



Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY

Engineering & Technical Services Division

Presented at the SIGGRAPH 1979 Sixth Annual Conference
on Computer Graphics and Interactive Techniques,
Chicago, IL, August 6-10, 1979

GRAPHICAL HUMAN INTERFACE TO A NEUTRAL BEAM SYSTEM

Victor Elischer and Jerry Fiddler

May 1979

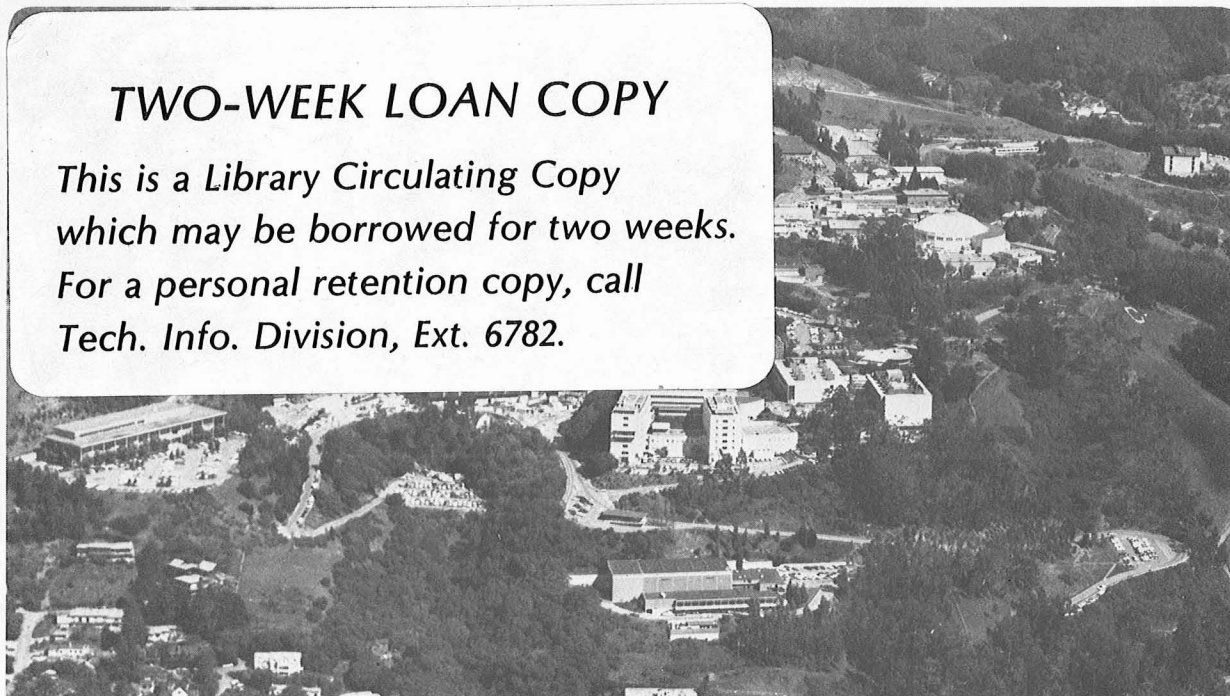
RECEIVED
LAWRENCE
BERKELEY LABORATORY

DEC 4 1979

LIBRARY AND
DOCUMENTS SECTION

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 6782.*



LBL-9146 c.2

Lawrence Berkeley Laboratory Library
University of California, Berkeley

GRAPHICAL HUMAN INTERFACE TO A NEUTRAL BEAM SYSTEM

Victor Elischer and Jerry Fiddler

Lawrence Berkeley Laboratory
Berkeley, California 94720

ABSTRACT

A color graphics system has been designed and implemented as the human interface to a 120KV, 60 amp neutral beam source being built at Lawrence Berkeley Laboratory for Tokamak Fusion Test Reactors. The system is based upon a 64 x 64 touch panel and allows a number of control and high level analysis displays to be brought up on any of three monitors. Included among the displays are a "Super-oscilloscope" package, a data acquisition timing package, numerical control displays and pictograms. The system succeeds in being very friendly and easy to learn for the operator. All control is via the touch panel and a set of six knobs and two keypads.

Key words and phrases: Computer graphics, Touch panel, Operator interface, Process control, Interactive graphics, Graphics portability.

