SLAC – PUB – 4425
October 1987
(I)

A CAMAC AND FASTBUS ENGINEERING TEST ENVIRONMENT SUPPORTED BY A MICROVAX/MICROVMS SYSTEM [*]

C. A. Logg

Stanford Linear Accelerator Center, Stanford University, Stanford, California 94305

Abstract

A flexible, multiuser engineering test environment has been established for the engineers in SLAC's Electronic Instrumentation Engineering group. The system hardware includes a standard MicroVAX II [1] and MicroVAX I with multiple CAMAC, FASTBUS, and GPIB instrumentation buses. The system software components include MicroVMS licenses with DECNET/SLACNET, FORTRAN, PASCAL, FORTH, and a versatile graphical display package. In addition, there are several software utilities available to facilitate FASTBUS and CAMAC prototype hardware debugging.

Introduction

For the past 10 years the engineering and technician staff of SLAC's Electronics Instrumentation Group (EIN) have been using stand-alone LSI-11 Q-bus systems [2] for their work which includes the prototype checkout, production test, and maintenance of high energy physics and accelerator control instrumentation. These LSI-11 systems usually run a stand-alone version of FORTH [3] developed at SLAC as their primary software system. Occasionally RT-11 is also used. The stand-alone characteristics of the LSI-11 FORTH have allowed direct access to the LSI-11 hardware and the interfaces to CAMAC [4] and FASTBUS [5] without fighting the protection of an operating system. Now however, given dramatically increased complexity of the instrumentation being developed, there is a need to run in a more sophisticated environment where there is access to the facilities of mathemetical libraries and languages better suited to instrumentation performance analysis.

Some of the problems involved in changing systems in this environment include user retraining, and expensive equipment purchases to support CAMAC and FASTBUS on a new system. VAXs have been heavily used at SLAC for years, and so when the MicroVAX line, with it Q-bus compatibility became available, it seemed an obvious choice. Due to budget constraints, it has been necessary to initially equip the MicroVAX systems with FASTBUS and CAMAC interface hardware salvaged from the LSI-11 Q-bus systems.

The following sections describe the system(s) which have evolved to date, as well as proposed plans for the future.

Overview

Currently EIN has about 600 square feet of laboratory space dedicated to 8 test stands. Four test stands (with a total of 3 CAMAC crates and 3 FASTBUS crates) are supported by a MicroVAX II (known as ELDU2). Two test stands (one with a CAMAC crate and one with a FASTBUS crate) are supported by a MicroVAX I (known as ELDU3), and one test stand (with one CAMAC crate) is supported by a stand-alone LSI-11/73 system. One test stand has no computer facilities associated with it, but has manual CAMAC and FASTBUS test boxes. The allocation of the CAMAC and FASTBUS crates on ELDU2 and ELDU3 is flexible so that they can be assigned to the engineers, technicians, and programmers (which will jointly be referred to as the users from hereon) as needed.

Hardware Configuration

ELDU2 (Fig. 1) is a standard MicroVAX II (in a BA123 box with a 12-slot backplane) with:

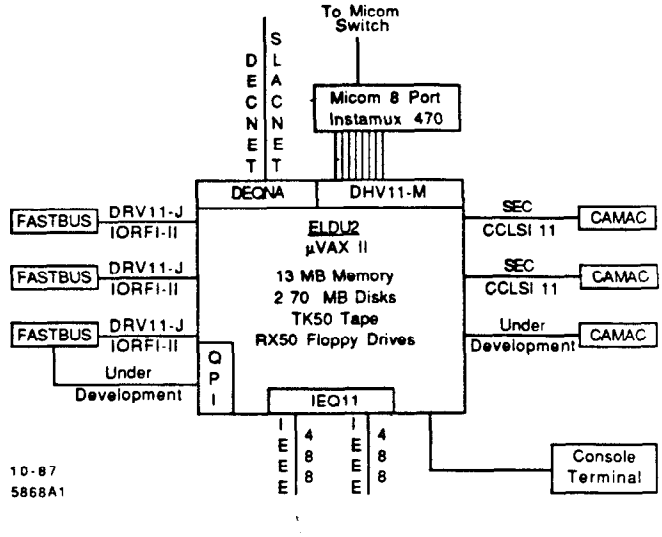

Fig. 1. ELDU2 MicroVAX II Configuration

13 MB memory
RQDX3 disk controller supporting:
two 70 MB RD53 Winchester disks
dual RX50 floppy disk drives
TK50 cartridge tape drive for system backup and software installation and distribution
DEQNA Ethernet interface
DHV11-M 8-port serial line multiplexer
IEQ11 which provides 2 independent IEEE-488 instrument buses
2 SEC (now DSP) CCLSI-11 CAMAC crate controllers
3 DRV11-J/IORFI-II FASTBUS Interfaces
QPI (Q-bus Processor Interface) for FASTBUS
a console terminal

