

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/3133311>

The CHI, a new Fastbus Interface and Processor

Article in IEEE Transactions on Nuclear Science · May 1990

DOI: 10.1109/23.106645 · Source: IEEE Xplore

CITATIONS

5

READS

16

8 authors, including:



Hans Muller

CERN

855 PUBLICATIONS 18,441 CITATIONS

[SEE PROFILE](#)



Tim Berners-Lee

Massachusetts Institute of Technology

155 PUBLICATIONS 23,896 CITATIONS

[SEE PROFILE](#)



Annemie Bogaerts

University of Antwerp

715 PUBLICATIONS 13,998 CITATIONS

[SEE PROFILE](#)



Arie Van Praag

CERN

65 PUBLICATIONS 40 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



LHCb DAQ (PhD thesis work) [View project](#)



Re-decentralizing the web [View project](#)

The CHI, a new Fastbus Interface and Processor

H. Müller, T. J. Berners-Lee, A. Bogaerts, D. Burckhart, R. Divià, K. Hollingworth, R. McLaren,
A. van Praag: CERN, 1211 Geneva 23, Switzerland

ABSTRACT

The CERN Host Interface (CHI) is a family of interfaces to interconnect Fastbus, VMEbus, and external host computers. The Fastbus interface consists of a processor board (CHI-P) and host-specific I/O ports allowing connection using fast parallel or serial interfaces. For efficiency in a data acquisition chain, the CHI-P contains a 1 MByte triple-port memory which allows concurrent access by Fastbus (as master or slave), the host link, and the 4.5 Mips on board processor.

The processor, an MC68030 with floating point coprocessor, also has 1 Mbyte of local memory and 1.25 Mbyte of EPROM. The hardware modularity allows the CHI-P to be used as an interface, as general purpose Fastbus test module, or as an embedded Fastbus processor. The resident software supports its use in each of these modes.

Remote Procedure Calls (RPC), an ISO style transport service, and the Standard Routines for Fastbus are provided on the host and on the CHI-P, allowing the migration of software between the two. Menu-driven test software, and an interactive interpreted/compiled language support its use in a test environment.

INTRODUCTION

The CERN Host Interface project [1], which was set up for the requirements for the LEP experiments, has created a new family of CERN- and industry supported hardware and software components. These are now available to interface Fastbus or VMEbus to VAX¹ computers, workstations or personal computers (Fig 1). In addition, Fastbus can be interfaced to VMEbus. In each case, fast parallel interfaces, an optical link, LANs, or serial lines are used.

HARDWARE MODULES

For Fastbus, the interface hardware consists of a new Fastbus processor module, the CHI-P, and various host interface daughter boards. The CHI-P board is a single width Fastbus master/slave which may operate on either a Fastbus cable or a crate segment. The daughter boards for the host connection can be plugged onto the CHI-P, forming a double width Fastbus module.

For high performance requirements, a proprietary VAX I/O register (DRB32, DRQ11 or DRE11 for the VAX/BI, Q-bus and Unibus respectively) may be connected to a CHI daughter board which includes DMA and byte swapping hardware. For test environments, "cheapernet" or V24 (RS232) lines can be

used. An SCSI daughter board provides a high speed read-out channel into a fast workstation, or allows direct writing to tape. The CHI-P can also be used without I/O daughter boards as a stand-alone processor for control or trigger purposes since its design concepts have integrated many ideas from the Delphi SGPM processor project [2].