CR Classifications: 3.17, 3.24, 3.82, 6.35, 8.2

INTRODUCTION

One of the engineering challenges inherent in the design of a magnetic confinement fusion reactor is the design for a neutral beam suitable to keep the plasma energy high enough to begin and continue fusion. The beam is composed of high energy hydrogen or deuterium atoms and is injected through the magnetic field into the contained plasma. LBL is designing and testing high power neutral beam sources destined to be used at Princeton in the Tokamak Fusion Test Reactor and at General Atomics in Doublet III. An operator control system was required for the beams and for the test stand. The operator interface design problem is a common one: how best to present large amounts of information about a complex system to a human operator without overloading him, and to receive instruction from the operator about the system's operation. Since the system is a new one, the problem is compounded by the fact that the kinds of information the operator will want to communicate with the system about are not exhaustively known, so the need for the ability to change the interaction is acute.

A critical factor in the design was that the operators who will use the system, having worked with less than friendly computer controlled systems in the past, tend now to be rather "anti-computer". To entice these operators, then, it was important to make the system responsive, fun, quickly learnable and visually attractive.

Since it is likely that a number of different persons will be writing displays, it was important that applications programmers be able to create some analysis displays relatively easily, without worrying about low level graphics, data acquisition or unit conversion. The interface to the application programs needed to be as simple and "bomb-proof" as possible.

Finally, the system needed to be fairly transportable and easily expandable to include, for instance, multiple CPU's.

The end product visible to the user is a set of three CRT's, one equipped with touch panel, on which several types of display are available. Control displays are available to control both the neutral beam and the display system itself. Pictogram, numerical and higher level analysis displays show the operator the state of the beam and its various subsystems. The operator need not be concerned at all with the fact that there is a computer handling his interaction with the neutral beam. His view of the system is completely graphical, and all his control of the system takes place from the touch panel on a set of menu type displays.

HARDWARE CONFIGURATION

The hardware layout breaks down logically into three sections; the main computer, a Modcomp IV/35 with 256K of memory, the graphics output hardware, and the graphical input hardware. The three subsystems connect together in a simple, standard, modular manner, allowing replacement of any of the three with minimum system perturbation.

Graphics Output System

Graphics output is handled by a Grinnell graphics controller driving three CRT's, two B&W and one color. The resolution is 512 x 512 on each, with four shades of grey available on the B&W's and sixteen colors selected from a palette of 4096 on the color monitor. The Grinnell receives its data from the computer via a simple block transfer.

Graphical Input System

The input hardware was all designed at LBL, except for the Intel 80/20 which acts as the input system controller. The Intel microcomputer communicates with the main computer via a single 9600 baud RS232 link. This makes debugging either

