SLAC-PUB-2859
December 1981
(E/I/A)

INTRODUCTION TO THE FASTBUS STANDARD DATA BUS*

R. S. Larsen
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305

ABSTRACT

The FASTBUS standard data bus is a development of the US NIM§ Committee of the National Bureau of Standards. Although originally conceived to provide an upgraded framework for the high data rate experimental apparatus of nuclear particle physics, the system is designed with sufficient generality to support a broad range of data acquisition as well as control applications. FASTBUS is based on an ECL interface technology and a generalized multiprocessor system architecture. The background and status of the development project are described.

(Submitted to Interfaces in Computing)

---

* Work supported by the Department of Energy, contract DE-AC03-76SF00515.
§ L. Costrell, Chairman, National Bureau of Standards, Department of Commerce, Washington, D.C.

BACKGROUND

The ideas for FASTBUS first began to germinate in 1976, when researchers in several high energy physics National Laboratories† began to anticipate experiments with upwards of 100,000 channels of high speed electronics and many distributed processors, some of them very high speed devices, operating in a single system. The current venerable standard, CAMAC [1] was seen to be restrictive in a number of its specifications, principally addressing range and data transfer speed. After a one-year review process, a committee chaired by the author issued a report [2] recommending the development of a new standard. It was proposed that the new standard have a wide range of basic features; the chief ones are listed as follows:

32 bit wide address and data fields.

Data word transfer speed <100 nsec (10X CAMAC).

Flexible addressing and data transfer modes (handshake and non-handshake).

Increased board size w.r.t. CAMAC.

Modular hardware specification.

Segmentation to support multiple parallel processors (masters).

Flexible architecture to support multiple masters and slaves in star, tree and ring topologies.

Error checking and diagnostics.

---

† Stanford Linear Accelerator Center (SLAC), Fermi National Accelerator Laboratory (FNAL), and Brookhaven National Laboratory (BNL).

Common segment and intersegment architectures.

Standardized power and logic levels.

Standardized, transportable basic software.

Economical mechanics and interface logic.

In 1977, a working committee was organized under the U.S. NIM committee. A large number of the participants had previous experience on the NIM-CAMAC standards committees but many were new people drawn from the National Laboratories, DOE laboratories, and universities in the U.S. and Canada. A special effort was made to include a representative group of physicists and software experts. A close liaison was developed with colleagues in the major European as well as in other foreign laboratories. The original group did not include official industrial participation, the policy being to work closely with the industrial community after the basic formulation of the specification and completion of some prototype testing. The ultimate aim of course, as with NIM and CAMAC, is to develop industrial support for FASTBUS components and systems, and even though the final specification has not been produced, this is already beginning to occur.

The work of the committee has been given financial support from the Department of Energy, Division of Research, and is reviewed by a DOE appointed user committee.¶

¶ Reporting to D. Sutter, DOE.

A FASTBUS PRIMER

FASTBUS, like NIM and CAMAC, is based on the concept of plug-compatible modules. In NIM, the modules are essentially independent and do not share a common data bus. In CAMAC, the modules communicate via a standardized 24-bit TTL parallel backplane data bus. FASTBUS is structured like CAMAC except the data bus is 32-bits wide; is multiplexed to provide a full 32 bit address field as well; is designed as a matched impedance bus ($Z_o \approx 82\Omega$, loaded); and utilizes an ECL transceiver interface. The ECL driver interface is an emitter-OR connection, and the receiver is generally a high impedance base connection. The ends of the bus are terminated by small plug-in cards on the backplane.

BASIC OPERATIONS

The address and data information are transmitted in a variety of possible modes. The most basic mode is the single operation handshake address-data cycle, i.e., a handshake address cycle followed by a handshake data cycle (see Fig. 1). Figure 1 shows the control signals for this operation, namely Address Sync and Address Acknowledge (AS,AK) and Data Sync and Data Acknowledge (DS,DK). The Read (RD) line controls the direction of flow of the data during the data cycle. Other modes are as follows:

Block Transfers

Another mode of operation of particular interest for high speed data transfer is a block transfer read or write. In this case, the controlling master module asserts consecutive data cycles following a

single address cycle (Fig. 2). Such transfers can be handshake, or, for long-line pipe-line types of transfers, nonhandshake; this is determined by the controlling master, which of course must have prior knowledge of the characteristics of the slave device.

Broadcast

A third type of operation, called Broadcast, is an operation which addresses a selected group of devices, and then transmits a data message to the entire group of listening devices. This mode can be extremely useful in preprocessing applications, for example, where the same data are needed by a group of distributed processors.

Control Space and Data Space

Two address spaces are defined in FASTBUS, Control Space and Data Space. Which of these two spaces is accessed depends on the status of control lines, called CL lines. The purpose of this distinction is primarily to protect control functions from accidental access, as well as to facilitate standardization of certain control and status register (CSR) functions. Implementation of CSR space allows extended addressing of 2^{32} control/status registers per device.

Extended Address

Extended address is basically a second address cycle immediately following the first which allows additional address space to be defined within a given module (device).

Mixed Operations

FASTBUS also defines mixed operations such as Read - Modify - Write. In this case, for example, after a normal Address and Read cycle, the RD line would be changed to reverse the direction of data flow, and the

modified data rewritten without having to readdress the device (see Fig. 3).

