SLAC-PUB-2637
November 1980
(I/A)

FASTBUS SNOOP DIAGNOSTIC MODULE*

Helmut V. Walz
Stanford Linear Accelerator Center
Stanford University, Stanford, California 94305
and
Robert Downing
University of Illinois
Urbana, Illinois

Abstract

Development of the FASTBUS Snoop Module, undertaken as part of the prototype program for the new interlaboratory data bus standard, is described. The Snoop Module resides on a FASTBUS crate segment and provides diagnostic monitoring and testing capability. Communication with a remote host computer is handled independent of FASTBUS through a serial link. The module consists of a high-speed ECL front-end to monitor and single-step FASTBUS cycles, a master-slave interface, and a control microprocessor with serial communication ports. Design details and performance specifications of the prototype module are reported.

I. Introduction to FASTBUS

FASTBUS is a proposed standard data bus for modular, high-speed data acquisition systems. It has been developed by the Fast System Design Group of the U.S. NIM Committee. FASTBUS systems can be configured from multiple bus segments. These segments are able to operate independently or link together selectively for exchange of data. Each segment accomodates multiple processors. The basic FASTBUS operating mode is asynchronous to allow for wide variations in operating speed of attached devices. Data block transfer rates of 100 MHz are expected possible between devices residing on the same bus segment.

A general description and status report of FASTBUS development work in progress is offered in two papers presented at this Symposium.[1,2] Those readers interested in complete specification details for FASTBUS may obtain a copy of the proposed bus standard from the National Bureau of Standards.[3]

II. Snoop Module Concept

The FASTBUS Snoop Module has been devised for diagnosing problems inside crate segments, and for monitoring communications from segment to segment in FASTBUS systems (Fig. 1). Since all bus-segment signal lines are accessible at each crate module location, such a diagnostic module may be used to monitor and record FASTBUS transactions within a crate segment. The FASTBUS wait line (WT) may be used to single-step bus cycles and implement programmable trap functions. The serial bus lines within each crate segment may be used as an independent communication path between diagnostic modules and the host computer, when connected to a serial network which bypasses all segment interconnect units.

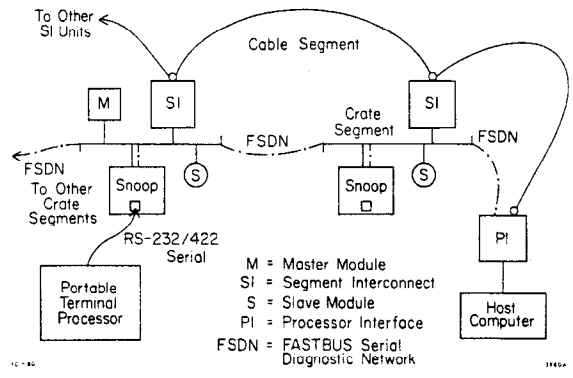

Fig. 1. Typical FASTBUS System with Snoop Modules

The basic hardware organization of the Snoop Module is shown in Fig. 2. A fast front-end section connects the module to the crate segment bus. This section handles diagnostic recording and control of the

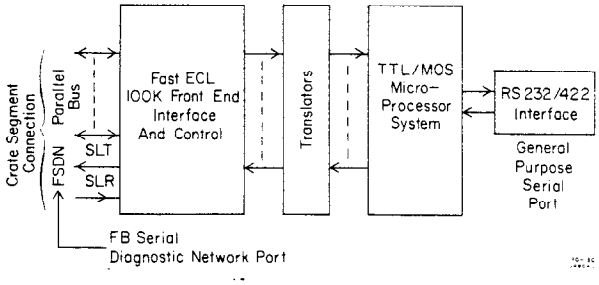

Fig. 2. Snoop Module Organization

crate segment, with response capability to match the fastest devices on the bus. It also provides interface and control for master-slave operation and connection to the serial bus lines. Hardware realization is based on emitter coupled logic (ECL) in general, with all speed-critical parts implemented with Fairchild 100 K ECL circuits and a high degree of parallelism. Control and supervision of the fast front-end section is handled by a compact microprocessor section, which includes a second, general-purpose, UART-type serial port. A powerful 16-bit CPU (MC 68000) has been selected to optimize handling of serial communications, interrupt driven control of the front-end diagnostic functions and master-slave operations, and replacement of random logic by firmware-based processor control throughout the module.

