AN MC68000 STAND-ALONE CAMAC MICROPROCESSOR SYSTEM

A. Rijllart

CERN, Geneva, Switzerland

ABSTRACT

An MC68000 stand-alone CAMAC microprocessor system has been designed for intelligent interfacing to CAMAC. It makes interfacing of complicated systems to CAMAC easier and offers the possibility of development and testing in stand-alone mode. Some of its most important features are the following:

- It is able to recognize all CAMAC functions and sub-addresses because the decoding can be done by software.
- The user is left the freedom of how to command it or how to make it respond.
- Its ability to interpret the CAMAC data also as a function or as a sub-address offers great flexibility.

(Submitted to Microprocessing and Microprogramming)

1. INTRODUCTION

This paper describes the MC68000 stand-alone CAMAC microprocessor system (68000 STAC), which has been designed for intelligent interfacing to CAMAC. CAMAC is a digital data-handling system in widespread use with on-line digital processors and computers [1]. The idea behind the work has been to make a versatile system that offers maximum flexibility and speed for the user, under minimum CAMAC control.

Maximum hardware flexibility is given by the modular structure of the processor system: a general processor module is the base, which does the CAMAC handling; around this module, interfaces can be built to take data or to control processes, for example.

Maximum software flexibility is offered by allowing users to define how to command the 68000 STAC through CAMAC and how to make it respond. Three suggested CAMAC-controlled modes are described in this paper.

The user can down-line load programs from a host computer and have powerful resident interactive languages, such as NODAL [2] and PILS [3].

Another important feature is that the 68000 STAC can work stand-alone, without CAMAC control, to do off-line testing and development.

Speed is also offered by the 68000 STAC, since it is able to transfer data in block mode at about 1/3 of the full CAMAC speed at the present time, and will be able to work at double this speed in one or two years' time.

The 68000 STAC has been developed out of the MC68000 CAMAC module (68KCM) [4], and the Independent Crate Controller (ICC) [5]. The experience and ideas obtained from the 68KCM have been used in order to transform the design of the ICC (based on MC68000) into a STAC, keeping the circuit board layout the same. This is a way of saving production effort and maximizing hardware and software compatibility. An Auxiliary Crate Controller (ACC) could be developed from the same design.

2. THE 68000 STAC MODULE

The 68000 STAC is built on a single CAMAC multiwire board, and is one slot wide.

On the circuit board are:

- i) 16 kbytes of Random Access Memory (RAM),
- ii) 32 kbytes of Erasable Programmable Read Only Memory (EPROM),
- iii) a CAMAC interface,
- iv) 2 serial V24 interfaces,
- v) a General Purpose Interface Bus (GPIB),
- vi) a simple programmable timer,
- vii) a hardware breakpoint-on-address facility,
- viii) a 60-pin flat cable connector, carrying the full processor bus.

To this processor bus, accessible from the front panel, a great variety of interfaces can be connected. An agreement has been achieved among MC68000 designers at CERN and Saclay on the connector type and pin assignment of this processor front-bus; however, further standardization would be desirable. There is still place for another connector on the front panel, with a maximum of 16 pins, to have more interrupt levels or other signals from an interface to the 68000 STAC. There is however no agreement on the use of this yet.

The timer can be programmed to interrupt the processor at fixed, but hardware adjustable, intervals. Interrupt driven monitors need such a timer, in which case the time interval usually is of the order of 10 ms.

Many advanced measuring instruments are equipped with a GPIB. Also microcomputer-oriented peripherals, like mini floppy disk drives, often have a GPIB and can therefore be connected directly to the 68000 STAC.

One of the V24 interfaces is for a terminal connection, the other one is to connect a host computer, for down-line loading of programs. These V24 interfaces are normally only used in stand-alone operation.

The CAMAC interface consists of two latches for the data and one for the function and sub-address; these are described in more detail in Section 4.

The hardware breakpoint-on-address facility is to be able to debug programs which are in EPROM. It is a 23-bit register which can be loaded by the processor. On the occurrence of the stored address an interrupt is generated. From this point in the program one could, for instance, trace (= single step) the following instructions.

On the front panel there is a 5-position switch with which one can choose the address from which the processor will start the execution after a power-on or reset. This gives the user the option of directing the processor into a monitor to operate stand-alone for testing or development or to make stand-alone measurements without CAMAC control, and starting again the 'CAMAC commanded' mode by simply selecting the desired mode with the switch and pushing the reset button.

The Z-line (initialize) of the CAMAC bus is indirectly connected to the processor reset line, similarly to the RESTART line from the function decoder. These lines generate a full initialization of the 68000 STAC and its peripherals, including interfaces connected to the processor front-bus.