The above operations illustrate the generality of Address and data manipulations in FASTBUS. The bus itself is therefore relatively easily adapted to a variety of processor bus structures or to the rather simple buses used in data acquisition modules. However, there are some additional features required to define a functional system:

Priorities and Priority Arbitration

A fundamental requirement of FASTBUS is to support a system which potentially involves many master devices, e.g., processors, or controllers, all active at the same time, and all potential users of the same data paths in the system (i.e., Crate Segments, or cable interconnections called Cable Segments). This is solved in the following general way: Each FASTBUS segment has a set of lines called priority arbitration lines, and each master device has a corresponding internal register containing a priority vector. On the FASTBUS backplane (or Cable Segment) there resides a special logic card called the Arbitration Timing Controller, and within each master module resides some priority comparison logic. Reduced to its simplest terms, priority levels may be raised at any time, the current master determines when an arbitration cycle can proceed and, once initiated, the arbitration logic proceeds rapidly to identify the master with the highest priority and awards the bus to this master. If the master now wishes to address a device on its own segment, it proceeds to do so. If it wishes to connect to a device on another segment, it initiates an address cycle which appears at a given SI, and the appropriate SI, recognizing the target device address, proceeds to arbitrate for the

next segment on behalf of the master. Priority levels are categorized as local or system priorities, so that high priority devices may be given high system priorities to enable them to quickly gain access to remote parts of the system.

Geographical and Logical Addressing

The above addressing description assumes that a logical address range is assigned to a given device. This can be done with hardware or with software. In a large system, the software solution is preferred. Therefore, a mechanism is needed to load the logical address; this requires an independent addressing mode. This is accomplished with a position code built into the backplane which allows each module to be uniquely accessed by its "geographical" position in the system. Initialization programs rely on this information.

Segment Interconnection

The basic backplane element in FASTBUS is called a Segment or Crate Segment, and a method is needed to interconnect Segments. The Segment Interconnect (SI) is a module which performs this function. The SI needs some unique properties: First, it must be able to control the range of addresses which it will accept in either direction in order to allow a variety of system structures (e.g., ring, tree or star interconnection of Segments). Thus it needs a programmable (hardware or software) table of address ranges, called a route map. Secondly, it requires a mechanism whereby different masters (controlling devices) on either of its ports can avoid collisions in competing for the same crate segment or segments. This is accomplished with contention resolution logic which examines priority levels of the competing masters and resolves conflicts according to preset rules.

OTHER FEATURES

FASTBUS has some other important features such as parity information for error detection, a special address mode known as Sparse Data Scan (for retrieving data from sparsely populated very large arrays of data registers), a service request line, a serial diagnostic line for independent access of diagnostic equipment regardless of the proper functioning of the main bus, and status lines to indicate various kinds of faults as well as standard responses. The table of all system lines is shown in Table I, and the standard meanings for the control and status lines in Tables II and III.

FASTBUS SYSTEMS

With the aforementioned basic features in mind, the construction of systems can be visualized. To summarize the foregoing, the basic building blocks are Crate Segments and Cable Segments, Modules, usually called Masters or Slaves (although a master clearly functions as both), Segment Interconnects, Arbitration Logic, and the Diagnostic Serial Line. Two other devices usually present are some kind of Host Processor, defined as that unit which contains the system description and initialization program, and a Snoop [3] module, which is a special diagnostic device. A simple system embodying these components is depicted in Fig. 4. Note that special interconnections of one segment to another can be made to provide high-priority access paths where warranted, namely, to relieve traffic on other parts of the system. The Cable Segment CS_{13} in Fig. 4 is such a connection, providing a direct, alternate path between the two segments.

Some specifics of a typical system are as follows:

Diagnostic Link

The serial diagnostic link is based on an Ethernet [5] compatible protocol, except slower in speed. Any device can utilize this single coaxial link, but mainly it will be used to interconnect Snoop devices, or to allow access to a specially-placed Snoop when system trouble-shooting.

Snoop Module [3],[4]

The Snoop is a special diagnostic module containing a very fast front-end "silo" which serves as a type of logic analyzer. It has the ability to generate WAIT (WT) in order to single-step FASTBUS operations, and the ability to act as a master to access FASTBUS modules from its location on a given segment. The Snoop in turn is controlled from an independent diagnostic station, communicating over the serial link. The Snoop itself also contains a processor so that diagnostic routines can be downloaded and exercised.

Module Hardware

The module is the most fundamental building block of FASTBUS. An example is shown in Fig. 5. The module is approximately 14.4 x 16 inches in size and utilizes a 130-pin post and header connector. The space above this connector is available for an auxiliary connector or connectors. The module can hold approximately 280 16-pin equivalent IC's and is designed to dissipate a nominal 75 watts maximum power. The basic module can be mounted in either an air or water-cooled crate.

Crate Hardware

A typical (air-cooled) crate is shown in Fig. 6. This particular design has 26 module locations, and an 8-layer press-fit backplane containing both Segment and an Auxiliary connector. The latter has a 3×44 pin‡ arrangement. The backplane contains separate planes or sections for +5, -5.2, -2, ±15, and 28 volts, as well as digital and analog grounds. The entire crate is designed to dissipate a nominal 1500 watts maximum.

