CERN-Data Handling Division
DD/84/17
E.M. Rimmer
October 1984

FASTBUS SOFTWARE IN EUROPE

(Invited Paper: IEEE 1984 Nuclear Science Symposium,
Orlando, Florida, USA, 31 October - 2 November, 1984)

DD-emr

FASTBUS SOFTWARE IN EUROPE

E.M.Rimmer

DD Division, CERN
1211 GENEVE 23, Switzerland

ABSTRACT

A large part of the European High-Energy Physics community, particularly that engaged with LEP, the future CERN Large Electron Positron Collider, has adopted FASTBUS as an instrumentation standard. Therefore, the focus of long-term FASTBUS support for hardware and software is at CERN, with end-users located both on-site and at collaborating institutes. The variety of Host machines and FASTBUS interfaces used, and the required mobility of the systems between development in home laboratories and ultimate incorporation at the experiment, has necessitated the maximum exploitation of standards wherever possible.

Introduction

The acceptance of the FASTBUS hardware standard [1] and draft software standard [2] by a large part of the European High Energy Physics community has resulted in the need to provide support for the hardware and associated software. Although the focus of this support is at CERN, teams at other European laboratories, such as LAPP-Annecy, LAL-Orsay, CEN-Saclay, the Rutherford Appleton Laboratory and the Swiss National Laboratory SIN, are developing FASTBUS systems with local support.

A major goal is support for LEP experiments scheduled for 1988/1989, however, present experiments and LEP test beam systems provide pilot project environments for module development and assessment, enabling a steady evolution towards a sophisticated use of FB.

To accelerate the adoption of FASTBUS and maximise user experience, and with only a small support staff, between 1 and 3 for FASTBUS software at CERN, it was decided to

exploit existing investment in standards such as CAMAC and FORTRAN, and to provide portable general purpose FASTBUS software. The support team and some of the end-users themselves have worked together towards this goal with considerable success. Given a solid software base, users are now able to concentrate effort on application dependent hardware and software.

System Hardware

In FASTBUS, the basic software component needed is for interfacing. Therefore, a summary of the present interfacing hardware used in Europe is given.

Host Machines

There is a clear trend towards using the VAX family for on-line data acquisition and the M68000 family for test systems and front-end processing. Despite this change to 32-bit Hosts, support is still required by many groups using PDP-11s or Norsk Data 100 16-bit machines. Therefore, software must be portable across this range.

Intermediate Bus

The above Hosts already had CAMAC interfaces supported by the standard NIM/ESONE CAMAC Routines [3] and, in the case of the mini-computers, by CAMAC Online Data Acquisition systems. Therefore, the CAMAC Dataway has been used in the short-term as an intermediate bus between the Hosts and FASTBUS, with CAMAC I/O registers coupling the Dataway to FASTBUS interface boards.

This approach has the added advantage that replacement of CAMAC registers by other I/O registers entails minimum modification to the FASTBUS Routines code, provided that driver software is available for the alternative registers. One of the CERN M68000 systems is housed in a VME crate, and a VME I/O register with emulated CAMAC calls has been implemented [4]; the conversion required no modification to the FB interfacing software. Discussions are underway to use UNIBUS or Q-bus DR11-type registers to bypass CAMAC on VAXs in a similar way.

FASTBUS Interfaces

Three FASTBUS Host Processor Interfaces are supported on all of the above Hosts; the FIORI (IORFI) FASTBUS I/O Register Interface [5]; the intelligent buffered CERN FASTBUS Interface CFI which has an on-board M68000 [6]; and a Fast Diagnostic Sequencer FSEQ [7]. In addition, a software interface, written as an educational and diagnostic aid, generates FASTBUS timing diagrams on a graphics terminal in lieu of executing FASTBUS operations [8]. The FASTBUS Routines are also being implemented for a M68000 based stand-alone General Purpose Master GPM [9].

A total of about 30 FIORIs and 20 CFIs will have been delivered by the end of 1984, which indicates the

magnitude of the support load.

Portable FASTBUS Routines

At the implementation level of the FASTBUS Routines, the aim was to achieve Host independence. To minimise the development and maintenance load, as much interface independence as possible was also desirable. The execution inefficiencies incurred by this approach can be reduced, where necessary, by recoding critical parts of the software.

The implementation is based entirely on List Processing techniques [10, 11]. The code is written in a subset of FORTRAN-77 and uses standard CAMAC Routines, emulated for VME. It consists of a host independent List Compiler plus a set of interface dependent but host independent List Processors. The Compiler generates intermediate code, named F-Code, from user calls to the Routines. F-Code Lists define sequences of FASTBUS operations which are interpreted and executed by List Processor software or firmware.

Because of the word-size and long-word/word addressing differences in the Host machines, target code cannot be completely Host independent. These slight differences are handled by a primitive Source Code Manager operating on unique input files. This will be replaced by a CERN-supported Code Manager in the near future.

