SLAC - PUB - 4416
September 1987
(I)

AN INTERACTIVE VISUAL INTERFACE FOR FASTBUS DIAGNOSTICS*

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

Abstract

Complex instruments which are not used regularly, such as diagnostic devices, are often avoided or inefficiently used because of the start-up effort required to figure out how to apply them. To solve this problem, we have developed a visual interface for the SLAC FASTBUS Snoop Module.[1,2]This interface is based on the Macintosh computer's standard human interface, using menus, dialogs and graphics to quickly bring to mind the range of possibilities and allow easy setup and manipulation of the Snoop module's powerful features.

Introduction

The Fastbus Snoop module is a specialized logic analyzer for diagnosing problems in Fastbus systems. The variety and interaction of possible operating modes makes the Snoop difficult to use, especially when it is treated as a diagnostic device which is used occasionally when there is trouble in the system. A regular user, such as a module repair technician using the Snoop as his main bench tool, would soon master it, but the proverbial "physicist with trouble at 3 AM" would be annoyed and frustrated by the complexity and unfriendliness of a traditional human interface.

We have developed a visual human interface which uses the graphic display and interactive capabilities of a microcomputer, the Apple Macintosh™. We chose the Macintosh because it has an outstanding reputation for being "user friendly". Its human interface design is so consistent and so powerful that users rarely need to use written documentation; most of the information needed can be found by interactively exploring the screen display. The Mac is also conveniently portable and inexpensive. We hope eventually to use its built-in network capability to control multiple Snoops and coordinate them for faster diagnosis of multi-segment Fastbus problems.

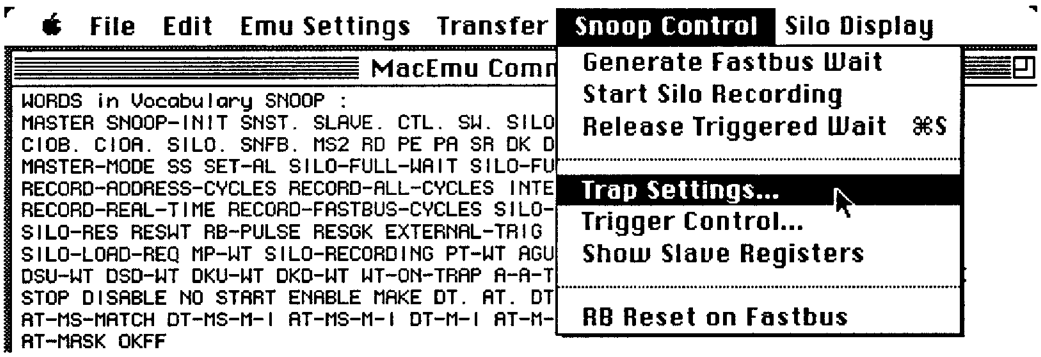
Snoop Hardware

The Snoop module contains a Motorola 68000-based microcomputer running a Forth-79 operating system which has been enhanced to provide control of the Fastbus diagnostic capabilities. The Snoop understands the Fastbus protocol, so it can save a complete record of each Fastbus cycle in a wide, very fast memory. It can also record sequential snapshots of the bus using a realtime clock. It has fast pattern-matching circuitry which allows it to react to specified events on the Fastbus; a match can start or stop recording, or stop activity on the bus. When the recording memory is full, recording can stop or continue (overwriting the oldest information), or the Fastbus activity itself can be halted while the fast memory is copied in order to obtain a complete record of activity larger than the fast memory can hold. The Snoop can generate Wait on the Fastbus in response to a pattern match, a sequence of matches, or transitions on Fastbus timing lines in order to cause bus operations to proceed in a slow single-step mode. The Snoop can act as a bus master, so it can initiate specific bus activity in order to test the response of some suspect device. The Snoop also acts as a slave when addressed.

Prior to this project, human interaction with the Snoop was provided by a "dumb" terminal emulator running on an IBM personal computer. In that environment the user would interact with the Snoop by either typing Forth commands or loading Forth blocks files from the IBM disk. Control of the Snoop Fastbus tools was provided by typing the appropriate command to the Snoop via this interface. We quickly found that this method required a good recall of both the Snoop commands (with exact spelling) and the details of Snoop operation.