The backplane signal lines are designed as transmission lines. Figure 7 illustrates the signal quality for an asymmetrically loaded bus with an extender card.

Power Supplies

A modular switching regulator, rack mounting power supply containing +5, -5.2, -2, and ±15 volts is now available commercially from two manufacturers. The 28V connection is intended for bulk power for optional on-board regulators.

Cooling

The crate shown relies on forced-air cooling. Another unit under development is designed for conduction water cooling via clip-type heat sinks in each module in contact with a cold plate.

System Software

There are two basically independent software systems involved in this example: (1) the System Manager, which resides in the host, contains a description of the system and generates the necessary route maps and address maps to initialize the system; and (2) the serial diagnostic

‡ This specification recently changed to a 2 row 130 pin connector.

software, designed to operate an autonomous Snoop-supported diagnostic system. The FASTBUS software committee is working on both systems.

R&D PROGRAMS

The above hardware and software components are being developed as prototypes to demonstrate the basic functionality and viability of the FASTBUS specification. The basic components needed to generate fast data transfers on the bus and exercise the major protocols have been completed and tested [6],[7],[8]. In performing these tests, several improvements to the specification were recommended and have been implemented. The second phase of the R&D tests, namely to operate multiple segments using the SI and Cable Segment, is now in progress. This has been a major design effort and the basic SI is now in construction [9]. An example of a processor interface has also been prototyped and tested [10].

In prototype applications, an earlier version of FASTBUS has been implemented in a single-crate system at BNL, and utilized to collect and pre-process data in a physics experiment [11]. The European collaborators have underway a number of activities, including actual experiments, as do other laboratories, principally FNAL in Illinois, TRIUMF in Canada, and SLAC [12].

Concurrent with these R&D programs the development of a detailed specification [13] has proceeded. The most recent version is now being finalized in order to be issued as a final published specification by March of 1982. It is intended to process the FASTBUS standard first as a DOE and an IEEE standard, and then to seek approval from both ANSI and IEC.

FUTURE PLANS

The immediate objectives for the FASTBUS development are threefold:

(1) Complete the formal R&D evaluations.

(2) Promote vendor support and participation, and

(3) Promote a broader range of applications.

The R&D evaluations are nearing completion and the final specification will be issued shortly. Vendor support in the high energy physics area is already developing and the rate at which this occurs depends mainly on demand (and budgets) of the various laboratories. For the longer range future, much of the viability of FASTBUS will depend upon its applicability and suitability for other applications, such as process control. For such applications, several further developments appear necessary or highly desirable.

(1) Development of high speed long distance serial links.

(2) Development of special protocol chips to support the FASTBUS ECL interface.*

(3) Development of system software supports, especially diagnostics.

(4) Development of "off the shelf" system modules and controllers.

The diagnostic link can in principle be upgraded to Ethernet quality, i.e., to a 10 MB baseband link. It is not clear whether this will suffice as a future serial control bus. Other local network techniques now emerging may prove more suitable.

In any event, it appears that a considerable additional effort both by vendors and the Standards Committee will be needed before FASTBUS

---

* The ECL interface is the current choice for FASTBUS. The current specification does not rule out future implementation using other logic families.

will be a serious contender in potentially broad controls applications. However, the basic architecture is highly suitable, and this avenue of development should receive serious attention in the future.

CONCLUSION

The formal phase of the development of the FASTBUS standard is nearing completion. Specific applications are beginning to grow, and vendor support is developing. Additional developments are needed to promote FASTBUS outside of the highly specialized area of high energy particle physics data acquisition and control. Future plans include processing as IEEE, ANSI and IEC standards, development of protocol chips, and development of high speed serial links.

ACKNOWLEDGEMENTS

The team working on FASTBUS is large and it is impossible to list individual contributions. Most of the technical work either has appeared or will eventually appear in print under the names of the chief contributors.

The work and continued supported of all of these people is gratefully acknowledged. This work is also supported by the Department of Energy, contract DE-AC03-76SF00515.

REFERENCES

1. IEEE Standard Modular Instrumentation and Digital Interface System (CAMAC*), IEEE Std. 583 (1975). [See also IEEE Stds. 596 (1976), 683 (1976), and 675 (1979).]

2. U.S. NIM Committee, "Future Data Bus Requirements for Laboratory High Speed Data Acquisition Systems," June 1977, TID-27621 (ERDA).

3. H. V. Walz and R. Downing, IEEE Trans. on Nucl. Sci., NS-28, No. 1 (1981), p. 380.

4. D. Gustavson, T. L. Holmes, L. Paffrath and J. P. Steffani, IEEE Trans. on Nucl. Sci., NS-28, No. 1 (1981), p. 380.

5. "The Ethernet, A Local Area Network: Data Link Layer and Physical Layer Specifications," Version 1.0, September 30, 1980, available from Intel Corp., 3065 Bowers Avenue, Santa Clara, CA 95051.

6. L. Paffrath, B. Bertolucci, S. Deiss, D. Gustavson, T. Holmes, D. Horelick, R. Larsen, C. Logg, H. Walz and B. Downing, "A FASTBUS Demonstration System," presented at the 1981 IEEE Nuclear Science Symposium, San Francisco.

7. B. Bertolucci and D. Horelick, "Design of a FASTBUS Programmable Sequencer Module and Memory Module," presented at the 1981 IEEE Nuclear Science Symposium, San Francisco.

