SLAC – PUB – 3784
September 1985
(I/A)

SLAC FASTBUS SNOOP MODULE – TEST RESULTS AND SUPPORT SOFTWARE*

DAVID B. GUSTAVSON AND HELMUT V. WALZ
Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94305

ABSTRACT

The development of a diagnostic module for FASTBUS has been completed. The Snoop Module is designed to reside on a Crate Segment and provide high-speed diagnostic monitoring and testing capabilities. Final hardware details and testing of production prototype modules are reported. Features of software under development for a stand-alone single Snoop diagnostic system and Multi-Snoop networks will be discussed.

1. INTRODUCTION

The SLAC Snoop Diagnostic Module for FASTBUS is a powerful single-width module which can act as an intelligent logic analyzer for debugging single- or multiple-crate FASTBUS systems. The Snoop can also be used as the computer for controlling small test systems, or several Snoops can be used in a coordinated fashion to diagnose segment interconnection problems.

The Snoop Module can be controlled by a computer or computer terminal connected to its front-panel connector, or by messages sent over the FASTBUS Serial Diagnostic Network, which attaches to the FASTBUS backplane. The network connection is particularly desirable when coordination of several Snoops is desired, or when remote control of Snoops is needed. The Snoop contains a 12 MHz 68000 microprocessor, with an operating program in ROM which allows simple access to Snoop facilities. An interpreter allows easy interactive entry and execution of new programs, such as special test routines, either from a terminal keyboard or from personal computer disk or diskette files.

2. SNOOP MODULE HARDWARE

Progress on prototype development of the Snoop Module has been reported previously.[1] This work has been completed, and five preproduction Snoop Modules have been built. The final hardware configuration of the module incorporates several improvements in the high-speed ECL front-end section and in the 68000 microprocessor section. Only minor modifications are planned before moving into full production.

A photograph of the Snoop Module is shown in Fig. 1, and a block diagram is shown in Fig. 2.

The ECL history-silo memory has been expanded four-fold to 1K words. Programming options for the address and data traps and the trigger modes of the silo memory have been made more versatile. One such trigger option allows real-time synchronization for recording by several Snoop Modules on different FASTBUS Segments via front panel interconnections.

The processor section utilizes a 12 MHz MC68000 CPU in a LCC package and provides 8 pairs of standard memory sockets. These may be used with various types of EPROMs and allow a choice of ROM/RAM configurations.

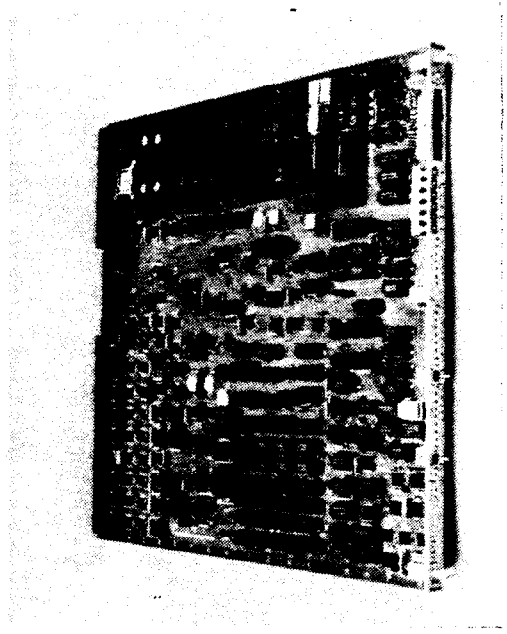

Fig. 1. The Snoop Module fits in a single crate slot.

A processor extension bus is available via the Module Auxiliary Connector. This bus may be utilized to expand processor memory, add processor peripherals, or implement additional FASTBUS diagnostic functions in an add-on module to extend Snoop Module capabilities as needed in the future. This extension bus also provides the possibility of linking two Snoop Modules back-to-back for Snoop testing and maintenance.