NEUTRAL BEAM SOURCE OPERATOR INTERFACE

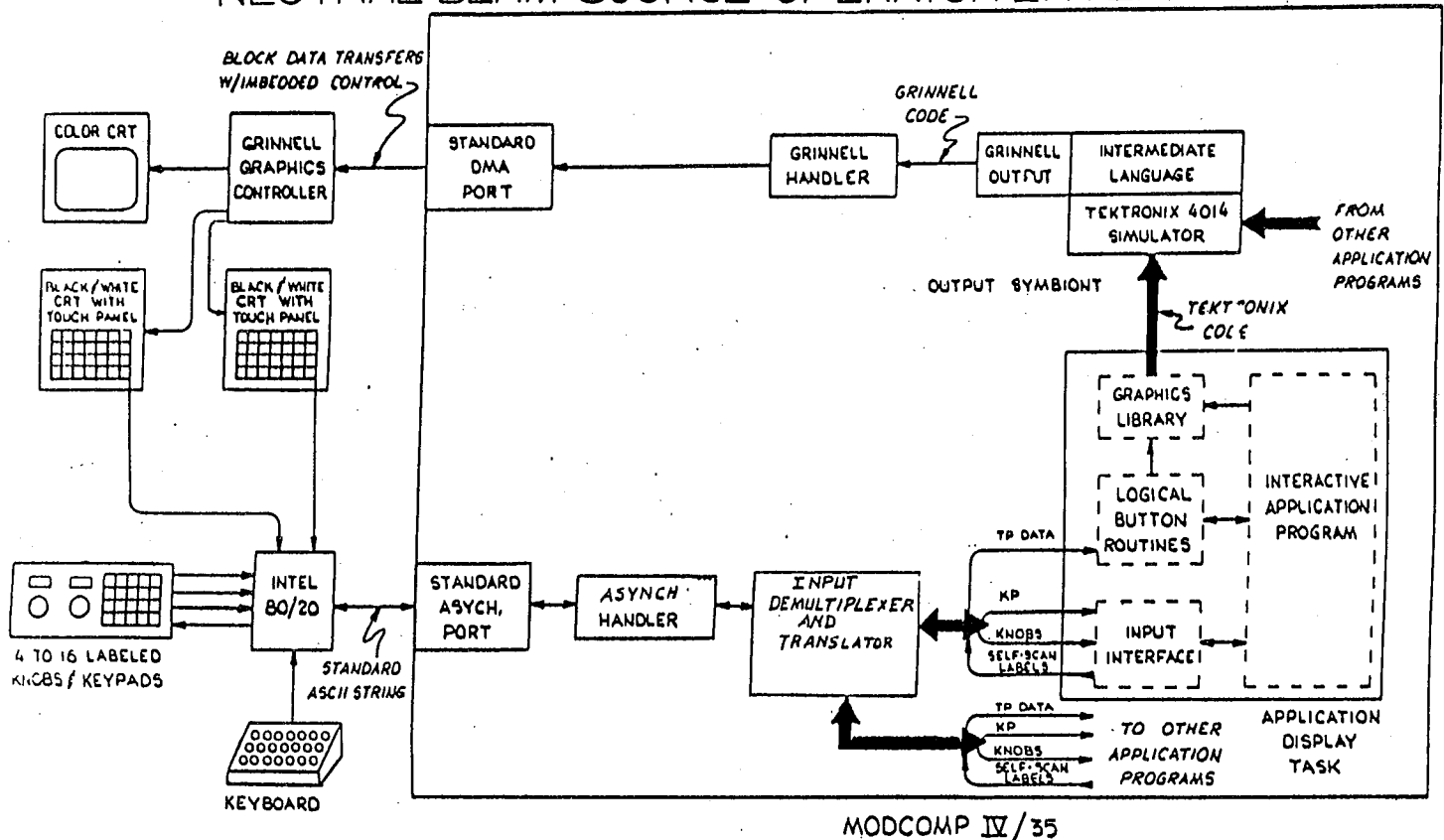


Figure 1

the host computer input or the micro computer output quite convenient, since a standard terminal can simulate either when debugging the other. This also promotes portability of the input system to any host computer.

All operator input devices are attached to the micro. Several types of devices have been implemented. Calculator-style keypads allow simple numerical entry, but are used quite rarely. Knobs are also available. When a knob is turned or a keypad pressed, an ASCII string describing the event is sent to the host computer by the Intel. Knobs are generally used for moving a cursor or display window, or for adjusting numerical values. Associated with each knob is a Burroughs Self-Scan display which can be used, by the program currently receiving knob data, to label the knob. A string sent from the host computer to the micro causes a label to be written. Although the capability to add a keyboard has been built into the system, there is currently none plugged in, nor is one anticipated.

The most important input device is the touch panel. This consists of two transparent sheets of mylar, each deposited with a thin film of gold and scribed into 32 conductive strips. This is placed over a CRT screen, allowing the operator to touch the image and cause an event in the

computer. When the touch panel is pressed, a string indicating the location of the center of of the hit is sent by the micro to the host computer. The micro also provides audible feedback.

Although the capability has been built into the system to handle many touch panels, knobs, keypads and a keyboard, we have found that a single touch panel and a set of four knobs is adequate for all the control and input for our system. By limiting the control to a small set of very powerful devices, potential for operator confusion is minimized, and interaction with the system is made to be rapid and consistent.

Input System Microcomputer

While a discussion of the internals of the microcomputer's firmware is beyond the scope of this paper, the range of its functions should be noted. All the complex input functions are translated to simple ASCII strings, ready for the Modcomp simply to read. For the touch panel, all the scanning, debouncing, translation and finding the center of a hit are handled by the micro's firmware. Most importantly, all the input devices are multiplexed and a single, simple consistent interface is presented to the main computer. This allows for simple expansion of the number

and types of input devices used. Using ASCII strings as the interface mode allows very simple transportability to a different host computer, and allows the use of a maximum amount of existing system software for string interpretation and manipulation.

SOFTWARE

The system structure is diagrammed in figure 1. The application task interfaces to the outside world through several packages of subroutines. The application programmer need only be concerned with high level graphics and graphical input.