8. B. Bertolucci, "Modules and Supporting Hardware for FASTBUS Test and Diagnostic Purposes," presented at the 1981 IEEE Nuclear Science Symposium, San Francisco.

9. R. Downing and M. Haney, "The FASTBUS Segment Interconnect and Cable Segment," presented at the 1981 IEEE Nuclear Science Symposium, San Francisco.

10. M. Larwill et al., IEEE Trans. Nucl. Sci., NS-28, No. 1 (1981), pp. 385-389.

11. L. B. Leipuner et al., IEEE Trans. Nucl. Sci. NS-28, No. 1 (1981), pp. 333-335.

12. R. S. Larsen, "Status and Future of FASTBUS," SLAC-PUB-2832 (1981), presented at the 1981 IEEE Nuclear Science Symposium, San Francisco.

13. FASTBUS Modular High Speed Data Acquisition System for High Energy Physics and other applications, Working Group Document, Tentative Sepcifications, U.S. NIM Committee, August 20, 1981. Available from L. Costrell, Department of Commerce, National Bureau of Standards, Washington, D.C. 20234.

TABLE I

FASTBUS Signals

Mnemonic	Signal Name	Use*	No.	Comments
AS	Address Sync	T/C	1	for addressing and reporting status of connection
AL	Address Acknowledge	T/C		
EG	Enable Geographical	C	1	
CL	Control Line	C	3	
RD	Read	C	1	
AD	Address/Data	I	32	
PA	Parity	I	1	
PE	Parity Enable	I/C	1	
SS	Slave Status	I	3	
DS	Data Sync	T	1	for data and control of data transfers
DK	Data Acknowledge	T	1	
WT	Wait	I/C	1	
SR	Service Request	I	1	
RB	Reset Bus	C	1	
BH	Bus Halted	C	1	
AG	Arbitration Grant	T	1	
AL	Arbitration Vector	I	1	
AR	Arbitration Request	C	1	for bus arbitration
AI	Arbitration Request Inhibit	C	1	
GK	Grant Acknowledge	T	1	
			60	
TX	Serial Line Transmit	I/C	1	for diagnostics (not on CABLE SEGMENT)
RX	Serial Line Receive		1	
GA	Geographical Address Pins (position encoded, not bussed) - simulated on CABLE SEGMENT	I	5	CRATE SEGMENT only
TP	T-Pin (not bussed)		1	
DL	Daisy Chain Left		3	
DR	Daisy Chain Right		3	
R	Reserved		24	

* T = Timing, C = Control, I = Information.

TABLE II

Control Line (CL) Interpretation

CL ADDRESS CYCLE INTERPRETATION

CL[2:0]	Connection Type	
CL = 0	Specific Device (Logical or Geographical Addressing)	Data Space
CL = 1	Specific Device (Logical or Geographical Addressing)	Control Space
CL = 2	Broadcast	Data Space
CL = 3	Broadcast	Data Space
CL = 4	Reserved	--

CL DATA CYCLE INTERPRETATION

CL2	CL1	CL0	Code Number	Interpretation
0	0	0	0	Random Data
0	0	1	1	Block Transfer, Handshake
0	1	0	2	Extended Address
0	1	1	3	Block Transfer, Non-handshake
1	X	X	4-7	Reserved

TABLE III

Status Line (SS) Interpretation

ADDRESS TIME SS RESPONSE WITH AK(u)

SS[2:0]	Interpretation
SS = 0	Address Recognized
SS = 1	Network Busy
SS = 2	Network Failure
SS = 3	Get Off
SS = 4	Invalid Internal Address (Optional)
SS = 5	Non-Standard
SS = 6	Non-Standard
SS = 7	Reserved

DATA TIME SS SLAVE RESPONSE TO DS(t)

SS[2:0]	Interpretation
SS = 0	Valid Address and Data
SS = 1	Retry
SS = 2	End of Transfer
SS = 3	Error
SS = 4	Requested Cycle Impossible
SS = 5	Non-Standard
SS = 6	Non-Standard
SS = 7	Faulty Data

FIGURE CAPTIONS

Fig. 1. Basic Handshake Read Operation (as seen at Master).

Fig. 2. Non-Handshake Operation Write Block Transfer (as seen at Master).

Fig. 3. Mixed Handshake Operation Read-Modify-Write (as seen by Master).

Fig. 4. Example System.

Fig. 5. Typical Module.

Fig. 6. Typical Crate.

Fig. 7. Backplane Waveform.

