Hitachi America, Ltd.

Application Note
E3000 emulator

Cristian Tomescu

E3000 features tutorial

INTRODUCTION

The E3000 is a powerful and compact emulation tool supporting the entire H8/300 family microcontrollers. Its diminutive size and user-friendly operating environment, combined with relatively powerful debugging features makes this device an ideal low-cost development tool for embedded system designers that utilize 8-bit Hitachi microcontrollers. The E3000 functions under 2 operating system platforms of the IBM PC: MS-DOS and MS-Windows. This tutorial will focus entirely on the usage of the Windows version of the emulator, the E3000W.

The emulator system consists of the E3000 emulator box, a user interface cable that connects the emulator box to the target system via a probe that plugs into the MCU socket, a RS-232 connection cable that links the emulator serially to one of the PC Communication ports, and an optional user external cable that allows monitoring of various signals on the target system. The operation of the emulator system is controlled by the E3000W Windows application software that must be installed into the Windows interface environment.

Emulation of different MCUs within the H8/300 family requires installment of the proper bond-out eval chip inside the emulator box. The table below shows the bond-out chips that support the various H8/300 family members.

Eval chip	H8 family
H8/325	H8/322, H8/323, H8/324, H8/325
H8/330	H8/330
H8/329/338	H8/326-329, H8/338-336
H8/350	H8/350
H8/3334	H8/3332 (IKAP), H8/3334

This tutorial will go through and explain all the debugging capabilities of the E3000, along with all necessary configuration and setup procedures. Two assembly demonstration programs will be utilized to exemplify the emulator features. No target system is needed for this tutorial as the demonstration programs will be loaded into and executed from the emulator memory.

E3000 CONFIGURATION AND SETUP

The E3000 should be linked to the PC via serial communication port 2 (COM 2) by default. If COM2 is used, a small window will appear on the screen stating that COM2 is unavailable. Click on the *OK* button, and the *Port Settings* window will display 2 serial port choices, COM1 and COM2, and 2 baud rate choices, 9600 and 19200 baud. Select COM1 and the desired communication speed, and click on the *OK* button.

Before downloading the executing code and starting actual emulation, the E3000 must be set up and properly configured for the MCU operating conditions. The following steps should **always** be undertaken before downloading the s-record file:

1. **CPU configuration**
   - select CPU type
   - select CPU operation mode
   - select CPU clock source
2. **Memory map setup**
3. **Clean up code-allocated memory area**

1. CPU Configuration

Access the CPU configuration dialog box by clicking on the *CPU configuration* option in the *File* menu (see Figure 1). Three list box options are displayed: *CPU*, *CPU Mode*, and *Clock* (see Figure 2). The *CPU* box will list all family members that can be emulated by the installed eval chip (ie. a H8/325 eval chip allows emulation of H8/322, H8/323, H8/324, and H8/325). The *CPU Mode* box will list one of the 3 possible modes of operation (single-chip, external mode with on-chip ROM enabled, and external mode with on-chip ROM disabled). The *Clock* box gives 4 options: emulator internal 20MHz clock, emulator internal 1.25MHz clock, target crystal (on emulator board), and target oscillator (on target board). Note that the emulator

cannot be run from a crystal oscillator on the target board - it must be wired at the XTAL inputs on the right side of the emulator front panel. After each selection, click on the *OK* button.

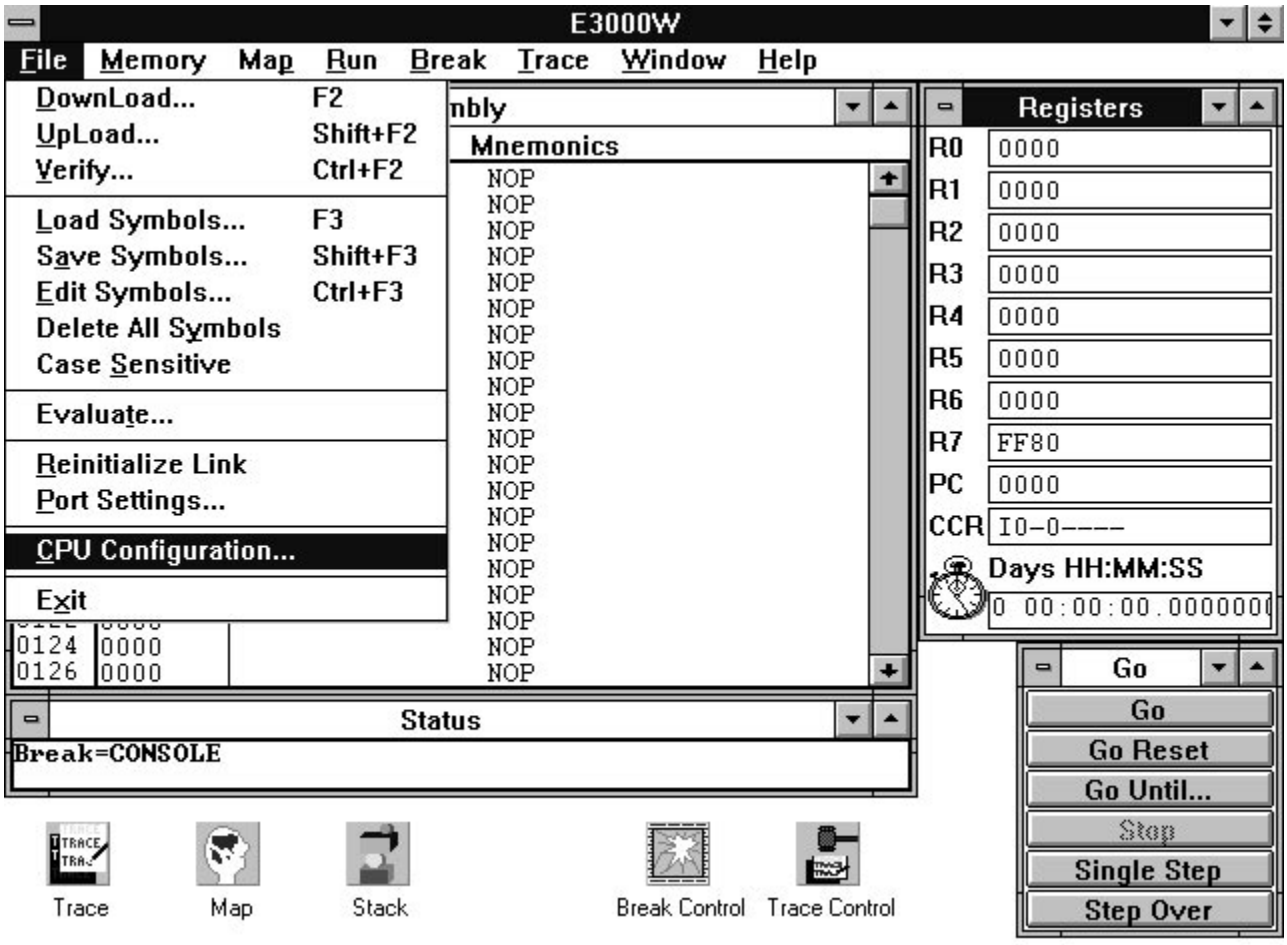


Figure 1.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

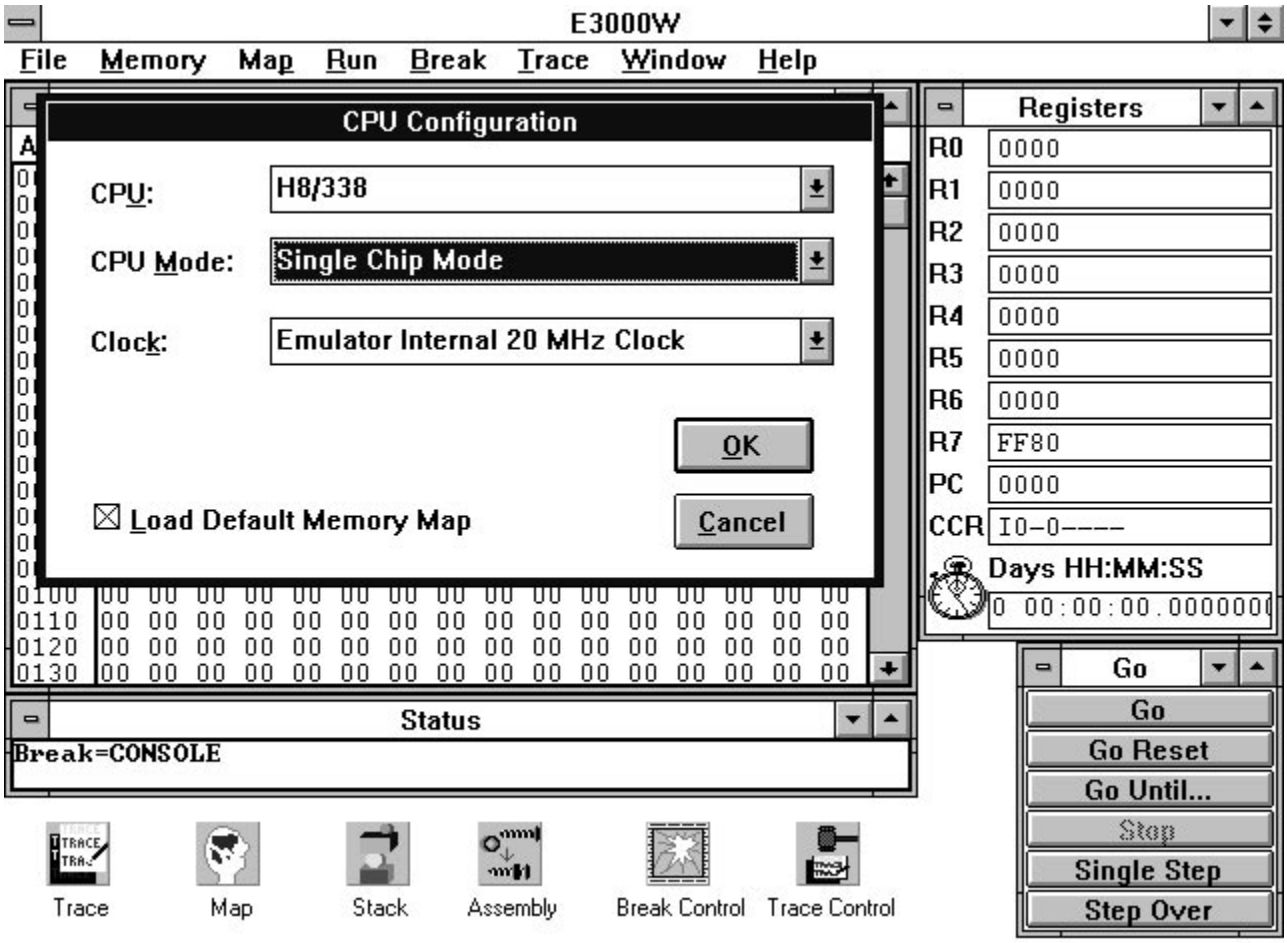


Figure 2.

2. Memory Map Setup

Access the Memory Map Setup dialog box by clicking on the *Edit Memory Map* option in the *Map* Menu (see Figures 3 and 4). The lower left box displays the memory map corresponding to the mode of operation chosen for the emulator (the default map). Internal ROM areas are configured as Emulator R/O (Read-Only) and internal RAM as well as the on-chip register areas are configured as Emularor R/W (Read-Write). External address spaces are defined as Target R/W areas. Any portions of the default memory map may be re-configured by setting the desired boundaries in the *Start Address* and *Stop Address/Length* boxes. Five options, displayed on the lower right side of the memory map dialog box, are available: *Target R/O, Target R/W, Emulator R/O, Emulator R/W, and No Memory.* Normally, the user would download his code into an emulator memory area configured as *Emulator R/W.* After the memory map has been selected, click on the *Set* and *Done* buttons.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