It is foreseen to use the higher density memory chips that will become available in the near future. Some extra space is left open on the card to replace the 24-pin sockets with 28-pin ones, which allows doubling the memory capacity still two or three times.

RAM and EPROM are interchangeable in 4 kbyte blocks for RAM and 8 kbyte blocks for EPROM, on account of the difference in density between the two memory types.

3. 68000 STAC OPERATION MODES

3.1 Stand-alone mode

In stand-alone mode a terminal can be connected to one of the V24 interfaces. A resident monitor allows the user to debug hardware and software by setting

breakpoints and displaying address contents or internal registers. A transparent mode can be entered to communicate, through the other V24 interface, directly with a host computer, and to down-line load cross-compiled software. This software can range from low-level assembler programs using a cross-assembler [6] to high-level language programs such as PASCAL. The terminal can also be serviced in interrupt mode, for test applications, using it as a single-key controller.

Other possibilities in stand-alone mode are to perform the same measurements as under CAMAC control when either the CAMAC controller or computer is down or used for test purposes of other equipment, or to run a powerful interactive language to develop interactive software.

3.2 CAMAC controlled mode

The CAMAC communication can be done in several ways. Three suggested modes are listed below.

- i) In real-time data taking it is preferred not to interrupt the processor from CAMAC, but to wait for the LAM signal which indicates that the processor wants to be serviced. From the processor side polling of the CAMAC function can be done. Function and sub-address can be decoded by software.
- ii) Whenever the CAMAC addresses the 68000 STAC an interrupt can be generated, after which the processor reads the CAMAC function. The decoding of the function and sub-address can still be done by software in order to maintain flexible function recognition.
- iii) An interrupt can also be generated after hardware decoding of the function and sub-address. Up to three different commands can interrupt the processor at three different levels. This mode can be used in combination with both previous modes.

Interrupts can also be generated by interface modules, which are connected to the processor front-bus. There is only one level available; however, there is a possibility for extension. On the front panel there are in addition 4 LEMO connectors for external interrupt generation.

4. 68000 STAC CAMAC HANDLING

The 68000 STAC has one latch on the CAMAC Read lines and one on the Write lines, both 24 bits wide. The X (command accepted) signal is generated by N (station number), because the 68000 STAC accepts, in principle, all commands. The Q (response) and L (look-at-me) signals are given by flip-flops, which can be set and reset by the processor and reset by CAMAC. The function and sub-address lines connect to a decoder, of which the three free outputs can be wired to an interrupt encoder if hardware function decoding of some functions is desired. All function and sub-address lines, including the N and I (inhibit), are also latched, available to the processor on the data lines at a certain address, and arranged in the following way:

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
x	x	x	x	x	I	A8	A4	A2	A1	F16	F8	F4	F2	F1	N

The bits b11-b15 are used to read the position of the mode select switch on the front panel.

One idea is, in order to have a good data exchange, to transfer a processor status word after every operation. The processor can do error coding, which simplifies error checking from the CAMAC side; however this is up to the user, of course. To show the possibilities of using software decoding some examples are described in the next sections in the following order:

- i) Standard CAMAC function and sub-address decoding,
- ii) Function decoding using the data lines,
- iii) Sub-address extension using the data lines.

4.1 Standard CAMAC function and sub-address decoding

A fast decoding can be achieved by using a 2 kbyte translation table, preferably in the first 64 kbytes of the memory, in order to be able to use short addressing mode. This table could be in EPROM, which would slow down the operation slightly, or in RAM, as desired.

The following program example, written in MC68000 assembler language, assumes that the address registers A1-A3 have been loaded beforehand with the information as listed below:

A1 - pointer to CAMAC function latch,
A2 - pointer to CAMAC data latch,
A3 - pointer to the CAMAC translation table.

NAF is the abbreviation for the CAMAC call at slot number N with function F and sub-address A.

```
*  
* DECODING THE CAMAC NAF, EITHER AFTER AN INTERRUPT OR  
* COMING FROM A POLLING LOOP:  
*  
NAFDECO  MOVE.W  (A1),D1          READ CAMAC NAF  
          AND.W   #$3FE,D1        MASK OFF I, N AND MODE BITS  
          ADDA.W  D1,A3           CALCULATE TABLE ADDRESS  
          MOVEA.W (A3),A3         LOAD VECTOR  
          JMP     (A3)            GO TO ROUTINE
```

The translation table has to be organized in such a way that each address contains a vector to a routine that performs the NAF shown in Fig. 1.

