The cern host interface

Article *in* IEEE Transactions on Nuclear Science · March 1988

DOI: 10.1109/23.12734 · Source: IEEE Xplore

15 authors, including:

Tim Berners-Lee
Massachusetts Institute of Technology
155 PUBLICATIONS **23,896** CITATIONS
SEE PROFILE

Daniele Jacobs
University of Trinidad and Tobago (UTT)
40 PUBLICATIONS **246** CITATIONS
SEE PROFILE

Hans Muller
CERN
855 PUBLICATIONS **18,441** CITATIONS
SEE PROFILE

Arie Van Praag
CERN
65 PUBLICATIONS **40** CITATIONS
SEE PROFILE

Data Handling Division
DD/87/22
October 1987

[redacted]

The CERN Host Interface

R.A. McLaren, T.J. Berners-Lee, D. Burckhardt, R. Divia, B. Heurley,
K. Hollingworth, D. Jacobs, H. Müller, C.F. Parkman, E. van der Bij,
A. van Praag
(CERN, 1211 Geneva 23, Switzerland)

A. Guglielmi
(Digital Equipment Corporation GmbH, Freischützstrasse 91,
8000 Munich 81, Germany)

T. Almeida, P. Gomes
(Laboratorio de Instrumentaçâo e Fisica Experimental de Particulas,
Av. Elias Garcia 14-1, 1000 Lisbon, Portugal)

P. Alves
(Instituto de Engenharia de Sistemas e Computadores,
Rue José Falcâo 15-6, 4000 Porto, Portugal)

(Presented at the Nuclear Science Symposium, San Francisco,
21-23 October 1987)

DD-bt

THE CERN HOST INTERFACE

R.A. McLaren, T.J. Berners-Lee, D. Burckhart, R. Divia, B. Heurley, K. Hollingworth, D. Jacobs, H. Müller,
C.F. Parkman, E. van der Bij, A. van Praag
CERN, 1211 Geneva 23, Switzerland

A. Guglielmi
Digital Equipment Corporation GmbH, Freischützstrasse 91, 8000 Munich 81, Germany

T. Almeida, P. Gomes
Laboratorio de Instrumentaçâo e Fisica Experimental de Particulas, Av. Elias Garcia 14-1 1000 Lisbon, Portugal

P. Alves
Instituto de Engenharia de Sistemas e Computadores, Rue José Falcâo 15-6, 4000 Porto, Portugal

Abstract

The CERN Host Interface project aims to provide modular interfaces between Digital Equipment Corporation's VAX series computers and the most popular high energy physics buses. These user-programmable interfaces contain a powerful central processing unit, large data memories and ports which allow the user to configure the interface for the required host computer and target bus.

Introduction

The CERN FASTBUS Interface (CFI) [1], has allowed us to acquire experience with interfacing to FASTBUS [2] and to discover the limitations of the design of the CFI. A study was initiated to collect the requirements of high energy physics experiments (mainly those planned for CERN's Large Electron Positron (LEP) accelerator), to review currently available hardware and to discuss the various interfacing architectures. The conclusion of the working group, endorsed by the LEP collaborations, was that work should begin on the CERN Host Interface (CHI), a series of user-programmable interfaces, which would interface FASTBUS and VMEbus [3] to Qbus-, UNIBUS- and VAXBI-based VAX computers from Digital Equipment Corporation (DEC).

For the implementation it was decided to retain the approach, used in the CFI, of splitting the interface into two sections connected by a high speed link. In the VAX, a standard DEC input/output register provides the interface to the internal VAX bus, whilst dedicated "ports" based in FASTBUS and VMEbus provide the logic for the target bus. A problem with this approach is the overhead for single word accesses to FASTBUS or VMEbus. This has been minimized by providing an infrastructure which allows user programs to run on the microprocessor in the interface and by incorporating the link protocol into the VAX/VMS driver.

Software support

Remote Procedure Call (RPC) [4] techniques allow user software to be split between the VAX and the Motorola MC68020 microprocessor incorporated in the CHI. Software may be developed on the VAX, for example, and time-critical modules may be cross-compiled for the microprocessor in the CHI. Utility programs on the VAX allow these routines to be loaded into the CHI, controlled, and linked transparently to routines on the VAX. 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 returned to the caller. The standard FASTBUS routines [5], for example, are provided on the MC68020 and are available via RPC on the VAX.