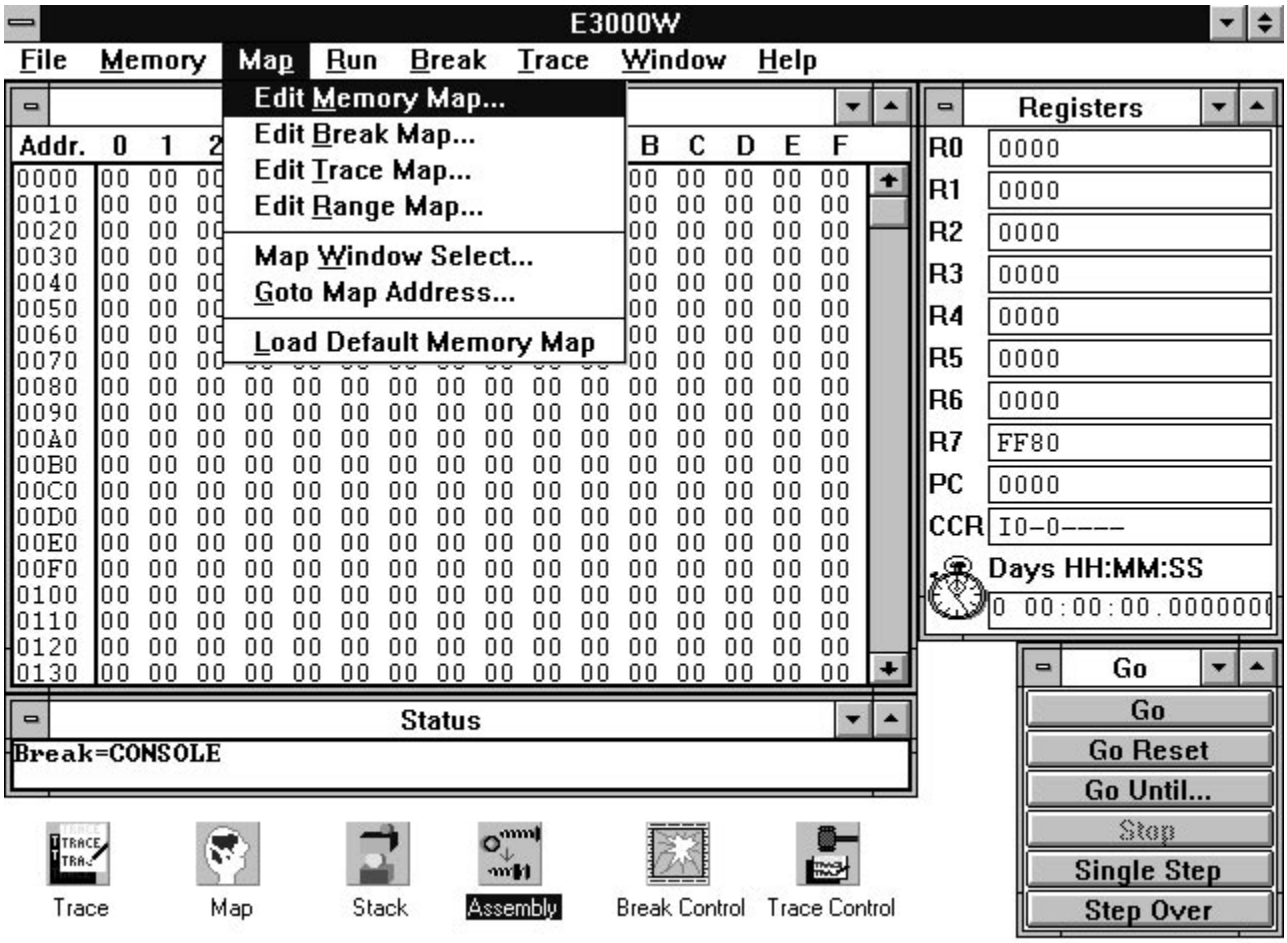


Figure 3.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

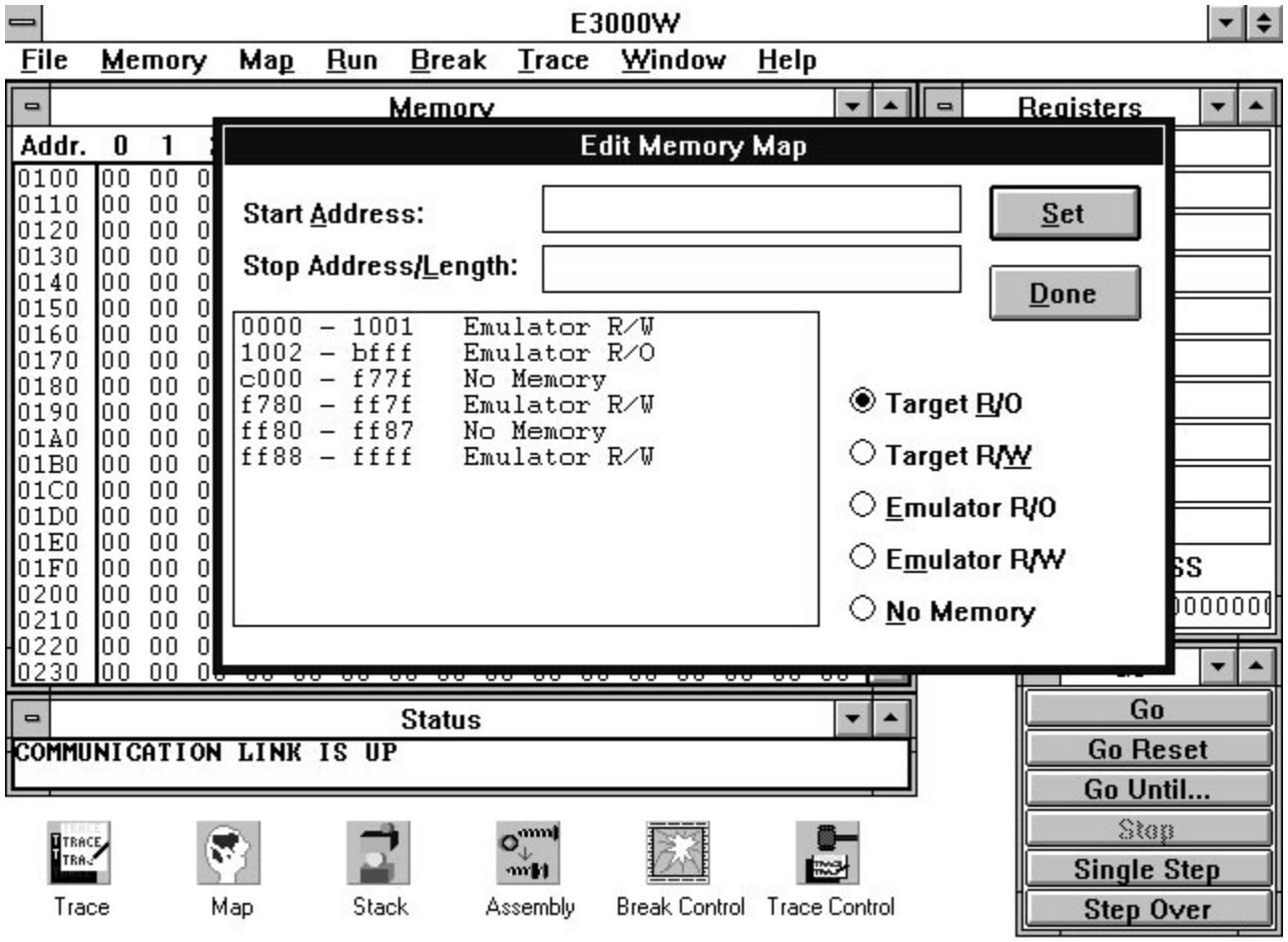


Figure 4.

3. Filling the memory with NOPs.

In order to avoid that any random characters present in the memory area that is to be loaded with the user code or data are processed as instructions, the user should clear this memory section by filling it with zero bytes (or NOPs). This is achieved in the Fill Memory dialog box by clicking on the *Fill* option in the *Memory* menu (see Figure 5). Type in the memory boundaries that will contain the user program in the *Start Address* and *Stop Address/Length* boxes, enter 0 in the *Value* box, and press the *OK* button.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

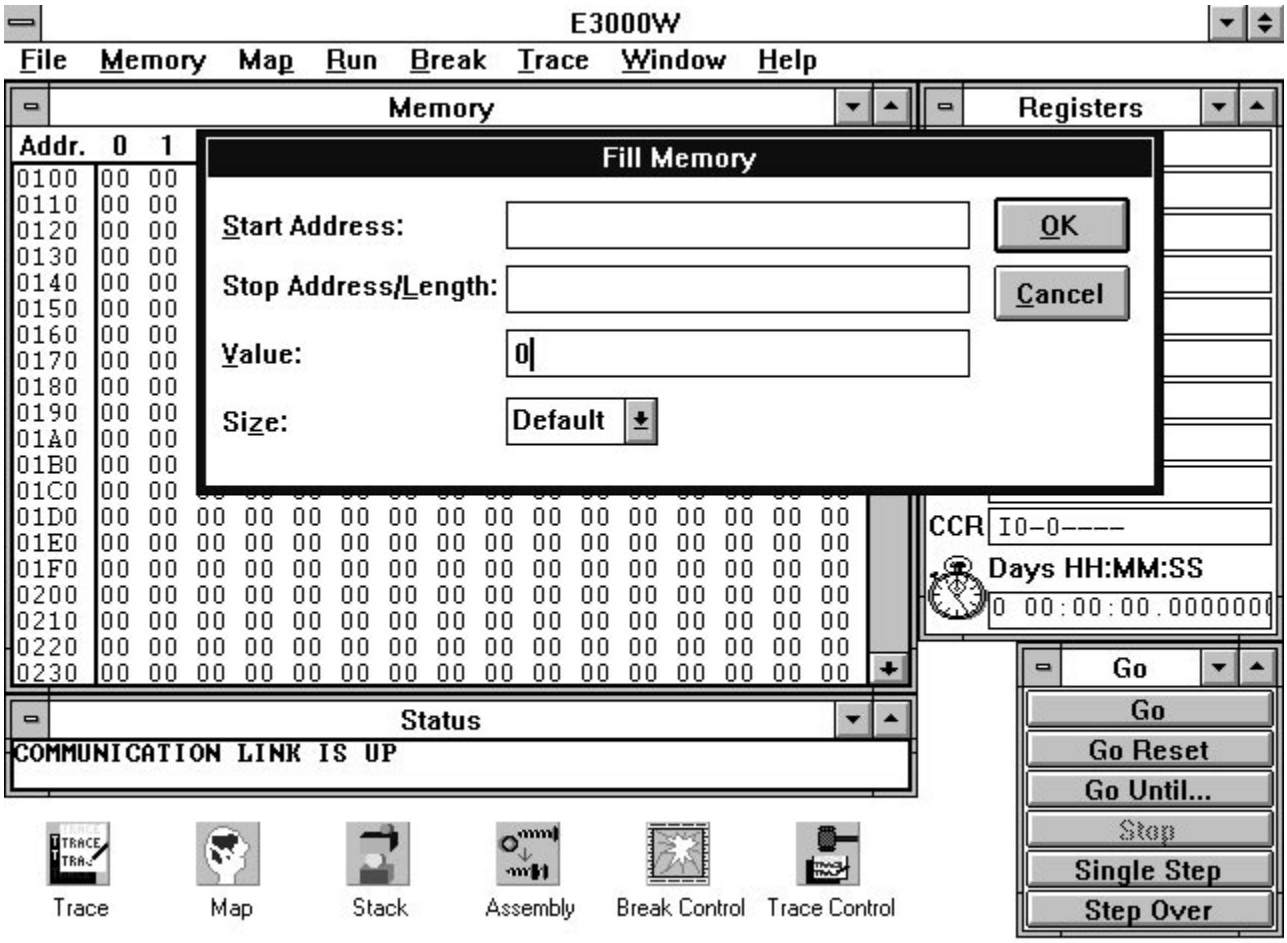


Figure 5.

DOWNLOADING THE PROGRAM

Once the E3000 has been configured and properly setup, the user file may be downloaded from its location in the PC directory into the assigned emulator R/W memory. The file must be in one of the 3 allowable formats: Motorola S, Intel Hex, or IEEE-695 absolute object file format. Pull down the Download dialog box by choosing the *Download* option from the *File Menu.* Then specify the location directory of the absolute file in the *Directories* option box, and double-click on the file name in the *Files* option box (see Figure 6). A small window will appear on the screen indicating the memory address boundaries of the downloaded program. Click on the *OK* button, and another small window will appear on the screen, indicating if an associated symbol file has been successfully downloaded or not. If there is no associated symbol file, an error message will be displayed in this window. Simply ignore this message and click on the *OK* button. Alternatively, the *Load symbols* option in the Download dialog window can be removed by clicking on its checkbox, and no error message will be displayed. The Down load dialog box also features an offset option box that will add an offset (maximum FFFF) to the normal load address; its default value is 0.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

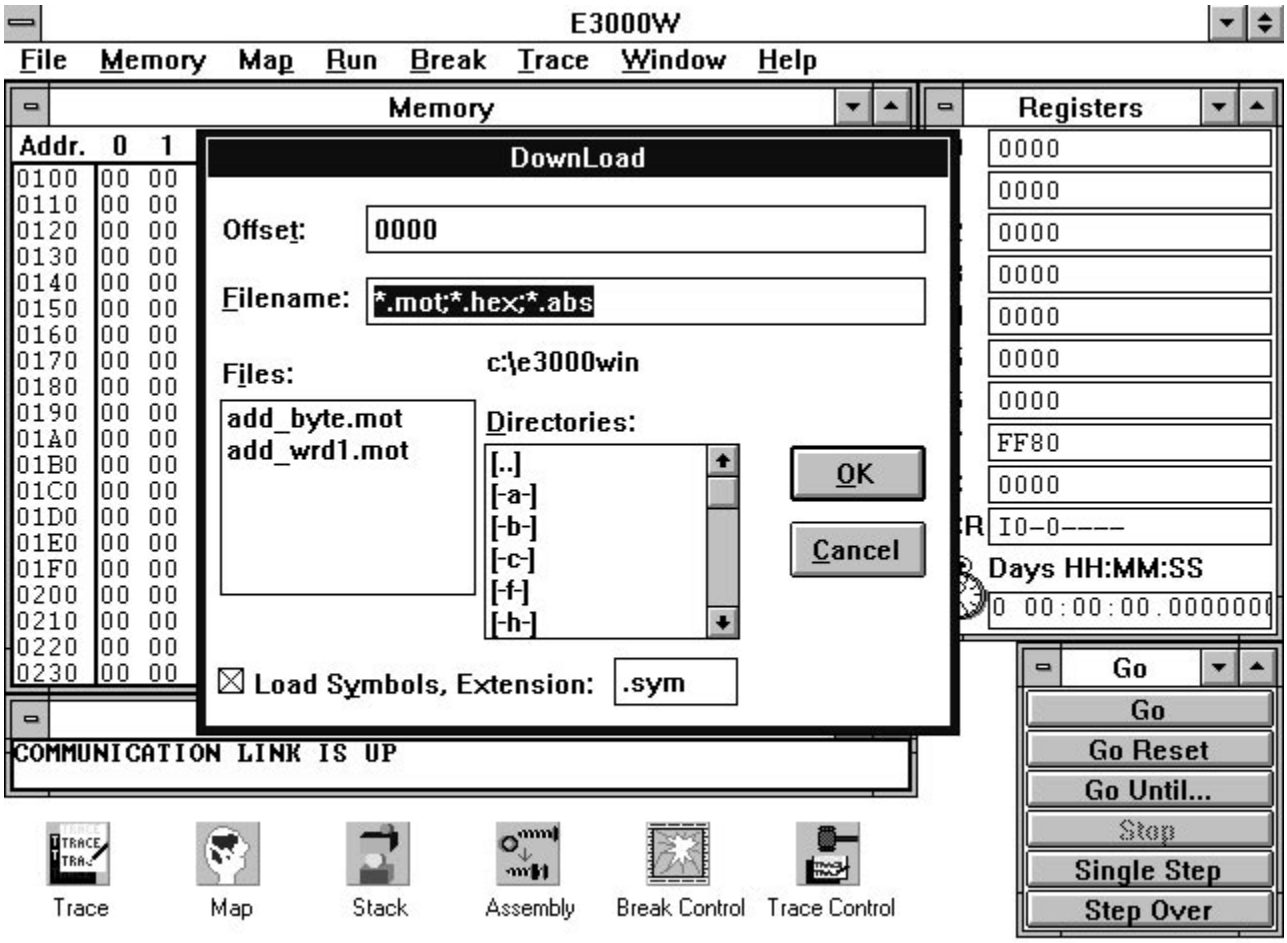


Figure 6.