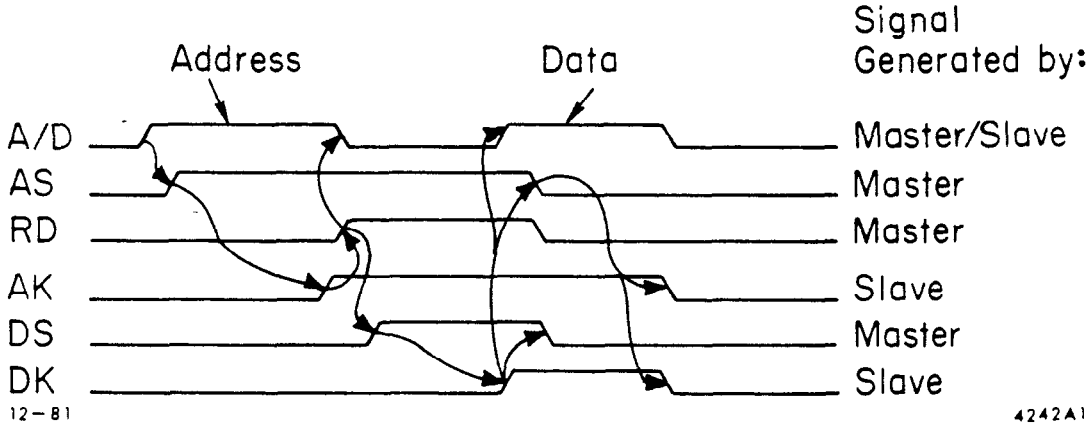

Address
Data
Signal
Generated by:
A/D
AS
RD
AK
DS
DK
Master/Slave
Master
Master
Slave
Master
Slave
12-81
4242A1

Fig. 1

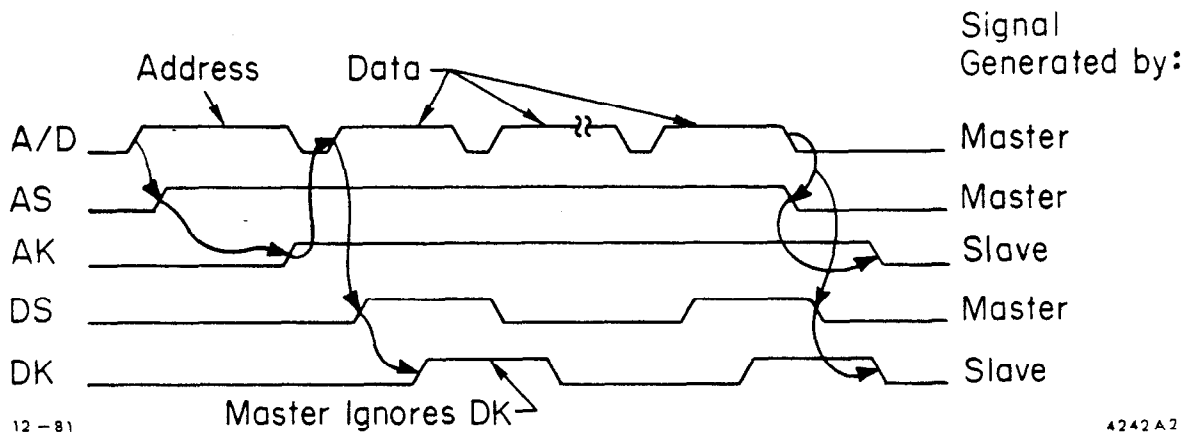

Signal
Generated by:
Address
Data
A/D
AS
AK
DS
DK
Master
Master
Slave
Master
Slave
Master Ignores DK
12-81
4242A2

Fig. 2

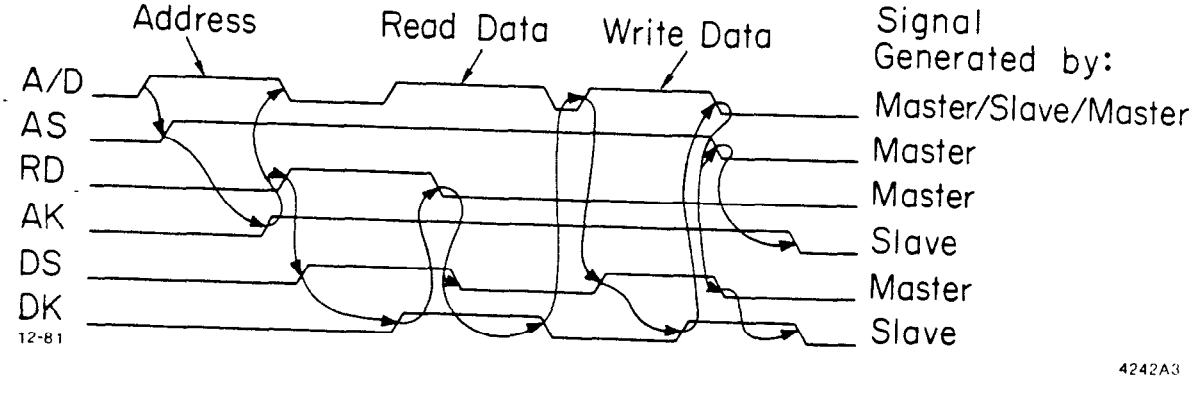

Address
Read Data
Write Data
Signal Generated by:
A/D
AS
RD
AK
DS
DK
Master/Slave/Master
Master
Master
Slave
Master
Slave
12-81
4242A3