Portable Application Programs

At the application level, the aim was to achieve both Host and interface independence. This requires that a set of Libraries exists for each Host machine, in addition to the FB Routines; CERN provides such common Libraries for histogramming, menu-handling and so on. These Libraries are not all portable, but

have identical interfaces to user programs.

The CERN implementation of the FASTBUS Routines can be called from any language with FORTRAN-compatible parameter passing. However, to achieve portability of application software, it was necessary to adopt a common interactive language for test systems and a common compiled language for production programs.

PILS, a Portable Interactive Language System [12], written in PASCAL, is used for test programs. PILS has Minimal BASIC as a sub-set and is available on VAX, M68000 and ND Hosts. It is not available on PDP-11s due to address limitations and there are no plans at present to provide a PDP version because of lack of personnel.

FORTRAN-77 is used for data acquisition and production programs. It is foreseen that FORTRAN data analysis programs, traditionally running in mainframes, will migrate into front-end processors to be used for monitoring and event filtering. Therefore, FORTRAN support for microprocessors, including cross-compilation and down-loading facilities, will continue to be necessary, even when time-critical program sections are coded in Assembler. The lack of a full FORTRAN-77 compiler for the PDP-11s means that a sub-set of the language must be used if programs are to be ported to these machines.

Programs using the FASTBUS Routines can drive any of the above interfaces without modification. Routines which are irrelevant for a particular interface are transparent, and errors are reported only if the interface in use cannot perform the specified operation; for example, the CFI is unable to access FASTBUS signal lines individually for diagnostics. Interface selection can be

made either at link-load time or dynamically at run-time. This means that a user can switch with ease between, say, the slow FIORI and the fast Sequencer for hardware debugging.

Some examples follow of portable packages which have been produced using the FASTBUS Routines.

FASTBUS Monitor program

A menu-driven program FBMON [13], written in FORTRAN, enables user to generate legal and illegal FASTBUS sequences for execution by any selected interface without the necessity of writing code. This program has had great success with design and maintenance engineers.

FASTBUS Display Module test

Another menu-driven FORTRAN program FDMTST [14] uses a CERN Display Module [15, 16] to check-out a FASTBUS crate for integrity of the bus, correct functioning of the protocols and so on. The kernel of this program is foreseen as part of future automated check-out during system initialisation.

SI test suite

The suite of SI test programs originally written at the University of Illinois in the FASTBUS Diagnostic Language FDL [17] have been translated into PILS for European consumption. This proved to be relatively easy because the FDL FASTBUS primitives, which do not conform to the FB software standard, map directly onto FASTBUS Routines which execute the corresponding single F-code instructions.

Data Acquisition

The pilot experiments which used FASTBUS in online data acquisition systems were those of the European

Muon Collaboration at CERN [18] and the SINDRUM Group at the Swiss National Laboratory SIN [19]. Both experiments, running in 1982/83 with IORFIs and PDP-11s, had fast trigger processors housed in FASTBUS. In the case of EMC, an IORFI was used exclusively for system initialisation, with the read-out performed via a CAMAC/ROMULUS system; in the case of SINDRUM, the IORFI was used both for initialisation and read-out.

The first use of the CFI in a data acquisition environment was with VAX-750s at the CERN NA31 experiment and at the ALEPH test beam during the summer of 1984. These systems were commissioned in collaboration with the DEC section of the DD Online group and with the FASTBUS interface section of the DD Experimental Development group [20, 21, 22].

The FASTBUS Routines were used in FORTRAN programs to generate event-building F-code Lists. The integrity of the Lists and their operation was verified by loading them into the CFI M68000 memory and executing them, after which the Lists were either archived on disc or left in the CFI memory. At run-start, the Lists were read from disc if necessary and loaded into the CFI without execution. On receipt of the CAMAC event interrupt, the Lists were activated by a call to the FB List Execution Routine in the case of NA31, and by CAMAC commands issued by the standard CERN VAX CAMAC data acquisition driver in the case of ALEPH.

The protocol used to dialogue between the Host resident part of the CFI List Processor and the on-board part in the M68000 has now been fully tested and stabilised. The DEC Online section plan to produce an upgraded CAMAC driver incorporating this protocol, to improve the CFI

performance by reducing system overheads.

Multi-Processing

Apart from tests and data acquisition using the standard FASTBUS Routines, another important FASTBUS software project is underway at CERN by a group from the University of California at Los Angeles. The VIRTUS project [23] uses an array of GPM M68000 modules in FASTBUS to perform parallel processing of physics events. The event data can be collected from FASTBUS front-end modules or injected into the system from a magnetic tape in off-line mode. The Host processor is a VAX 11-750, and it is interfaced to the FASTBUS via a DR11W register driving a modified GPM. The VAX software is written in FORTRAN and the M68000 software in Assembler.

Data Flow Studies

Some preliminary computational studies have been made of methods to treat data flow and dead-time problems through FASTBUS systems which contain processing elements and associated data buffers. Three separate approaches have been taken; the use of the IBM General Purpose Simulation System GPSS [24]; the use of formal queuing theory within a specially-written FORTRAN program [25]; and an investigation of using the ADA rendezvous to simulate real-time events [26].