Link Protocol

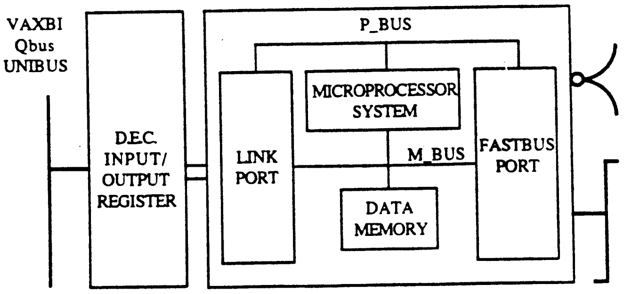
A protocol is run over the link to accomplish:

- Control of the direction of the (basically half-duplex) link;
- Multiplexing of data between different tasks at either end, using multiple logical connections;
- Flow control, so that data are never sent unless a receiving buffer exists.

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.

On the MC68020 side, the software distinguishes between transfers of arrays of bytes, words and long-words. Advantage is then taken of the byte swapping ability of the hardware to keep each data type intact after transfer to or from the VAX.

The FASTBUS Implementation of the CHI



Block diagram of the VAX to FASTBUS interface

The VAX to FASTBUS interface is a single width FASTBUS module with four major components: the microprocessor system, the data memory, the parallel link port which connects to the input/output register in the VAX, and the FASTBUS port which contains the Master and Slave logic for FASTBUS. Two separate address and data buses (the P_bus and the M_bus) interconnect the components. The P_bus allows the microprocessor to access control registers and to trigger single word data transfers in the link port and FASTBUS port. The M_bus allows Direct Memory Access (DMA) from the ports to the multi-ported data memory and access by the microprocessor to the data memory.

The Microprocessor System

The microprocessor system is a 16 MHz Motorola MC68020 with 1 MByte of local random access memory, 1 Mbyte of erasable programmable read only memory, a real time clock and two serial RS232 ports for communication with a terminal and a host computer during debugging.

The Data Memory

The 1 Mbyte multi-port data memory can be accessed from the microprocessor, the link port, and both the Master and Slave logic of the FASTBUS port. Arbitration for the data memory is normally performed on a cycle-by-cycle basis but can also be "locked" to allow unique access by one of the ports which is required, for example, during FASTBUS pipeline transfers.

The Parallel Link Port

The link to the minicomputer must be adaptable. Whereas 700 series VAXes and microVAXes have 16-bit full-duplex parallel input/output registers [7], the 8000 series machines have 32-bit half-duplex registers [6]. The link port hardware can multiplex 32-bit words onto 16 or 8-bit wide paths, putting the bytes in any order. A set of adapter boards then provide the necessary choice of physical interface and drive levels. The microprocessor can read which adapter board is connected, and so configure the hardware appropriately.

The link port may be accessed by microprocessor instructions, or data can be transferred directly to or from data memory or FASTBUS using a 24-bit word counter. A pipeline register allows transfers to proceed at a theoretical bandwidth of 15 Mbytes/sec. In practice, however, the bandwidth is at present limited to 5.5 Mbytes/sec on 8000 series VAXes and 1.2 Mbytes/sec on the microVAX.

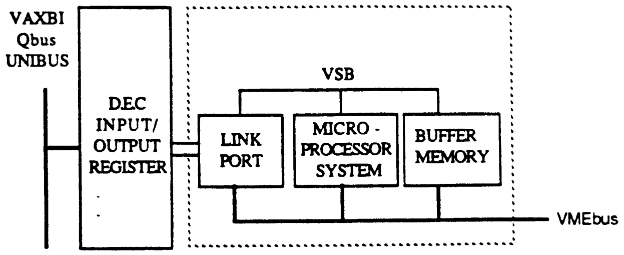
The FASTBUS Port

There are three main components in the FASTBUS port: the Master, the Slave, and the Interrupt handler. The Master logic is based on the principle used in the General Purpose Master [8]. Operations on FASTBUS are triggered by accessing "key" addresses in the MC68020 address space, operands for these operations are transferred on the data lines. Any sequence of FASTBUS operations may then be constructed by a number of MOVE instructions. These instructions may also be stretched by allowing the MC68020 cycle to terminate only when the FASTBUS operation is complete. FASTBUS block transfers are handled by a dedicated DMA controller which allows transfers of up to 20 Mbytes/sec. FASTBUS pipeline transfer rates are programmable. In the case of non-zero SS responses, parity errors, or a timeout on the FASTBUS operation, an exception handler is invoked.

In the FASTBUS Slave port, the Control and Status Registers (CSR) for FASTBUS Interrupt Messages are implemented in hardware. Dedicated DMA logic transfers the message to the data memory. When the message is complete, an interrupt is sent to the microprocessor and a flag set. The interrupt service routine resets the flag, which enables subsequent interrupt messages to be received. FASTBUS Service Requests interrupt the MC68020 directly.

All other Slave CSR and DATA registers are emulated by microprocessor software.