Since the Snoop Module is expected to see widespread usage in FASTBUS systems, this module organization combines the low functional complexity and extensive hardware parallelism of the fast ECL front-end with the high level of integration of the processor section, achieving a single-width module implementation. A detailed block diagram is shown in Fig. 3. Based on this block diagram, a description of important design details is offered in the following two sections.

III. Fast ECL Front-End

The fast ECL front-end contains the basic diagnostic functions of the Snoop Module: programmable wait-step logic, traps for address, address-data, and parity-error detection, activity history silo memory, and master-slave interface logic. To allow processor

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

(Contributed to the 1980 Nuclear Science and Nuclear Power System Symposium, Orlando, Florida, November 5-7, 1980.)

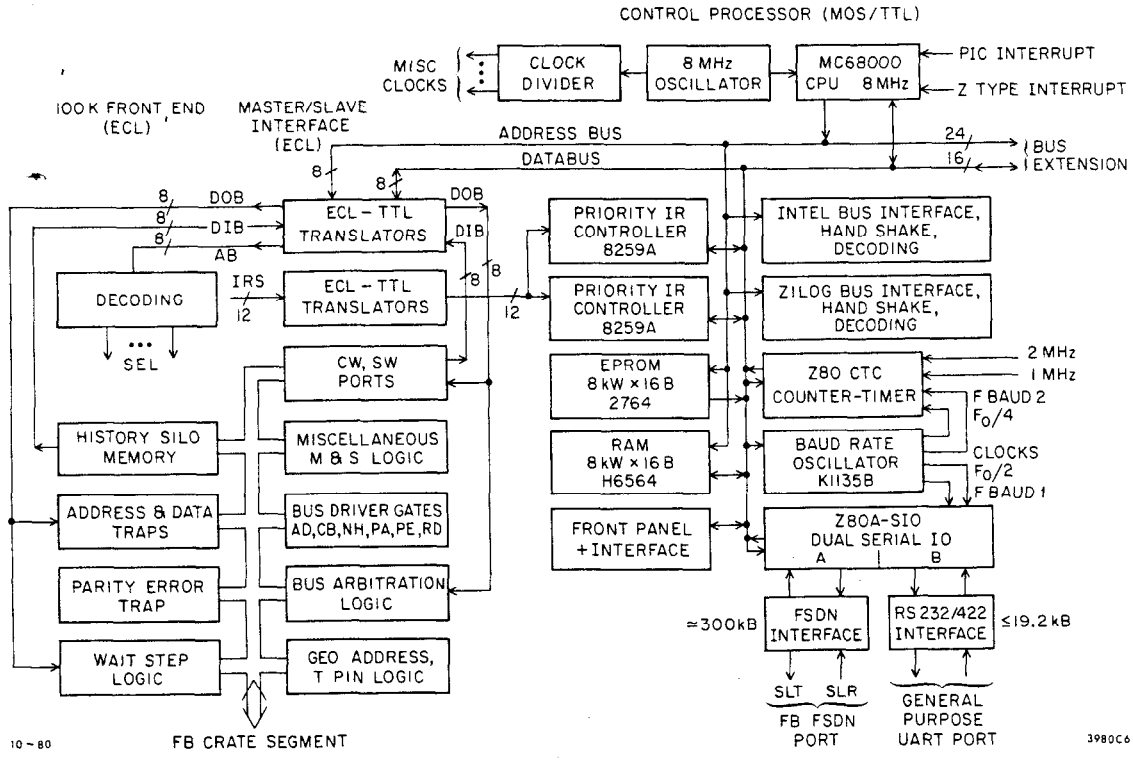

Fig. 3. Snoop Block Diagram

control of this section, control and status word registers, address decoding, and ECL-TTL level conversion are also provided. This section is implemented with a mixture of 100 K and 10 K ECL integrated circuits.