Control

The approach we take with the Macintosh instead is to provide a visual representation of the various capabilities of the Snoop. Related functions are grouped together so the user is aware of the range of possibilities and their interrelation. For example, to control Fastbus Wait generation, the user chooses the TRAP SETTINGS... menu item (see Figure 1). The current settings are retrieved from the Snoop and the Macintosh dialog box pictured in Figure 2 appears, filled in to correctly reflect those settings. The user then checks (or unchecks) the appropriate boxes to control the desired Snoop behavior. In the case of the "Trap Mode" function, the three choices are mutually exclusive and the radio button controls work accordingly. Additionally the user may specify hexadecimal numbers for use as addresses to be matched on the Fastbus and may select the action to be taken when matches occur. When the user specifies these addresses in this dialog, the input is filtered so that only valid numbers can be provided. After the user is satisfied with his selections, the SET button is pressed to change the Snoop parameters.

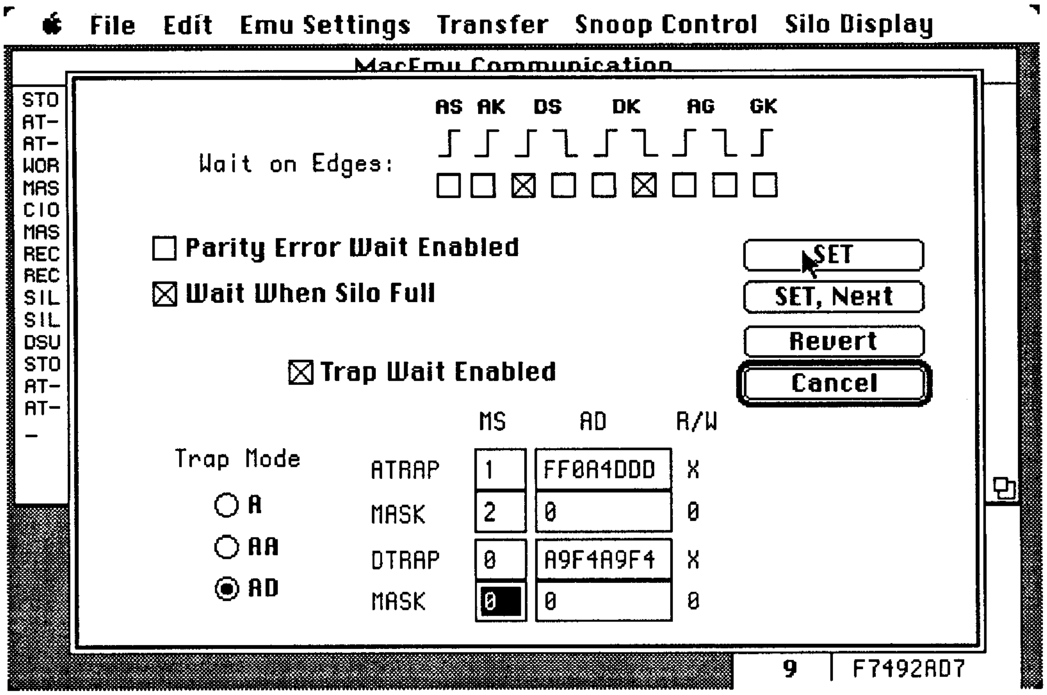


```
WORDS in Vocabulary SNOOP :
MASTER SNOOP-INIT SNST. SLAVE. CTL. SW. SILO
CIOB. CIOA. SILO. SNFB. MS2 RD PE PA SR DK D
MASTER-MODE SS SET-AL SILO-FULL-WAIT SILO-FU
RECORD-ADDRESS-CYCLES RECORD-ALL-CYCLES INTE
RECORD-REAL-TIME RECORD-FASTBUS-CYCLES SILO-
SILO-RES RESWT RB-PULSE RESGK EXTERNAL-TRIG
SILO-LOAD-REQ MP-WT SILO-RECORDING PT-WT AGU
DSU-WT DSD-WT DKU-WT DKD-WT WT-ON-TRAP A-A-T
STOP DISABLE NO START ENABLE MAKE DT. AT. DT
AT-MS-MATCH DT-MS-M-I AT-MS-M-I DT-M-I AT-M-
AT-MASK OKFF
```

Fig. 1. A menu used for controlling Snoop actions. When items with trailing ... are selected, a dialog is presented for further interaction.

