willer's group) of DD CERN Division wich produces compiler and kindly gives all the assistance to user it.

REFERENCES.

(1) - Family systems design and data book (Texas Instruments). TMS 9900 Microprocessor data manual.
(2) - TM 990/302 software development board (Texas Instruments).
(3) - TM 990/206 expansion memory board (Texas Instruments).
(4) - TM 990/100 microcomputer user's guide (Texas Instruments).
(5) - TMS 9902 asynchronous communications controller manual (Texas Instruments).
(6) - TMS 9901 programmable system interface data manual (Texas Instruments).
(7) - EUR 4100 CAMAC - A modulare instrumentation system for data handling.
(8) - EUR 4600 CAMAC - Organization of multicrate systems.
(9) - PDP 11 peripherals handbook, Chapter 5. DEC library.
(10) - PDP 11 family. DEC library.
(11) - How to undump the CDC BCPL distribution tape, I. Willers, CERN DD Division, User's Manual.
(12) - The BCPL mini manual, S. Mellor and I. Willers, CERN DD Division (1978).