FASTBUS SOFTWARE STATUS

SLAC-PUB-2639
October 1980
(I)

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

ABSTRACT

Computer software will be needed in addition to the mechanical, electrical, protocol and timing specifications of the FASTBUS, in order to facilitate the use of this flexible new multiprocessor and multisegment data acquisition and processing system. Software considerations have been important in the FASTBUS design, but standard subroutines and recommended algorithms will be needed as the FASTBUS comes into use. This paper summarizes current FASTBUS software projects, goals and status.

INTRODUCTION

The FASTBUS Software Working Group was created to consider the software implications of the evolving hardware standard and to develop software standards which will facilitate the use of the FASTBUS system<1>. It also serves as a focal point for the coordination of development efforts among the various laboratories.

The major categories of this work are the development of official software standards and recommended practices, coordination of the production of shareable support software of general utility, coordination of simulation and theoretical studies, and provision of software feedback for proposed hardware designs.

Since most of this work is still in progress, little has been published and much of the available documentation is informal and changing rapidly. In order to help the reader find the latest information on a given topic, the relevant institution names are mentioned in the text. The corresponding contact person named below can then be contacted to get the latest information. Copies of working documents and minutes of meetings are distributed to working group members, and to others on request, by R. S. Larsen (SLAC).

SOFTWARE STANDARDS

One of the primary standards needed for FASTBUS is a set of subroutine calling sequences similar to that developed for CAMAC, which provide access to all FASTBUS functions in a standard way from high level languages such as FORTRAN, Pascal, and Basic. Such a standard greatly facilitates portability of useful programs, makes it easier to understand what other people's programs are doing, and facilitates the movement of people from one laboratory to another. In addition, a standard set of FORTH FASTBUS words would be useful, since this language is gaining popularity in the testbench and diagnostic area.

Work is in progress in this area at CERN, Fermilab, the University of Illinois and SLAC.

Another subject of standardization is the assignment of addresses and bits in Control/Status Register space. If uniform conventions for the assignment of these bits were not adopted, management of large systems would be very difficult to automate. Though many modules will have their own special bits with special uses, the goal is to standardize the common functions in a way which is compatible with the requirements of both software and hardware economy. Much of this work has been done and is included in the current FASTBUS specifications, but additions and evolutionary changes will be needed, as well as clarification and refinement of bit definitions.

Some FASTBUS protocols also need standardization, such as the format of interrupt messages and the handling of detected transmission errors. The interrupt format is included in the current specification draft, but our understanding of error recovery is still developing. The general principle is that erroneous data should not be considered to have been transferred, so the failed cycle can be simply repeated. This has difficulties in the case of read cycles, since the source of the data has no way of knowing that it was badly received by the master. The solution is for the master to stop and re-address the slave to reread the data, but this is not possible in the case of slaves which implement destructive readout, such as read-and-clear, some I/O port designs, or FIFO buffers which have no associated internal address. So far the best solution for these devices appears to require a CSR register which saves the last datum read. Since destructive readout devices also cause difficulty in diagnosing problems in large systems, a strong argument could also be made for avoiding such designs.

A data file is needed for management of large systems, in order to keep track of the types and locations of modules, to assign names to them for use by symbolic references in programs (and thus to serve as a kind of global symbol table), and to provide information needed for system initialization, address allocation and route map generation. It may be possible to standardize the format of this file in order to facilitate the creation of a variety of useful utility programs for managing the system. Currently one experimental implementation has just been completed at SLAC. More experience will be required before standardization will be possible.

Standard procedures for accessing the global system information and for linking programs for the multiple processors in the FASTBUS system would be very useful. Some investigation of the problems has been done at the University of Illinois and at SLAC, but real progress awaits a deeper understanding.

The FASTBUS Serial Diagnostic Network will require standard software protocols as well. Message formats and contents will need standardization in order to permit communication among devices from different vendors, and of course the particular implementation of the FSDN in hardware must be standardized in a way which allows the needs of the system to be met. Network studies are under way at LASL and SLAC.

---

*Work supported by the Department of Energy under contract number DE-AC03-76SF00515.

Contributed to the 1980 IEEE Nuclear Science and Nuclear Power Systems Symposium, Orlando, Florida Nov. 5-7, 1980.

SOFTWARE COORDINATION

The FASTBUS Software Working Group serves as a forum for discussion of proposals and plans for work by the various laboratories currently active in FASTBUS. Though the FBSWG cannot itself become directly involved in the actual production of such large quantities of software, it can help avoid unnecessary duplication of effort and may be able to suggest useful directions for willing producers.

These coordination activities consist of: the distribution of relevant documents; informal review of work in progress in the various groups; and topical meetings, the latest of which was devoted to examination of the Unibus Processor Interface under construction at Fermilab. Future work can be foreseen in the areas of special fast processors customized for FASTBUS operation, Event Builders (Fermilab), and a variety of processor interfaces and buffered interconnects (CERN).

A variety of theoretical studies are needed for a better understanding of FASTBUS systems. Two are presently under way, an ISPS (Instruction Set Processor Simulator) simulation being done at CERN and a Simula simulation at the University of Illinois.

Additional work is needed in several areas, especially in performance measurement. It will be useful to be able to measure the traffic in various parts of a FASTBUS system in order to find and eliminate bottlenecks and in order to optimize the system configuration. The information available for this purpose will probably be snapshots taken by the Snoop<2> diagnostic modules, and methods of deducing meaningful traffic estimates from this information are needed.