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

Poster paper presented at the Nuclear Science Symposium, San Francisco, CA, October 21-23, 1987



Wait on Edges: AS AK DS DK AG GK

□ □ ☒ □ □ ☒ □ □ □

□ Parity Error Wait Enabled

☒ Wait When Silo Full

☒ Trap Wait Enabled

Trap Mode

○ A

○ AA

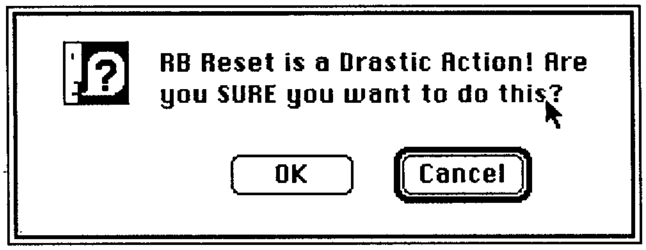
● AD

	MS	AD	R/W
ATRAP	1	FF0A4DDD	X
MASK	2	0	0
DTRAP	0	A9F4A9F4	X
MASK	0	0	0

9 | F7492AD7

Fig. 2. Dialog which provides Snoop control of Fastbus Wait generation. User checks boxes and provides numerical data, then presses the SET button to change the parameters on the Snoop.

Some of the Snoop functions are pulsed actions. Examples of this are the start- and stop-recording for the fast Silo memory, manual generation of Wait on the bus, and a full reset of the Fastbus. Menus provide a simple and convenient method of choosing these functions. Since resetting the Fastbus is a potentially disastrous action, the user is warned before the action is taken and a way out is provided (see Figure 3).



AB Reset is a Drastic Action! Are you SURE you want to do this?

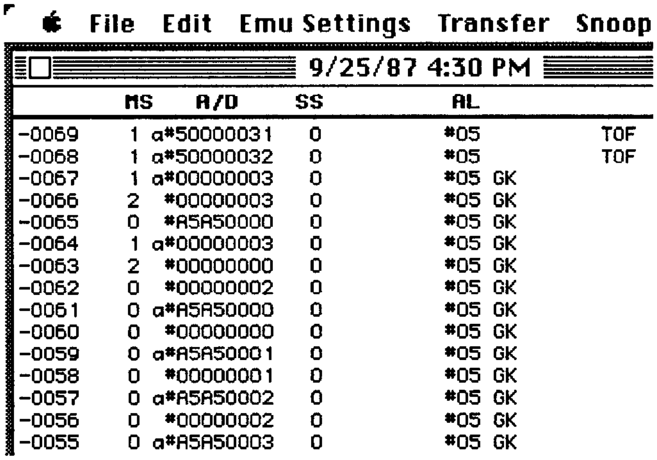
Fig. 3. Alert box warning the user before a drastic action is taken. This helps prevent accidental use of functions which can cause problems.

Display

The fast Silo memory is used to record cycles on the Fastbus, and viewing the Silo is a crucial part of using the Snoop. A Snoop user merely chooses the GET NEW SILO menu item: the full contents of this memory is transfered to the Mac, and a standard Macintosh scrolling window containing the data appears (see Figure 4). This method allows the user to view the entire contents of the Silo without retransmitting the data to the Macintosh. The user may choose to save all or part of this Silo snapshot to a Macintosh disk for later review.