A. Programmable Wait-Step Logic

The wait-step logic asserts the WAIT line in response to bus timing and control signal transitions. The selection of signals used as trigger inputs is made by setting enable bits in two control word registers. Available trigger sources are AS, AK, DS, DK, AG, GK, address-data trap, and parity-error trap.

A typical wait circuit is shown in Fig. 4. The estimated wait response delay is 5 ns.

B. Address-Data and Parity-Error Traps

The address-data trap is illustrated in Fig. 5. The 32 AD lines, CB and NH are compared with a pair of

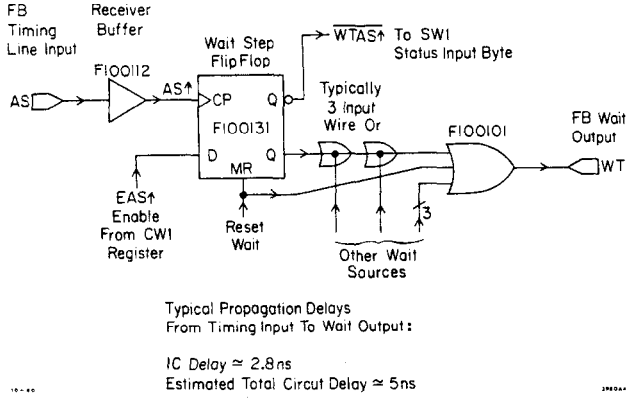

Fig. 4. Typical Wait-Step Logic

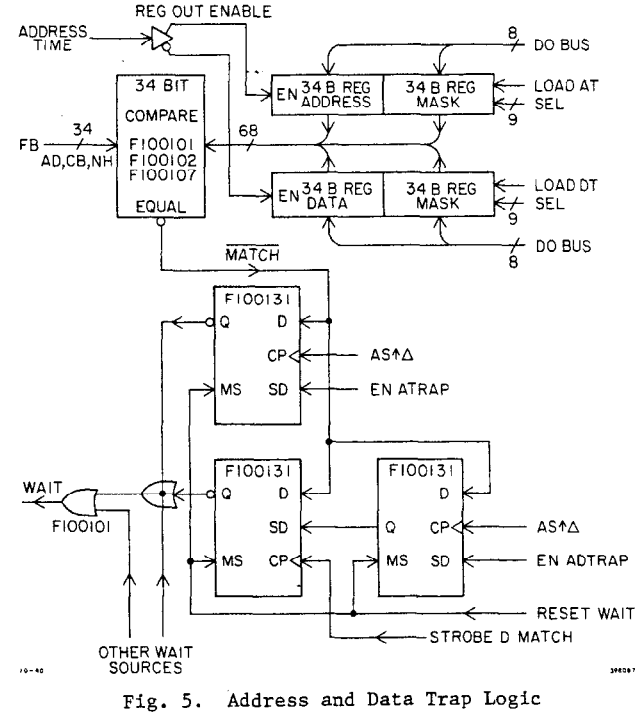

Fig. 5. Address and Data Trap Logic

34 bit registers at address and data sync times. The contents of these registers allows each bit to be specified as 0, 1 or "don't care". The trap may be used as an address trap or an address-data combination trap. The combination trap can be utilized to detect extended address cycles.

The expected response time for address or data detection is 8 ns. The parity-error trap consists of a 33 bit parity checker (F100160) driving an array of five parallel flip-flops. These flip-flops are clocked by appropriately delayed timing signals and their outputs are used as wait sources driving the WAIT output line. The estimated response time for parity-error detection is 15 ns.

C. Activity History Silo Memory

The silo memory is able to record 256 FASTBUS cycles with a speed in excess of 100 MHz (Fig. 6).