THE E3000 APPLICATION WINDOW

Before examining the various debug features of the E3000, a brief description of the application window is necessary. Figure 7 shows the E3000 application window. On the top of the screen, the Main Menu Bar is displayed, containing the following 8 menu items: *File, Memory, Map, Run, Break, Trace, Window,* and *Help.* Below, there are 4 default windows displayed: the *Memory* window, the *Registers* window, the *Status* window, and the *Go* window. Brief descritions of these windows are given below.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

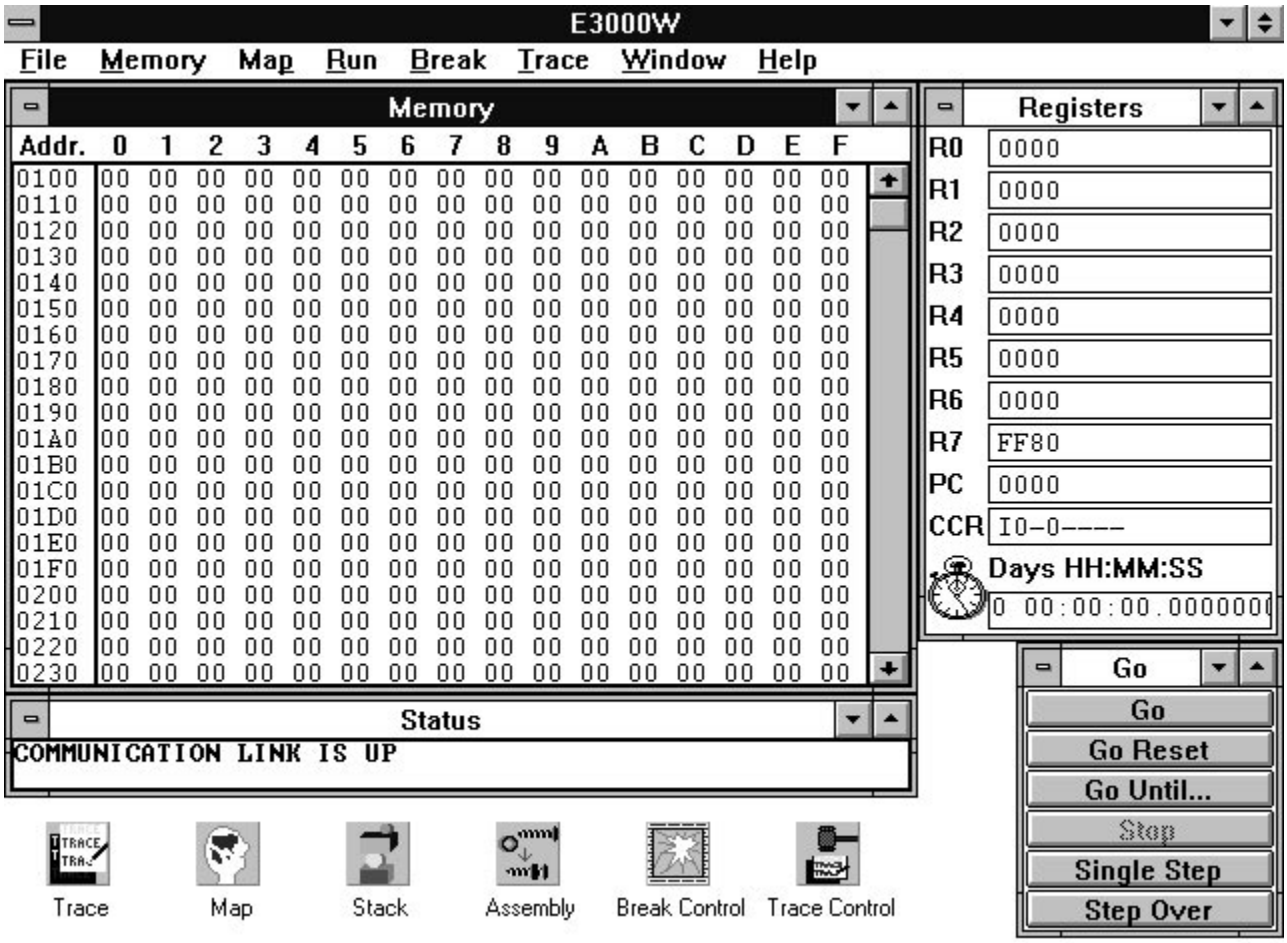


Figure 7.

The *Memory* window displays the data contents in each memory location in a tabular format. By default, the memory contents starting at address 0 are shown. If the contents of a different memory section need to be displayed, the user can either scroll down by dragging the cursor along the scroll bar, or double-click on the *Addr.* column and the *Goto Memory Address* dialog box will be displayed. Enter the desired address, click on the *OK* button, and the contents of a memory section starting at the indicated address will be displayed.

The *Registers* window shows the contents of all CPU registers (R0 - R7, PC, and CCR), and the E3000 timer value. The contents of the general-purpose registers are indicated in a word format with the 2 most significant digits representing the contents of RnH and the 2 lowest significant digits representing the contents of RnL. R7 should always indicate the location of the stackpointer. The contents of the CCR are designated bit-by-bit. If any flag is set, the corresponding starting letter will be shown. The 2 user bits (bit 6 and 4) are designated as either 1's or 0's. By default, bit 7 (the I bit) is set, the 2 user bits are 0's, and the rest of the flag bits are cleared. The contents of each CPU register are updated on the screen during program execution or single-stepping. The E3000 timer value is displayed under the CCR contents, and shows the elapsed time between 2 events during trace execution. This option is further explained and demonstrated under the TRACING OPERATION section of this tutorial.

The *Status* window shows the status of the communication link (up or down), break and trace events, and the status of the CPU. The break and trace events displayed are discussed in the BREAKING OPERATION and TRACING OPERATION sections of this tutorial. The CPU status is shown by pressing the

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

ENTER key on the keyboard while the focus is on the *Status* window.

The *Go* window provides the following 6 control options to run and stop the emulator: *Go, Go Reset, Go Until..., Stop, Single Step,* and *Step Over*. Each alternative is explained in the RUNNING THE PROGRAM section.

In addition to the Main Menu Bar and the 4 Default Windows explained above, the application screen also contains the following 6 function icons: *Trace, Map, Stack, Assembly, Break Control,* and *Trace Control.* The *Trace, Trace Control,* and *Break Control* functions are discussed in context with the break and trace operations. By double clicking on the *Stack* icon, the contents of the stack memory (starting at FF80) are displayed in a tabular format. Any stack address can be displayed by positioning the cursor on the *Addrs* column inside the *Stack* window, double-clicking the mouse, and entering the desired address. Activating the *Map* icon displays the memory map of the emulator, that is the mapping status of each memory location as explained in the Memory Map section of the E3000 CONFIGURATION AND SETUP. By activating the *Assembly* icon, the *Assembly* window will overlap the default *Memory* window. This window contains 4 columns labeled *Addrs, Op-Code, Symbol,* and *Mneumonics,* and it is used to view the downloaded program. The *Addrs* column shows the memory addresses, the *Op-Code* column shows the data contents at each address, the *Symbol* column shows any program labels, and the *Mneumonics* column displays the assembly instruction corresponding to the data contents at each memory location.

RUNNING THE PROGRAM

Once the program has been loaded into the emulator memory, it can be viewed in the *Assembly* window (which is activated by double-clicking on the *Assembly* icon). To position the first program instruction at the top in the *Assembly* window, double-click on the *Addrs* column and specify the start address of the program in the *Goto Memory Address* pop-up window. Next, the program counter (PC) must point to the start address of the program. This step is done by moving the mouse pointer to the box containing the value of the PC, holding down the left mouse button, dragging the pointer across the PC register value, and entering the start address of the program. Figure 8 shows a typical E3000 application window just before running the program.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

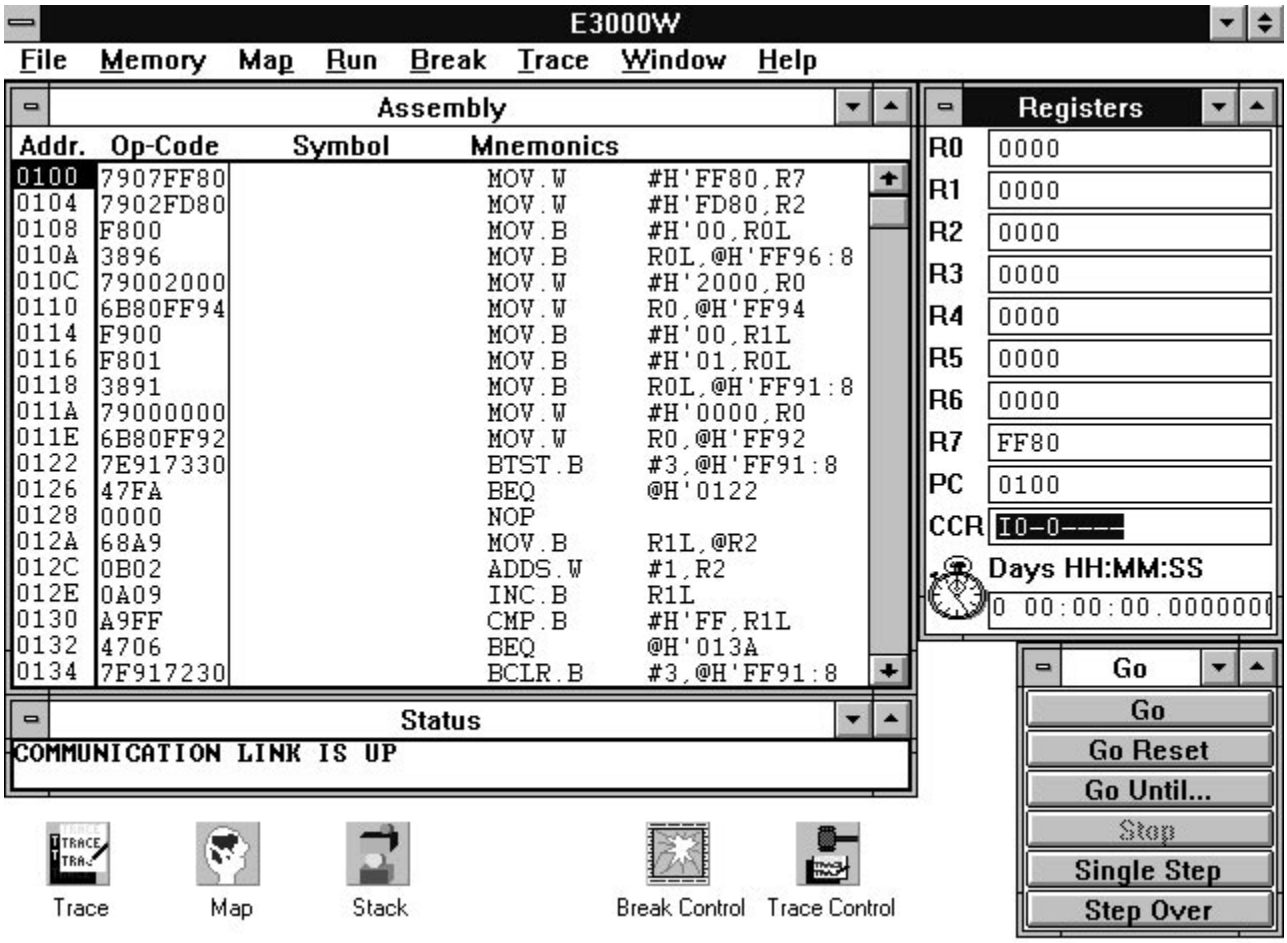


Figure 8.

The user program is now ready to be executed. There are several options that allow real-time program execution, and they can be activated from either the *Go* default window on the bottom right of the application screen, or by clicking on the *Run* menu item on the top of the screen. These options are: *Go*, *Go Reset*, and *Go Until*.

1. The *Go* command starts program execution from the memory location specified in the PC.
2. The *Go Reset* command performs a hardware reset before executing the program. As a result, all on-chip registers are reset, the I-bit of the CCR is set (thus disabling the interrupts), and the PC is loaded with the data contents at address H'0000.
3. The *Go Until* command starts program execution and then stops at a given address within the program address range. Click on the *Go Until* button, specify the stop address, and click on the *OK* button to start program execution.

During program execution, the *Assembly* window address portion as well as the *Go* buttons cannot be activated. The *Status* window will show the message STATE = RUN, and the PC value change as the program cycles through. The rest of the CPU registers in the *Registers* window are updated as well. The program execution can be cancelled by clicking on the *Stop* button in the *Go* default window or in the *Run* menu.

The program can also be run in a non real-time mode, namely through single-stepping. This feature is a powerful debugging tool that allows the program execution to be scrutinized step-by-step. The instruction currently pointed at by the PC is executed, and the program stops with the PC pointing at the next instruction. The effects of each instruction execution

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

upon the CPU registers, the on-chip peripheral registers, or memory can thus be monitored. It is activated by clicking on the *Single Step* button in the *Go* default window or in the *Run* menu while the program is not in running mode. In addition, 2 additional options for single-stepping are available: *Step count* and *Step over calls. Step count* allows the user to execute a certain amount of instructions per each step. *Step over calls* performs single-stepping according to the specified step count except within subroutines. If a BSR or JSR instruction is reached, this command will step over the entire subroutine, thus treating it like a regular instruction.

DEBUGGING THE PROGRAM

The E3000 offers 2 main debugging features: **Breakpoints** and **Real-Time Trace**. Both of these conditions can be triggered upon either address or address range conditions, data conditions, user system triggered external events, event status options, or "complex" events (based upon a combination of the previous conditions). Each of these options will be described in detail under the following 2 sections, BREAK OPERATION and TRACE OPERATION.

BREAK OPERATION

The E3000 break function enables the user to stop the program flow given a condition or a set of conditions are met. This feature allows the programmer to examine CPU and on-chip peripheral register contents, memory contents, and to monitor for unexpected causes that would inhibit correct program operation. The E3000 has 7 types of breakpoints: Access Breakpoints, Write-Protect Breakpoints, Range Breakpoints, Software Breakpoints, Hardware Breakpoints, External Event Breakpoints, and External Breakpoints.