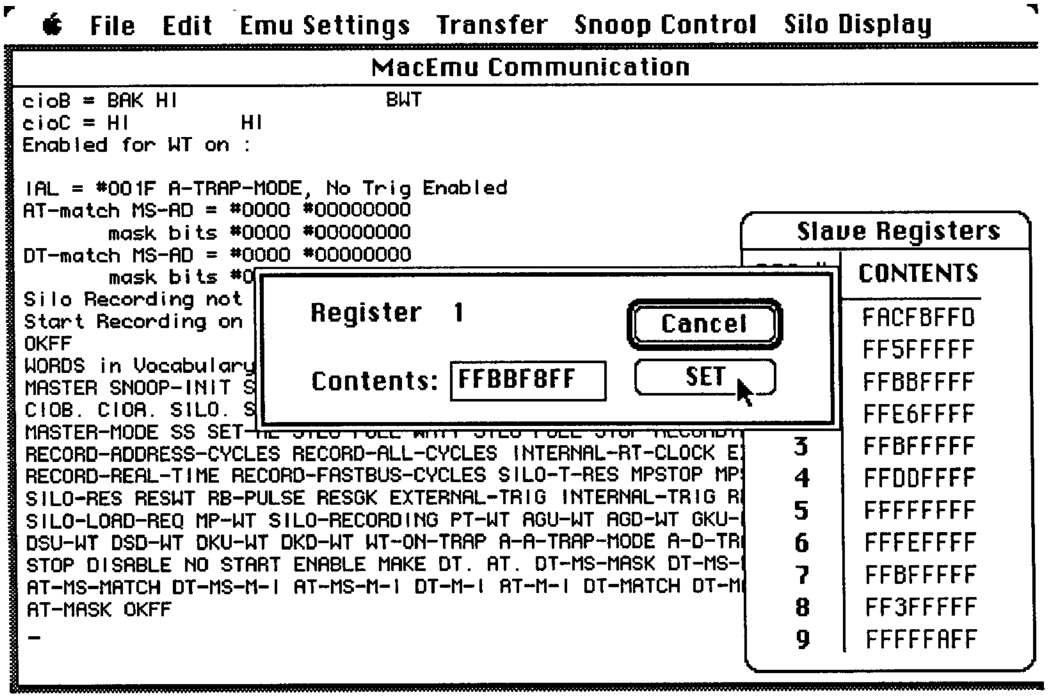
The power of the Macintosh interface makes it possible for users to avoid needless or conflicting effort. In many cases, a specific operation is not available due to the current Snoop status. For example, while the Silo is recording it is not possible to read out the Silo memory. If this is attempted from a dumb terminal, the user receives a message stating that the Silo is recording. When using the Macintosh interface, the GET NEW SILO menu item is dimmed and unselectable in the cases where this action is not allowed. This type of control is used throughout our Macintosh interface and prevents incompatible actions by the user while providing visual clues as to alternate operating modes and the current Snoop status.

Dynamic status information is also used for a "Real-Time" status display of the Slave registers in the Snoop. If these registers are changed by some other Master on the bus by addressing the Snoop and writing data, the Macintosh display is updated accordingly. Additionally the



```
9/25/87 4:30 PM
         MS    A/D       SS          AL
-0069     1  a*50000031   0          *05         TOF
-0068     1  a*50000032   0          *05         TOF
-0067     1  a*00000003   0          *05  GK
-0066     2   *00000003   0          *05  GK
-0065     0   *A5A50000   0          *05  GK
-0064     1  a*00000003   0          *05  GK
-0063     2   *00000000   0          *05  GK
-0062     0   *00000002   0          *05  GK
-0061     0  a*A5A50000   0          *05  GK
-0060     0   *00000000   0          *05  GK
-0059     0  a*A5A50001   0          *05  GK
-0058     0   *00000001   0          *05  GK
-0057     0  a*A5A50002   0          *05  GK
-0056     0   *00000002   0          *05  GK
-0055     0  a*A5A50003   0          *05  GK
```

Fig. 4. Window containing data from the fast Silo memory of the Snoop. This data window can be scrolled to view the entire contents and saved to disk if desired.



```
cioB = BAK HI                    BWT
cioC = HI           HI
Enabled for WT on :

IAL = #001F A-TRAP-MODE, No Trig Enabled
AT-match MS-AD = #0000 #00000000
       mask bits #0000 #00000000
DT-match MS-AD = #0000 #00000000
       mask bits #0
Silo Recording not
Start Recording on
OKFF
WORDS in Vocabulary
MASTER SNOOP-INIT S
CIOB. CIOA. SILO. S
MASTER-MODE SS SET-
RECORD-ADDRESS-CYCLES RECORD-ALL-CYCLES INTERNAL-RT-CLOCK E
RECORD-REAL-TIME RECORD-FASTBUS-CYCLES SILO-T-RES MPSTOP MP
SILO-RES RESWT RB-PULSE RESGK EXTERNAL-TRIG INTERNAL-TRIG R
SILO-LOAD-REQ MP-WT SILO-RECORDING PT-WT AGU-WT AGD-WT GKU-
DSU-WT DSD-WT DKU-WT DKD-WT WT-ON-TRAP A-A-TRAP-MODE A-D-TR
STOP DISABLE NO START ENABLE MAKE DT. AT. DT-MS-MASK DT-MS-
AT-MS-MATCH DT-MS-M-I AT-MS-M-I DT-M-I AT-M-I DT-MATCH DT-M
AT-MASK OKFF
-
```

