CERN-Data Handling Division
DD/89/5
D. Burckhart et al.
February 1989

SOFTWARE SUPPORT FOR THE CERN HOST INTERFACE

Presented at the International Conference on Impact of Digital Microelectronics and Microprocessors on Particle Physics, Trieste, Italy, 28-30 March 1988

DD-bt

SOFTWARE SUPPORT FOR THE CERN HOST INTERFACE

D. Burckhart, T.J. Berners-Lee, C. Bizeau, R. Divia, P. Gallno,
B. Heurley, K. Hollingworth, D. Jacobs, R.A. McLaren, H. Müller,
C. Parkman, E. van de Bij, A. van Praag (CERN); A. Guglielmi,
M. Rance (DEC); T. Almeida, P. Gomes (LIP); P. Alves (INESC)

ABSTRACT

A system of software modules has been designed for the CERN Host Interface (CHI) [1] project, a set of interfaces between VAX series computers and both FASTBUS [2] and VMEbus [3]. The CHI hardware allows the software to be split between its microprocessor and the VAX. Provision has been made for multi-user support on the host and communication software as well as library packages. Software tools assist the user in preparing his application software.

1. INTRODUCTION

In order to ensure good coordination of the hardware and software projects, a software system was designed in parallel with the hardware development of the CHI. By taking advantage of the common hardware architecture for the FASTBUS and VMEbus versions of the CHI, and by using modular software, a significant part of the support software can be shared for both versions. Remote Procedure Calls (RPC) [4] form the standard communication technique for system and application software and allow the use of library packages such as the NIM FASTBUS Standard Routines [5] locally on the processor or remotely on the host. Fast data transfers between the host and the instrumentation bus are supported making use of Direct Memory Access facilities designed into the CHI. An enhanced VAX/VMS driver reduces operating system overheads.

2. REQUIREMENTS

In conventional data acquisition for CAMAC [6] based experiments the driver formed part of the data acquisition kernel. Sequential readout lists had to be pre-prepared using a specialized language for FASTBUS in the CERN FASTBUS Interface (CFI) [6]. For the current generation of High Energy Physics experiments at LEP, the online software architecture had to be extended for new instrumentation buses which make heavy use of embedded processing for their multi-level trigger stages and for data reduction. The size and complexity of the readout architecture of those experiments require more flexible readout algorithms than are possible with readout lists. This can only be provided by the close coupling of a user programmable general processor to the instrumentation bus in use. On board intelligence provides the user with maximum flexibility for his particular event readout software. During the development

phase of test or data acquisition programs a user-friendly and time-saving debugging environment is required, whereas at run time, fast execution and data transfer are important.

3. HOST TO INTERFACE COMMUNICATION

A software protocol run over the hardware link accomplishes control of the direction of the link, multiplexing of data between different tasks at either end, and flow control, so that data are never sent unless a receiving buffer exists. In collaboration with Digital Equipment Corporation the original VAX/VMS drivers have been enhanced to include concurrent access by many tasks. Send and receive transactions are handled in a single request in order to reduce operating system overhead for the communication protocol. The functionality provided to the user is that of an ISO standard transport service, allowing the creation and deletion of logical task-task connections. These connections are used by the RPC system, and are also directly available to application programs.

4. DEVELOPMENT ENVIRONMENT

Application software may be developed and debugged on the VAX, e.g. using CHI-resident routines remotely. At a later stage time-critical modules may be loaded into the CHI, and may then call the same library routines locally. At run time, when a program on one processor calls a subroutine on the other, parameters are passed to the subroutine, the subroutine is executed, and status and data are returned to the caller.

5. FASTBUS ACCESS

The NIM FASTBUS Standard routines have been implemented as a CHI-resident library which is based on the implementation of the Standard FASTBUS routines for the GPM [8] [9]. Use is made of the "key" address technique, i.e. the FASTBUS action of a single cycle routine consists of one MC68020 MOVE instruction to a FASTBUS key address. FASTBUS data may be transferred to and from CHI memory, or the "external buffer" mode of the Standard Routines may be used to transfer it directly to or from the VAX. Service Request and Interrupt Messages, accepted by the CHI FASTBUS port, result in the execution of the corresponding Standard service routines.

6. CONCLUSIONS

The CHI support software provides a flexible, user-friendly environment for application code running on the host or on the processor. Taking advantage of its built-in hardware features like DMA facilities and FASTBUS key addresses, high efficiency is achieved. Remote

Procedure Calls are used as a communication technique and an enhanced DEC Driver minimizes communication overheads.

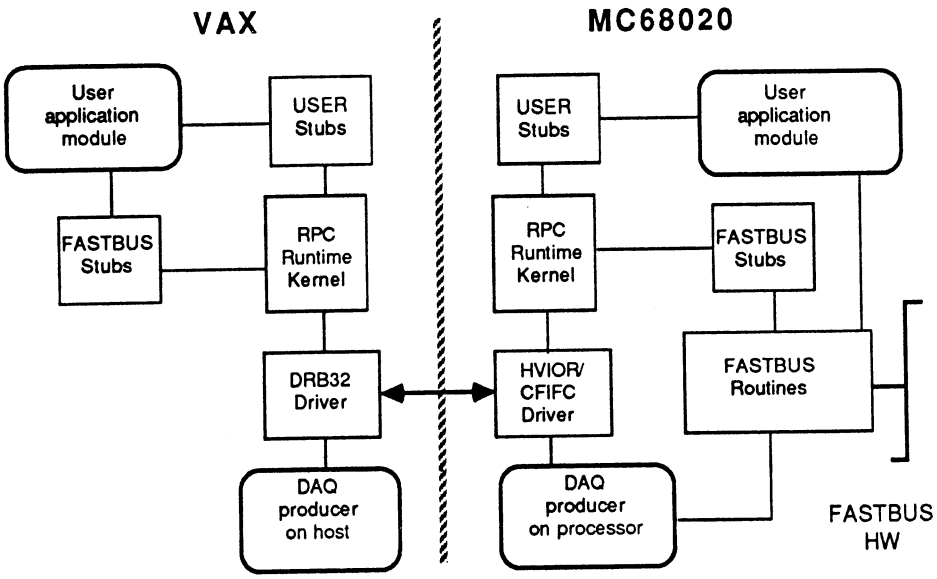

Fig. 1: CHI FASTBUS software overview

7. REFERENCES

[1] McLaren, R.A. et al., "The CERN Host Interface". Presented at the Nuclear Science Symposium, San Francisco, 21-23 October 1987

[2] ANSI/IEEE Std 960-1986, "IEEE Standard FASTBUS Modular High-Speed Data Acquisition and Control System"

[3] IEEE Standard 1014, "VMEbus: A Standard Specification for a Versatile Backplane Bus"

[4] Berners-Lee, T.J., "Experience with Remote Procedure Call in Data Acquisition and Control", Proceedings of the 5th Conference on Real-Time Computer Applications in Nuclear, Particle and Plasma Physics, San Francisco, May 1987

[5] U.S. NIM Committee, "FASTBUS Standard Routines", U.S. Department of Energy DOE/ER-0324, March 1987

[6] CAMAC, updated specification, Esone Committe, EUR8500, 1983

[7] Jacobs, D. et al., "CFI user guide", private communication, January 1985

[8] Müller, H., "A General Purpose Master (GPM) and Memory Module (DSM) for Online Applications", IEEE Transactions on Nuclear Science NS-33, no.1, February 1986

[9] Kozlowski, T. and Foreman, W.M., "An implementation of the new IEEE Standard Routines for FASTBUS". IEEE Transactions on Nuclear Science, NS-34, no.4, August 1987.