Graphics Output

The graphics library is an extended form of TCS, Tektronix's graphics package. This package provides fairly low level graphics support, allowing one to draw lines, points and characters. TCS is well supported and widely used. We have extended TCS to take advantage of the added capabilities of the Grinnell graphics controller, such as color. In the future, we hope to switch to a more advanced graphics package which will allow more advanced capabilities, such as 3-D and segmentation.

A major problem with TCS, and with most graphics packages, is the lack of run time device independence. At best, many packages allow one to select from among a choice of output device types at link time. TCS outputs Tektronix display code. To allow run time device selection, and to allow us to use the Grinnell controller with TCS, an output symbiont was written. A symbiont is a Modcomp artifact, a task to which one can read or write as if it were a device. The function of this particular symbiont is to translate some graphics device dependent code to another device dependent code. In its present form, it consists of a 4014 simulator on its input, which translates Tektronix display code into a more general intermediate language, and a Grinnell output section which translates the intermediate language into Grinnell display code. Any number of input simulators and output drivers may be added, allowing run time device selection among graphics devices with different properties, even though using a device dependent graphics package. Another major advantage of using the symbiont approach is that when we change graphics packages we can use the new package's Tektronix output driver, rather than writing our own Grinnell driver.

Graphical Input

Input from the touch panel, knobs, keypads and/or keyboard is all multiplexed over a single RS232 line. This line is read, through the system asynch handler, by an input demultiplexor and translation task. This task breaks up the single stream into a number of logical streams, one for each input device. Devices are assigned on an individual basis to application tasks requesting them. The demultiplexor task communi-

cates with the applications tasks through an intertask message service over virtual circuits, allowing a much more flexible interface than the standard I/O system. Additionally, this task receives self-scan labeling requests from the applications tasks and ships them out to the appropriate knob label, via the microcomputer.

An input interface library to which the applications task links takes care of translation of the message format coming from the input demultiplexor, formats and sends self-scan labeling requests to the demultiplexor, and handles general message protocol with the demultiplexor. The applications program is simply told, "Knob 4 was turned 5 ticks to the right," or whatever.

Buttons

The logical button routines comprise a package of higher level routines which handles touch panel input translation. The applications code creates a logical button on the screen with a call to MAKE_BUTTON. This call specifies the location and size of the button, in pixels, and whether or not to indicate the button's existence by drawing a border around it, coloring it in or writing some text in it. This call also assigns to the button an ID number. When the touch panel is pressed in the area defined as the button, the applications task is notified that the button has been pressed. The button package returns the assigned ID of the button to the task. By default, the button package also provides visual feedback to the operator that the button has been pressed by drawing a thick border around it. This can be disabled by the calling program if desired. The microcomputer supplies audible feedback using a beeper. The button package also translates between screen pixel coordinates and touch panel coordinates using variables stored on disc indicating their relationship. The operator may at any time align the touch panel to the screen by changing these variables using a simple "Press the dot" program.

EXAMPLE DISPLAYS

Figure 2 shows the system. By pressing buttons on the central menu, the operator can call any display on to any screen, or can call other displays to replace the menu display. Below the CRT's are knob panels labeled with self-scan displays. As one can see from the picture, the position of the displays currently requires one to reach up to press a button. Hopefully, the CRT's will soon be moved into a more convenient arrangement.

Two sample display programs are shown in figures 3-8. Figures 3-5 are photographs of the Timing Control program. All the system timing is generated by programmable pulse

generators which are downloaded from the computer. The operator sets up the timing from a number of displays like the one shown. In figure 3, the operator is selecting a particular signal to adjust. When he turns the knobs associated with the display, he will change that signal and will see the numbers associated with that signal change. He will also see the actual picture of the signal move on the screen, and the pulse will widen or narrow, to show the real effects of what he is doing. By hitting the "Set Limits" button and then touching two points on the screen, the operator defines a window which he can examine more closely (figure 4). The signal area on the screen is defined as a number of buttons which are "invisible". In figure 5, he has expanded his viewing to the window previously defined and has decided to go out and change the modules. When he presses the "Change Modules" button, two buttons appear in its place to make sure the change is really desired. The operator can still back down from here with no changes being made to the timing hardware. Since these photographs have been taken, the signals and the displays have been given more meaningful names.