Fig. 3

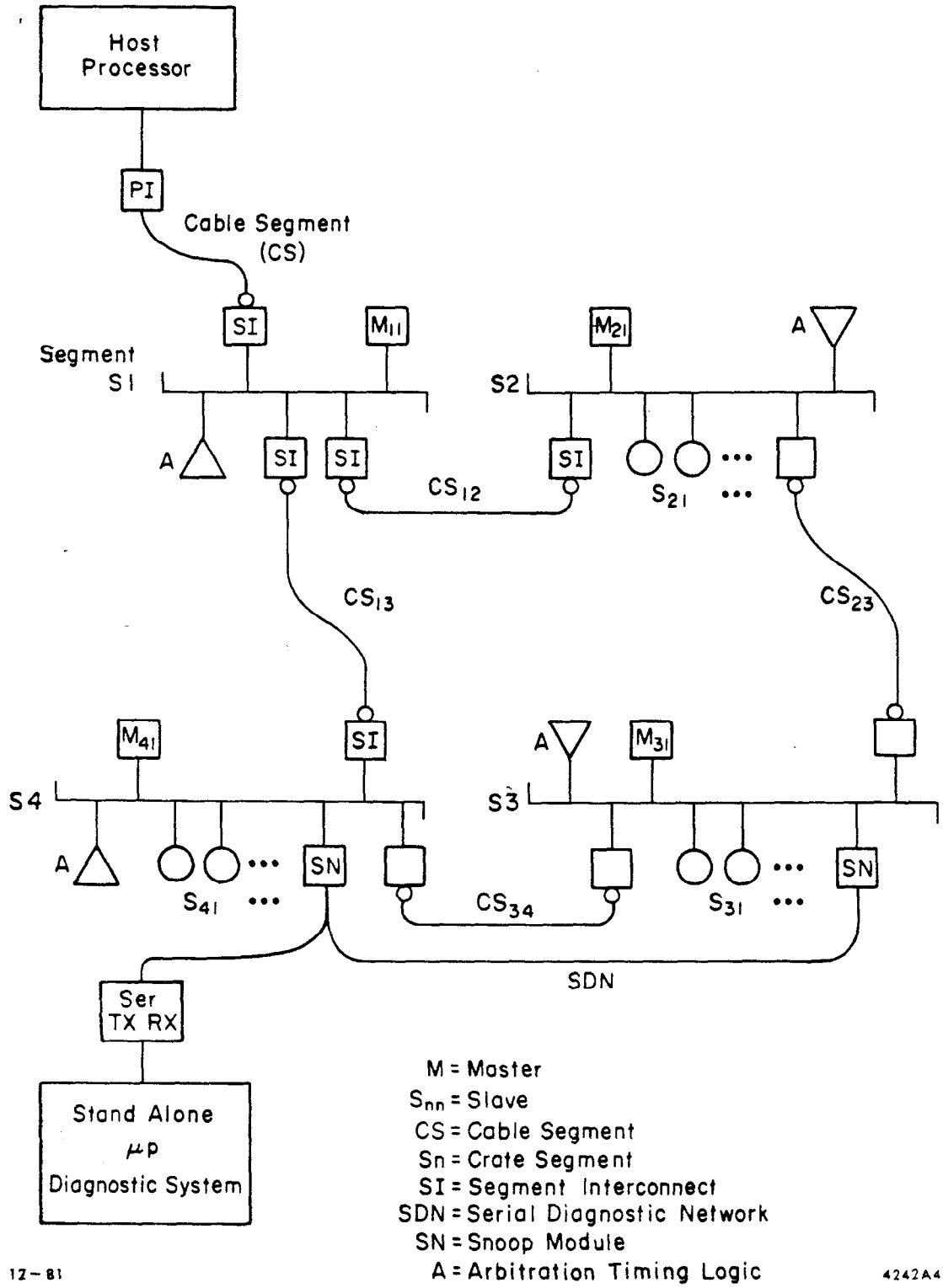
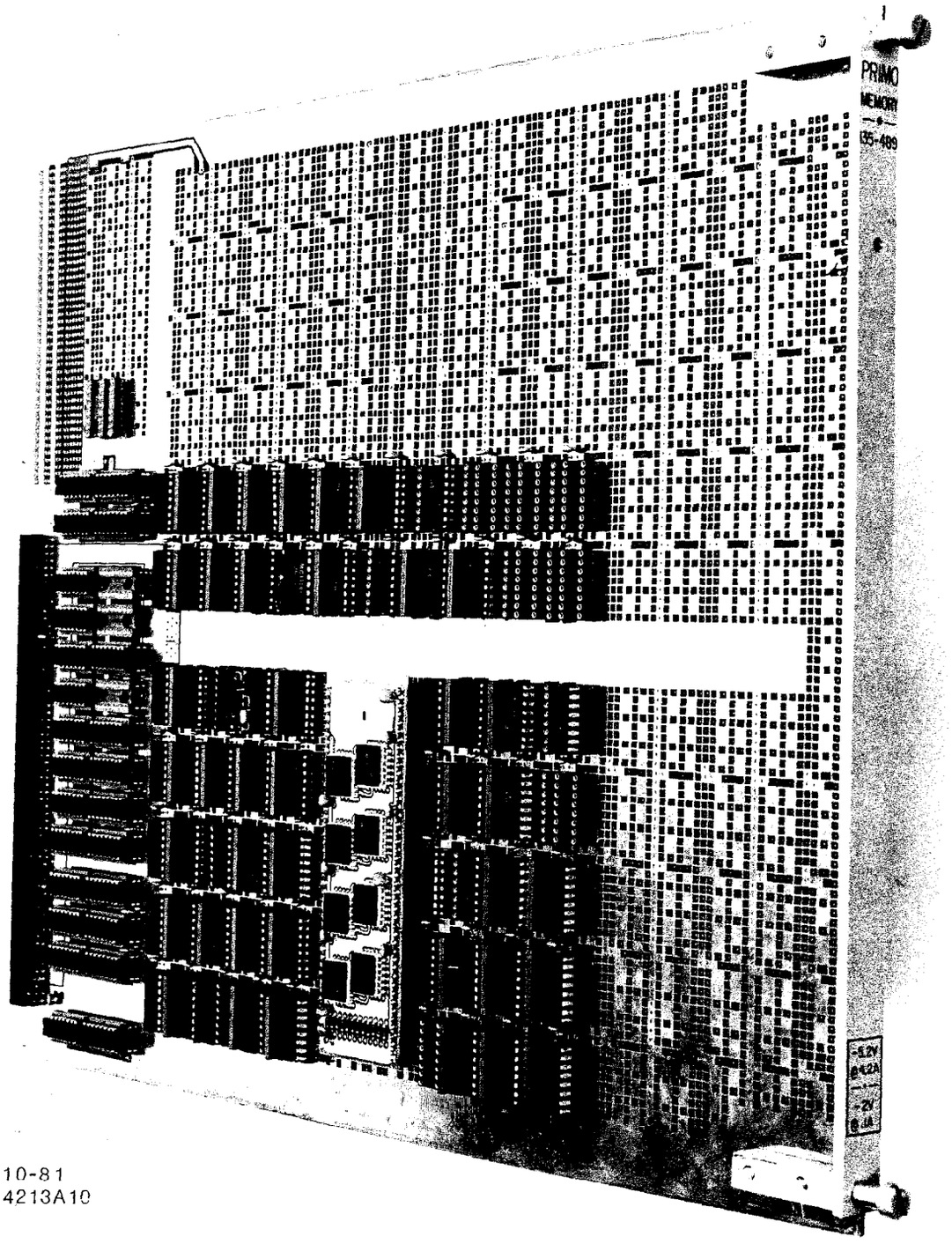