Another possibility is to decode only the CAMAC function with a translation table. The same decoding program can be used, masking off also the sub-address bits. The corresponding translation table, shown in Fig. 2, needs only to be 64 bytes long. The sub-address can be considered after decoding, if it is relevant. This way of decoding takes a bit more time than the previous method, if the sub-address has to be considered, but uses less memory space. It could be preferred in many applications because often it will not be necessary to use the sub-address to transfer data or to command the 68000 STAC.

4.2 Function decoding using the data lines

Flexibility is offered by the 68000 STAC in order to use the data bits also as function information, as defined by the user. Using the data lines to define many more user functions could be done by reserving a certain CAMAC function to indicate this case. One could either decode this function with a separate test, before using the decoding program example, or afterwards do a second decoding for the data. This second decode step, for the data, would be similar to that of the program example, only A1 will change then into A2 and masking off bits would not be necessary. This method of decoding can be used in combination with the standard CAMAC NAF one. The translation table for the data can be similar to that shown in Fig. 2.

4.3 Sub-address extension using the data lines

The data lines can also be chosen to contain sub-address information, for instance for read-out of an interface with many channels. The sub-address information on the data lines could then be decoded in a similar way to that in the first steps of the program example, with the translation table address being the address of the multi-channel interface and the content being the channel data. One possibility is to employ only some of the data lines as the sub-address or sub-address extension, using the existing sub-address lines at the same time. Especially with the 24-bit transfer that would leave enough bits for the data. Using all data bits as sub-address, one would need a second NAF to transfer the data. This method of decoding can be used with all standard CAMAC functions, or one could reserve a special function for it without losing the possibility of the previously described function-decoding using the data lines.

5. RESPONSE TIME OF THE 68000 STAC

The execution times mentioned in this section are based on the speed of the 8 MHz version of the MC68000. A CAMAC instruction (NAF) takes 1 μ s. The MC68000 executes one 16-bit MOVE instruction in about that time. From the time the NAF has become valid the controller expects the X and Q signals to be given within

250 ns. It is clear that in this time the processor is not able to decode the function, act, and respond. To give an idea, each instruction in the program example takes 1 μ s. Two possible solutions of this timing problem are given below.

- i) If CAMAC gives only a single NAF, which would be an instruction taking quite some time for the processor to complete, the flip-flops could be prepared by the processor to give the response immediately. After having finished the job, the processor could set a LAM and prepare for the next response. Anyway the Q signal should always be checked and the NAF repeated in case of no response, but usually there will not be any delay.
- ii) If CAMAC requires short operations from the 68000 STAC, such as several un-ordered words having to be transferred, the same NAF should be repeated several times until a Q-response has been received.

The word un-ordered was used in point (ii) because ordered data can be transferred, in both directions, using the block-transfer in the Q-repeat mode. The Q-stop mode can only be used with controllers whose speed is slower than that of the 68000 STAC. The block-transfer with the 68000 STAC can be about 1/3 of the full CAMAC speed for 16-bit word transfers, and 1/4 of the speed for 24 bits.

Today 12 MHz versions are available already and probably 16 MHz versions will be produced in one or two years. This will double the transfer speed of today, bringing it close to the maximum of CAMAC.

6. INTERFACE POSSIBILITIES AND APPLICATIONS

There are many possible applications for the 68000 STAC in data taking and compression, before transferring the data over the CAMAC dataway. With a single CAMAC instruction a complicated task can be commanded of the 68000 STAC and the result could be a short block-transfer of compressed data from the 68000 STAC to CAMAC. The data taking becomes more versatile in this way, because the 68000 STAC is a modular and programmable system, and has large interface capabilities.

Fast data links in between computers could also be driven by the 68000 STAC. A modulator-demodulator (MODEM) link could be serviced in interrupt mode, while data exchange between CAMAC and 68000 STAC could be done in one of the CAMAC-controlled modes described in this paper. A dual port memory could be added, accessible to CAMAC and 68000 STAC, in order to increase the transfer speed and to provide a large data buffer.

The 68000 STAC could also easily interface advanced measuring instruments to CAMAC, such as complex network analysers, spectrum analysers, frequency counters, synthesizers, vacuum instruments, etc.

6.1 Optimized measurements

We plan to use the 68000 STAC in the future for measuring low temperature with resistance sensors. An interface will multiplex up to 8 sensor channels and digitize the alternating voltage across the sensors. The 68000 STAC will then digitally demodulate, filter, and convert the voltage values into temperature and display them. With the 68000 STAC in this set-up, one can have the system operating stand alone as well as reading out the temperatures by CAMAC. The processor can be programmed for optimum noise suppression and for averaging a number of values before outputting the mean temperature value and its standard deviation. The system will perform a periodic calibration cycle using known resistances.