and one CAMAC crate connected by crate controllers currently under evaluation.

The 8 DHV11-M ports are connected via an 8-port Micom Instamux to the SLAC Micom switch [6]. In the EIN laboratory area with the test stands are several terminals which are also connected to the SLAC Micom switch, as are the terminals in the users' offices. The test stands supported by ELDU2 can thus be accessed and utilized by the users in the EIN laboratory itself, from their offices, or from any other terminals at SLAC which are connected to the Micom switch.

ELDU3 is a standard MicroVAX I [in a BA23 (8-slot backplane) rack mount chassis] with:

4 MB memory
RQDX2 disk controller supporting:
one 30 MB RD52 Winchester disk
one 10 MB RD51 Winchester disk
dual RX50 floppy disk drives
DEQNA Ethernet interface
DZV11 4-port serial interface
DSD-880 30 MB Winchester (which looks like 3 RL02s to VMS)
1 SEC (now DSP) CCLSI-11 CAMAC crate controller
1 DRV11-J/IORFI-II FASTBUS Interface
a console terminal

Two of the ELDU3 DZV11 ports are connected to the SLAC Micom switch and thus ELDU3 has all the user access capabilities discussed for ELDU2 above.

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

Presented at the Nuclear Science Symposium, San Francisco, California, October 23-25, 1987

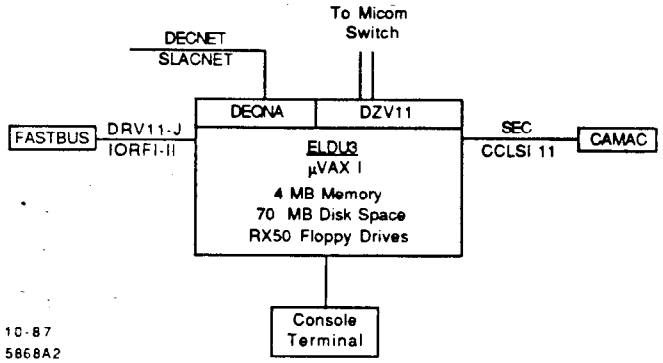

Fig. 2. ELDU3 MicroVAX I Configuration

Hardcopy (text and graphics) output for ELDU2 and ELDU3 are provided via DECNET/SLACNET [7] file transfer facilities which allow a file to be transferred for printing to any printer connected to the SLAC network. Locally, in our building, an IMAGEN printer located down the hall from the EIN laboratory is usually used.

Software Configuration

Both systems are running VMS, and have DECNET/SLACNET, FORTH [8], and the versatile SLAC histogram and graphical display packages of HANDYPAK [9] and Unified Graphics [10] available. ELDU2 also has FORTRAN, PASCAL, and the IEQ11 VMS driver licenses. Currently the only drivers available on ELDU2 and ELDU3 for CAMAC and FASTBUS are FORTH drivers. Formal VMS drivers are in the process of being imported and/or developed.

FORTH

A version of FORTH which runs under VMS has been developed at SLAC. Its model is closest to the 79–Standard. Given the powerful Record Management Services of VMS, the traditional BLOCK/SCREEN structuring of FORTH has been replaced by a more natural sequential file organization with variable length records. Full access to most of the features of VMS is permitted. For example, when debugging a FORTH program, a user can SPAWN a task and access his/her favorite editor, modify and/or generate more code, and then after returning to FORTH, use the FORTH 'FORGET' feature, and reload the modified code. Run-time linkage from FORTH to entry points in VMS sharable images is permitted, and thus the user can link to procedures written in MACRO–32, FORTRAN, and PASCAL. The VMS Debugger can even be used to debug FORTH code.