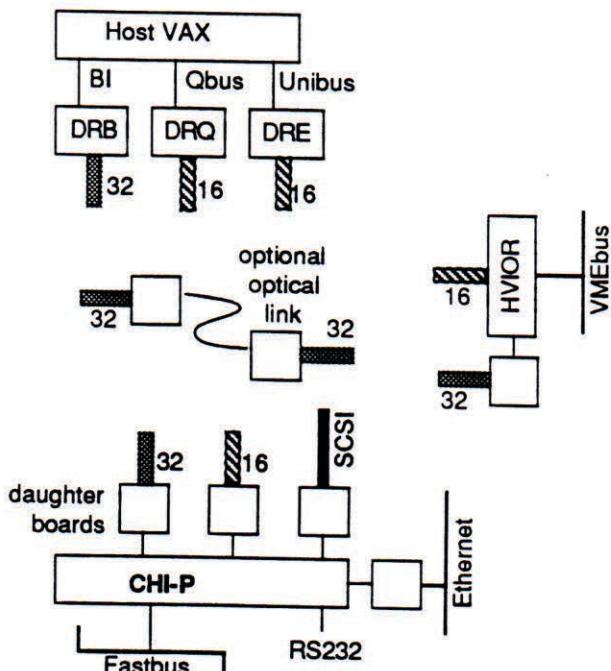


Fig. 1: CERN Host Interface Overview

The optional optical link [3] extends a 32 bit wide parallel link to distances of up to 10 km.

For the VMEbus, the interface consists of a parallel interface card (the "HVIOR"), and a standard VMEbus processor.

With this family of components, interfaces can be well adapted to price, performance and environment requirements.

SOFTWARE ARCHITECTURE

The CHI concept makes use of high performance processors situated in Fastbus or VMEbus, which can run large user programs locally. The host computer is mainly used for program development and downloading, as well as for monitoring or writing data to mass storage. The CHI software modules [4] are split between bus processor modules and the host processor modules. Libraries are shared between the two processors, using the Remote Procedure Call (RPC) technique [5], running over a connection oriented transport protocol. The transport service functions are in the ISO style [6] and may

¹ VAX, VAX/BI, Q-bus and Unibus are trademarks of Digital Equipment Corporation

also be called directly using the CERN "CATS" [7] calling sequence. In order to reduce system overheads under VAX/VMS, an enhanced (DEC supported) driver was developed which also allows simultaneous access from many processes.

With these techniques, application programs can be run either locally in the CHI-P or remotely on the host, and data can be transferred in both directions. (Fig 2)

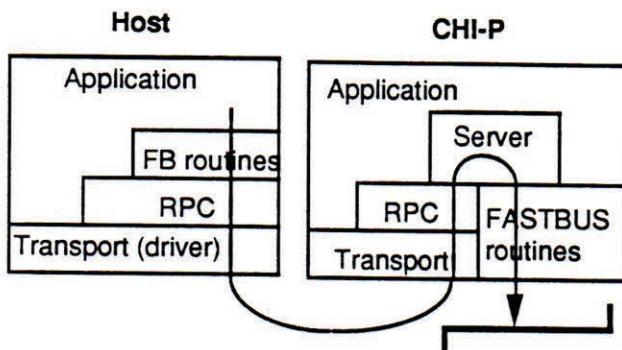


Fig 2. Software layering of the remote Fastbus routines

The VALET-plus software environment [8] for VMEbus is also available for Fastbus systems. The host software allows remote logon, and provides a file server and graphics server to the CHI processor. The CHI processor software includes a resident PILS language interpreter and compiler [9].

CHI-P PROCESSOR

In the following, this paper restricts itself to a discussion of FASTBUS interfacing and processing with an emphasis on hardware architecture, and application environments.