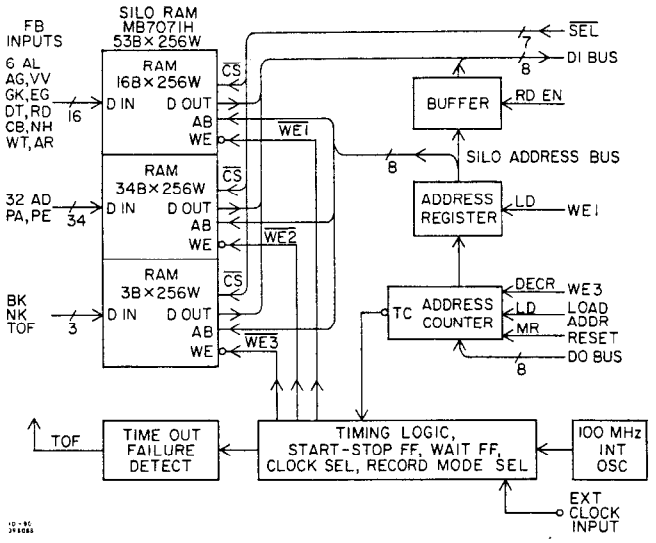

Fig. 6. Activity History Silo Memory

For each cycle, 52 bus signals and a time-out failure bit (TOF) are recorded. Several programmable modes for start and stop of silo recording are available. For example, the address-data trap, described previously, may be used to start recording with automatic stop and wait generation when the memory is filled. Choices of FASTBUS-synchronized or real-time clock recording modes are available. For read-out the silo is addressed from the processor and each word is multiplexed onto the 8-bit data bus to the processor in 7 bytes. The data path from the crate segment bus lines through the silo memories onto the data input bus to the processor is also used during WAIT = 1 to read the bus status.

D. Master-Slave Interface Logic

To support master-slave capability of the Snoop Module, hardware is provided for bus priority arbitration with a control word register for the module arbitration level. For slave mode, geographic address recognition is implemented. FASTBUS protocol response and generation is handled with processor interrupts, status word input buffers and control word output registers. Master-slave operation is described in more detail in Section V.

IV. Control Processor and Serial Ports

Processor design is the result of the following Snoop Module requirements:

- highly compact implementation in order to accommodate all module hardware on one PC board;
- multi-channel, programmable, high-speed interrupt handling;
- approximate 300 K bit serial port handling for FASTBUS Serial Diagnostic Network;
- maximum CPU execution speed to optimize throughput and minimize memory requirements;
- 32-bit CPU registers and instructions to handle 32-bit wide FASTBUS data.

The implementation uses approximately 25 integrated circuit packages. For the CPU the MC68000 (8 MHz) microprocessor was chosen. Memory has a word width of 16 bits and consists of 8 K words of EPROM and 8 K words of RAM. To minimize package count, 64 K bit EPROM and hybrid static RAM units are utilized.

The programmable priority interrupt structure accommodates 25 interrupt vectors. Three interrupt handling schemes are combined. The 15 interrupt inputs from the fast front-end section and the module front panel are processed by 8259A interrupt controllers (Intel). Interrupt sources from the dual serial IO port and 4-channel counter-timer (10 sources) are connected into a Zilog-type daisy chain configuration. Both interrupt handlers are then connected to the interrupt control inputs of the CPU. Some special logic is used to combine all three interrupt schemes. The four-channel counter-timer subsystem has two real-time clock inputs (1 and 2 MHz) and two serial port clock inputs (1.2672 MHz and a programmable baud rate generator clock with 0.8 to 316.8 kHz). This system is provided for implementing real-time clock, elapsed timer, and serial communication related timer functions.

Two serial ports are available from a Z80A-SIO controller unit. The general purpose UART port operates asynchronously with 16 programmable baud rates from 50 baud to 19.2 K baud. By means of jumpers the serial interface standard is selected from RS232 and RS422 formats. Standard modem control signals are also available. Receiver, transmitter and status interrupts are generated to the CPU. The second SIO channel is used as port for the FASTBUS Serial Diagnostic Network (FSDN).[4] FSDN will be used for remote control of Snoop Modules (and other diagnostic modules) and communication among Snoop Modules for diagnosis of segment interconnect and cable segment problems. The prototype FSDN is an Ethernet-like carrier-sense-multiple-access network with collision detection. Manchester coding is used to combine data and clock and achieve constant 50% duty-cycle serial output modulation. The synchronous transmission rate is 316.8 K bits per second.