Software Utilities

Several software utilities have been written in FORTH for use in CAMAC and FASTBUS instrumentation checkout. These include scope loop utilities for CAMAC and FASTBUS, and general FASTBUS module and memory tests for FASTBUS instrumentation.

CAMAC Utilities

For CAMAC there is the menu shown in Fig. 3 for specifying C,N,A,F functions which can then be used in various combinations in loops for scoping.

FASTBUS Utilities

The FASTBUS utilities are more extensive, and include not only line wigglers for scope loops, but also some general FASTBUS interface and memory tests.

```
*************************************************************
***MENU-CAMAC Line Wigglers ( restart with: *CLW )

C,N,A,F,W DEFAULTS FOR THESE TESTS ARE:0 0 0 0 #00000000
CNAF1,DATA1= 0 0 0 0 #00000000
CNAF2,DATA2= 0 0 0 0 #00000000

 ITEM #     DESCRIPTION
 ------     -----------
   1 ------ SYSTEM-HELP
   2 ------ GO-TO-FORTH
   3 ------ EXIT-THIS-MENU
   4 ------ SET-CNAF
   5 ------ SET-CNAF1
   6 ------ SET-CNAF2
   7 ------ MAP-A-MODULE
   8 ------ C-Z-LOOP
   9 ------ WIGGLE-A-LINES
  10 ------ WIGGLE-F-LINES
  11 ------ WIGGLE-W-LINES
  12 ------ W-LINES-CNAF2-LOOP
  13 ------ CNAF1-LOOP
  14 ------ CNAF1-CNAF2-LOOP
  15 ------ INC-THRU-A
  16 ------ INC-THRU-F

*************************************************************
        MENU ITEM # OR <cr> FOR MENU =
```

Fig. 3. CAMAC Line Wiggler Utility

Figure 4 shows the menu for the line wigglers (for scope loops). This menu allows the user to specify primary and secondary address cycles, and the type of read and write cycles that are to be performed in high speed loops. It also enables the user to specify various combinations (even illegal combinations) of protocol lines which can then be wiggled in a loop.

```
*********************************************************************
***MENU-FASTBUS Line Wigglers ( restart with: *FLW )

FASTBUS Crate Port # = 1   Module Slot # = 15   Module ID = #00190000
OPTIONS:
Precede test with CS Primary Address Cycle ( Primary Address = 15 )
Precede test with a write NTA Cycle:  ( NTA = #00000000 )
Use AD = #00000000 for control line test
Control lines toggling during test are: DS SS=0  MS=0
Lines up are: AS SS=0  MS=0


ITEM #    DESCRIPTION
------    -----------
 1 ------ SYSTEM-HELP
 2 ------ GO-TO-FORTH
 3 ------ EXIT-THIS-MENU
 4 ------ *SET-FASTBUS_Crate_Port_#
 5 ------ *SET-Module_Slot_#
 6 ------ *SET-Address_Cycle_Option
 7 ------ *SET-NTA_Option
 8 ------ *SET-NTA
 9 ------ *SET-Test_AD_Values
10 ------ *SET-Line Test States

11 ------ FBINIT
12 ------ FBMAP

13 ------ CONTROL_LINE_WIGGLER
14 ------ AD_LINE_WIGGLER
15 ------ Loop over: address cycle, optional NTA cycle, write & read
16 ------ Write and read cycle loop
17 ------ SELECTIVE_SET&CLEAR_LOOP
18 ------ Loop over all lines, wiggling each, one at a time

*********************************************************************
       MENU ITEM # OR <cr> FOR MENU =
```