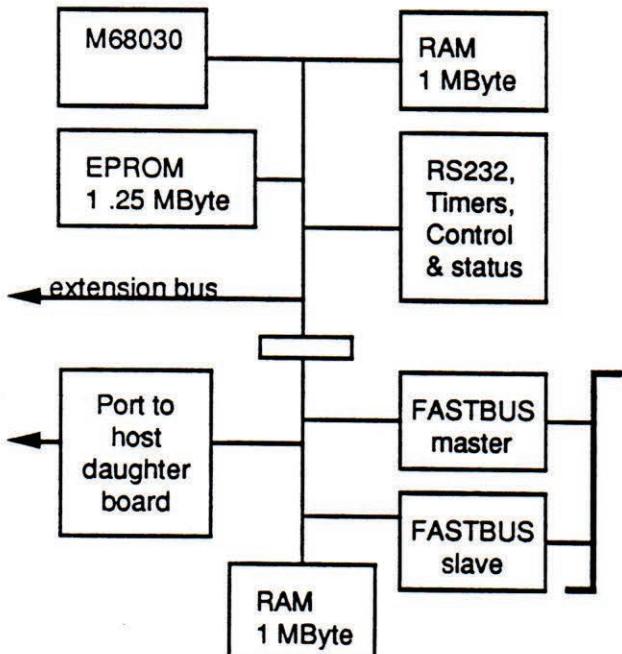


Fig. 3: CHI-P internal architecture

Processor

In order to satisfy the demand for computing power and for memory protection, a Motorola MC68030 microprocessor with MC68882 floating point coprocessor was chosen for the CHI-P module. The standard CHI-P Fastbus module runs at 16.0 MHz clock rate without wait states when accessing 100ns static RAM. Its performance was measured as roughly 4.5 Mips

The processor system in the CHI-P is connected to two serial I/O channels via RS232 lines, based on a Motorola MC68681 DUART chip. This multifunction chip also integrates a timer and general-purpose I/O signals for the CHI-P front panel. A real time clock is available for system timing routines. Currently, 75 hardware interrupts are implemented in the CHI-P. Two of processor's interrupt levels are used by the 16-bit interrupt controllers of the Fastbus port and the host I/O port. One level is used by the slave port protocol emulation and one level is connected to a software generated interrupt for the VAX/VMS-like Asynchronous System Trap (AST) emulation.

Memory Map

The CHI-P (Fig. 3) has four primary address sections which are private to the processor: a 1 Mbyte (future 4 Mbyte) private RAM memory, two EPROM sections for 32 bit and 8 bit EPROMs, and an 8 bit port for RS232 lines, timers, control and status registers and breakpoint memory.

The four external address sections provide processor access to the Fastbus port, an optional I/O host port, the X-bus extension, and a 1 Mbyte shared data memory. This 32 bit wide memory is shared between the I/O host port, the Fastbus ports and the processor, which each arbitrate for access. (A four port, 'first-come-first-served' arbiter with 25ns time sampling is used.) The MC68030 processor can also access DMA- and interrupt devices as well as other Fastbus port specific registers through this address space.

The X-bus connector extends the MC68030 bus for expansion options which reside on the CHI-P mother board. Flat mounting, eurocard sized plug-in boards can be used, the first one being the "cheapernet" interface.

The I/O host port provides access to the shared data memory for the daughter boards. This memory port could as well be used in the future for newly designed number-crunching coprocessors for trigger applications.

The Fastbus master port

The Fastbus master can execute MC68030 instructions in Fastbus using the "key" address technique. Certain address ranges provide access to all Fastbus port internal devices and registers. Others are used to generate Fastbus cycles and operations. Others select the Fastbus protocols and are acknowledged by the Fastbus handshake. These key addresses provoke actions such as FASTBUS operations, the action

being defined by the address used. For example, fields in the Primary Address key address control the Fastbus arbitration method and level used (Fig 4). The data bus transfers the 32 bit value to be used on the AD lines, using the full power of the processor's addressing modes.

AP	AL5	AL4	AL3	AL2	AL1	AL0
Arbitration level						
0	0	Normal				
0	1	Priority				
1	0	Assured				
1	1	Disable				

Fig. 4: Arbitration parameters are contained within the key address.

Each cycle is accompanied by a programmable counter operation for long and short Fastbus timeouts. Timeouts and SS responses are terminated by a MC68030 bus error, with flag bits to facilitate error handling. Asynchronous or synchronous Fastbus block transfers are autonomously executed via a DMA controller which can transfer data between Fastbus and the data memory at up to 20 Mbytes/s. During this operation, the processor can compete in accessing the data memory. The maximum length of the block is 1 Mbyte. Word count and address comparison registers allow termination at any memory boundary. The CHI-P can generate any sequence of Fastbus cycles, standard operations, or complicated compound operations since these are composed as individual read or write assembler instructions. Powerful MC68030 instructions, performing read-modify-write operations, can be used in Fastbus.

A wide range of options are implemented, such as generation of reserved MS codes, suppression of individual SS error handling, and suppression of timeouts.

