Fermi National Accelerator Laboratory

FN-387
2330.000

FASTBUS SOFTWARE REVIEW*

R. Pordes

May 1983

*Paper submitted to the Third Biennial Conference on Real-Time Computer Applications in Nuclear and Particle Physics, May 16-19, 1983, Berkeley, California 94720

FASTBUS SOFTWARE REVIEW

R. Pordes *
Fermi National Accelerator Laboratory
P.O. Box 500, Batavia
Illinois 60510

Abstract

This paper presents a summary of progress in the definition and implementation of general purpose software to use the data-bus system FASTBUS.

It concentrates on the topics considered and developed by the FASTBUS Software Working Group, and implementations of software at the institutions represented in this group.

Introduction

This paper gives an overview of some of the developments in software for FASTBUS since the last Real-Time Computer Applications in Particle Physics Conference in 1981. It will cover much of the same material as presented by D. Gustavson in his paper "FASTBUS Software Progress" at the IEEE Conference in October 1982 (1), with some difference in emphasis. Other papers to be presented at this conference will concentrate on specific implementations and uses of existing FASTBUS software (2).

The FASTBUS Software Working Group (FSWG) has provided a forum for the discussion of software related topics in FASTBUS and for the coordination and communication of software developments in the United States and Europe. The FSWG has reviewed the FASTBUS Hardware Specification (3) at all stages. It has made recommendations, many of which have been adopted, to the hardware committee on the definition of error-status codes, definitions of control-status registers and the internal format of FASTBUS Interrupt Messages.

Standard Routine Specifications

The specifitations for Standard Routines for FASTBUS (4) proposed by the FSWG are ready for publication under the auspices of the NIM FASTBUS Committee as a "trial-use" standard. The document defining and describing the standard is available in its pre-publication form (5).

The specifications provide a common vocabulary and syntax for implementing a FASTBUS Software Package. The aim is to avoid unnecessary and unproductive differences in software at

---

* Work supported by the U.S Department of Energy, Contract DE-AC02-76ZH03000.

different installations. It is recognized that differences in computers and FASTBUS hardware interfaces make it impossible to standardize the software in complete detail. The specifications are, however, designed to be as computer and language independent as possible.

The draft specifications of eighteen months ago have been implemented on several computer systems using a variety of FASTBUS interfaces. At Brookhaven the implementation was for a VAX-11/780 using the DDI FASTBUS Interface (6); at CERN the implementation was through CAMAC to the IORFI FASTBUS Interface (7); and at Fermilab the implementation was for a PDP-11 RT-11 System using the UPI FASTBUS Interface (8). CERN has converted to the present pre-release standard to interface to FASTBUS. Fermilab is now implementing the pre-release version for a VAX-11 computer using the UPI.

The implementations have all provided access to the FASTBUS operation routines from FORTRAN and typically languages such as BASIC or PASCAL. The implementations are described in more detail in other papers (1,2). FASTBUS systems containing many segments will require further software development. It is expected that this will lead to additions to the standard specifications in order to provide a standard interface to the software procedures needed. For example, a standard routine and parameter set will be defined for obtaining information on the topology and logical address assignments of the FASTBUS system.

FASTBUS Software Resource Guide

Members of the FSWG are collaborating on the writing of the FASTBUS Software Resource Guide (FSRG) (9). This document contains a general description of all software procedures needed in a FASTBUS system, followed by a brief description of all existing general software and software-related hardware modules (for example computer-FASTBUS interfaces) that are known to members of the group. It also includes a glossary of FASTBUS terms and a bibliography of publications.

The guide is intended to help in the communication of FASTBUS software knowledge and already existing software tools. It is to be published as an informal reference document rather than as a formal standard, and it is planned to make frequent updates to it.

The FSRG is presented in greater detail in a poster paper submitted to the conference by R. M. Brown et al. (10).

Segment Interconnect

FASTBUS crates and cable segments are connected by Segment Interconnects (SIs). Each SI connects a crate to a cable FASTBUS segment, and has a route-table and a set of Control-Status registers for each 'side'. A FASTBUS address may be passed through an SI unchanged (untransformed) or with the high-order Group Field changed (transformed) in a way specified by the SI's route table.

A system including four prototype SIs was tested in two different topologies at the University of Illinois, Urbana-Champaign. Software procedures for initializing such a multi-segment system have been written and tested. This included:

1. Downloading of route maps for each side of each SI,
2. Enabling the SIs to recognize and pass FASTBUS addresses (in both non-transforming and transforming modes),
3. Placing the SIs in the "route-trace" mode to perform system diagnostics.

The tests included software controlled FASTBUS transactions which passed through several SIs en-route between a Master on one segment and a Slave on another. This software is reported in more detail in a poster paper submitted to this conference by D. D. Lesny et al. (11).

Diagnostics

Although special hardware modules are being designed to aid in detecting and diagnosing FASTBUS system errors, much of the burden and emphasis must rest on software diagnostics. A powerful, flexible, easy to use diagnostic program will be a necessity in tracing system errors in multi-segment FASTBUS systems. The development of software diagnostic tools that will be equally effective for a single segment FASTBUS system and for a more complex multi-segment system has been a major goal of the FASTBUS software development.

