FASTBUS SOFTWARE PROGRESS*

David B. Gustavson
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

The FASTBUS Software Working Group has met several times since the 1983 Nuclear Science Symposium, including a meeting just preceding this conference. This paper summarizes the decisions taken and progress made, including the results of an ambitious summer program of experimentation with new ways of implementing the draft standard subroutines for FASTBUS.

Introduction

FASTBUS is entering a new stage of life, moving from development to application in the laboratories and spreading into industry. As this happens, special development funding decreases and further development work has to be supported by the "customer", typically a laboratory experiment. While this progression is both normal and desirable, it is difficult to keep long-range strategies and goals in mind while facing the usual tight schedules and budgetary problems of modern experiments.

Despite these problems, there has been some progress toward long-range FASTBUS goals this year. For example, thinking about Buffered Interconnects has begun; the IEEE Computer Society has completed work on a standard for relocatable object code (IEEE P695) which may be useful for solving the multiprocessor system-wide linking problem; and a major effort for implementing FASTBUS standard subroutines in an experiment environment has resulted in a thorough reexamination of the proposed subroutine standard.

Standard Subroutines

A draft proposed standard for subroutines interfacing to FASTBUS has been in use for more than a year, and has been distributed along with the hardware specification. Editorial efforts are underway to improve the format of the document before final standardization, in particular to make it more concise and easier to maintain by reducing the repetition of information. Meanwhile, members of the Collider Detector Facility project at Fermilab noted some problem areas when trying to implement the routines in a multi-user multi-tasking environment.

The CDF members generated a proposal for changes to the standard routines which would eliminate the observed problems while retaining the basic flavor of the existing draft. In order to evaluate this proposal in a timely way, a scheduled editing subcommittee meeting on May 15-16 at Fermilab was changed (on rather short notice) to a full FASTBUS Software Working Group meeting. After examining the proposal, the Working Group decided to proceed on two parallel paths through the summer.

In order not to delay progress toward formal standardization, Ken Dawson and an editorial subcommittee continued work on the existing draft format problems, making few technical changes, while an experimental implementation of the proposed revisions was made and tested at the CDF with collaboration from other interested parties.

Results of both efforts were distributed to the full Working Group in September, for consideration at the Working Group meeting held immediately preceding this Nuclear Science Symposium, on October 29-30 in Orlando. The Working Group faced a dilemma: the proposed revisions, as evolved, appear very desirable but result in significant changes, which will delay the completion of standardization of the routines. Furthermore, existing implementations will require significant reworking to accommodate the revisions. User (application) programs will also require some changes. Should there be two standards, or one standard now followed by a major revision in a year, or should there be one standard but produced six months later?

It was pointed out in the discussions that any such standard is very ambitious, because it tries to cover an enormous multi-dimensional range of applications: 1) Machine resources varying from microcoded controllers through microprocessors to minicomputers or supercomputers. 2) Environment varying from single crate testbench to multiple crate, multiple user, multiple task systems. 3) Interfaces of drastically different architectures, from programmed I/O (IORFI) through DMA (DDI) to list-driven controllers (UPI). It is probably not possible to solve all problems well. On the other hand, success can be had in various degrees, varying from the increased programmer portability due to common naming conventions, to complete portability of application code.

After deliberation, the group decided to move as rapidly as possible to implement the revisions before issuing a single standard. The present draft document will continue to be distributed, but with a warning. The CDF implementation is largely portable, and will be available to other implementors. FASTBUS users who need subroutines immediately are advised to use the existing draft version, while those who can wait six months or a year should plan on using the revised version. There was general agreement that we could afford to make a change of this sort once, but only once, and only if we can do it quickly enough.

The goals of the revisions are: 1) Improved program portability. 2) Multi-user capability. 3) Additional buffer addressing modes. 4) Greatly enhanced error reporting.

The most obvious change seen by the user is that system control structures will be hidden. In the present draft, the user was urged not to access the structures directly, but he had to allocate the memory for them in his own program. The revised version allocates them internally and hides them from the user. This helps to prevent implementation-detail differences (perhaps caused by different interface hardware) from showing up in user programs and reducing their portability.