1. Access Breakpoints automatically occur as a result of the user attempting to execute from a memory location mapped in the *Memory Map* file menu as *No Memory*. For example, an H8/329 operating in mode 3 has, by default, no memory allocated within the H'2000 - H'FE7F range. If any address within this range is accessed by the user program, an access break will occur.
2. Write-Protect Breakpoints automatically occur if the user program attempts to execute a write operation to any memory location mapped in the *Memory Map* file menu as either *Target Read-Only (R/O)* or *Emulator Read-Only (R/O)*. For example, an H8/330 operating in mode 3 has, by default, memory area H'0000 - H'3FFF mapped to emulator read-only (since it is allocated to the on-chip ROM). If an attempt to write to any location within this range is made, a write-protect break will occur.
3. Range Breakpoints occur if the user program accesses any memory location set to trigger a break in the *Edit Break Map* dialog box. To arbitrarily set a range of memory to cause a program break, click on the *Edit Break Map* option in the *Map* file menu, and the *Edit Break Map* dialog box will be displayed (see Figure 9). The default break map indicates that the break function is disabled throughout the entire memory map. The user can enter the memory boundary or boundaries in which an access will trigger a break by filling the *Start Address* and *Stop Address/Length* boxes, and clicking on the *Set* and *Done* switches.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

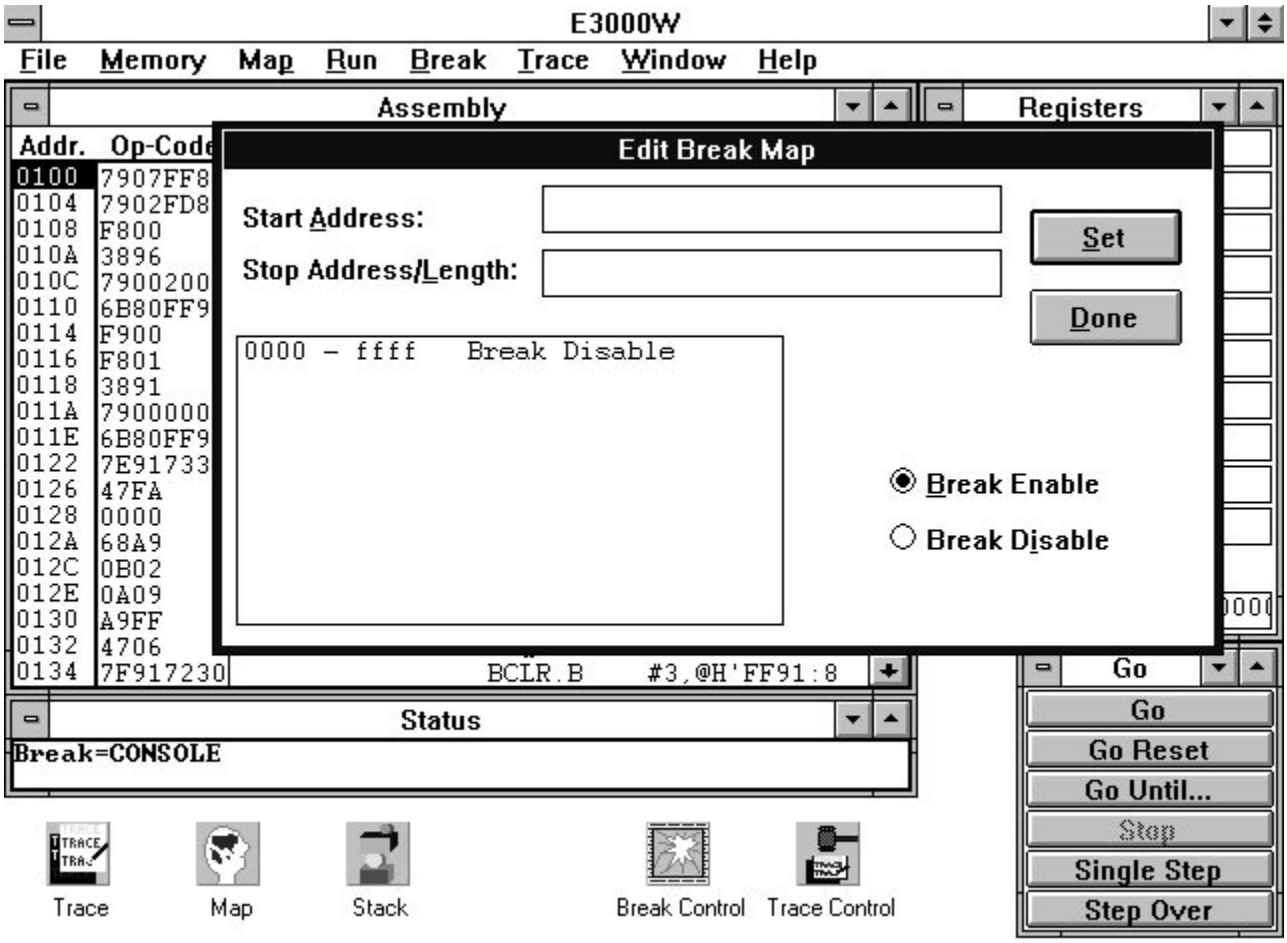


Figure 9.

4. Software Breakpoints - A maximum of 31 software breakpoints can be specified. A software breakpoint causes the program to break on a set address(es) upon instruction execution at that location (but not upon instrcuction pre-fetch). To set a software breakpoint, access the *Instruction Break* dialog box by choosing the *Break* file menu and clicking on the *Instruction Breaks* option (see Figure 10). Alternatively, the *Instruction Break* dialog box can be accessed by clicking on the *Break Control* icon, and then double-clicking on the *Instruction Break* switch. In the column marked *Expression*, enter the desired address(es) upon which a breakpoint should be triggered, click on the corresponding *Status* button(s) to turn the option(s) on, and then click on the *OK* switch.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

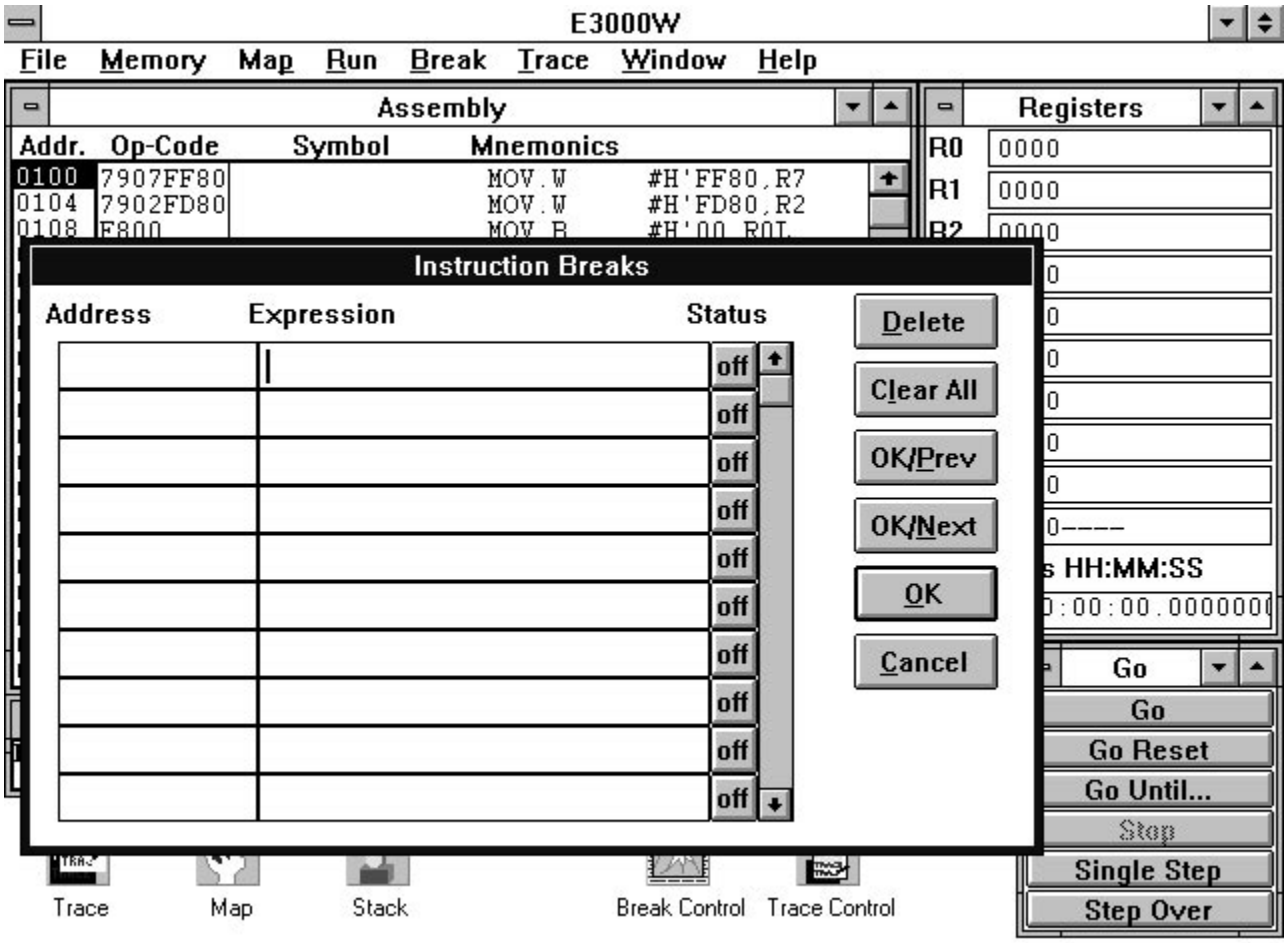


Figure 10.

5. Hardware Breakpoints - A maximum of 4 hardware breakpoints can be specified. A hardware breakpoint causes the program to break on a condition or set of conditions that are specified in the *Break Control* window, which is activated by clicking on the *Break Control* icon (see Figure 11). A hardware breakpoint can occur upon specification of an **event**, a **trigger**, and/or a **sequence.**

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

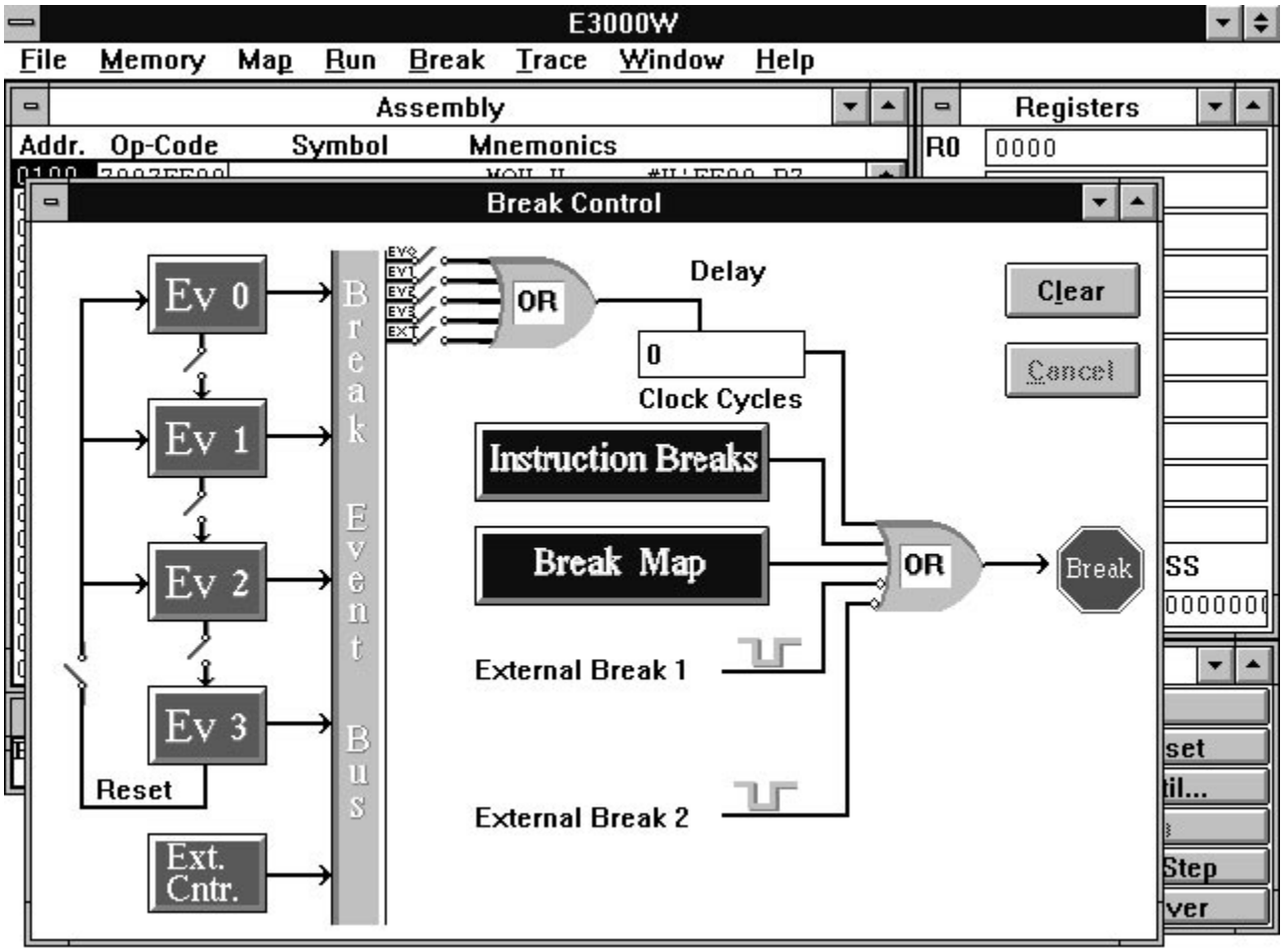


Figure 11.

An **event** is specified by clicking on one of the 4 event boxes (*EV0 - EV3)* on the left of the screen in the *Break Control* window; hence, up to 4 hardware breakpoints can be simultaneoulsy set. To set the event conditions, double-click on the event box, and the *Break Event (0-3)* window will be displayed (see Figure 12).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

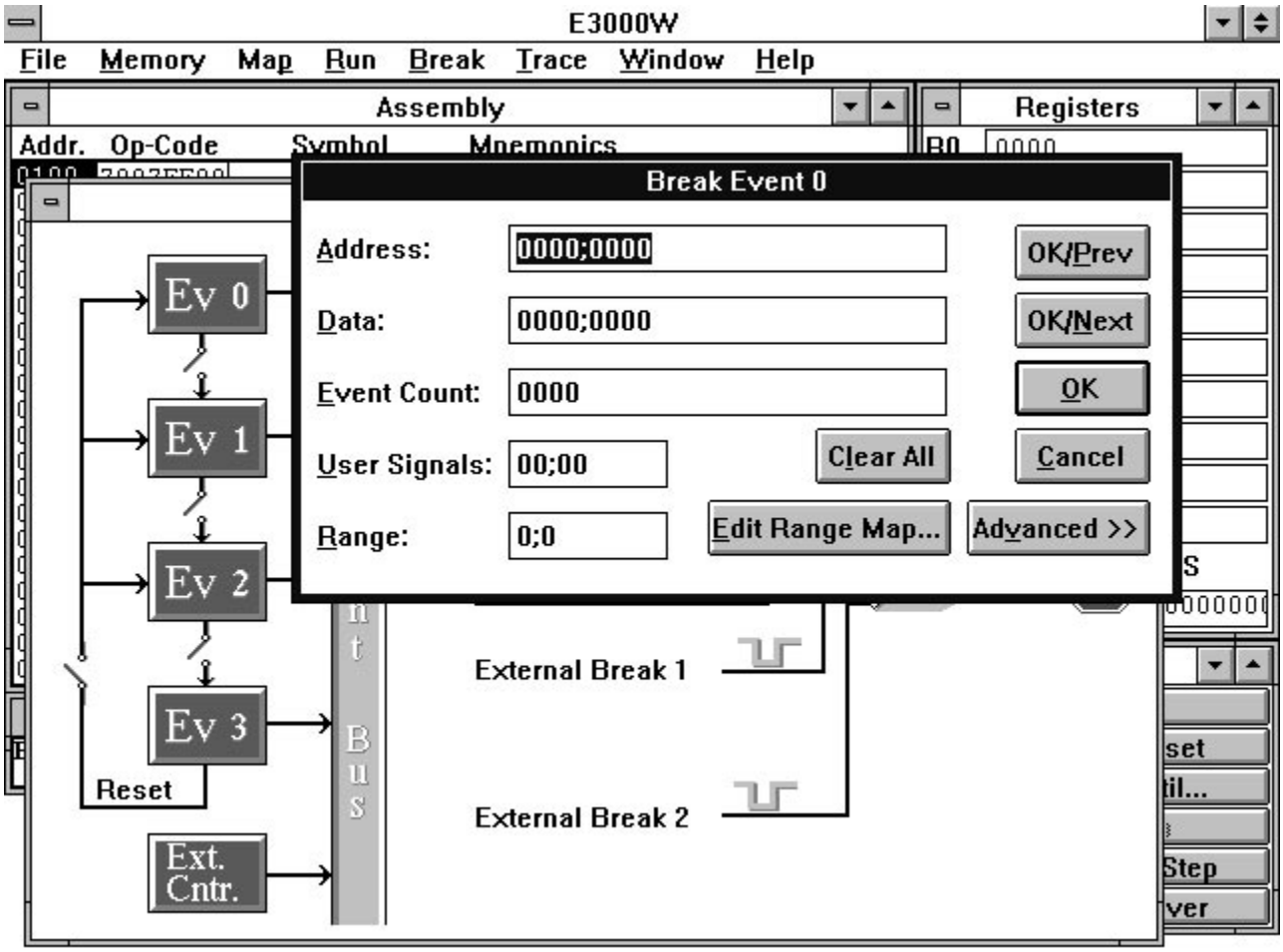


Figure 12.

