INTELLIGENT PCI CONTROLLER WITH EMBEDDED MICROPROCESSOR FOR CRATE CONTROLLERS KK009, KK012 AND K106.

Yu.B. Semenov, V.E. Zhuchko, D.V. Kamanin, V.G. Tishcneko

A wide range of CAMAC blocks available nowadays on the market allows assembling virtually any electronic setup for modern experiments in nuclear physics. It allows configuring flexible and powerful systems for acquisition and storage of experimental data. To control such systems one normally uses inexpensive but powerful personal computers (PC) running under various operating systems (OS). The choice of OS depends on peculiarities of the task being solved. Very often, a fast response of PC is needed to handle the interrupt request from experimental hardware (Look-At-Me or LAM signal from CAMAC). This requirement can be satisfied in part by using a real-time OS (like QNX, OS-9, VxWorks, etc.). However, this approach has some disadvantages. First of all, all commercial real-time OSes are typically very expensive. The second disadvantage is connected with the low performance of CAMAC interface in comparison with the usual performance of a modern PC (the internal CAMAC cycle amounts to 1 μs). The difference in clock rates of PC and CAMAC results in ineffective scheduling of a processor (CPU) time. The bulk of time the CPU would have to keep waiting for the end of a CNAF command execution and for arrival of a new data chunk. Thus, starting from a certain rate of LAM requests, the CPU will be busy with handling of interrupts only and the execution of the OS as well as of user programs will be blocked. In fact, as we understand now, the bulk of this time the CPU will just stand idle. We offer a more effective method to solve the problem in question. It is based on the approach frequently employed by National Instruments. Namely, we propose to move the procedure of data taking from execution context of CPU to a special-purpose interface PCI card being installed into PC. The card is equipped by a microprocessor build on the top of programmable logic arrays and by a FIFO (First In – First Out) memory. The card issuers the interrupt request to CPU after reading of a whole event (or even several events depending on user settings). The advantages of this approach are evident. First of all, the CPU gets free of routine procedure of data taking from individual CAMAC blocks per each interrupt. Secondly, the real-time regime is provided in a straightforward way. It makes it possible to run any OS including widely used OSes from MS-Windows family. The response time to CAMAC interrupt is fixed and amounts to 30 ns, which is much less than the time guaranteed by commercial real-time OSes.

The developed PCI card is compatible with any of widely used crate controllers like KK009, KK012, K106. It has two modes of operation:

i) Passive mode, where the card emulates the work of its prototype [1] depending on the type of plugged crate controller (KK009, KK012, K106). This mode is provided for compatibility with already existing software. A minor modification of software is needed due to usage of PCI bus instead of outdated ISA bus, which is hardly available in modern PCs.

ii) Programmable mode, where a set of instructions stored in the RAM of PCI card starts to execute when an interrupt is fired in the CAMAC crate been serviced.

The core of the proposed PCI card is based on a programmable microchip ALTERA of ACEX series EPIC100QC208-3. This card is a further development of our former solution [2].

From software point of view the card occupies a memory region of FFFFh bytes. The lower memory space from base address to the address 7FFFh is mapped into the address space of crate controller KK009 (or 0000h - 1FFFh for K106). The memory from 8000h to 87FFh (2048 bytes) is reserved for the memory of embedded microprocessor. The microprocessor has eight 16-bit general-purpose registers and one register accessible both from PC and from microprocessor. The whole memory including FIFO is organized in the frame of one ALTERA chip. Such an approach provides the fastest memory access.

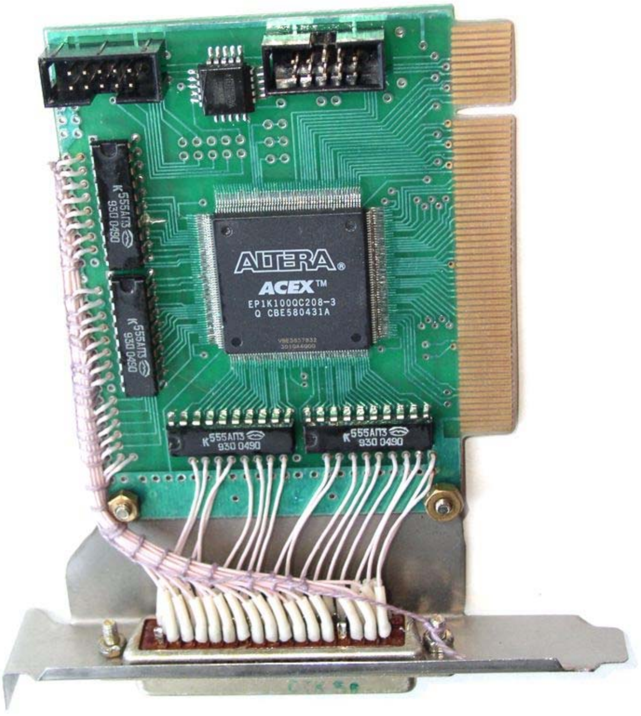

Fig.1 Photo of a production-sample PCI card.

To program the microprocessor a set of about 30 commands is provided including the commands for reading the CAMAC blocks into FIFO memory, simple arithmetic, conditional jump instructions, conditional branch instructions, commands to issuer interrupt to CPU. This set of commands allows organizing an intelligent data acquisition system, which will only read those blocks have to be read accordingly to the structure of a physical event. Such implementation of a software trigger (second level trigger) allows, on the one hand, to reduce the data flow, and, on the other hand, to free the CPU of PC for executing of other tasks for data storage and/or on-line data analysis. To make the microprocessor programming easier a special cross-compiler is developed. With its help one can build a task corresponding to the actual hardware configuration, to load the task into memory of PCI card, to test the task, and to apply corrections if needed. A more detailed description of assembler instructions for microprocessor, as well as other relevant technical documentation for users and developers is available at Internet page of the FOBOS group [3].

The approach described above can be successfully adapted to any crate controller. In particular, it can be used with controllers based on VME or VSB bus standards, like in acquisition system of the FOBOS spectrometer [4].

A further development of our approach will be a programmable CAMAC controller with integrated functionality of our PCI card, which will be controlled via local network. In that case, there will be no need to place a special data taking PC close to experimental setup. It opens up new perspectives for usage of portable computers in experiments and wireless connections between experimental hardware and control PC.

.

REFERENCES

1. Antyukhov V.A. et al., Digital CAMAC Modules (Issue XV), P10-87-928, JINR, 1987.
2. Yu.B.Semenov et al. “Heavy Ion Physics” FLNR JINR scientific report 2001-2002, Dubna, 2003, Ed. by A.G.Popeko, p.233
3. http://fobos.jinr.ru/solutions
4. Yu.B.Semenov et al. “Heavy Ion Physics” FLNR JINR scientific report 1999-2000, Dubna, 2001, Ed. by A.G.Popeko, p.197