SLAC-PUB-2703
March 1981
(E/I/A)

APPLICABILITY OF THE FASTBUS STANDARD TO DISTRIBUTED CONTROL*

S. R. Deiss, R. W. Downing,† D. B. Gustavson, R. S. Larsen, C. A. Logg, L. Paffrath
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

Abstract

The new FASTBUS standard has been designed to provide a framework for distributed processing in both experimental data acquisition and accelerator control. The features of FASTBUS which support distributed control are a priority arbitration scheme which allows intercrate as well as intracrate message flow between processors and slave devices; and a high bandwidth to permit efficient sharing of the data paths by high-speed devices. Sophisticated diagnostic aids permit system-wide error checking and/or correction. Software has been developed for large distributed systems. This consists of a system data base description, and initialization algorithms to allocate address space and establish preferred message routes. A diagnostics package is also being developed, based on an independent Ethernet-like serial link. The paper describes available hardware and software, on-going developments, and current applications.

Introduction

In late 1976, a group of laboratory physics users in collaboration with members of the U.S. NIM Committee, the standards group responsible for NIM§ and CAMAC¶ in the United States, decided to study the feasibility of a next-generation standard data bus for laboratory data acquisition and control. This effort eventually led to the development of a new standard, FASTBUS, the specification of which is nearing completion. As part of the standardization effort, prototypes are being built and tested as a demonstration proof of the system.

Although FASTBUS was conceived primarily to solve high-speed data acquisition problems, its architects from the outset also recognized the importance of accommodating multiple processors both in specialized data acquisition as well as control applications. In large-scale laboratory systems in particular, these needs often coincide in the same physical system, or as different branches of a connected system. Standardization of hardware and software in such situations in invaluable, as has already been demonstrated by existing standards.

From a control system point of view, the FASTBUS standard hardware and software will support a wide range of system architectures, as well as system diagnostics. FASTBUS consists of a high performance hardware system capable of supporting the latest and most powerful microprocessor and bit-slice processor systems, plus basic software tools for configuration and diagnosing complex connected systems. In this sense, the development effort is attempting to achieve a much higher level of standardization than exists with any current standard. At the same time, simple subsets are both possible and economically attractive.

Future Control System Considerations

Due to the advent of the microprocessor, all future large systems will involve increased intelligence at different levels throughout the system. In some cases this extra intelligence will be relatively innocuous; e.g., in the control of front-end devices, power supplies and machine controls and the like, by firmware-resident microprograms.

At intermediate levels, however, where either sections of a machine or entire sub-systems of a process need to be controlled, the increased intelligence required will usually dictate a multiple-processor solution, where the processors are slaved in some way to each other or to a high level control system supervisor. These intermediate levels require the most specialized hardware, communication interfaces, and diagnostics design. The multiple minicomputers used in current systems can be expected to be replaced by arrays of microprocessors. CAMAC and similar systems have been used quite successfully at these intermediate levels. However, for distributed systems where intercrate communication is needed, FASTBUS offers a more viable alternative.

For the future, then, the trend toward more computing/control capability in much smaller packages argues for a modular hardware system which will not be bandwidth-limited when supporting a large number of subprocessors, or in modes of operation where high-speed burst of data need to be passed through the system to some remote station or host. A likely requirement is the need to sample on-line diagnostic data of a fairly detailed sort, such as waveform information, and to transmit this to a diagnostics supervisor. This type of built-in diagnostics capability will become more common, in order to constantly monitor the health of the system and its critical components, detect degradation in performance prior to actual failure, and take steps to avert subsequent lost operation.

Figure 1 shows a common control system arrangement, which includes a redundant host processor containing all decision-making capabilities, and an array of sub-system slave controllers. Here it has been assumed that an array of crates CAMAC is linked either via serial or parallel data path, and that CAMAC modules provide the parallel interface format to the remote devices.