Future Plans

The CERN FASTBUS software support team plan to start development of FASTBUS system management software during 1985. Two major projects are being launched; one to define and implement a set of lightweight message protocols for use between distributed processors in FASTBUS; the other to build up database support for FASTBUS system configuration and

initialisation. This database will eventually be integrated with the on-line data acquisition systems.

In addition, initial studies are underway to specify a second generation high performance FASTBUS interface coupled to a 32-bit VAX bus.

Conclusions

The economics of exploiting hardware standards via accompanying software standards has, once again, been well demonstrated. Development and maintenance requirements are kept to a minimum and limited resources can be usefully expended to tune the systems in critical areas.

FASTBUS in Europe is now well supported by portable basic software which will consolidate FASTBUS use over an increasing range of applications. User groups are able to concentrate their efforts on software specific to their experiments. These user programs and the overall FASTBUS system management support software will form the main fraction of our work in the immediate future.

Acknowledgements

The FASTBUS Routines were implemented by C.Bizeau, D.Burckhart, R.Divia and E.M.Rimmer. We thank all of our colleagues who have participated in our discussions and provided many useful ideas. It is a pleasure to offer special thanks to P.Baggia, Y.Bertsch, P.Gavillet, H.Mueller, K.Peach, R.Ritter, C.M.Story and D.Yaeger.

References

- [1] U.S.NIM Committee, FASTBUS, A Modular High-Speed Data Acquisition System for High-Energy Physics and other Applications, DOE/ER-0189, December 1983
- [2] U.S.NIM Committee, Standard

Routines for FASTBUS, Draft, September 1984

- [3] ESONE and NIM Committees, Subroutines for CAMAC, ESONE/SR/01; DOE/EV-0016; IEEE Std.758

- [4] C.F.Parkman, VIOR, VME Bus Compatible Dual 16 Bit I/O Register, CERN:DD MICRO-VME/N9, June 1983

- [5] C.F.Parkman and E.M.Rimmer, FIORI, CERN FASTBUS-to-I/O Register Interface, CERN DD:FB/MAN3, October 1983

The FIORI is a CERN derivative of the IORFI; C.Logg and L.Paffrath, SLAC

- [6] D.Jacobs et al., CFI, CERN FASTBUS Interface, CERN DD:FB/MAN1, December 1983

- [7] G.Fremont, EP Division, CERN; report in preparation

- [8] R.Divia, DD Division, CERN and Istituto di Scienza dell'Informazione, Universita di Torino, Italy; report in preparation

- [9] H.Mueller, The GPM, a General Purpose Programmable Master, CERN:EP Electronics Note, May 1984

- [10] E.M.Rimmer et al., CERN Software for FASTBUS, CERN DD:FB/MAN2, August 1984

- [11] E.M.Rimmer, CERN's Approach to FASTBUS Software, Proceedings of the 3rd Biennial Conference on Real-Time Computing in Nuclear and Particle Physics, IEEE Trans. on Nucl. Sci., NS-30, No.5, October 1983, pp.3968-3971

- [12] R.D.Russell, D.O.Williams and L.J.Tremblet, PILS Reference

- Manual, CERN DD:GEN/MAN2, July 1984
- [13] Y.Bertsch et al., Laboratoire de Physique des Particules, Annecy-le-Vieux, France
 - [14] C.M.Story, FASTBUS Diagnostic Program, CERN:DD FB/NOTE2, August 1984
 - [15] H.Mueller, The FDM, a Diagnostic FASTBUS Display Module, CERN:EP Electronics Note 82-07, July 1983
 - [16] H.Mueller and D.Burckhart, FASTBUS Display Module for Segment Diagnosis, Interfaces in Computing, 2 (1984), pp.241-248
 - [17] D.Lesny, FASTBUS Diagnostic Language Users Manual, COO-1195-478, Department of Physics, University of Illinois, April 1983
 - [18] L.Gustafsson et al., Proc. of Information Meeting on FASTBUS, CERN-EP Electronics Note 82-03
 - [19] W.Bertl et al., Proc. of FASTBUS Interest Group Meeting, CERN-EP Electronics Note F393, June 1984
 - [20] K.Peach and members of the CERN NA31 experiment; private communication
 - [21] D.M.Sendall and members of the CERN DD Online Computing DEC section; private communication
 - [22] R.McLaren and members of the CERN DD Experimental Development Group; private communication
 - [23] J.Ellett, R.Ritter, D.Yaeger and J.Zweizig, VIRTUS Software Design Document, UCLA/VIRT/8407, October 1984
 - [24] J-F.Renardy, DPhPe, CEN-Saclay, France; private communication
 - [25] B.Bouquet, LAL, Orsay, France; private communication
 - [26] A.Pellier, LAPP, Annecy-le-Vieux, France; private communication