

WindoWorks

A Flexible Program for Computerized Testing of Accelerator Control System Electronic Circuit Boards

Jeff Utterback

Fermi National Accelerator Laboratory*
POB 500, MS 307, Batavia, IL 60510 USA

Abstract

Since most accelerator control system circuit boards reside in a commercial bus architecture, such as CAMAC or VMEbus, a computerized test station is needed for exercising the boards. This test station is needed for the development of newly designed prototypes, for commissioning newly manufactured boards, for diagnosing boards which have failed in service, and for long term testing of boards with intermittent failure problems. WindoWorks was created to address these needs. It is a flexible program which runs on a PC compatible computer and uses a PC to bus crate interface. WindoWorks was designed to give the user a flexible way to test circuit boards. Each test is encapsulated into a window. By bringing up several different windows the user can run several different tests simultaneously. The windows are sizable, and moveable. They have data entry boxes so that the test can be customized to the users preference. The windows can be used in conjunction with each other in order to create super-tests. There are several windows which are generic. They can be used to test basic functions on any VME (or CAMAC) board. There are other windows which have been created to test specific boards. New windows for testing specific boards can be easily created by a Pascal programmer using the WindoWorks framework.

I. MOTIVATION

Most modern accelerator control system circuit boards reside in a commercial bus crate such as CAMAC, VMEbus, or Multibus. Due to the nature of the bus architecture, an intelligent bus master is required to exercise the boards in the crate. An intelligent bus master is a circuit board which either has an on board CPU or is controlled by a remote computer.

When developing a newly designed circuit board, commissioning a newly manufactured circuit board, or diagnosing a board which has failed, some sort of testing facility is required. A minimum testing facility configuration consists of a test crate, a bus master, and a computer program which allows a human to conduct various tests on the circuit board of interest.

The test software should be easy to use and flexible enough for the operator to devise tests in unique ways in order to solve particular problems. The test software should have basic functions such as reading from and writing to an address, as well as more sophisticated tests for exercising

overall circuit board functions. The test software should have tests which constantly repeat so as to provide stimulus when using an oscilloscope to diagnose a failure. It should also be able to run unattended for long periods of time so that intermittent failures can be found.

When investigating suitable computer programs for conducting tests, very few commercial products were found. The products which are available, such as VMEbug [1] are rather tedious to use and limited in capabilities. When no suitable commercial programs could be found for testing circuit boards, WindoWorks was created.

II. HARDWARE

The original WindoWorks test station was configured to test newly designed VMEbus circuit boards. Later, a CAMAC version of the test system was also created.

A. VMEbus Test Station

The hardware for the VMEbus test station consists of a VME crate, a BiT3 PC to VMEbus adaptor [2], and a PC/AT compatible computer with mouse.

A PC computer is used in the test station because they are readily available, relatively inexpensive, and have a wide selection of software tools available. Also, the PC's color graphics and mouse helps to make a nice user interface.

The BiT3 PC to VMEbus adaptor consists of a circuit board which goes into one of the PC's expansion slots, another circuit board which goes into the VME crate, and an interconnecting cable. The BiT3 interface has several command and status registers to configure and use the system. A program running on the PC can access any address on the VME crate using a paged memory mapping scheme.

The BiT3 board which resides in the VME crate acts as a bus master to access all other boards in the crate. It can also be configured to be the "slot one controller". All types of bus cycles can be performed including interrupt acknowledge cycles. Both normal and "short I/O" address space can be accessed. The VME address modifier can be set to any value using a command register.

B. CAMAC Test Station

The hardware for the CAMAC test station consists of a CAMAC crate, a Tevatron Serial Crate Controller [3], a TSCC Link Driver [4], a PC to Link Driver interface card [5], and a PC/AT compatible computer with mouse.

The PC/AT was chosen for the same reasons as in the VMEbus test station, and to keep the standard human interface that users had become accustomed to using with the VMEbus test station. The TSCC and Link Driver are

*Operated by the Universities Research Association under contract with the U.S. Department of Energy.

standard Fermilab equipment used for all CAMAC crates in the field. The PC to Link Driver interface card was designed at Fermilab for the sole purpose of creating flexible modern CAMAC test stations. It resides in a PC expansion slot and is connected to the Link Driver via ribbon cables.