Snoop FASTBUS Segment drive capability in Master and Slave modes utilizes some of the existing logic from diagnostic functions. The bus-driving hardware is a mixture of 10K and 100K ECL devices. However, 10k ECL drivers are used on all timing lines (AS, AK, DS, DK, AR, GK) where signal transitions convey information. Drivers of 100K ECL type are restricted to control and information signals (AD, MS, SS, etc.), where settling time is provided before sampling of their levels.

3. SNOOP SOFTWARE

The ROM-resident software for the Snoop Module has been written in the Forth language, providing a compact yet powerful on-board system that includes a compiler, interpreter, and assembler and also supports multi-tasking and interrupt handling. The system is particularly well suited to a hardware-debugging environment, because it is fully interactive and provides nearly instantaneous turn-around for writing and modifying programs to exercise particular facets of the hardware. Forth bypasses the usual edit, compile, link, load, execute cycle required by most batch-oriented language environments, and runs much faster than the BASIC-style interpretive languages. This speed is greatly appreciated when a program loop is needed for generating a repetitive sequence of electrical signals for observation by oscilloscope. Because a 68000 assembler is integrated into the system, it is easy to generate the ultimate-speed machine-coded instructions in the few cases where that extreme is needed.

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

Poster paper presented at the Nuclear Science Symposium, San Francisco, CA, October 23-25, 1985.