The waveform display and analysis program is shown in figures 6-8. After each shot, the operator is normally shown a display like that in figure 6, which shows the key waveforms in



BBC 791-582

Figure 2

the neutral beam source. This particular display presents a large amount of information, but several levels of display are easily available, from that shown to one with waveforms and grids but no numbers at all. Using a menu on another screen (not shown) he can select one or more of these waveforms for further study. In figure 7, the operator has selected a particular waveform and, using the menu and knobs, has positioned a cursor for a window to be enlarged. In figure 8, this enlargement has been performed. The operator can also specify, from the menu, that he always wants to see the waveform displayed in this way, and then whenever the display in figure 6 is called, the waveform in question will be displayed expanded in the proper way.

RESULTS

The most important view of the system must always be the operator's. He sees a system which is completely graphical, both input and output. The system that he sees is quite consistent, since all his interaction is through the same set of button routines and input interface routines. All his control is from one central location through a set of uncluttered, unambiguous menu and control displays.

The operator is able to think, not in terms of a computer, but in terms of magic pictures which change and react when he touches them. This is the crux of the design. The overriding philosophy has been to simplify all interaction and to use a few powerful input devices, so as to let the operator concentrate on running the beam rather than running the computer. Lots of feedback is provided, both audible and visual. This is vital if the operator is to have faith in the system. Lack of feedback could easily promote frustration. Additionally, the responsiveness of the system makes it fun to use. Anyone who enjoys working with his tools will do a better job.

The system is organized in levels so that by changing only the top level display programs and the data structure, entirely different types of machines could be controlled, without changing the mode of operator interaction. The underlying hardware and software structure comprises a general purpose operator interface system that is friendly and flexible.

ACKNOWLEDGEMENTS

Van Jacobson did much of the initial design and wrote the output symbiont. Steve Mellor provided critical reading of the paper. Larry Brusse and Mark Nyman designed the input hardware. Robert Upshaw designed and coded the microcomputer firmware. Kirk Thege wrote the waveform displays used here as examples.

This work was supported by the U. S. Department of Energy under Contract No. W-7405-ENG-48.