Up to four main event conditions can be specified by entering data into the *Address, Data, User Signals,* and/or *Range* condition fields. The *Address* field indicates the program address upon which a break will occur. The *Data field* specifies the data that has to be present on the emulator data bus for a break condition to be issued. The *User Signals* field shows which of the 8 user external probes (0-7) will trigger a break, and upon which logic level. Two hexadecimal numbers separated by a semi-colon indicate this condition. The first number shows which probe(s) is active by having its corresponding bit number high; the second number indicates a low trigger condition if its corresponding bit is 0, and a high trigger condition if its corresponding bit is 1. For example, if a break is to be issued when a low signal is present at probe 3, enter 08;00. The *Range* field indicates an address range upon which execution a break is issued. Up to 8 address ranges can be specified (0-7). To set one or several address ranges, click on the *Edit Range Map* option in the *Map* file menu or inside the *Break Event* window, and the *Edit Range Map* window will be displayed. Range 0 (H'0000-H'FFFF) is the default range. To enter a different range, speciify the range boundaries in the *Start Address* and *Stop Address/Length* fields, choose the desired range number, and click on the *Set* and *Done* switches. For example, if a program starts at H'100 and an address range between H'110 - H'200 is specified, the program will break right after the instruction at the last address within the range is executed. In addition, 5 more hardware break conditions can be specified by clicking on the *Advanced* switch inside the *Break Event* dialog box (see Figure 13).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

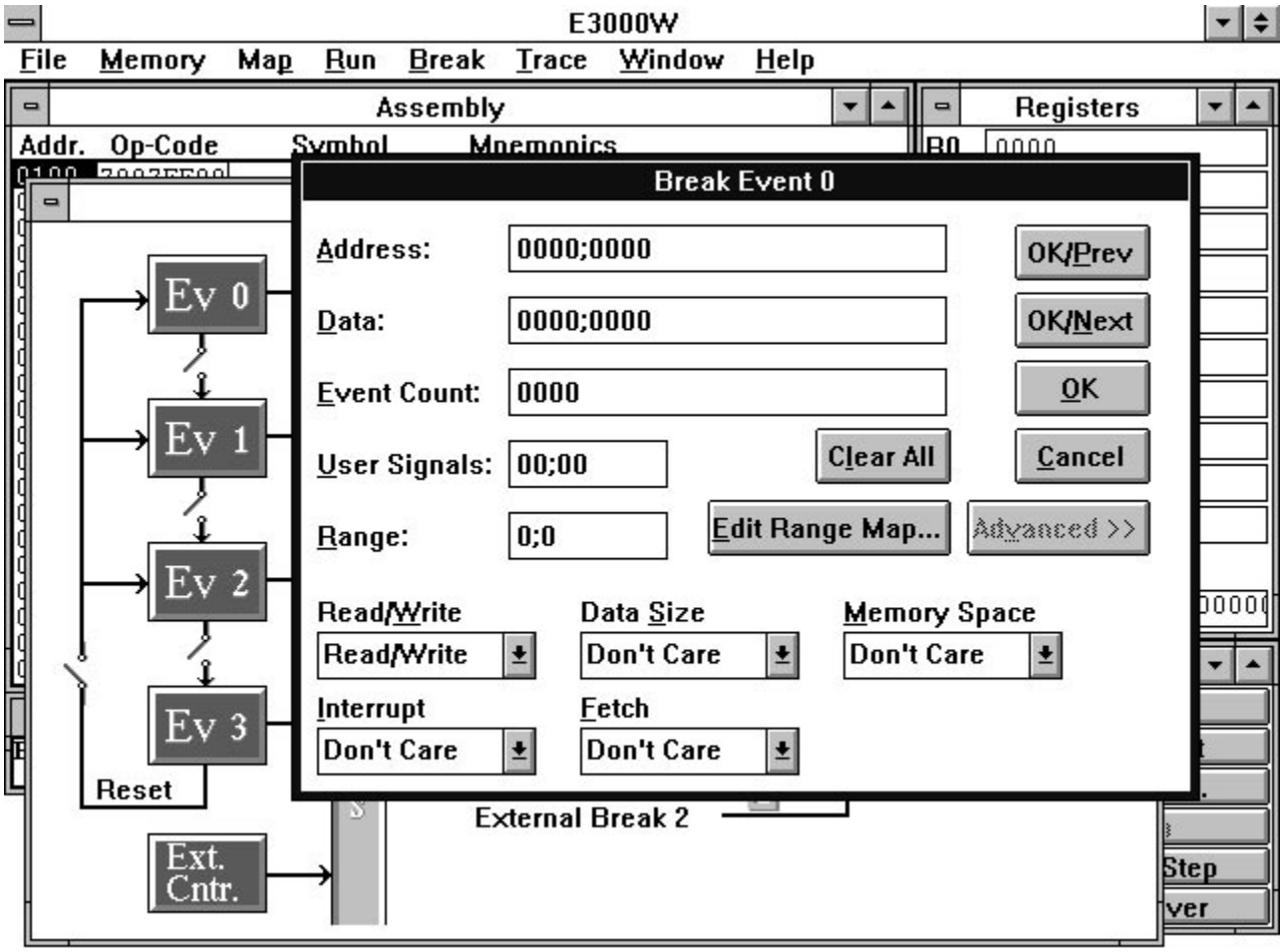


Figure 13.

These options are labeled *Read/Write*, *Data Size*, *Memory Space*, *Interrupt*, and *Fetch*. The *Read/Write* option specifies if an event will occur on a read-only, write-only, or either a read or write operation. The *Data Size* option indicates if a break will occur if a byte data, word data, or any data is present on the data bus during a memory cycle. The *Memory Space* field indicates a break will occur upon a 2-state, 3-state, or any length state memory access. The *Interrupt* field determines if a break will occur only if the eval chip interupt acknowledge bit (IACK) is set (that is when an interrupt service routine is requested). Finally, the *Fetch* switch determines if a break will occur on an instruction or a data fetch.

A **trigger** is specified in the *Event Count* field within the *Break Event* window (see Figure 12). A specified event break will not be triggered until it occurs for the number of times indicated in the *Event Count* field.

A **sequence** is specified in the *Break Control* window by closing the appropriate switches that connect neighbouring event boxes *EV 0-3* (see Figure 12). In this case, a break condition will only occur if the chosen events occur in the specified order. For example, if the switches connecting *EV0* to *EV1*, *EV1* to *EV2*, and *EV2* to *EV3* are closed, a break will occur only if the *EV0*, *EV1*, and *EV2* conditions will occur in order (and after *EV2*).

Once the event conditions and/or the trigger condition for each event have been set, close the corresponding event switch at the OR-gate in the *Break Control* window, and close the window. Optionally, a delay may be specified in the *Delay* field within the *Break Control* window, and indicates the number of clock cycles from the break

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

occurence that must be executed before actually performing the break.

6. External Event Breakpoints are monitored by the counted break external probe (probe 8) in the user system, and are specified (one at a time) in the *External Event Counter* dialog box. Click on the *Ext Cntr*. box on the lower-left corner inside the *Break Control* window, and the *External Event Counter* window will be displayed (see Figure 14). Specify the break condition in the *Trigger at* field as either a low level or a high level at the probe, the number of times this condition must occur before a break is issued in the *External Counter Value* field, and then click on the *OK* switch. A maximum count of 64K is possible.

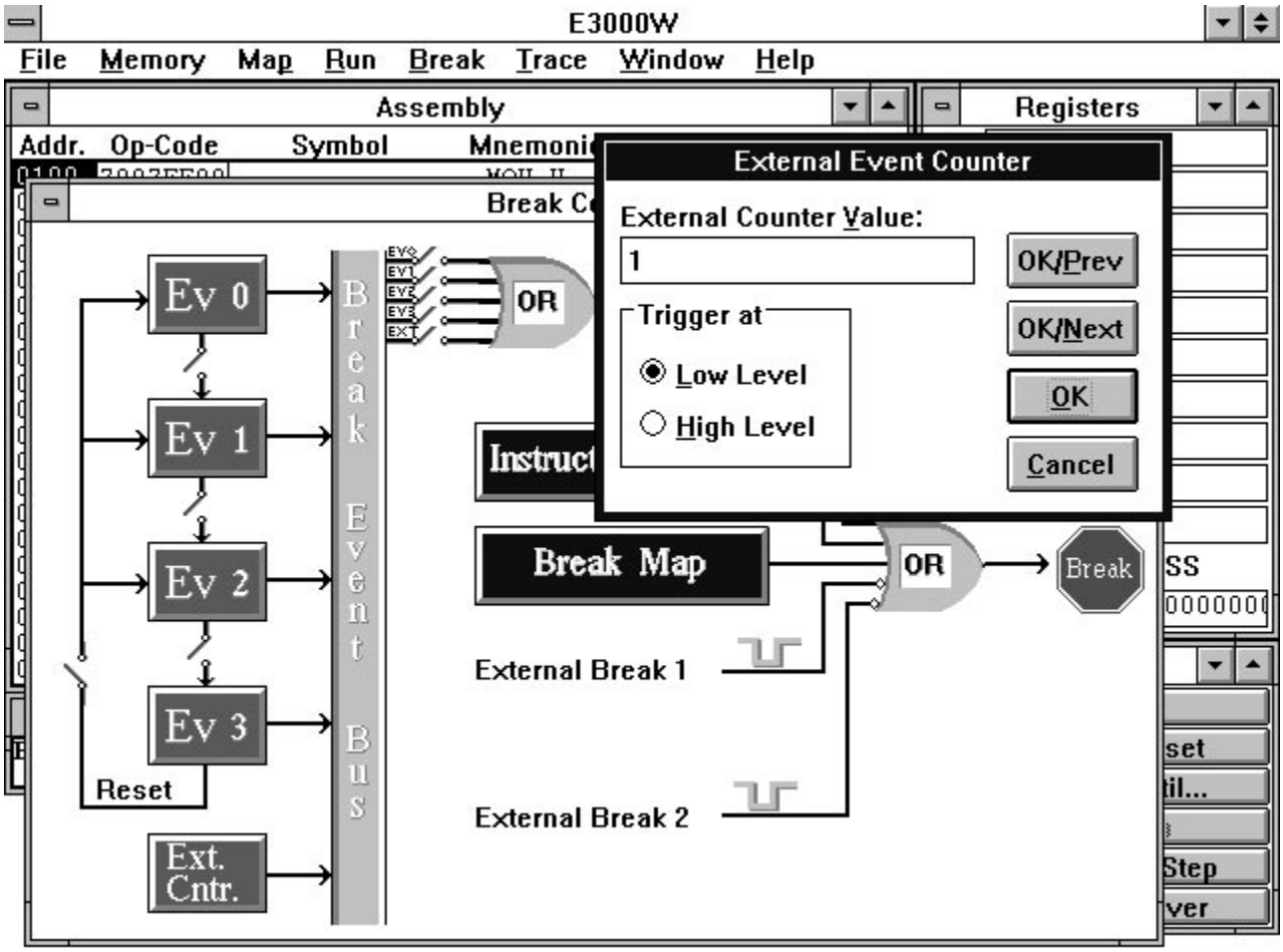


Figure 14.

7. External Breakpoints are monitored in the user system by the 2 external probes, probe 9 (XB1) and probe 10 (XB2). A low condition at either of these probes will cause an immediate and unmaskable break.

TRACE OPERATION

The E3000 trace operation allows the programmer to examine and debug the operation of his/her H8/300-based system through keeping a record on the effects of each instruction executed upon the CPU registers and memory. Critical portions of a program can be stored in the trace buffer memory in the emulator, and up to 1800 trace entries (54 bits x 1.7Kbyte trace memory) can be stored in the buffer. Since the E3000 has the capability of loading and examining trace data while the program is running in real-time, the operation of time-critical code portions such as interrupt service routines can be analyzed.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

TRACE OPERATION SETUP

Before running the portion of the program that needs to be traced, the trace conditions must be initialized by appropriately setting the **trace map**, **trace clock**, and **trace trigger conditions**.

1. Setting the trace map enables or disables trace events (ie. data or instructions) from being accumulated into the trace buffer. Click on the *Edit Trace Map* option in the *Map* file menu to display the *Edit Trace Map* window, or click on the *Trace Map* option in the *Trace* file menu to display the *Trace Map* window and configure the trace map (see Figure 15). The default trace map is enabled throughout the entire MPU address range. To select only a portion of the program to be stored in the trace buffer, specify the boundaries of this portion of the program in the *Start Address* and *Stop Address/Length* fields in the *Edit Trace Map* dialog box, click on the *Trace Enable* button, and click on the *Set* and *Done* switches.