Fastbus Slave port

This port supports geographical addressing and several broadcast protocols, such as the T-pin scan. The secondary addresses are handled by a 24 bit NTA register. The maximum data rate for accessing the data memory in the data space is 15 Mbytes/s. Fastbus Interrupt Messages to the 16 interrupt receivers are implemented in CSR space. All other CSR registers are emulated in software: Interrupts are generated for each Fastbus data cycle not implemented in hardware, and an interrupt routine supplies the required functionality.

FASTBUS SOFTWARE

The NIM Fastbus Standard Routines [10], developed from the Fortran-compatible Los Alamos implementation for the GPM [11], have been ported and optimised for the CHI processor. The Fortran compatibility has been maintained. New features of the CHI-P, such as individual line handling and Fastbus Interrupt Message reception have been added to the routines. Even with full parameter checking, the routines provide much improved speed as compared to the previous

CERN Fastbus Interface CFI. The resident Fastbus library can be used in programs executing on the host computer, by Remote Procedure Call.

Making use of the CERN Cross software running on a VAX computer, user applications can be developed in Fortran, C or Pascal, with embedded Fastbus calls.

CHI-P resident software

The CHI-P resident system libraries [12] fit into roughly 0.5 Mbyte of EPROM memory. With the addition of the PILS language, test routines, and manufacturer's test firmware, a total of 1.25 Mbyte of EPROM are currently used. The libraries consists of RPC libraries, and AST emulator, a CERN stand-alone kernel (MoniCa), the Fastbus standard routines, and CATS libraries with drivers for DRB32, DRQ-11/DRE-11, V24 and Ethernet.

Test and diagnostic features

Menu based test software for Fastbus is contained in the standard EPROM together with a "learn" feature to provide a simple way of composing a series of Fastbus operation without programming. Basic validation tests for the CHI are contained in EPROM and activated during power-up. More complete conformance tests are loaded as PILS programs. The PILS language system allows both compiled and immediate program execution, and easy evolution of test programs. The standard Fastbus line handling routines are implemented for diagnostic backplane tests from high level languages.

As Fastbus key address cycles and operations are executed as individual instructions of the MC68030 microprocessor, breakpoints or single-step utilities of the MoniCa kernel can be used to efficiently debug Fastbus systems on a cycle-by-cycle basis. Other MoniCa utilities display the Fastbus line status, initialise Segment interconnects, or perform a Reset Bus. Latched and non-latched diagnostic bus status registers are available for 'bus-snooping' purposes.

CHI-P APPLICATIONS

The CHI is the main host interface in the DELPHI and L3 experiments. In Delphi, a program written in the C language and using the standard Fastbus routines accumulates events and messages in a shared data memory while simultaneously transferring data to a server program running on the remote 8700 VAX (Fig. 5). The optical link is used to interconnect the CHI-P in the underground area to the host on the surface level where data are written onto IBM 3480 compatible cartridge tapes. Data volumes of up to 200 Mbytes/event at several events per second have to be transferred.

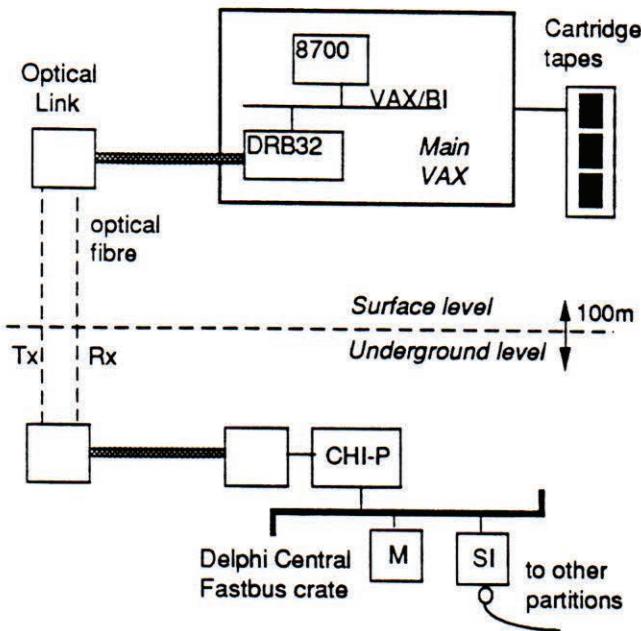


Fig. 5: The CHI in the Delphi experiment

In the L3 experiment, direct CHI to VAX/BI bus connections can run at a sustained 3 Mbyte/s data rate, with readout procedures written in Fortran. A 10 km optical link, transferring data at roughly 1 Mbyte/s has also been established between a VAX and a HVIOR VME interface, connected to an IBM 3090.