Fig. 4. FASTBUS Line Wiggler Utility

```
*************************************************************************
***MENU-General FASTBUS Memory Test ( restart with: *GMT )

FASTBUS Crate Port # = 1      Module Slot # = 15      Module ID = #00190000
Selected test pattern is: Alternating address and complement of the address.
Least Significant bit is 0      Most significant bit is: 31
Starting NTA is: #00000000 in DATA Space.
Number of FASTBUS words to be tested is: #00000000

 ITEM #     DESCRIPTION
 ------     -----------
   1 ------ SYSTEM-HELP
   2 ------ GO-TO-FORTH
   3 ------ EXIT-THIS-MENU
   4 ------ *SET-FASTBUS_Crate_Port_#
   5 ------ *SET-Module_Slot_#
   6 ------ *SET-SLOW_Block_Transfers
   7 ------ *SET-VERY-FAST_Block_Transfers
   8 ------ *SET-Parameters for the Memory Test

   9 ------ FBINIT
  10 ------ FBMAP
  11 ------ TRACE-ON
  12 ------ TRACE-OFF
  13 ------ PARITY-ON
  14 ------ PARITY-OFF

  15 ------ Select pattern: Incremental pattern
  16 ------ Select pattern: Sliding 0 in field of 1's
  17 ------ Select pattern: Sliding 1 in field of 0's
  18 ------ Select pattern: Alternating 1's and 0's
  19 ------ Select pattern: Alternating addr and complement of addr

  20 ------ Test_Memory_W/Block_Transfers
  21 ------ Test_Memory_WO/Block_Transfers
  22 ------ Test_Memory_W/All_Patterns
  23 ------ Loop on the Memory Test W/Block Transfers
  24 ------ Loop on the Memory Test WO/Block Transfers

*************************************************************************
```

Fig. 5. General Memory Test Utility for FASTBUS Instrumentation

```
*************************************************************************
***MENU-Basic FASTBUS Module Tests ( restart with: *BMT )

FASTBUS Crate Port # = 1      Module Slot # = 15      Module ID = #00190000

 ITEM #     DESCRIPTION
 ------     -----------
   1 ------ SYSTEM-HELP
   2 ------ GO-TO-FORTH
   3 ------ EXIT-THIS-MENU
   4 ------ *SET-FASTBUS_Crate_Port_#
   5 ------ *SET-Module_Slot_#
   6 ------ Display_the_Configuration
   7 ------ FBINIT
   8 ------ FBMAP
   9 ------ TRACE-ON
  10 ------ TRACE-OFF
  11 ------ Check the Crate Segment and all Geographic Addressing

  12 ------ Test_CSR_Space_Addressing
  13 ------ Test_CSR_Space_NTA_Register
  14 ------ Check_CSR_0
  15 ------ Map_Lower_CSR_Space_(0-#200)
  16 ------ Test_CSR_Selective_Set&Clear
  17 ------ Test_Single_CSR_Registers
  18 ------ Test CSR Register and Memory Ranges

  19 ------ Test_Data_Space_Addressing
  20 ------ Test_Data_Space_NTA_Register
  21 ------ Test_Data_Space_Memory

*************************************************************************
```

Fig. 6. Basic FASTBUS Module Test Utility

Figure 5 displays the menu which interfaces the user to the general purpose memory test utility.

These memory tests are currently implemented via a DRV11-J/IORFI-II [11] interface to FASTBUS. The 'SLOW' block transfers referred to in the menu perform full SS code and protocol checking, while the 'VERY-FAST' block transfers are just DS toggles. The user specifies the width of the memory to be tested, the starting NTA, the number of FASTBUS words to be tested and the type of pattern to be used. The memory tests can also be done via random write and read cycles. Ultimately these tests will also be implemented using some type of high speed FASTBUS Master to test FASTBUS module memories at full FASTBUS speeds.

Figure 6 displays the menu which interfaces the user to the Basic Module tests.

The information for the Basic Module Tests is specified in a configuration file which indicates in what space and address range(s) the module's memory lies, the memory's width, which CSR registers have selective set and clear bits, and what other CSR registers (and their widths) a module has. Note that there may be special conditions which affect the operation of things such as the selective set and clear bits, which cannot be taken into account by this general program.

These utilities are expected to continue to evolve as they receive use and their shortcomings are recognized. Currently they only operate through a DRV11-J/IORFI-II interface to FASTBUS. In the near future the FASTBUS general module and memory tests will be implemented for some high speed FASTBUS Master.

Timing Information

For the FASTBUS scope loops (run via the menu shown in Figure 4), the table shown in Fig. 7 lists some of the timing differences between ELDU2 (the MicroVAX II) and ELDU3 (the MicroVAX I) using a DRV11-J/IORFI-II FASTBUS Interface. Both systems were running exactly the same code.

These timing statistics are significant as they indicate that MicroVAX Is, which many people may consider obsolete, can be effectively used to support an instrumentation test stand.