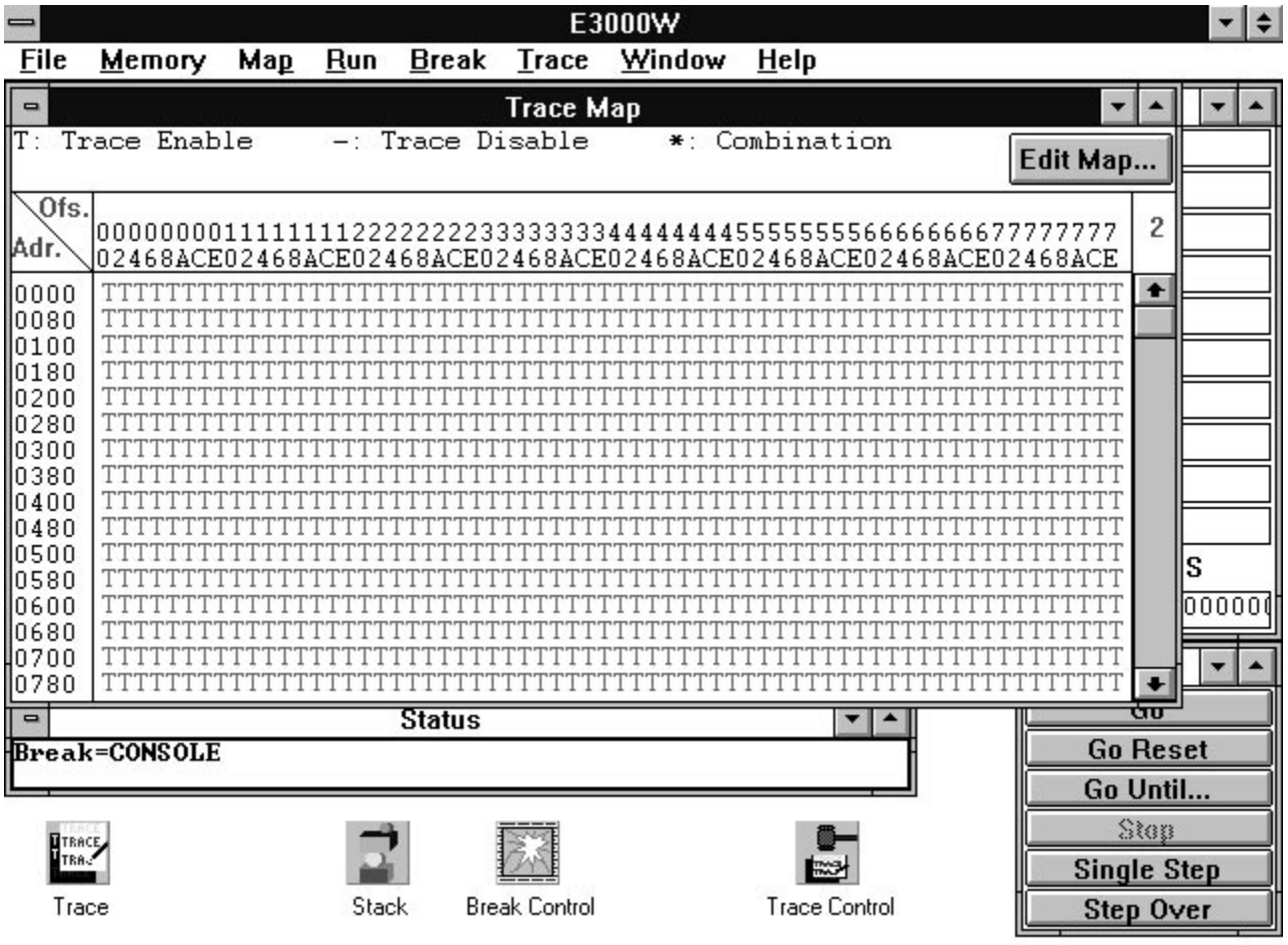


Figure 15.

2. Setting the trace clock determines what trace information will be captured in the trace buffer. Click on the *Set Trace Clock* option in the *Trace* file menu (see Figure 16). Four options are available: *Read/Write, Read, Write,* and *Phi.* The *Read/Write* option will cause all read and write cycles within the enabled portion of the trace map to be stored in the trace buffer. The *Read* option stores only the read cycles, and the *Write* option stores only the write cycles. The *Phi* option stores the machine cycles in addition to the read/write cycles. The default setting is on *Read/Write.* After making the selection, click on the *OK* switch and close the window.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

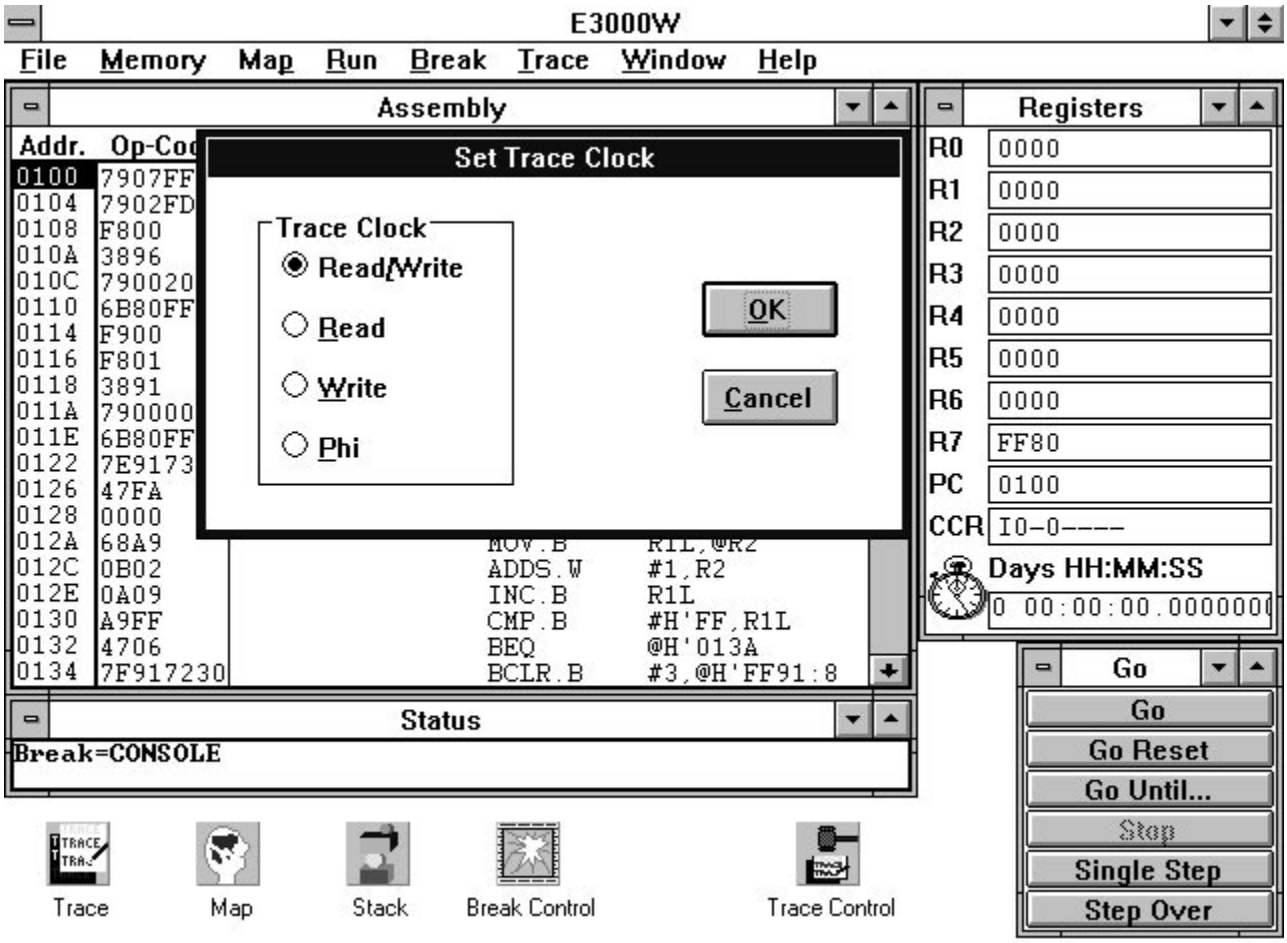


Figure 16.

3. Setting the trace trigger conditions can be done in 2 ways: **simple trace control** and **advanced trace control**.
   a. **The simple trace control** is invoked by clicking on the *Trace Control* icon, and the *Trace Control* window is displayed (see Figure 17).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

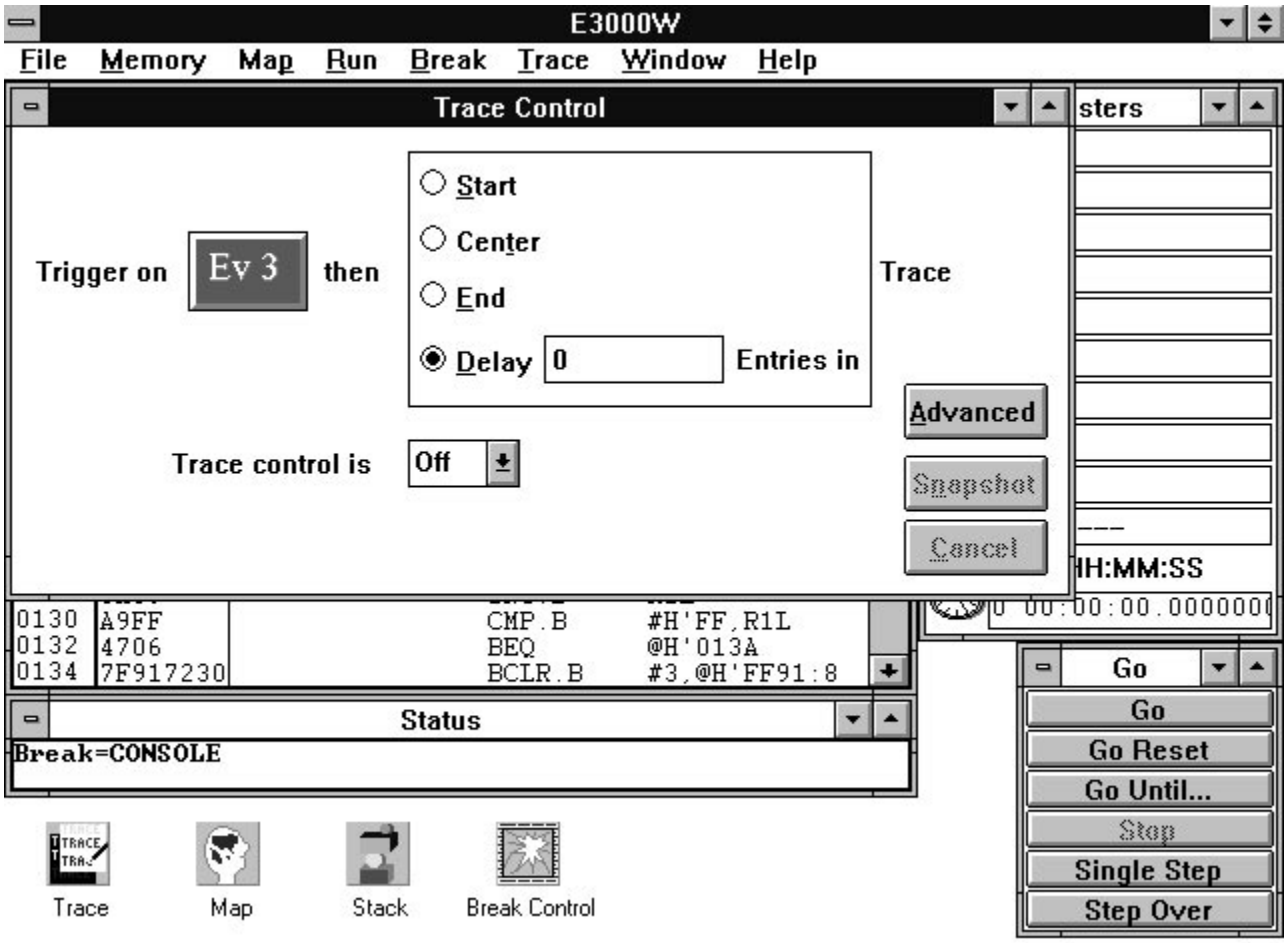


Figure 17.

Click on the *EV3* box and the *Trace Event 3* window will be displayed to select the desired trigger conditions of the trace (see Figure 18). The trace conditions displayed in this window are and can be set up identically to the hardware breakpoint conditions explained in the previous section. That is, a trace can be triggered upon a desired address, data, user signal(s), read/write operation, data size, memory cycle length, interrupt, and/or fetch operation. Also, a trace can be triggered upon a specified event count as well. After the conditions have been configured, click on the *OK* switch and the *Trace Control* window will be displayed again. The trace buffer can be set up to collect additional trace events past the trace stop boundary by clicking on the *Delay* button and specifying the number of additional trace events in the adjacent field. Finally, clicking on the O*n* option of the *Trace Control is* field will enable the trace operation.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

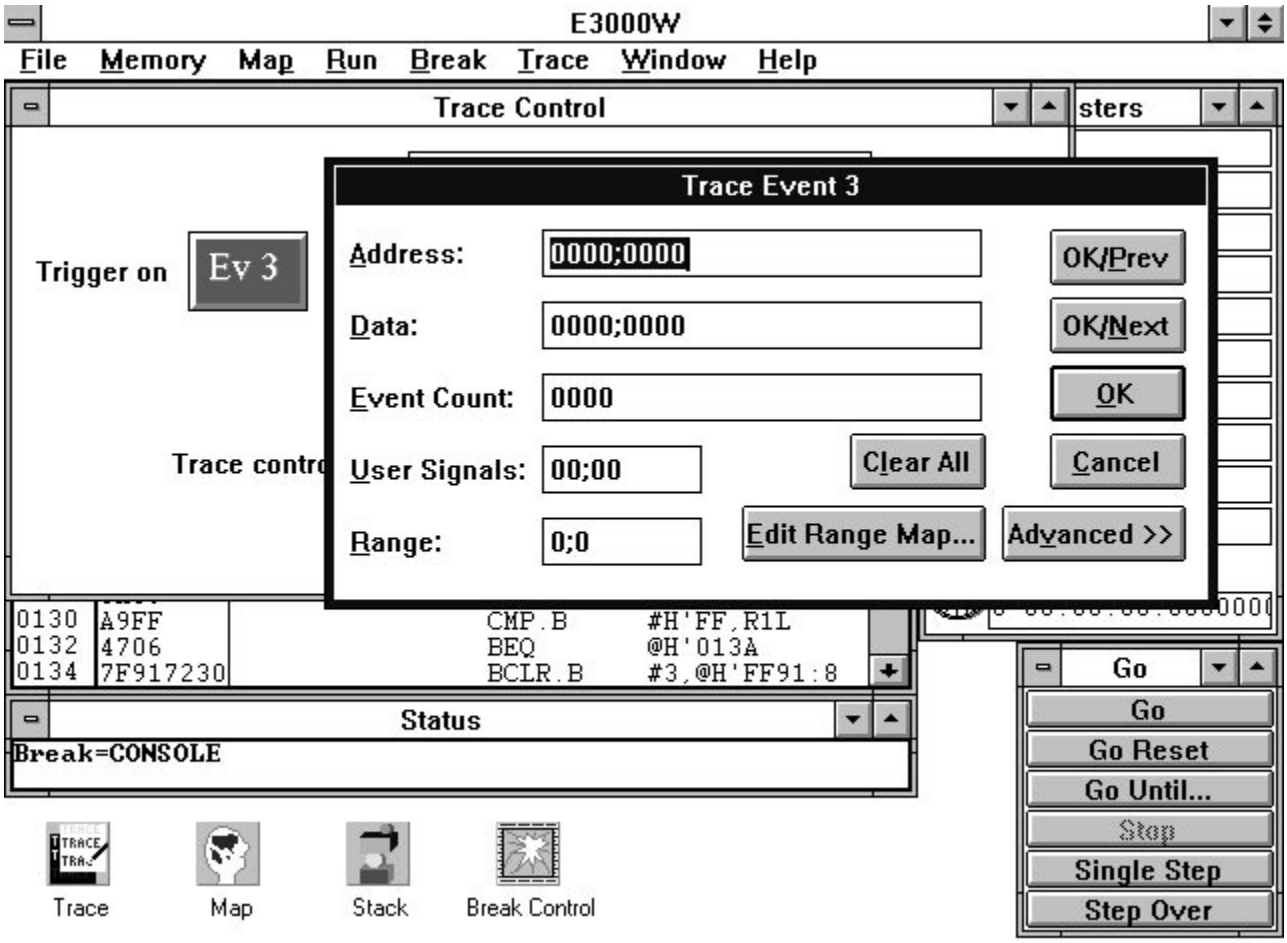


Figure 18.