The SIO channel, operating in SDLC mode, takes care of network address recognition, cyclic redundancy code generation and checking, and vectored interrupt handling. As shown in Fig. 7, two registered PROMs are

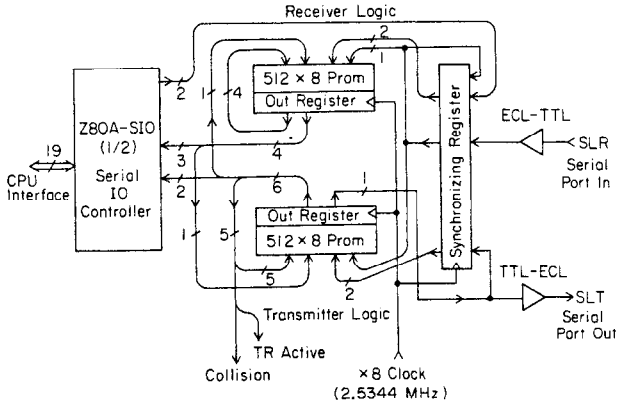

Fig. 7. FSDN Serial Port Interface

used to form Mealy finite-state logic machines to encode transmitted data, separate received data and clock information, detect collisions when several stations try to transmit simultaneously, and perform miscellaneous logic functions. Another register is used for staticizing asynchronous signals. Translator circuits and an ECL driver and receiver couple the FSDN port to the crate segment serial bus lines.

The module front panel is shown in Fig. 8. The 16-bit status LED display is CPU driven as an output

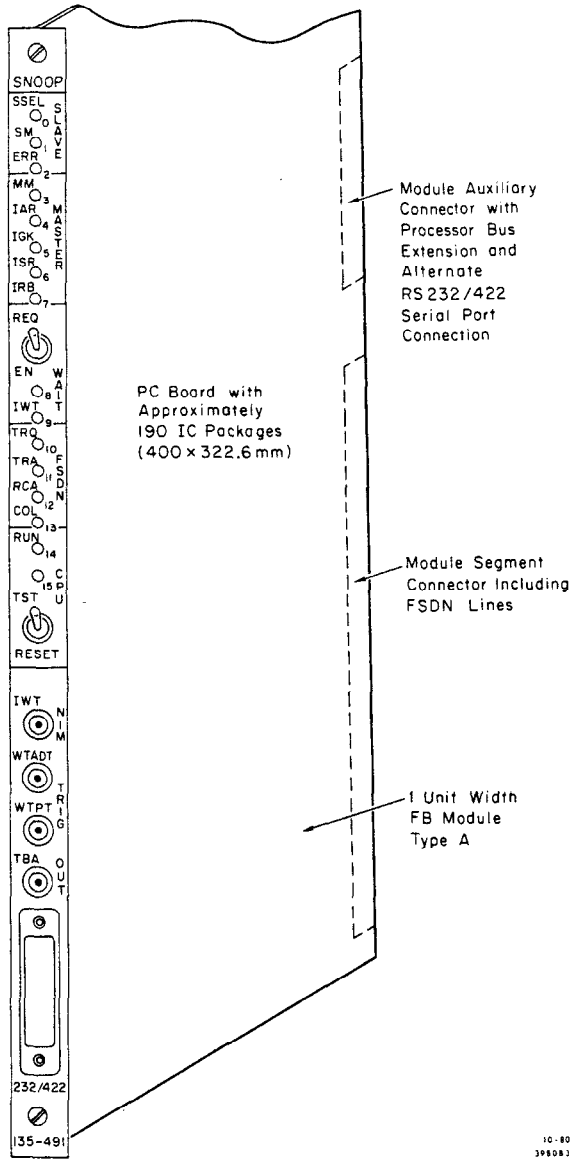

Fig. 8. Illustration of Module with Front Panel

port register. For testing work, it may also be used as a 16-bit data display. Two switch functions generate CPU interrupts for manual FB WAIT generation and for a processor self-test. The third switch function performs a direct hardware reset of the CPU. Finally, the processor section contains a small amount of random logic to combine the different interrupt schemes, to generate the handshake responses for the MC68000 CPU, and decoding required for addressing all processor peripheral devices.