A program running on the PC uses registers on the interface card to invoke the standard CAMAC function codes. In this manner, the PC can send commands to or read data from the circuit boards in the crate.

III. SOFTWARE

The program which runs on the PC was written in Pascal using the Turbo Pascal [6] compiler and development environment. The program also makes use of a windows programming package called MetaWINDOW [7].

Because of the differing nature between CAMAC and VME two separate versions of the software have been created. The user interface is the same on both versions but the hardware drivers and test windows are necessarily different. In order to avoid confusion the rest of this paper will be constrained to the discussion of the VMEbus version.

A. User Interface

WindoWorks was created with a modern "mouse and windows" user interface. The interface eliminates the need for tediously typing in commands. It utilizes pull down menus and scroll boxes.

Each test is encapsulated into a window. The user can create and delete window tests at will. The windows are moveable and sizeable. Most of the test windows only need a small area on the CRT. This allows the user to run several tests at the same time or in conjunction with each other. The tests running in the windows generally keep repeating until stopped by the user. This feature is useful when stimulus is needed while diagnosing a problem with an oscilloscope. The windows execute their tests in a round robin fashion. Usually the tests execute fast enough so that all windows appear to be running simultaneously.

Once a suite of windows has been set up and configured, the entire setup can be saved on the hard disk. This allows for quick retrieval of preconfigured setups.

B. Generic Windows

Several test windows have been designed so that they would be useful for testing any VMEbus circuit board. These windows test basic functions which are common to all VMEbus boards. The generic VMEbus windows include the Hex Memory Dump Window, the Binary Memory Dump Window, the Memory Test Window, the Block Fill Window, the Write Burst Window, the Watcher Window, the Matcher Window, the Interrupt Acknowledge Window, and the Text Window.

The Memory Dump Window is the most used of all the generic windows. The Memory Dump Window constantly reads an area of memory and displays it in the window. The user chooses the starting address and the size of the memory

block displayed. The memory block size is chosen by changing the size of the window. The user can display one byte of data by making the window very small or several hundred bytes by making the window large. The user can choose to display the data in bytes, words, or longwords, in hex or in binary. If any data changes in the memory, the new value is immediately displayed in the window. The user can also write new data to a specific address by simply clicking the mouse on the data of interest. After choosing the data location to alter, the user can type in new data, or scroll the data up and down by using the left and right mouse buttons. The Memory Dump Window is useful for testing basic circuit board read and write functionality. It is also useful for watching the progress of other test windows such as the Memory Test Window or the Block Fill Window. Since the Memory Dump Window runs constantly, the user can diagnose problems with an oscilloscope without having to keep entering read commands like some commercial test programs require.

The Memory Test Window writes data to a user specified block of memory and then reads the data back to verify the write-read functionality. If the read back process finds an error, an error message is displayed in the window with the offending address and data. The user can specify byte, word, and longword access. The user can also specify one of 3 test schemes. The first scheme writes the same user specified value to each location on each pass. The second scheme writes the same value to each location, but chooses a new random value on each successive pass. The third and most stalwart scheme writes a different random value to each location on every pass. Each of the three schemes can be useful depending on the particular problem that is being diagnosed.

The Block Fill Window writes user specified data to a block of memory. This window is useful for finding write cycle problems and for initializing memory or command registers. The window continues to keep writing the block of data over and over until stopped by the user so that diagnosis can be done. This window can be used in conjunction with the Memory Dump Window to verify that the data was successfully written.

The Write Burst Window writes user specified data to a single user specified address. The user can choose exactly how many times the data is written, from 1 to infinity. This can be useful when testing a circuit which produces a signal each time a command is written. The user may need to verify that a precise number of signals occur in conjunction with the number of commands sent.

The Watcher Window continuously reads one user specified address. The window will display the number of times the data changed and will also display the highest and lowest data value ever read from that location. This window has a wide range of uses and is very useful for diagnosing intermittent problems.

The Matcher Window continuously reads one user specified address and compares the data read to a user specified value. If the data read matches the value, the window increments a

counter and displays the number of occurrences when the data matched the value of interest. The user can use "don't care" X's in the value to allow more flexibility. There is a hex and a binary version of this window.