Figure 2 proposes an alternate arrangement. Here, any processor connected to the main system bus (cable) can in principle initiate communications. At the subsystem level each controlled device is accessed over a serial link, normally a relatively slow link but possibly a higher speed (e.g., fiber optic) link where special

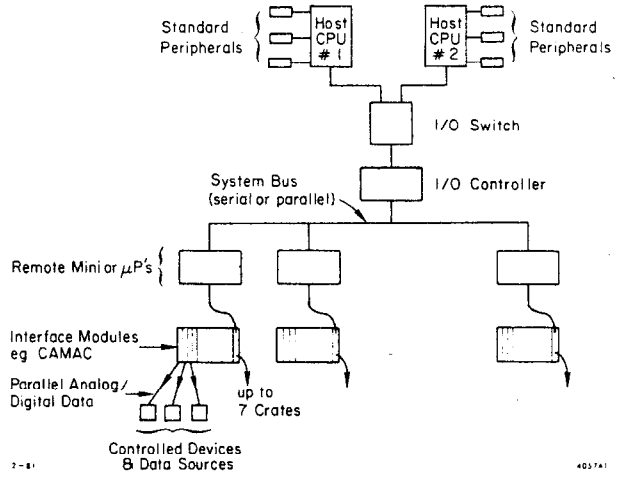

Fig. 1. Centralized Control System Using CAMAC.

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.
† Work supported by the Department of Energy, contract DE-AC02-76ER01195 (University of Illinois).
§ Nuclear Instrument Module Standard, ERDA Report TID20893 (Rev. 4) 7/74.
¶ Computer Automated Measurement and Controls; IEEE Standards 583-1975, 595-1976, 596-1976, 683-1976.

(Contributed to the 1981 Particle Accelerator Conference, Washington, D.C., March 11-13, 1981.)

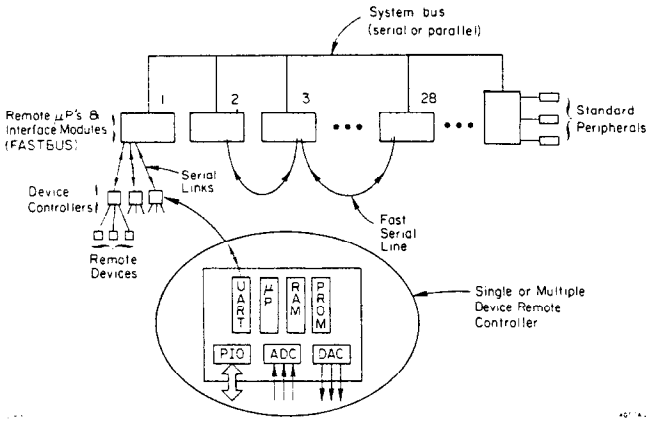

Fig. 2. Distributed Control System Using FASTBUS.

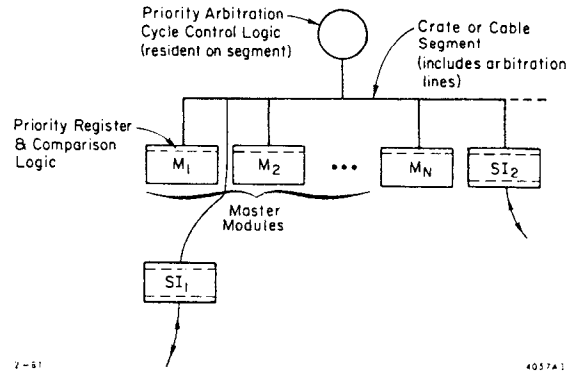

Fig. 3. Bus Arbitration Scheme.

requirements dictate more bandwidth. This serial approach minimizes the physical cable plant. The main architectural feature, however, is that any processor, including the host, can communicate with any other processor over the system bus. (In Fig. 1, the system bus is under the absolute control of the host which would have to manage any such device-to-device data transfers.) Furthermore, in Fig. 2 it is also possible to connect parts of the system which frequently need to communicate with one another, allowing these parts to communicate autonomously while other message traffic is taking place on the main system bus. For example, Station 28 in Fig. 2 is a special diagnostics processor which needs high-speed data in burst mode from Stations 2 and 3, while the main system bus is a high-speed serial data link which supports normal system traffic.

In Fig. 2, each station shown is a FASTBUS crate, or Crate Segment, which has the capacity for a large number of intelligent microprocessors and associated memory, I/O modules, buffers, serial data modules or special interfaces. The FASTBUS design allows autonomous communication within a module, between modules in a crate, or between crates via the system bus. This flexible framework is essential for systems of the future.