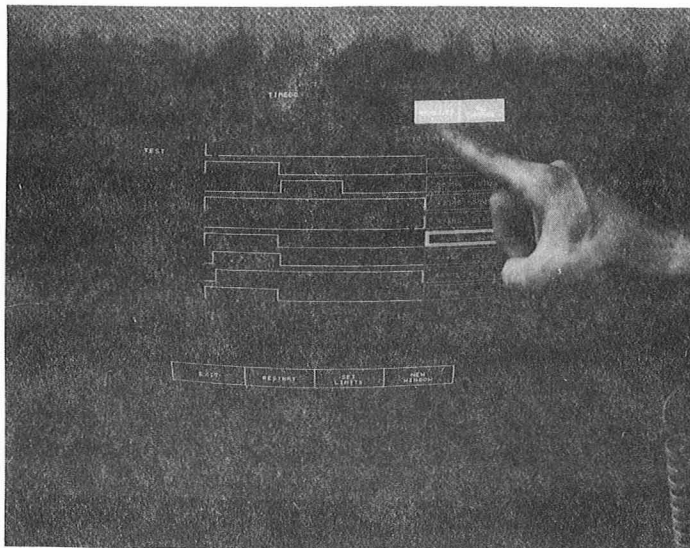


Figure 3 XBB 793-3448

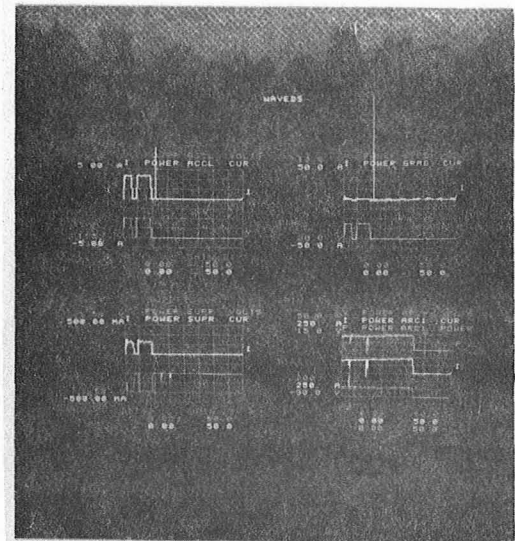


Figure 6 BBC 793-3438

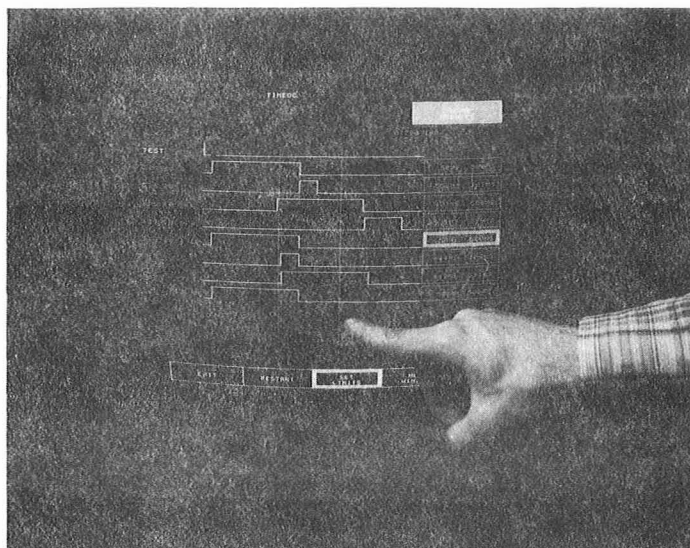


Figure 4 XBB 793-3447

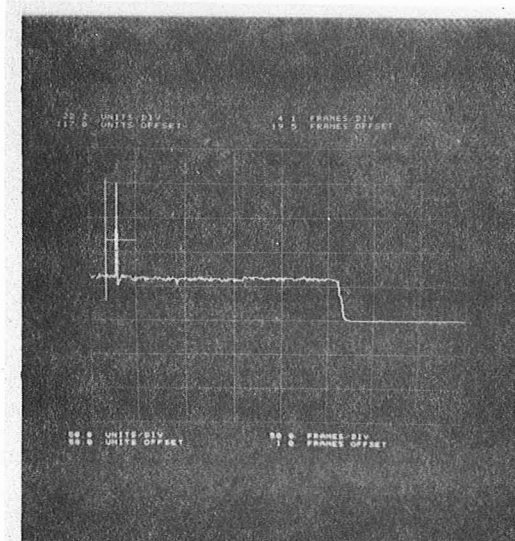


Figure 7 BBC 793-3434

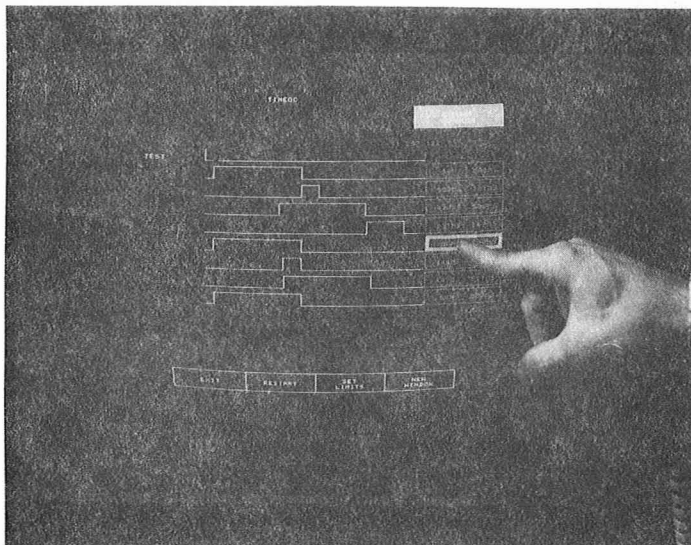


Figure 5 XBB 793-3446

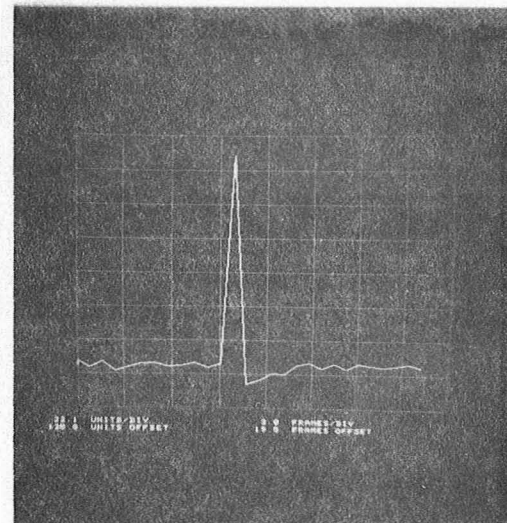


Figure 8 BBC 793-3442

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

TECHNICAL INFORMATION DEPARTMENT
LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720