Other fruitful areas for study include the optimal choice of routes and address allocation given various constraints and combinations of equipment.

SUPPORT SOFTWARE

A package loosely referred to as the FASTBUS General System Software is a major requirement for easy use of FASTBUS. This package is built around the system description file mentioned above, providing the utilities for creation and maintenance of that database, comparing the database with the real hardware, setting up route maps for the Segment Interconnect modules, assigning addresses and priorities, determining the broadcast message paths, and initializing the hardware.

A package with these features is nearing completion at SLAC. It is written in UCSD Pascal and runs on a small LSI/11 computer with floppy disks. This small system is capable of handling several dozen FASTBUS segments, mainly being limited by storage capacity and speed of the floppy disk drives. Memory usage is adapted to the needs of the problem by a simulated virtual memory technique, storing as much needed data in memory as possible and referring to disk storage only when necessary.

Though the hardware and interfaces necessary for checking the initialization algorithms are not yet available, printed simulations are being produced now.

Work on a System Description Language and other parts of the general system software problem is also being done at the Illinois Institute of Technology, under contract to Fermilab.

General purpose diagnostic software is another area of effort. Work is under way at the University of Illinois on the adaptation of the Camac Diagnostic Language to FASTBUS.

Several diagnostic systems based on the FORTH language and operating system are under development. FORTH is being used at SLAC on LSI/11 systems with CAMAC at present, and a general purpose set of "words" (FORTH's analogue to the concept of "subroutine" used in other languages) is being developed to provide convenient control of the SLAC I/O Register Interface to FASTBUS. This system will be ready in time for the initial module tests, which will begin soon.

Another SLAC FORTH project is supporting the Snoop diagnostic module. In addition to the general FASTBUS words of the above system, the Snoop system will be multitasking and will include an interface to the FASTBUS Serial Diagnostic Network and an optional display terminal with mini-floppy drive for storage of diagnostic programs. This system is being designed to provide a convenient human interface<3> to FASTBUS, especially on the test bench or in small systems. With its fast logic-analyzer capabilities it will also be invaluable in troubleshooting.

The multitasking FORTH system for the display terminal is now in operation, running on a Z80, and the FORTH kernel has been written for the Snoop's 68000 processor but not tested yet.

Adaptation of CAMAC data acquisition system software (Multi, Supergram, etc.) to a FASTBUS environment is also being considered, particularly at Fermilab, the University of Illinois, and the University of Michigan.

Support software not directly related to FASTBUS, such as cross compilers, linkers and assemblers for the new microprocessors, is also being developed in various places, notably CERN, Fermilab and SLAC.

CONTACTS FOR FURTHER INFORMATION

CERN:
E. Margaret Rimmer
D. D. Division
CERN
1211 Geneve 23
Switzerland
Tel: Geneve 83.30.45

Fermilab:
Jeffrey A. Appel
Fermi National Accelerator Laboratory
P. O. Box 500
Batavia, IL 60510
Tel: (312) 840-3922

University of Illinois:
Richard M. Brown
Loomis Laboratory of Physics
University of Illinois
Urbana, IL 61801
Tel: (217) 333-0074

LASL:
C. Dwayne Ethridge
Los Alamos Scientific Laboratory
Los Alamos, NM 87545
Tel: (505) 667-2676

University of Michigan:
Carl Akerlof
Physics Department
University of Michigan
Ann Arbor, MI 48109
Tel: (313) 764-9278

SLAC:
David B. Gustavson
Stanford Linear Accelerator Center
P. O. Box 4349
Stanford, CA 94305
Tel: (415) 854-3300 x2863

ACKNOWLEDGMENTS

The following persons have contributed significantly to the work of the FASTBUS Software Working Group during the past year, through personal attendance or written communication:

E. M. Rimmer of CERN

Jeff Appel, Al Brenner, Steve Gannon, Marvin Johnson, Terry Lagerlund and Lou Taff of Fermilab

Tom Christopher, Martha Evens and W. Kabat of the Illinois Institute of Technology

Richard M. Brown, Dave Lesny, Keith Nater and Jerry Wray of the University of Illinois at Urbana

Ken Dawson, Dwayne Ethridge and Dennis Perry of the Los Alamos Scientific Laboratory

Carl Akerlof of the University of Michigan

Steve Deiss, Dave Gustavson, Terry Holmes, Connie Logg and John Steffani of the Stanford Linear Accelerator Center

REFERENCES

1. FASTBUS Tentative Specification, July 1980 with update September 1980, U. S. NIM Committee.

   Status of the FASTBUS Standard Data Bus, R. S. Larsen, invited paper presented at the 1980 IEEE Nuclear Science and Nuclear Power Systems Symposium, Orlando, Florida Nov. 5-7, 1980.

2. FASTBUS Snoop Diagnostic Module, R. Downing and H. Walz, contributed to the 1980 IEEE Nuclear Science and Nuclear Power Systems Symposium, Orlando, Florida, Nov. 5-7, 1980.

3. A "Front Panel" Human Interface for FASTBUS, D. B. Gustavson, T. L. Holmes, L. Paffrath and J. P. Steffani, contributed to the 1980 IEEE Nuclear Science and Nuclear Power Systems Symposium, Orlando, Florida, Nov. 5-7, 1980.