Observations Concerning DMA and Non-DMA FASTBUS and CAMAC Interfaces

In establishing an electronics instrumentation test environment such as is described here, careful consideration should be given to the functional variations provided by DMA and non-DMA CAMAC and FASTBUS interfaces/controllers. There are advantages and disadvantages to both, and they fulfill different needs. I have had experience with the following FASTBUS interfaces: the DRV11-J/IORFI-II which is a non-DMA FASTBUS interface, and the QPI [12] which is a DMA FASTBUS interface; and the following CAMAC interfaces: SEC (now DSP) CCLSI11 crate controller [13] which is a non-DMA CAMAC interface, and the Kinetic Systems 2922/3922 [14] which is a DMA CAMAC interface/crate controller. FASTBUS and CAMAC DMA interfaces are currently under evaluation.

FASTBUS Loop Contents	MicroVAX I	MicroVAX II
Random Cycle (Read or Write)	12.9 Microseconds	8.7 Microseconds
Random Write Cycle followed by a Random Read Cycle	18.4 Microseconds	14.0 Microseconds
Primary Address, NTA write, Random Write Cycle, Random Read Cycle	98.8 Microseconds	90.9 Microseconds

Fig. 7. Some MicroVAX I and MicroVAX II Timing Comparisons

In the test environment described herein, the non-DMA devices are dedicated to a single user and mapped directly into the user's program space. This avoids the protection of the operating system and provides the same direct access to and absolute control of the FASTBUS and CAMAC instrumentation buses as is available on the LSI-11 test stand systems. In the case of the DRV11-J/IORFI-II, one is able to manipulate the FASTBUS protocol and data lines in almost any manner desired. Almost any combination (legal and illegal) of FASTBUS protocol lines can be manipulated, and it is possible to generate bad parity in the data to test parity detection by the equipment under test. Other advantages of non-DMA devices include: no fancy drivers are needed, and they are generally cheaper than DMA devices. Disadvantages of non-DMA devices include: they are generally much slower and require more CPU cycles to operate.

Advantages of DMA interfaces include: they are much faster than non-DMA devices, and usually require fewer CPU cycles (compared to non-DMA devices) to transfer large blocks of data. Disadvantages include: they are usually more expensive, and require expensive software drivers, even if used in a single user mode. DMA devices with sophisticated drivers can be use in almost any environment as they generally protect the users from each other. In our environment, with our non-DMA interfaces and simple FORTH drivers, we rely heavily on the benevolent nature of our users.

DMA FASTBUS and CAMAC interfaces are currently under evaluation and development. Ultimately it is hoped to provide a variety of DMA (for high speed) and non-DMA (for flexibility and absolute control) access to the FASTBUS and CAMAC crates in the system.

Disadvantages and Advantages of this Test Environment

Some of the disadvantages of this test environment include:

- On ELDU2 the scope traces can get faint when there are multiple users on the system chewing significant amounts of CPU cycles. So far, in our test environment, this has not been a problem. Most test bench computer usage tends to be at human interaction speeds rather than all out number crunching speeds. Such a consideration should however, be taken into account if number crunching applications are being considered for porting to a test environment support system.
- There are two VMS systems (ELDU2 and ELDU3) to manage. System management includes disk maintenance, operating system and layered product updates and maintenance, user account maintenance, system and user file backup, and peripheral hardware trouble shooting and repair, just to name a few. The complexity of VMS and the associated system hardware makes trouble shooting much more difficult than it is on the stand-alone LSI-11 systems.
- On ELDU2, the BA123 box (and 12-slot backplane) is completely full and can not be expanded. Internally, the system is a cabling nightmare due to the large number of cables required to connect and support the tape, disk, terminal ports, FASTBUS, CAMAC, and IEQ11 interfaces and peripherals.
- When a MicroVAX system is down, there are no computer facilities for its associated test stands.
- The test stands are not portable. They must remain in close proximity to the CPU. The LSI-11 test stands are self contained and can be easily moved anywhere they were needed.