The WA89 experiment at CERN uses a CHI-P with SCSI interface to emulate a magnetic tape device. This is connected to and read out by a fast workstation.

Test systems are being set up at CERN consisting of a personal computer, host computer and a CHI-P running PILS.

CHI-P as stand-alone processor

The CHI-P Fastbus board has all features required for embedded processing in Fastbus, i.e. general purpose master and slave ports with Fastbus Interrupt Message receivers. Its fast I/O, large buffer space and the possibility to generate any Fastbus operation are available to collect, dispatch and process Fastbus data very efficiently. Applications as trigger processing can be envisaged, in particular because of the computing power of the MC68030 and its floating point coprocessor. In addition, a number crunching coprocessor with direct access to the data memory via the host port could be designed for such purposes.

CHI performance

The CHI-P executes Fastbus cycles asynchronously at the speed of individual MC68030 instructions or, when using autonomous DMA, at the speed limit of the data memory in use. With the standard 100 ns data memory, 20 Mbytes/s of data transfer rates can be generated by the CHI master port on the local segment. The slave port can handle rates up to 15

Mbytes/s. Test routines, written in assembly language without any checking can execute in less than 2 μ s on Fastbus with up to 4 μ s repetition, whilst standard NIM routines with full checking require 6 μ s bus occupancy for FRD/FWD operations with repetition rates in the order of 75 μ s.

CONCLUSION

The CERN Host Interface concept provides a unified way of using powerful processors in both Fastbus and VMEbus which can execute large application programs locally. Their communication with a host computer is based on bidirectional data transfers and remote procedure execution. Several types of link are available with many options for speed and distance and compatibility with different systems. The CHI hardware and software provide a complete and fully tested family.

REFERENCES

- [1] McLaren, R.A. et al., "The CERN Host Interface", IEEE Trans. Nucl. Sci. Vol 35, No 1 (1987), 321
- [2] Müller, H., "The evolution of the General Purpose Fastbus Master", IEEE Trans. Nucl. Sci. Vol 35, No 1 (1987), 324
- [3] McLaren, R.A. et al., "The CERN Host interface and optical link", Proc. Internat. Conf. on the Impact of Digital Microelectronics and Microprocessors on Particle Physics", Trieste, (1988), 280
- [4] Burckhart, D. et al. "Software support for the CERN Host Interface", Proc. Internat. Conf. on the Impact of Digital Microelectronics and Microprocessors on Particle Physics", Trieste, (1988), 280
- [5] Berners-Lee, T.J., "Experience with Remote Procedure Calls in Data Acquisition and Control", Proceed. 5th Conf. on Real Time Computer Appl." San Francisco 1987
- [6] International Organisation for Standardisation, "Information Processing Systems - Open Systems - Transport Service Definition" ISO/DIS 8072
- [7] Berners-Lee, T.J., G. Heiman 'The CATS Transport Calling Sequence" CERN document RPC/USER/CATS
- [8] Perrin, Y. et al., "The VALET-PLUS embedded into large physics experiments" Proceed. ESONE "VMEBUS in Research", Zürich 1988
- [9] Russel, R.D. et al., "PILS a portable, interactive language system", Proc. 4th Biennal Conference on Real Time Computing, Chicago (1985)
- [10] U.S. NIM Committee, "Fastbus Standard Routines", U.S. Department of Energy document DOE/ER-0367, May 1988
- [11] Kozlowski, T. and Foreman, W.M., "An implementation of the new IEEE Standard Routines for Fastbus", IEEE Trans. Nucl. Sci. Vol 34, No. 4 (1987)
- [12] Divia, R. et al "The CHI, software aspects", Minutes of the 30th meeting on Fastbus developments, CERN EP