b. **The advanced trace control** is invoked by clicking on the *Advanced* switch in the *Trace Control* default window. Or, alternatively, check off the *Advanced* option in the *Trace* file menu, click on the *Trace Control* icon, and the advanced *Trace Control* is displayed (see Figure 19).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

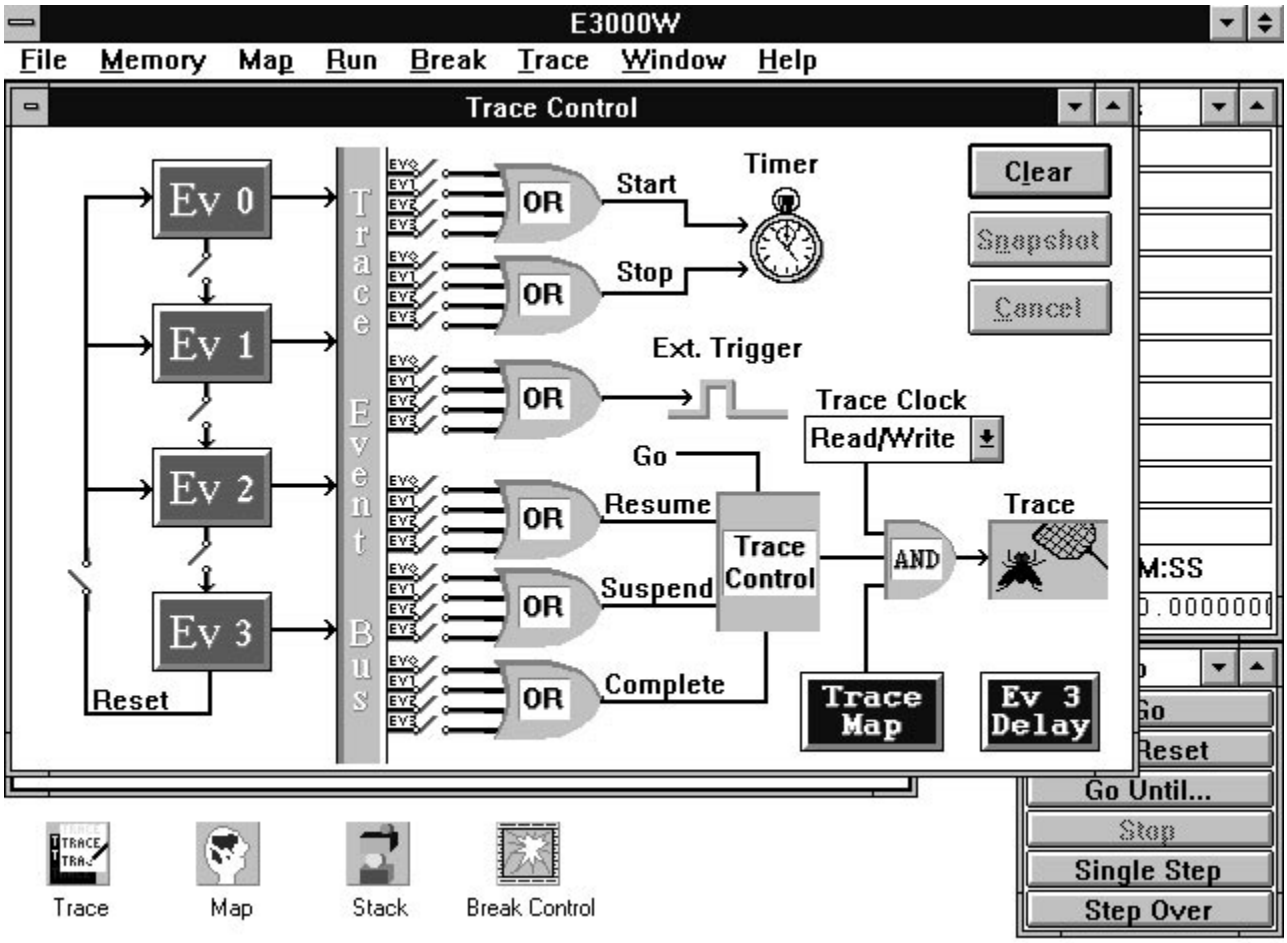


Figure 19.

A maximum of 4 tracepoints (trace trigger conditions) can be specified by clicking on anyone of the 4 *EV 0-3* boxes on the left screen, and the corresponding *Trace Event* dialog box will be shown (see Figure 20). Each trace point can be specified using the same condition parameters as for the hardware breakpoints, namely by event, trigger, and sequence (see the BREAK OPERATION section for acomplete description). In addition, 3 trace control operations are possible once trace points have been defined. **First**, the time elapsed between 2 trace events can be recorded in the timer field below the *Registers* default window (within the main application window). Simply close the switch to the OR-gate corresponding to the *Start* trace event, and do likewise to the switch of the OR-gate corresponding to the *Stop* trace event, and close the *Trace Control* window. **Second,** a trace event can be set to trigger an output pulse to an oscilloscope. Select the desired trace event by closing the corresponding switch to the OR-gate whose output is labeled *Ext Trigger*. **Third,** a collection of code events loaded into the trace buffer memory can be suspended (stop trace), resumed (re-start trace from where it stopped), or completed (no further code events are recorded upon a specified trace event condition). Simply close the corresponding trace event switches to the appropriate OR-gate. Note that a trace event cannot trigger more than one of the above 3 operations. For example, a code section execution can be captured in the trace buffer upon trace event EV1, suspended upon trace event EV2, resumed upon trace event EV3, and completed upon trace event EV0. EV1 can be triggered upon an address, EV2 can be triggered

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

upon an interrupt, EV3 can be activated upon a user signal at an external probe in the target system, and EV0 can be activated upon a data byte present on the data bus.

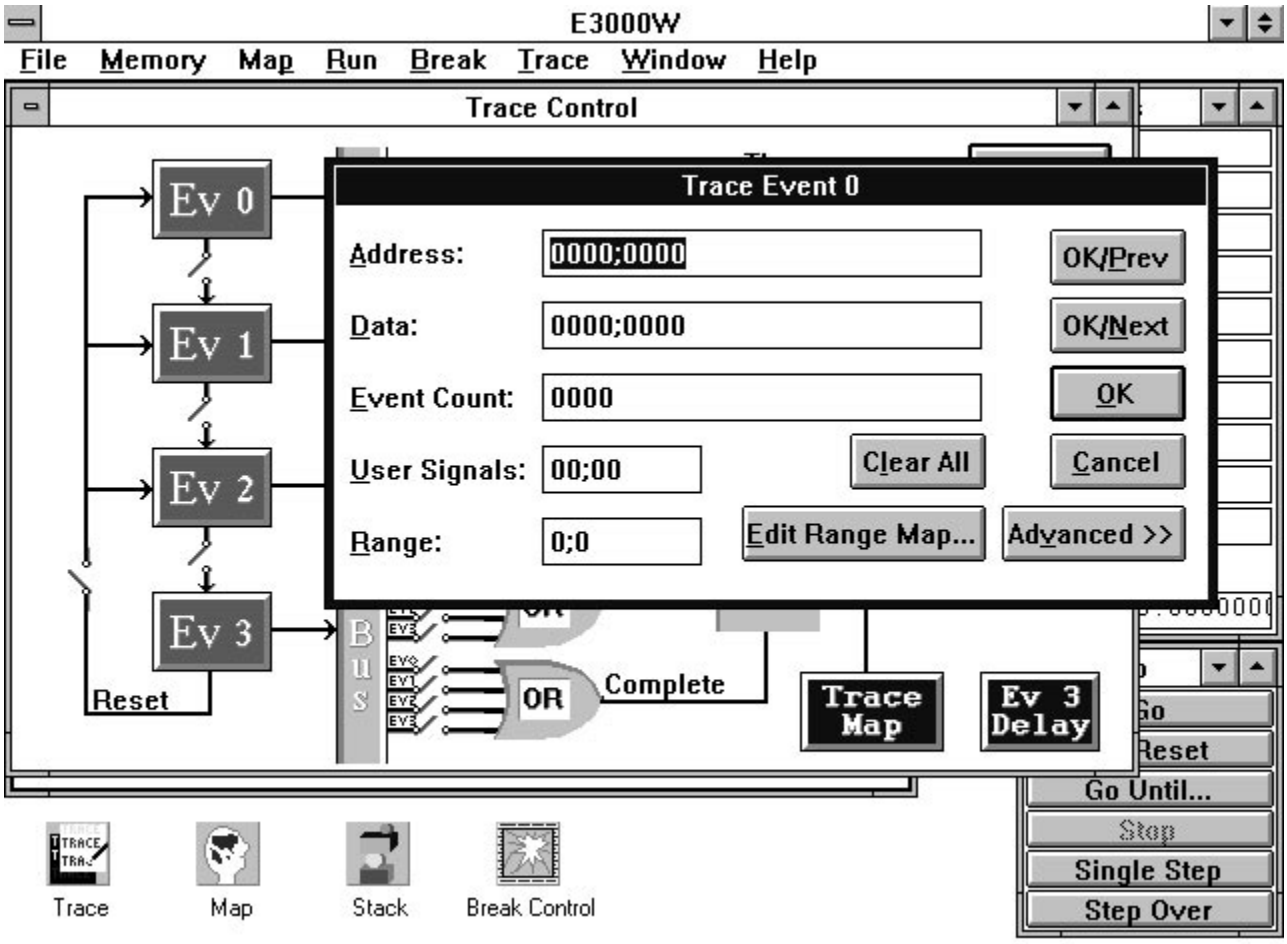


Figure 20.

TRACE DISPLAY

The trace buffer can be displayed anytime after a break or a trace complete condition by double-clicking on the *Trace* icon, and the *Trace (clock =)* dialog box is displayed (see Figure 21).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

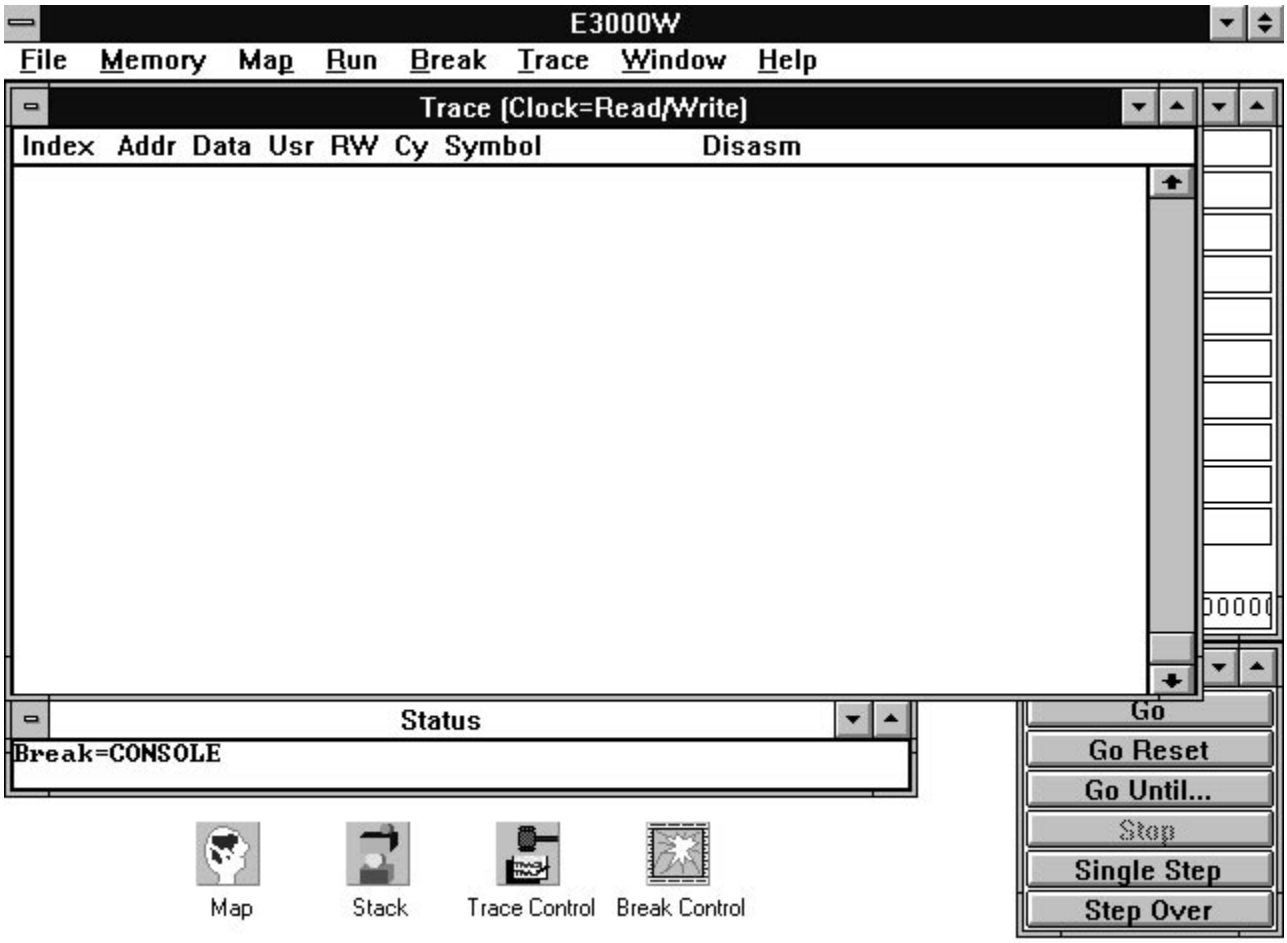


Figure 21.