In general the diagnostic programs that have been developed provide facilities for the control of single FASTBUS lines, for the performing of single FASTBUS cycles and complete operations, and for the definition and execution of sets of compound operations.

The SLAC Forth-FBDOS system (12) has been used to test the FASTBUS protocol and debug and test modules such as the SLAC Snoop Diagnostic Module (13). CERN has embedded calls to their implementation of the standard routine specifications in a version of BASIC (14). The University of Illinois has developed a "BASIC-like" language, The FASTBUS Diagnostic Language (FDL), whose vocabulary explicitly includes FASTBUS operations and whose syntax is matched to the parameters needed to control FASTBUS (15).

FASTBUS Interrupt Mechanisms

The FSWG has been concerned with the definition and use of the two FASTBUS Interrupt Mechanisms. It has helped to define procedures to follow when a computer interface detects the Service Request (SR) line asserted (the 'OR' of SR from all the modules on a segment).

A FASTBUS Interrupt Message is the transfer of one or more data words to a special block of registers in a Slave module. To help provide some uniformity in such messages the FSWG is proposing a recommended format for the first three data words of a FASTBUS Interrupt Message.

System Management Software

In any FASTBUS system containing several segments, software is needed to perform the following system management functions:

1. Provide a detailed system description. This includes the system topology, the number and type of segments, modules and SIs, the address space requirements of each module and segment, and the connections required in the system.
2. Generate SI route tables, module logical address assignments and broadcast tree structures.
3. Perform the actual system initialization and verification.

A paper by S. Deiss describing a prototype FASTBUS System Manager is published in the IEEE Proceedings from the October 1982 Nuclear Science Symposium (16). Based on this experience, another implementation is now in progress at Fermilab (17).

System management software for FASTBUS will soon see use in real multi-segment systems, and will be developed further in the next year.

Summary

The next few years will see the development of more complex multi-segment FASTBUS systems. Much software development will be needed to control and diagnose them.

With the continuing commitment from the member institutions, the FASTBUS Software Working Group can provide coordination for the important functions of collecting, discussing and disseminating of software knowledge and resources for the growing FASTBUS user community.

Acknowledgments

This paper gives a summary of the work and contributions of many people. A list of those who have contributed to the work of the FSWG is given in D. Gustavson's paper, FASTBUS Software Progress (1).

References

1. FASTBUS Software Progress, D. Gustavson, IEEE Transactions on Nuclear Science, Vol NS-30, February 1983. See also other papers on FASTBUS in these proceedings.
2. See papers by J. A. Appel and W. K. Dawson, G. Benenson et al., et al., D. D. Lesny and J. J Wray.
3. FASTBUS Specification, November 1982, U. S. NIM Committee available from L. Costrell Department of Commerce, National Bureau of Standards, Washington D. C. 20234.
4. Specifications for Standard Routines for FASTBUS, FSDG085, April 1983, R. Pordes (Editor), FSWG, for availability see reference below.
5. Available from L. Costrell, Department of Commerce, National Bureau of Standards, Washington D. C. 20234, or from R. Pordes, Fermilab, MS 120, P. O. Box 500, Batavia, Illinois 60510.
6. FASTBUS Host Interface, E. J. Siskind, IEEE Transactions on Nuclear Science, Vol NS-30, February 1983.
7. CERN's Approach to FASTBUS Software, E. M. Rimmer, submitted to this conference.
8. Standard Software Routines for FASTBUS and their implementation for a PDP-11 RT-11 system using the UNIBUS Procesor Interface, R. Pordes, IEEE Transactions on Nuclear Science, Vol NS-30, February 1983.
9. FASTBUS Software Resource Guide, R. M. Brown (Editor), FSWG, COO-1195-480, Available from R. M. Brown, Loomis Laboratory of Physics, University of Illinois, Urbana-Champaign, Illinois 61801.
10. The FASTBUS Software Resource Guide, R. M. Brown et al., submitted to this conference.
11. Using FDL to test a Multi-Master Multi-Segment FASTBUS System, D. D. Lesny et al., submitted to this conference.
12. FASTBUS Diagnostic Operating System (FBDOS), August 1982, C. Logg, SLAC, P. O. Box 4349, California 94305.
13. Progress on the SLAC SNOOP Diagnostic Module for FASTBUS, H. V. Waltz et al., IEEE Transactions on Nuclear Science, Vol NS-30, February 1983.
14. See paper submitted to this conference by E. M. Rimmer, CERN.
15. FASTBUS Diagnostic Language, UIUC COO-1195-478 D. D. Lesny, Loomis Laboratory, University of Illinois Urbana-Champaign, Illinois 61801
16. Software for Managing Multi-Crate FASTBUS Systems, S. R. Deiss and D. B. Gustavson, IEEE Transactions on Nuclear Science, Vol NS-30, February 1983.
17. FBN015, FBN017, Unpublished, D. B. Burch, S. Gannon, 60510. R. Pordes, Fermilab, P. O. Box 500, Batavia, Illinois