The VMEbus Implementation of the CHI

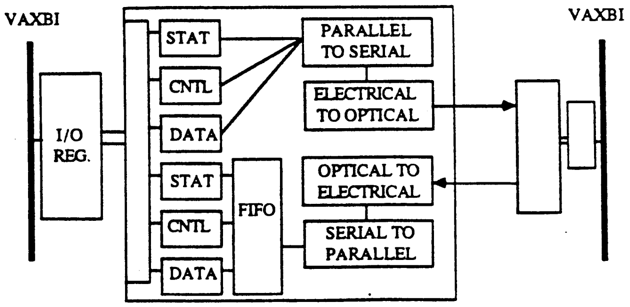


Block diagram of the VAX to VMEbus interface

Two factors influenced the decision to implement the VMEbus interface in a multi-board configuration. Firstly the board size is limited in VMEbus and secondly, VMEbus microprocessor modules and data memory are available commercially. The only module to be designed was a VMEbus input/output register to connect to the VAX [9]. This module has similar functionality to the parallel link port on the FASTBUS implementation, providing a link to the input/output register in the VAX via plug-on adaptors and byte and word swap logic. VMEbus logic provides a Master/Slave port with DMA capability. Data can be transferred with standard (A24) or extended (A32) addressing and D16 or D32 data widths. Block mode data transfer is supported to allow higher bandwidth. A VME Subsystem Bus [10] Master port is also implemented.

The CHI Optical Connection

The LEP experiments will have a distance of up to 500m between the data acquisition computers and the detectors. This can be bridged in several ways: for example, by a FASTBUS to FASTBUS link, or by providing an optical extension module to the parallel link of the CHI. The latter is being implemented as part of the CHI project.



Block diagram of the CHI Optical Link

Transmission over the optical link is full duplex with identical receive and transmit logic at both ends of a pair of optical fibers. Cables from the input/output register in the VAX enter the optical extension module and the signals are divided into three groups; control, data and status. These are serialised and transmitted over a pair of optical fibers to the remote module where the serial information is converted back into standard signals. Flow control, necessary for data transfer, is provided by a stop/go semaphore which signals the sender of a block of data when the receiving end can no longer accept transfers. A FIFO at the receive end buffers data from the transmitter. Only parity (one bit per byte) is used on the link and results of both component and prototype tests, on a VAX to VAX connection, will determine whether more sophisticated error detection and correction are required.

Conclusions

The CHI is an example of how modular hardware, in the form of ports, can be interconnected using a well-defined, high speed link. Although conceived with the aim of interfacing VAXes to FASTBUS and VMEbus, the CHI can also provide a VMEbus to FASTBUS connection. In addition, fast VAX to VAX, FASTBUS to FASTBUS or VMEbus to VMEbus links can be implemented with, or without, the optical connection.

User code may run in the interface in a familiar software environment which includes tra-.sport connections to tasks on the host machine, and the standard FASTBUS routines. Used in this way, the CHI provides flexibility unavailable with the previous generation of interfaces.

Acknowledgements

The work described in this paper is part of a joint project between Digital Equipment Corporation and CERN. We would like to thank E. Gerelle, DEC's Joint Project Manager and F. Gagliardi (CERN) for setting up the framework for the project. At the LIP, in Portugal, we are indebted to G. Barreira who organised the infrastructure required to construct the optical link. At CERN, we would like to thank J. Allaby, J.A. Bogaerts, D.M. Sendall and H. Verweij for their support and K.J. Peach, P. Ponting and E.M. Rimmer for their advice. Last, but by no means least, we would like to thank that essential group of people, our users.

Trademarks

VAX, VMS, DRB32, DRQ-11, DRE-11, VAXBI, UNIBUS and Qbus are trademarks of Digital Equipment Corporation.

References

[1] D. Jacobs et al., CERN DD/ED, "CFI User Guide", October 1983, private communication

[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] T.J. Berners-Lee, CERN, "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] "DRB32 Technical Manual", Digital Equipment Corporation, October 1986

[7] "DRU-11C, DRE-11C, DRQ-11C Alternate Buffer DMA Interfaces", Digital Equipment Corporation, Computer Special Systems, Munich, February 1984

[8] "A General Purpose Master (GPM) and memory module (DSM) for online applications", IEEE Transactions on Nuclear Science, vol. NS-33, no. 1, February 1986

[9] C.F. Parkman, CERN DD/OC, "The VMEbus/Host Interface I/O Register Module 'HyperVior' User's Manual", April 1987, private communication

[10] IEC BUS 822 - Parallel Sub System Bus for the IEC BUS 821 (VSB) International Electrotechnical Commission 47B (Central Office) 22.