Register 1

Contents: FFBBF8FF

	CONTENTS
	FACF8FFD
	FF5FFFFF
	FFBBFFFF
	FFE6FFFF
3	FFBFFFFF
4	FFDDFFFF
5	FFFFFFFF
6	FFFEFFFF
7	FFBFFFFF
8	FF3FFFFF
9	FFFFFAFF

Fig. 5. Interactive modification of a Snoop Slave register. On the right is the window display of all the Slave registers. This window is dynamically updated to reflect the current register contents.

Snoop user can interactively change these registers for readout by another module. Instead of issuing direct Forth commands, the user can edit the value of a register on the Macintosh and it will be rewritten to the Snoop (see Figure 5).

"Dumb" Terminal Window

From within this Macintosh environment, we continue to provide users with a "dumb" terminal interface which enables execution of Snoop Forth commands. A powerful capability of the Snoop is the ability of users to write programs in Forth and execute them on the Snoop. The ability to write Forth programs is invaluable in module-test situations. For example it is easy to write a loop which will produce counting sequences of data and write them into the module under test:

```
: TST-LOOP 0 BEGIN 1+ DUP <reg> <addr> FWC
?TERMINAL UNTIL DROP ;
```

"FWC" is the name of the *Fastbus Standard Routines*[3] subroutine which performs a Write to Control space. The user can use a standard text editor on the Macintosh to edit the Forth programs. Selection of a menu command will then transfer and load these programs into the Snoop. This approach provides the user with a familiar text editing environment and eliminates the need for using the restrictive Forth blocks mechanism.

Implementation

The Macintosh computer has a well deserved reputation for being difficult to program. To speed the development process, we chose the Mach 2 Forth development system by Palo Alto Shipping Company. This excellent system consisted of an interactive environment which made prototyping and testing of code modules quick and efficient. In addition, Mach 2 provided a multi-tasking framework which aided program design and added extra functionality to our finished program. Mach 2 is the only Forth for the Macintosh which compiles to true 68000 instructions (traditional Forth interpreters compile to pseudo–code and use an inner interpreter for execution), and this was important since speedy execution is crucial for smoothness in our Macintosh interface. The execution time of the resulting compiled code from Mach 2 is comparable to that of code from languages such as C and Pascal.[4] We used a Macintosh Plus computer, but the code will run on any Macintosh from a Mac 512KE to a Macintosh II.

Conclusions

The development of our interface is still ongoing. Field testing of the Snoop/Macintosh package is just beginning and we expect the valuable feedback we receive to help us refine the interface further. We already find that the use of the Macintosh greatly simplifies our use of the Snoop module. While using the Snoop in this Macintosh environment, we are not forced to consider the full complexities of the Snoop module; rather we can concentrate on the debugging tasks at hand. Nevertheless, the full Forth system is still available to us whenever we choose it.

Our experience with the Snoop/Macintosh system has convinced us that our approach is sound; the system is much easier to use with the Macintosh interface than with a command style interface. Similar results could be obtained by using a menu interface on some other computer, but the Macintosh's graphical interface is superior; we have not reached its limits yet.

References

1) ANSI/IEEE Standard 960-1986, "FASTBUS Modular High-Speed Data Acquisition and Control System for High Energy Physics and Other Applications".

2) D.B.Gustavson and H.V.Walz, "SLAC FASTBUS Snoop Module-Test Results and Support Software", IEEE Trans. on Nucl. Sci. NS-33, No.1, p. 811 (1986).

3) U.S. NIM Committee, Fastbus Standard Routines, Department of Energy Report DOE/ER-0325, March 1987.

4) L.Chavez, "A Fast Forth for the 68000", Dr. Dobbs Journal of Software Tools, vol. 12, pp. 32-36, October 1987.