The Interrupt Acknowledge Window is used for debugging boards which generate interrupts. When a board in the crate produces an interrupt, the window performs an interrupt acknowledge cycle and then displays the vector returned by the interrupting board.

The Text Window is used for documenting the users test setup. After a set of windows are created and configured for testing, the user may document the test setup by entering text into one or more Text Windows. The entire test setup can then be saved on the hard disk so that it may be recalled at a later date. The text windows serve as an aid to understanding the tests which were previously configured.

C. Special Windows

The generic windows can be quite useful for testing circuit boards and in some cases the generic windows are all that is needed, but for special purpose circuit boards, special purpose test windows are often desired. Special purpose test windows can be created by a Pascal programmer using the WindoWorks frame work. A programmer may use the generic windows as a template for creating windows to test specific circuit boards.

Some examples of special purpose windows which have been created are:

A window to display and change the dots on a board with a dot matrix display.

A window to run long term bit errors tests on a transmitter-receiver pair.

A window to configure the registers of a video chip.

A window to display the time and date read from a clock calendar chip.

D. Using Windows Together

One of the nice features of WindoWorks is that the windows can be used in conjunction with each other to perform super-tests. Each window provides a basic building block which gives the user power to create unique tests in order to solve particular problems. For example, the user can watch the outcome of the Memory Test Window with the Memory Dump Window. The user may want to use the Watcher Window to see if the Write Burst Window had the desired effect. It is amazing how creative the user becomes when using WindoWorks to test and diagnose hardware. Sometimes the user is able to create tests which were never envisioned by the software designer.

E. Long Term Tests

It is often necessary to run long term tests on circuit boards in order to catch intermittent failures or to determine error rates. The long term tests must be able to run unattended, for days or weeks at a time. If an error occurs when no human is present the test program must be able to keep a log of failures.

WindoWorks has the capability to be used for long term tests. Several of the generic windows display and hold errors that occur. The PC's hard disk makes a convenient place to keep a log of failures. Error log text files can easily be created and updated by the WindoWorks program so the user can read the file at his leisure.

F. Future Upgrades

WindoWorks was originally created before the advent of Microsoft Windows [8]. Since the popularity of MS Windows 3.1 is now evident, it may be wise to port the WindoWorks program to the MS Windows format. There are several new programming packages that propose to help a programmer develop MS Windows applications. Using MS Windows may be advantageous because of its standard GUI and its true multitasking ability.

IV. REFERENCES

[1] VMEbug is a trademark of Motorola Inc. Microcomputer Division, 2900 South Diablo Way, Tempe, Arizona 85282 VMEbug is a program which runs from PROM on Motorola VMEbus CPU boards. It uses a dumb terminal for the user interface. VMEbug was designed for downloading and debugging programs which run on the CPU board, but it is often used to exercise circuit boards in the crate when no other debugging facility is available.

[2] BiT3 Computer Corp. 8120 Penn Avenue South, Minneapolis, MN, 55431-1393 Phone (612) 881-6955

[3] R. Ducar, "Tevatron Serial Crate Controller," *Fermilab Controls Hardware Release No. 2*, (1981)

The Tevatron Serial Crate Controller (TSCC) was designed at Fermilab and is used in all of the CAMAC crates in the Fermilab Tevatron control system.

[4] R. Ducar, "TSCC Link Driver," *Fermilab Controls Hardware Release No. 8.1*, (1987)

The TSCC Link Driver (TSLD) was designed at Fermilab to accommodate the user with a convenient parallel interface to the serial protocol demanded by the TSCC.

[5] The PC to Link Driver Interface card was designed at Fermilab by R. Marquardt, J. Smolucha, and J. Zagel.

[6] Turbo Pascal is a registered trademark of Borland International, Inc. 4585 Scotts Valley Drive, Scotts Valley, CA 95066

[7] MetaWINDOW is a trademark of Metagraphics Software Corporation, 4575 Scotts Valley Drive, POB 66779, Scotts Valley, CA 95066

[8] Microsoft, and MS are registered trademarks, and Windows is a trademark of Microsoft Corporation.