V. FASTBUS Master-Slave Capability

A simple master-slave capability for the Snoop Module has been implemented by utilizing part of the fast front-end hardware of the basic Snoop Module and by adding some dedicated logic. Control of master-slave operations is handled by the processor through interrupts and IO instructions. The history silo memory serves as an input buffer for 32 AD lines and some control and information signal lines (Fig. 6). Output of AD lines, CB and NH is accomplished with two of the four trap registers and added driving gates onto the segment bus. One register holds the information for an address cycle, the other for a data cycle. This sharing of hardware implies that silo recording and address-data trap operation is not available during master-slave operation.

The required dedicated hardware consists of arbitration gating logic with module priority register, AR and GK flip-flops, geographic address decoder and SEL flip-flop, broadcast TP output logic, processor control word output registers and interrupt input handling for timing and control signals. Operation as bus master works simply by acquiring mastership through priority arbitration, and then executing single FASTBUS cycles with processor IO instructions. The asynchronous handshake protocol of FASTBUS allows execution at microprocessor speed. Slave mode operation is more critical in order to avoid time-out failures. Extensive use of processor interrupts and hardware-generated FASTBUS WAIT states is required to synchronize the Snoop processor as a slave.

No logical addressing is available in slave mode. All control and data space registers are emulated by software.

VI. Prototype Module Development Effort

Two major considerations dictated the construction of the Snoop Module hardware:

- all logic circuitry to be contained in a single-width FASTBUS type A module;
- minimum propagation delay requirements for fast ECL 100 K front-end section.

The Snoop Module design requires approximately 190 integrated circuit packages of mixed sizes. This calls for a high density, multilayer printed circuit board. The PC board has power distribution and ground on the two interior layers. All signal traces are on the outer two sides. Fairchild 100 K ECL devices used for parts of the fast front-end section are in 24-pin flat packages. This optimizes circuit speed and board density. The leads of these flat packs are formed into "dog-leg" shape and reflow-soldered to the surface of the board. A finned heat sink is epoxied to the ceramic case. For best heat transfer results the IC flat package has been inverted to locate the die as close as possible to the heat sink. Flat packs are mounted on a 0.55 x 0.55 inch grid. PC board layout is based on 0.010 inch signal trace width with a center-to-center spacing of 0.025 inch. This allows one trace between flat pack leads. Signal traces are microstrip lines with a characteristic impedance of approximately 85 ohms. Special care has been taken to minimize interconnecting line lengths and keep stub lengths from the FASTBUS bus connector as short as possible.

In parallel with this hardware effort, software for the Snoop control processor and a FASTBUS diagnostic system with several Snoop Modules linked by the FSDN is being developed. [5]

VII. Conclusion

This paper has described a high performance diagnostic module for FASTBUS. Since the basic Snoop Module functions were developed as part of the FASTBUS Standard Specification, this module is expected to find extensive usage in future FASTBUS systems. Beyond the standard application of the Snoop Module, numerous other possibilities may be of interest. Some of these possibilities are illustrated in Fig. 9. For reference purposes, features and basic specifications of the prototype Snoop Module are summarized in Table I.

Complete fabrication and extensive testing of a prototype module are expected during the next three months.