The predecessor of the 68000 STAC, the 68KCM, has been used for fast and accurate measurement of nuclear spin polarization [4]. A nuclear magnetic resonance (NMR) interface was built to control a digital frequency synthesizer and to digitize the measured signal with a fast 14-bit analog-to-digital converter (ADC). The settling time of the analog circuitry (15-20 μ s) and the conversion time of the ADC (12 μ s) are of the same order of time as that in which the processor can accumulate the data, calculate the next frequency, and provide some synchronization signals. The measurement is fully CAMAC-controlled, and the averaged data (a few hundred 16-bit words) can be sent across using a block transfer. The speed and accuracy in the polarization measurement are essential in systems where

feedback is needed for automatic polarization growth maximization. The parameters to optimize are the frequency and power of a microwave source.

Another application for the 68000 STAC will be measuring $^3\text{He}/^4\text{He}$ isotopic concentrations with an acoustic transducer. The frequency of an acoustic resonator filled with a $^3\text{He}/^4\text{He}$ gas mixture depends on the ratio of the two isotopes. An interface has to be made to measure the resonance frequency. Corrections must be applied for the temperature and pressure of the transducer also measured by the 68000 STAC. Then the $^3\text{He}/^4\text{He}$ ratio will be calculated, displayed, and made available for transfer through CAMAC. A continuous calibration is possible, using the frequency of a reference resonator filled with ^4He as a time standard.

We also plan to use the 68000 STAC for the frequency measurement and control of a microwave source. The frequency can be measured by down-conversions, and the processor will bring it to the capture range of a phase-locked loop.

6.2 Distributed control and optimization

The most important use of a powerful stand-alone computer is in the distributed control of complicated systems. In such systems important savings can be achieved, timewise and moneywise, because of the inherent ability of independent debugging of subsystems.

A multiprocessor system can easily be assembled using an ICC and several STACs. As an example one ICC and two STACs are needed in a dynamic nuclear polarization growth maximization system, as mentioned earlier. The first STAC would measure and control the microwave frequency, and the second would measure the polarization using NMR, as described in the previous section. The ICC then commands the first STAC to set the microwave frequency at a known start value. From the second STAC it reads the data about the polarization during certain intervals, thus determining the speed of the polarization growth. With a hill-climbing optimization program in the ICC, which continuously adjusts the microwave frequency, one can maximize the polarization growth. Similarly the power of the microwave source can be optimized.

The 68000 STAC would also be ideally suitable for interfacing dipole and quadrupole mass spectrometers. It would require an interface to do the digitization of the measured signal and supply digital mass control. The 68000 STAC can then scan through a desired mass range, display the measured signal, and calculate the surface areas of the mass peaks. Using these, the 68000 STAC could provide signals to control various plants or processes such as ion sources, isotope separators, and ultra high vacuum equipment.

6.3 Intelligent man-machine interfacing

The 68000 STAC will be part of the man-machine interface in the future CERN Large Electron Positron collider (LEP) control room. One of its tasks will be to interface between the operator and the computer that controls the machine. Other tasks are under study: one is to be a personal processor for the operator to execute small computing tasks; another is as a fast display driver for a colour TV monitor, accepting high-level graphics commands through CAMAC.

Acknowledgements

I wish to thank B. Stumpe and G.C. Shering of the CERN Super Proton Synchrotron Division for their collaboration in the design and production of the 68000 STAC, and H. von Eicken of the Data Handling Division for software support. I also wish to thank T.O. Niinikoski of the Experimental Physics Division for many ideas and discussions on the 68000 STAC applications, and assistance in writing this paper.

REFERENCES

- [1] Atomic Energy Committee on Nuclear Instrument Modules, CAMAC a modular instrumentation system for data handling, TID-25875, EUR 4100e (1972) (Office for Official Publications of the European Communities, Luxembourg).
- [2] M.C. Crowley-Milling and G.C. Shering, The NODAL system for the SPS, CERN 78-07 (1978).
- [3] R.D. Russell and D.O. Williams, PILS -- A Portable Interactive Language System, CERN DD/OC/82/4 (1982). See also paper presented at the Second International Symposium INRIA, Real time data '82, Versailles (France), 1982.
- [4] T.O. Niinikoski and A. Rijllart, An MC68000 microprocessor CAMAC system for NMR measurement of polarization, Nucl. Instrum. Methods 199, 485 (1982).
- [5] B. Stumpe, Hardware description of an MC68000 independent crate controller - ICC 68000 (SPS 3350), CERN SPS/AOP/Note/81-13 (1981).
- [6] H. von Eicken, M68MIL, A cross macro assembler for the Motorola M68000, CERN DD/80/26 (1980).

Figure captions

Fig. 1 : CAMAC calls translation table.

Fig. 2 : CAMAC functions translation table.

Fig. 1

Fig. 2