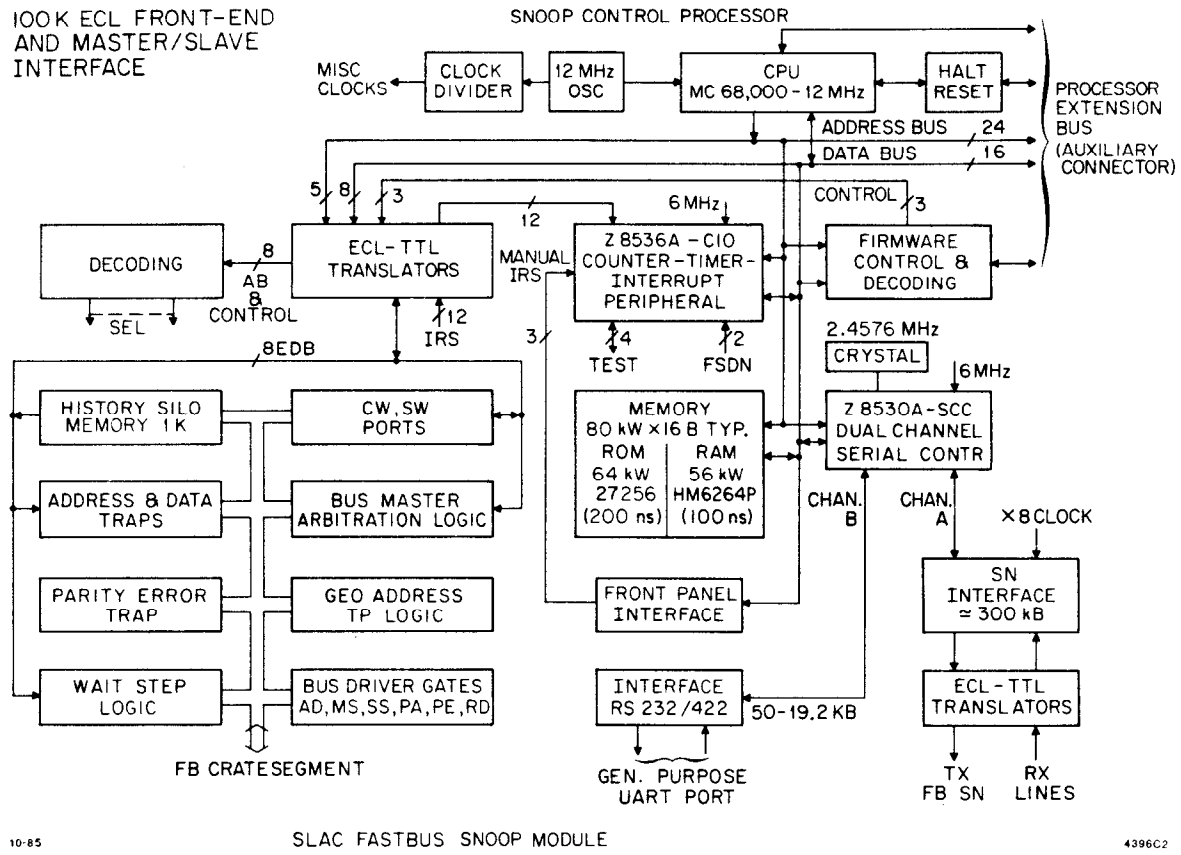

Fig. 2. A block diagram of the Snoop Module.

The Snoop Module has served as its own development system. Once the processor, memory and terminal I/O were working, ROM containing a small Forth nucleus was installed. Additional features were tested and added incrementally, first by keyboard entry, next by downloading from a smart terminal's diskette drive, and finally by incorporation into the ROM. The smart terminal currently used is an IBM PC.

The ROM contains a simple but comprehensive user interface, which allows complete control of the Snoop by means of a simple ASCII terminal. Menus are available to help the inexperienced user, and the full Forth programming capability is available as well.

Using a PC (or other computer) instead of a simple terminal adds disk storage capability, making it possible to create a stored collection of useful diagnostic routines which support trouble-shooting work on particular modules, and to share such routines with others. The PC also adds hardcopy printing capability to the Snoop, and even test-data storage.

The Forth system provides complete access to all Snoop hardware features for those who need it, while presenting a simple and clean model which hides the dirty details from the normal user. For example,

MAKE ASU-WT	causes WT to be generated by the Snoop when AS goes Up, and
NO ASU-WT	turns that feature off.
#3FF. AT-IGNORE #123800. AT-MATCH MAKE WT-ON-TRAP A-TRAP-MODE	sets up the comparator masking bits to ignore the low-order 10 AD bits, sets the AD value to be matched at hexidecimal 123800, then enables WT generation whenever the match occurs during an address cycle.
SNST.	displays the current SNoop STatus in text form.
5 0 SILO.	displays the 5 FASTBUS cycles last recorded in the silo memory, and
3 5 SILO.	displays the 3 cycles before those.

In addition to these simple interfaces to the Snoop Module's logic-analyzer capabilities, there are interfaces which allow use of the Snoop as a FASTBUS master, so that it can exercise other FASTBUS devices, or perform simple data-collection and analysis chores. The Snoop acts as a FASTBUS slave using microprocessor simulation with hardware help for (geographic) address recognition. The slave registers can also be accessed by the user, so that (for example) another master could be set up to read or write data to the Snoop "slave" for testing purposes.

The FASTBUS Master interface uses command names compatible with the proposed FASTBUS Standard Subroutines. Only a subset of those functions will be provided in ROM, but the user has building-block routines available for generating any additional functions he might wish to have for a particular application.

4. MULTI-SNOOP NETWORKS

The Snoop Module is also intended for use in large multi-segment systems: for diagnosing interconnect problems; for acting as a remotely-controlled master to collect data about, or to stimulate, a broken system; for collecting samples of segment activity without disturbing a running system; and for gathering correlated information from intermittently-connected segments.

These functions require a method of controlling and communicating with multiple Snoops in a coordinated way, something more effective than separate terminals connected to each Snoop. The FASTBUS Diagnostic Serial Network was envisioned for this purpose, and an experimental version is being implemented in the Snoop hardware. The FASTBUS specification has not defined this network yet, largely because of a desire to adopt some industry standard rather than using our own unique (and thus expensive) design. Prototype work on the network has been carried out using a Manchester-coded low-speed Ethernet-like scheme, while waiting for industry to come up with the right answer for FASTBUS.