FASTBUS Features for Distributed Control

FASTBUS has been described in the literature (see Ref. 1 for the most recent status report and a summary of relevant papers) and in a Draft Specification.[2] Only the salient features will be summarized here; the reader is referred to the references for details.

A. Bus Arbitration System

The most important architectural feature of FASTBUS is its arbitration scheme, which works as follows: Imagine a collection of processors attempting to communicate over the same bus segment as shown in Fig. 3. Each processor is assigned a unique priority vector, or number, represented by a 6-bit binary code. When a processor wishes to gain access to the bus, it waits for the first opportunity when the bus is not busy, and then asserts its priority code on a special set of lines. All processors (master modules) desiring use of the bus do this simultaneously. The competing modules then monitor the lines to see if a higher code than theirs is present. If a higher level code is present, the requesting module withdraws its request. After a very short time, only the highest requesting master module remains, and is granted use of the bus by the Arbitration Timing Control Logic, which resides on the segment. When the winning module is finished its task, it relinquishes the bus, and the priority competition begins anew.

In addition to this priority mechanism, a fairness mechanism is included which can be used to prevent a high priority master from hogging the bus.

The importance of a standardized arbitration scheme in future designs cannot be overstated when one considers the expanding size and complexity of large processing and control systems. All FASTBUS master modules, e.g., processors, will be designed to allow participation in some form of the arbitration scheme. Systems which require more concentrated control of the bus can of course be easily implemented by the choice of arbitration priority levels, or by giving one or more processors absolute mastership over subsections of the system.

B. Bus Structure

FASTBUS is a high-speed, 32-bit multiplexed address and data bus. The speed of the bus, which uses an ECL interface, is designed to accommodate the latest generation of 32-bit microprocessor chips, as well as the special-purpose processors common to laboratory data acquisition or diagnostic applications. The basic speed limitation of the bus and ECL interface is <100 nsec per 32-bit word transfer, in full handshake mode.

The normal FASTBUS random access mode is a full handshake on one or more address cycles followed by one or more data cycles. A second commonly used mode is the block transfer handshake mode. A broadcast mode is defined which allows a number of receiving devices to accept the same message. Geographic (module position dependent) addressing is used for initialization and for a fast scan mode known as Sparse Data Scan. For special applications, nonhandshake operation is possible.

FASTBUS has not yet incorporated a standard protocol for high-speed long-distance serial transmission, although this is under consideration. FASTBUS does, however, include a serial diagnostic link which is designed to operate using an Ethernet-like protocol, at intermediate speeds (~100 KB max). This link is compatible with future Ethernet chip technology now under development. (Ethernet has been adopted by DEC, Intel, Xerox and Hewlett Packard as a future standard for local networks.)

The basic FASTBUS bus element is the Crate Segment, or Crate (Fig. 4). This is a standard EIA rack width enclosure with a backplane containing all address and data, control, arbitration, etc., and power lines for supporting a collection of module devices. Various power and cooling schemes can be accommodated. The serial line is also resident on the backplane.

Collections of devices on a crate segment can communicate with other segments via a full parallel bus (Cable Segment), or via serial link. The parallel link is interfaced via a standard device called a Segment Interconnect (SI). A system interconnection using Segment Interconnects and a Cable Segment is shown in Fig. 5. Also shown is one example of a serial interconnection, referred to as a Buffered Interconnect (BI). This

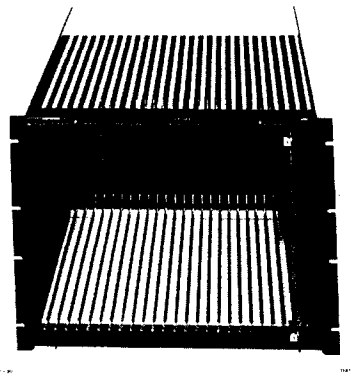

Fig. 4. FASTBUS Crate - Type A.