Some of the advantages (particularly over the LSI-11 systems) are:

- It is much easier to keep everyone using the latest version of the FORTH system and support code. With the LSI-11, there were 30 or more copies to keep up to date.
- It is easier for the users to share code and utilities among themselves.
- Since the VMS operating system is being used, a greater variety of software tools can be made available to the users.
- The FORTH-FORTRAN connection means that basic hardware exercising and data collection can be done in FORTH, and the data can be passed to FORTRAN programs for analysis.
- It is now possible to run on our systems and in our test environment, some of the same code which is run by the experimenters. Also, our code can now, in many cases, be transported to their VMS systems if desired.

Current Shortcomings and of Future Improvements

The primary shortcomings at this point have to do with the lack of DMA CAMAC and FASTBUS interfaces and an established way to test FASTBUS modules at full FASTBUS speeds. In addition, due to the lack of a DMA CAMAC interface, it is impractical to test large memory boards (for which we usually build CAMAC testers) on the MicroVAX systems. This is however, being looked into. In the near future it is hoped to have established support facilities for high speed FASTBUS Masters such as SLAC Scanner Processors [15] and Aleph Event Builders [16]. We also plan to provide DMA FASTBUS and CAMAC interfaces and their VMS drivers.

Acknowledgments

During the development of this test environment, I have received valuable assistance from many of my colleagues including: Dave Andres, Dave Bozarth, Pat Clancey, Tom Dean, Teresa Downey, Fred Griffith, Mike Huffer, Terry Hung, Steve MacKenzie, Jeff Olsen, Owen Saxton, Mike Sullenberger, Cathy VanIngen, and others. In particular, I would like to thank Leo Paffrath for his support and encouragement.

References

[1] MicroVAX I, MicroVAX II, MicroVMS, DECNET, LSI-11, RT-11, VAX, and VMS are registered trademarks of Digital Equipment Corp.

[2] J. Kieffer, C. A. Logg, D. Farwell, "The Support and Utilization of the LSI-11 Processor Family at SLAC," Proc. of the Digital Equipment Computer Users Soc., Spring 1981, SLAC-PUB-2727, Apr 1981.

[3] C. A. Logg, "SLAC ELD LSI-11 FORTH User's Manual," ELDDOC # 4, Oct. 1985, SLAC.

[4] ANSI, "CAMAC Instrumentation and Interface Standards," ANSI/IEEE STANDARD 583-1982.

[5] ANSI, "IEEE Standard FASTBUS Modular High-Speed Data Acquisition and Control Standard," ANSI/IEEE Standard 960-1986.

[6] R. L. A.Cottrell, Hans Frese, Tim Streater, "SCS Networking Guide," Oct. 1985, SCS Networking Group, SLAC.

[7] R. L. A. Cottrell, T. Downey, H. Frese, C. Granieri, M. Huffer, L. Moss, T. Streater, O. Saxton, D. Wiser, "SLACnet—Implementation and Experiences," SLAC-PUB-3894, Mar. 1986.

[8] C. A. Logg, "VAX/VMS CALFORTH Reference Manual and User's Guide," ELDDOC # 70, SLAC.

[9] A. Boyarski, "HANDYPAK—A Histogram and Display Package (Release 6.3)," SLAC-PUB-234 (Rev), June 1986.

[10] Robert Beach, "Unified Graphics," CGTM No. 170, Computation Group, SLAC, 1976.

[11] C. A. Logg, "I/O Register to FASTBUS Interface II (IORFI II) Description," ELDDOC # 68, Nov 1984, SLAC. This was adapted from the one by Rotolo and Chappa of FERMILAB.

[12] KineticSystems Corporation, "FASTBUS Q-bus Processor Interface F914," KineticSystems Corporation, Lockport IL.

[13] DSP Technology, "CC-LSI-11 User's Manual," May 1978, DSP, Fremont, CA.

[14] KineticSystems Corporation, "Model 2922 Q-bus Adapter" and "Model 3922-Z1A Crate Controller," 1987, KineticSystems Corporation, Lockport IL.

[15] H. Brafman, T. Glanzman, A. J. Lankford, J. Olsen, L. Paffrath, "The SLAC Scanner Processor: A FASTBUS Module for Data Collection and Processing," IEEE Transactions on Nuclear Science (NS-32), No. 1, Feb 1985.

[16] R. Benetta, A. Marchioro, G. McPherson, W. von Ruden, "The ALEPH Event Builder," Proc. of the FASTBUS Software Workshop at CERN, Sep 1985.