Host Processor
PI
Cable Segment (CS)
SI
M11
M21
A
Segment S1
S2
A
SI
SI
SI
CS12
S21
CS13
CS23
M41
SI
A
M31
S4
S3
A
SN
SN
S41
CS34
S31
SDN
Ser TX RX
Stand Alone μp Diagnostic System
M = Master
Snn = Slave
CS = Cable Segment
Sn = Crate Segment
SI = Segment Interconnect
SDN = Serial Diagnostic Network
SN = Snoop Module
A = Arbitration Timing Logic
12-81
4242A4

Fig. 4



10-81
4213A10

Fig. 5

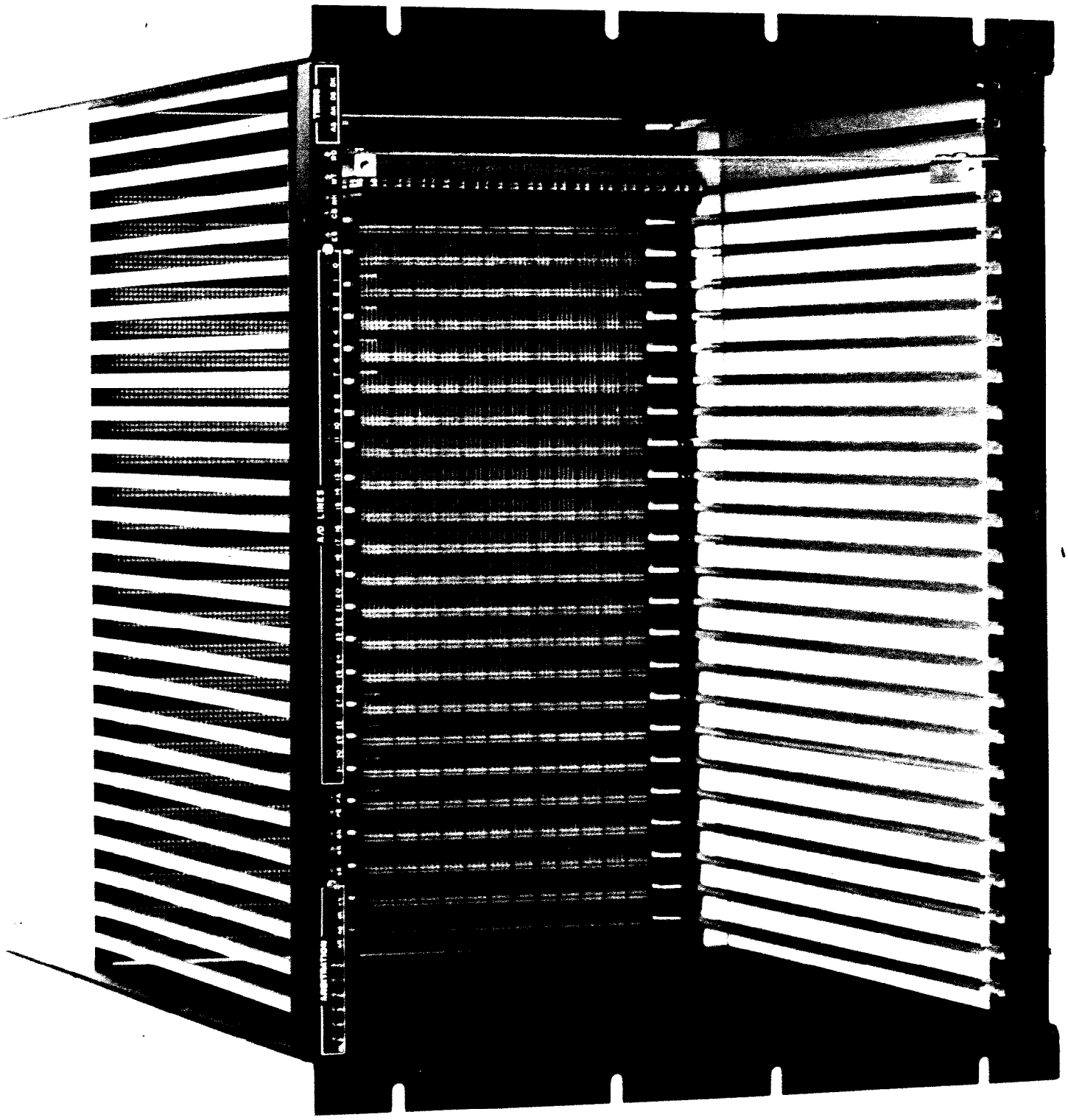
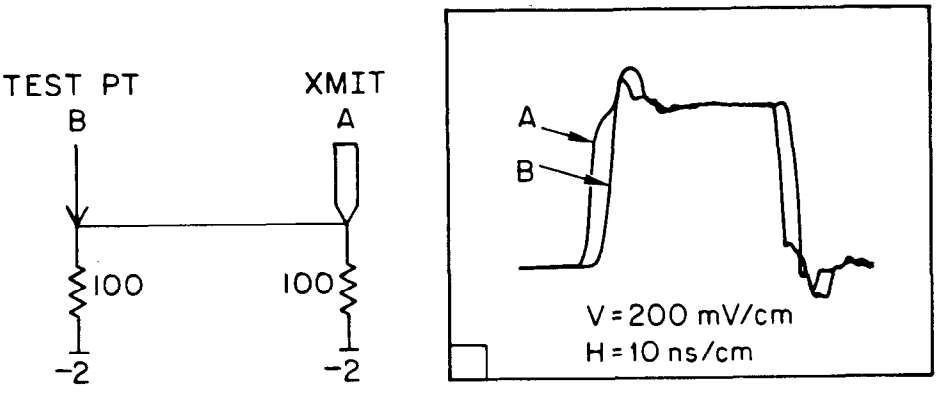
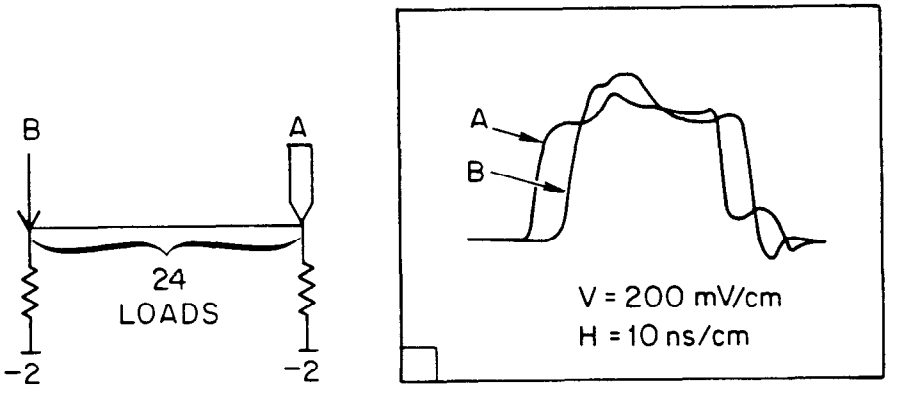


Fig. 6

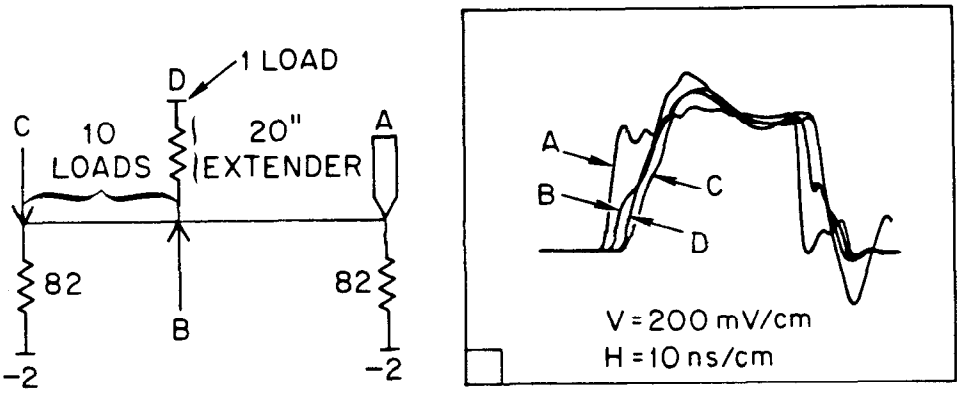
BACKPLANE WAVEFORMS



(a) BUS UNLOADED, $R_T = 100\,\Omega$



(b) BUS FULLY LOADED, $R_T = 100\,\Omega$



10-80
398784

(c) BUS HALF LOADED, EXTENDER, $R_T = 82\,\Omega$

Fig. 7