Table I

FASTBUS Snoop Module Features and Basic Specifications

- Fast front-end implemented with 100 K ECL devices in 24-pin flatpaks.
- Programmable wait-step-logic with 5 ns response time.
- Address and address-data combination traps with 8 ns response time; each trap with 34-bit trap and mask registers.
- Parity-error trap with 15 ns response time.
- Activity history silo with 53-bit × 256 word RAM; 100 MHz recording speed; programmable recording modes; FB or real-time synchronized; logic analyzer mode with internal or external clock source.
- Simple master-slave capability with geographic address recognition, bus arbribration, IO register programmed protocol control, software emulation of CSR slave registers.
- MC68000 CPU (8 MHz).
- 25 level programmable priority interrupt structure.
- 16-bit × 8 K word static RAM (350/480 ns access/cycle time).
- 16-bit × 8 K word EPROM (250 ns).
- 4-channel, programmable counter-timer subsystem with real-time and serial interface related clock inputs.
- 2-channel, programmable serial interface.
- RS232/422 asynchronous serial port with programmable baud rates (50-19.2 KB).
- Synchronous serial port interface to FSDN with 316.8 KB.
- Processor driven front panel with status display; manual wait execution switch; interrupt sense switch for processor selftest; CPU reset switch; NIM level scope trigger outputs from wait-step logic.
- 1-unit wide FB module type A with 190 IC packages on a 4-layer PC board; estimated 85 Watt power dissipation.

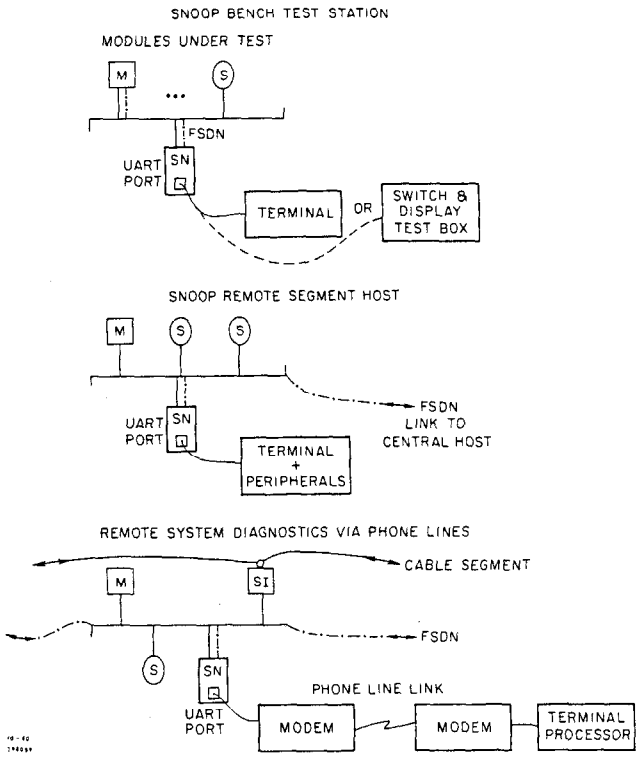

Fig. 9. Some Snoop Module Applications

Acknowledgments

The authors of this paper would like to express their appreciation for contributions to design and specifications of the Snoop Module by D. Gustavson and L. Paffrath. The effort on PC board detail layout by Ruth Kastner of the University of Illinois deserves special recognition. Finally, the leadership and encouragement by R. Larsen was a vital ingredient to the success of this project.

References

1. Status of the FASTBUS Standard Data Bus, R. S. Larsen, Stanford Linear Accelerator Center; invited paper presented at the IEEE Nuclear Science Symposium, Orlando, Florida, November 1980.
2. FASTBUS Software Status, D. B. Gustavson, Stanford Linear Accelerator Center; paper presented at the IEEE Nuclear Science Symposium, Orlando, Florida, November 1980.
3. FASTBUS Draft Specification, September 1980, available from L. Costrell, Department of Commerce, National Bureau of Standards, Washing, D.C. 20234.
4. Design Considerations for the FASTBUS Serial Diagnostic Network, D. B. Gustavson, Stanford Linear Accelerator Center, FSDG-089 Internal Report, 20 October 1980.
5. A "Front Panel" Human Interface for FASTBUS, D. B. Gustavson, T. L. Holmes, L. Paffrath and J. P. Steffani, Stanford Linear Accelerator Center; paper contributed to the IEEE Nuclear Science Symposium, Orlando, Florida, November 1980.