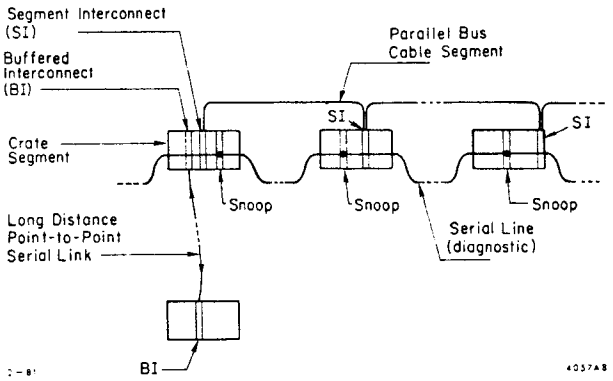

Fig. 5. System Interconnection Using SI's.

particular example shows a point-to-point serial connection. The third bus shown is the serial diagnostic line (SL) which is a true, multi-drop Ethernet-compatible serial bus.

It is also possible to add direct links, or to employ special local networks for segment intercommunication. However, the real strength of the standard will be realized by future standardization of a specific serial protocol and hardware. (This is a secondary goal of FASTBUS, since the parallel segment is of highest interest at the present time.)

C. Bus Control

The parallel bus supports handshake, nonhandshake, random access, block transfer, broadcast, etc., data transfers. The most common mode will be a full handshake address cycle followed by a full handshake data cycle. The normal address cycle can be followed by a second address cycle (Extended Address) in order to access greater than 2^{32} system addresses. All normal addressing is logical; however, geographical addressing via coded lines in the backplane is incorporated for initialization and special scan modes.

The bus structure also contains parity and parity enable lines, implementable at the discretion of the designer. Most control systems will of course include error checking and/or correction.

D. Diagnostics

A very large effort has been expended on the design of diagnostic hardware and software. The Serial Line will initially be used to communicate between SNOOP devices. SNOOP is a special module with a fast front-end history silo, and the ability to insert wait states under program control in order to single-step the bus. A diagnostic microprocessor (M68000), serial link interface, and basic FASTBUS master capability are included. The purpose of SNOOP is to assist both in initial checkout as well as system trouble-shooting of operating systems. SNOOP and its attendant system controller can be made portable and independent of the main system bus.

E. Control and Status Register Standardization

The most basic level of standardization of a data bus is the hardware pin assignments for data and control functional lines. In CAMAC, some of the basic operations were defined in terms of 5 coded lines (F lines), e.g., Read, Write, Read and Clear, etc. However, many common functions were not defined in either hardware or software, which led to incompatibility of even very similar functional modules at both the hardware and software levels.

In FASTBUS, to improve this situation, an attempt has been made to standardize the usage of certain control/status registers, including some specific bit assignments. Control and Status (CSR) space is divided into four general categories: Normal, Program, Parameter, and User CSR space. Normal space contains status and control registers, Program space is a protected program storage area, Parameter space contains status information such as module inventory number, calibration constants, etc., and User space is for free use by the device designer or implementor.

F. Segment Interconnect (SI)

A basic specification has been developed for the SI so that a standard device will be available for intercrate communication. This device has two major capabilities: First, it allows a range of acceptable addresses to be stored on each of its ports, to define allowable data paths between various parts of the system; and second, it contains two-way arbitration circuitry of its own, thus enabling it to pose as a master to the remote Segment from either direction, and to arbitrate for the remote Segment on behalf of the requesting master.

G. Software

Standard software being developed is of two general types. The first is required to initialize the system, including setting up of "route maps" between various parts of the system. Because of the necessity to allocate address space and priorities in the system, software is being designed to perform this task automatically.

The second major software effort is a diagnostics package to support the SNOOP system. This system has two basic uses: One, to trouble-shoot complex connected systems; and two, to support bench testing of FASTBUS modules and subsystems.

H. Other

FASTBUS has many other features of interest to future system designers. One is a versatile power supply system. which accommodates standard voltages of +5, -5.2, -2, and ±15 volts. In addition, a 28V dc line is provided to serve as a bulk supply connection for on-board regulation. The FASTBUS card has a large auxiliary connector space in the rear, to be used for user-defined I/O connections, as well as front panel space.

Conclusion

The FASTBUS development is nearing completion, and commercial availability of basic modules in 1982 appears feasible. FASTBUS offers a viable alternative for future control systems as well as high-speed data acquisition.

References

1. R. S. Larsen, "Status of the FASTBUS Standard Data Bus", IEEE Nucl. Science Symposium, Nov. 1980.
2. FASTBUS - Tentative Specification, U.S. NIM Committee, September 1980.