Another change is in the handling of interface parameters. The present draft expected certain parameters to be set once for the given installation and then left alone during subsequent program execution. That mechanism is now felt to be inadequate for real multi-user systems, so the revised version provides for parameter changes to be recorded in sequence in the lists of FASTBUS operations which drive the interface.

For more details of the proposed changes, please contact one of the proposers of the revisions: Dave Hanssen, Jeff Marsh and Catharine van Ingen of the CDF at Fermilab; Steve Gannon of the Computing Department at Fermilab; Dave Sherden of SLAC; and Eric Siskind of NYCB Real-Time Computing, Inc.

The Working Group hopes to have a new draft standard incorporating the technical revisions by Spring 1985. Ken Dawson of TRIUMF is serving as editor for this effort.

Relevant Non-FASTBUS Work

Several activities of the IEEE Computer Society may be of use to us in solving FASTBUS problems. First, the IEEE P695 Relocatable Object Code project has based its work on CUFOM, the CERN Uniform Format for Object Modules. The resulting Microprocessor Uniform Format for Object Modules, MUFOM, is a generalization and extension of CUFOM, and is much more generally applicable than its name indicates. It may be of use in any environment concerned with cross-assembly or compilation, and in particular it may be helpful in solving the problem of resolving symbolic references to FASTBUS device addresses which are shared by multiple processors.

Second, there is a project under the IEEE P896 Futurebus committee, called the Systems Architecture Study Group, which is addressing allocation of CSR space in standard ways, thinking about standardized initialization procedures, and generally doing things similar to the FASTBUS SWG activities. The goal is to make a standard layer of low-level software which will make various buses, such as NuBus, 896, and Multibus-II look alike to user applications. Representatives from these other buses are participating in the effort.

Directly following from this P896 work is a project now backed by the Computer Society Standards Activities Board, to develop a layered model for "closed systems" (buses) analogous to the 7-layer ISO-OSI "open systems" model so prevalent in networking now.

Current FASTBUS Software Activities

Fermilab: CDF is implementing the revised subroutine package on a VAX and on a PDP-11, writing an FDL (FASTBUS Diagnostic Language) to Pascal translator, and writing a FASTBUS segment initializer in FORTRAN. Contact Catharine van Ingen or Jeff Marsh for details. The Computing Department is supporting several experiments, test beams, and detector development with the existing subroutine package, and is collaborating in the CDF effort. It is also adding CAMAC extensions to FDL, provides RTMULTI acquisition systems with FASTBUS extensions, is working on a FASTBUS database manager, and examines manufacturer's modules and processor interfaces. Contact Dave Ritchie for details.

Los Alamos National Lab: has FDL running with a CERN subroutine package supporting the CERN FIORI interface in a module-test application. Experimenting with commercial interfaces. Contact Ron Nelson for details.

SLAC: SLD is collaborating with CDF in the subroutine revision effort. Contact Dave Sherden. Mark-II is using the DDI VAX interface with standard subroutines, has a MicroVAX with IORFI interpreting and emulating the DDI control lists, is hoping for a FASTBUS interface for the 3081/E processor soon, and is programming the SLAC Scanner Processor in FORTRAN (with some assembler language subroutines). Contact Andy Lankford. FORTH test bench support systems and MicroVAX FORTH are available. Contact Connie Logg. Work is proceeding on the Snoop board checkout and operating system. Contact Dave Gustavson.

University of Illinois, Urbana/Champaign: Maintaining and developing FDL. Incorporating Fermilab CAMAC enhancements due to Ruth Pordes et al., in the style of CDL. Planning to port FDL to VAX, implement IORFI drivers in Pascal, possibly try new subroutine standard. Contact Dave Lesny or Jerry Wray. Working on Buffered Interconnects for FASTBUS. Contact Bob Dobinson.

Summary

In spite of a general shortage of manpower, some work with long-term general utility is still being done. Experiments and other FASTBUS "customers" are urged to budget some development support beyond that needed to achieve their short-term goals, for the general welfare. Every effort should be made to improve global efficiency by writing general-purpose programs which can be used by others, to help avoid duplication of effort.

FASTBUS documentation, including copies of the current standard subroutines draft, may be obtained from Louis Costrell, National Bureau of Standards, Center for Radiation Research, Gaithersburg, MD 20899, telephone (301) 921-2518.