We are now converting to the AppleTalk network[2] used by Apple Corporation for its Macintosh computers and Laserwriter printer. The AppleTalk network uses even less board space than our prototype network, and a subset of the original hardware, so we can implement it by simply removing components and inserting jumpers. AppleTalk uses the FM0 modulation capabilities of the Zilog 8530 Serial Communications Controller, and a Carrier Sense Multiple Access/Collision Avoidance scheme (CSMA/CA) instead of our prototype Manchester modulation with Carrier Sense Multiple Access/Collision Detection (CSMA/CD) scheme.

We are, of course, concerned about the future of the AppleTalk network in the face of competition from higher performance IEEE-Standard networks such as the Ethernet (IEEE 802.3), Token Bus (IEEE 802.4) and Token Ring (IEEE 802.5). We would in principle prefer to use one of those standards, but several obstacles have made that impractical up to the present time. First, reliable silicon support for those standards, with adequate density to meet our board space limitations, is just now becoming available. Unfortunately, the chip sets are generally not designed in a way which permits wire-OR connection of multiple nodes on the FASTBUS backplane (an absolutely essential requirement for any eventual Diagnostic Network standard), and the more compact implementations are rather processor-specific.

In addition, the high data rates are a problem in two ways: the demands on the processor may be so great as to require a co-processor to handle the load of the network, costing additional board space; and, high data rates place stringent requirements on cable length and type which are seriously inconvenient in a typical FASTBUS environment.

It seems clear that there is already sufficient support for AppleTalk that it will be around for some time, and therefore that it has significant advantages compared to using our own unique design. In particular, gateways which connect AppleTalk with Ethernet are becoming available commercially, and this path is probably the simplest and cheapest way to link the Snoops with the DEC VAXes and other computers commonly used in data acquisition systems today.

A software effort is currently beginning at SLAC to implement the AppleTalk network on the Snoop, using Macintosh computers as development tools. Protocols appropriate to the Snoop's functions remain to be understood, selected and implemented. We intend to include the necessary hooks in the Snoop ROM software so that a Snoop can be activated by either a terminal or a PC connected to its front panel, or by the FASTBUS Diagnostic Network connection on the backplane.

The CDF group at the Fermi National Accelerator Laboratory has debugged and used two Snoop Modules under control of a program running on a VAX, using the front-panel terminal connection to the Snoop.[3]

5. CONCLUSION

Basic functional testing of the Snoop Module prototype is almost completed. The module performs very reliably. Additional testing in more complex and higher-speed FASTBUS environments still remains to be done, including actual field testing by users.

During the next several months, minor hardware corrections and completion of the documentation package are planned. A module production run is scheduled during the next 6 months. It is hoped that this pilot production will be performed under a commercial contract. SLAC is planning to make sample quantities from this production run available to the FASTBUS community. The cost for tested Snoop Modules with software on-board is expected to be U.S. $ 6K to 8K.

With the first large multisegment FASTBUS systems now being assembled, we hope that this Snoop Module will become a valuable tool for diagnosing and solving FASTBUS system problems.

ACKNOWLEDGEMENT

We are grateful to R. S. Larsen for his support throughout this project; to Ed Austin for his work on the PC layout; to V. Itani for his work on the drawings; to Fred Schinn for his assistance in the hardware checkout, coordination support, engineering support, wirelist creation and maintenance; and to Sergio Zimmermann of FNAL for his collaboration and assistance in the hardware testing.

REFERENCES

1. H. V. Walz and D. B. Gustavson, "Status of the SLAC Snoop Diagnostic Module for FASTBUS", IEEE Transactions on Nucl. Sci. NS-30, No. 4, 2276 (1983).
2. "Inside AppleTalk", Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014.
3. S. Zimmermann, Fermi National Accelerator Laboratory, "A FASTBUS Logic Analyzer Based on SLAC Snoop Diagnostic Modules", paper presented at this conference.