There are 2 types of trace displays depending upon the chosen trace clock (phi or any other option). If the *Read/Write*, *Read*, or *Write* trace clock option is chosen, the trace displayed contains the following elements:

1. An **index** which indicates the number of trace entries, and which is normally a negative number. Positive index numbers are the trace entries that accumulated after the trace event 3 (EV3) condition has triggered (according to the specified delay in the *Trace Control* dialog box).
2. An **address** column indicating the contents of the program counter during each CPU cycle.
3. A **data** column indicating the data bus contents during the CPU cycle.
4. A **user** column indicating the status of the user external signals.
5. A **read/write cycle (R/W)** column showing the type of CPU cycle and the number of cycle states.
6. An **access (CY)** column indicating the type of access (internal, external, EEPROM, or no EEPROM).
7. A **symbol column** showing the corresponding symbol at that address.
8. A **disassembly** column showing the traced assembly code.

If the phi trace clock option is used, machine cycles are recorded into the trace buffer memory as well, and the trace display does not contain the symbol and dissasembly columns, but a **control** column instead (see Figure 22). This column has 7 entry flags that indicate (from left to right): evaluation chip wait state (W), opcode fetch (R), opcode decode (D), interrupt acknowledge (I), external 3-state access or internal 2-state access (3/2), branch not taken flag (B), and any type of jump taken (J).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

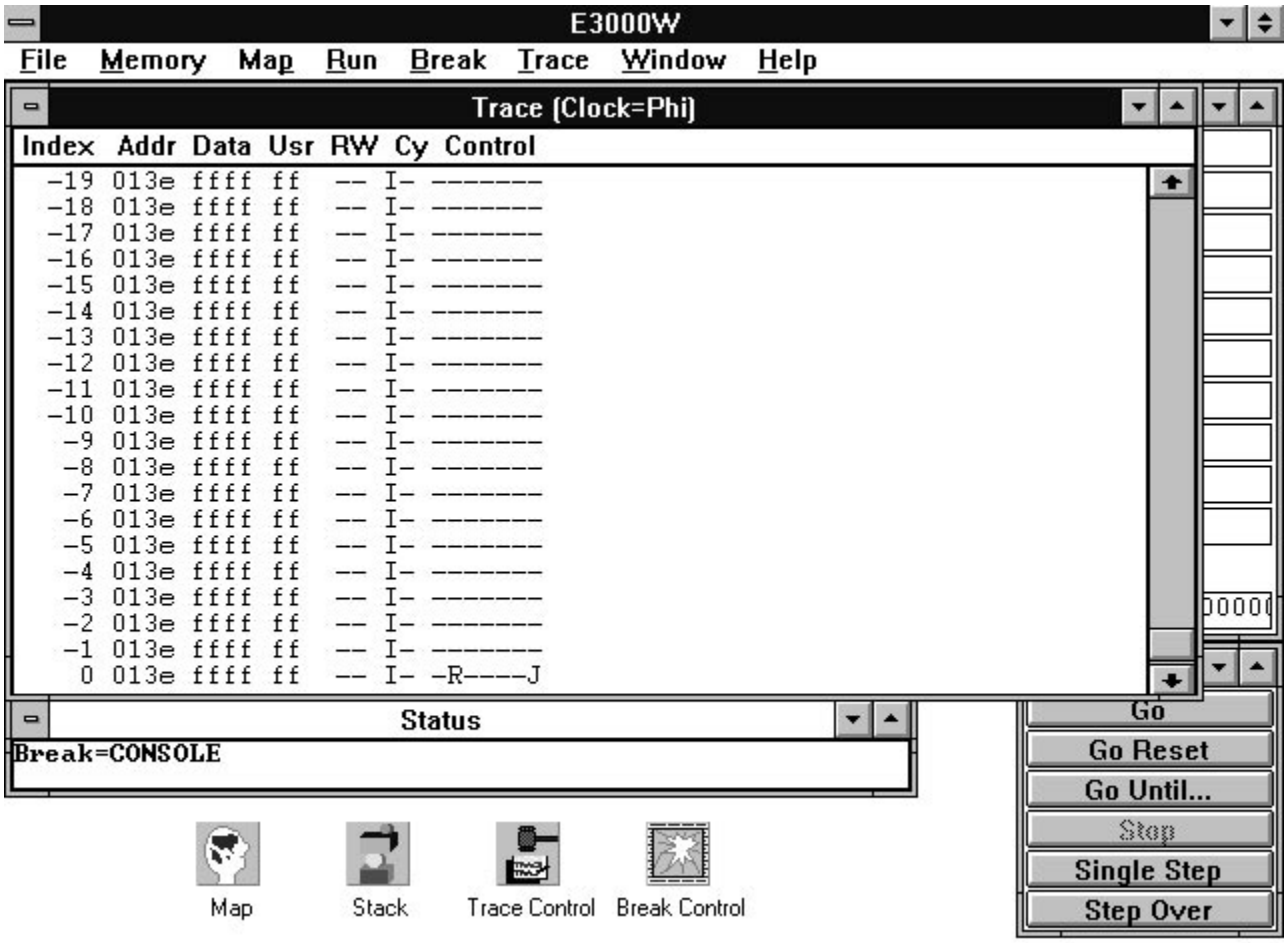


Figure 22.

PROGRAM EXAMPLES AND LISTINGS

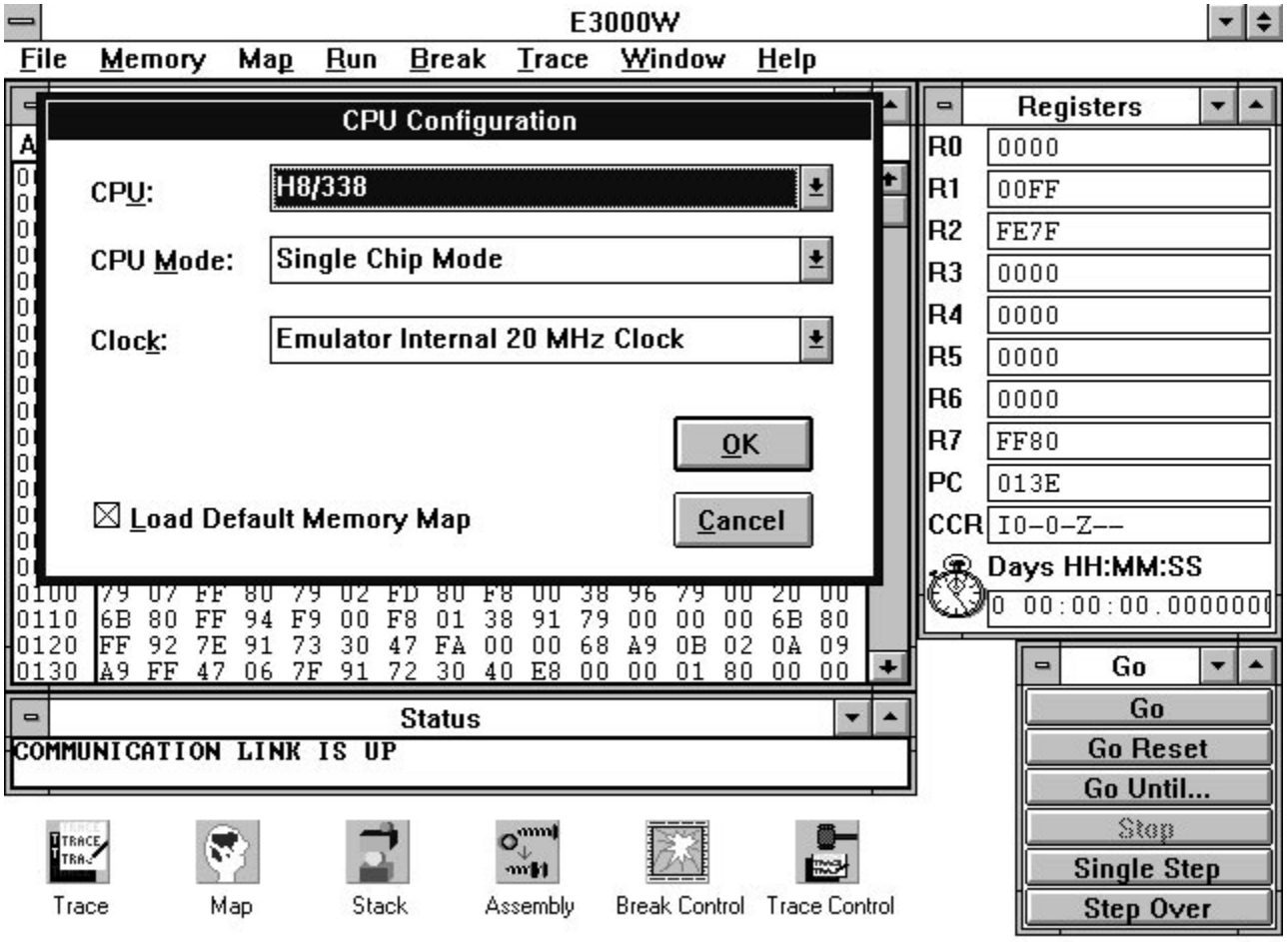
All the setup, download, and debug features of the E3000 explained previously will be exemplified in detail by using 2 program examples. **Program 1** (tut_pol) fills a 256-byte emulator memory block between H'FD80 - H'FE7E with incrementing byte data from H'00 to H'FE using the polling method. The 16-bit timer module of the H8/3XX is used to sequentially load the data into the memory on a specific count (every 1.6384ms with a 5MHz timer clock). When the incremented data reaches the value of H'FF, the program enters the sleep mode and execution stops. **Program 2** (tut_int) achieves the same result but uses the end-of-count timer interrupt to load the data into the emulator memory. A listing of both programs is given in Appendix A at pages 56-57.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

HANDS-ON EXERCISES

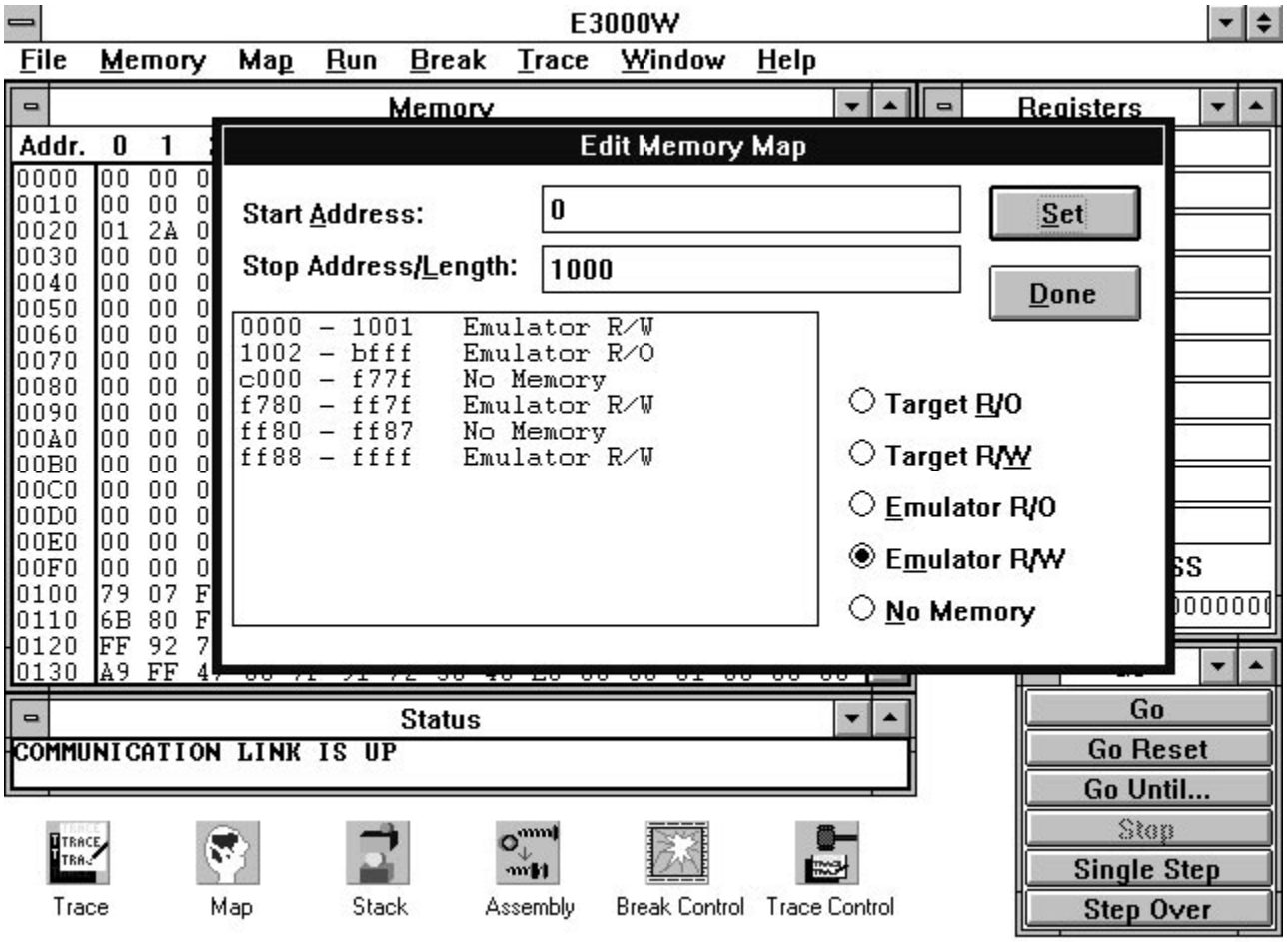
This section uses the above-listed program examples in order to show how a user can typically utilize the features of the E3000 in order to download, run, and use the debug tools explained in the previous sections. Before starting the exercises, the user must have the E3000 correctly connected to the PC according to the indications given in the E3000 Setup section. The green *PWR* LED on the front panel of the E3000 must be lit indicating that serial connection is established. The 2 programs listed must be assembled, linked, and in s-record format file (.abs).

1. **Configure the E3000 for emulation:**
   a. Select CPU type to H8/338: *File* menu - *CPU Configuration* - *CPU* (choose H8/338) - *CPU Mode* (choose single-chip) - *CPU Clock* (choose 20MHz internal).



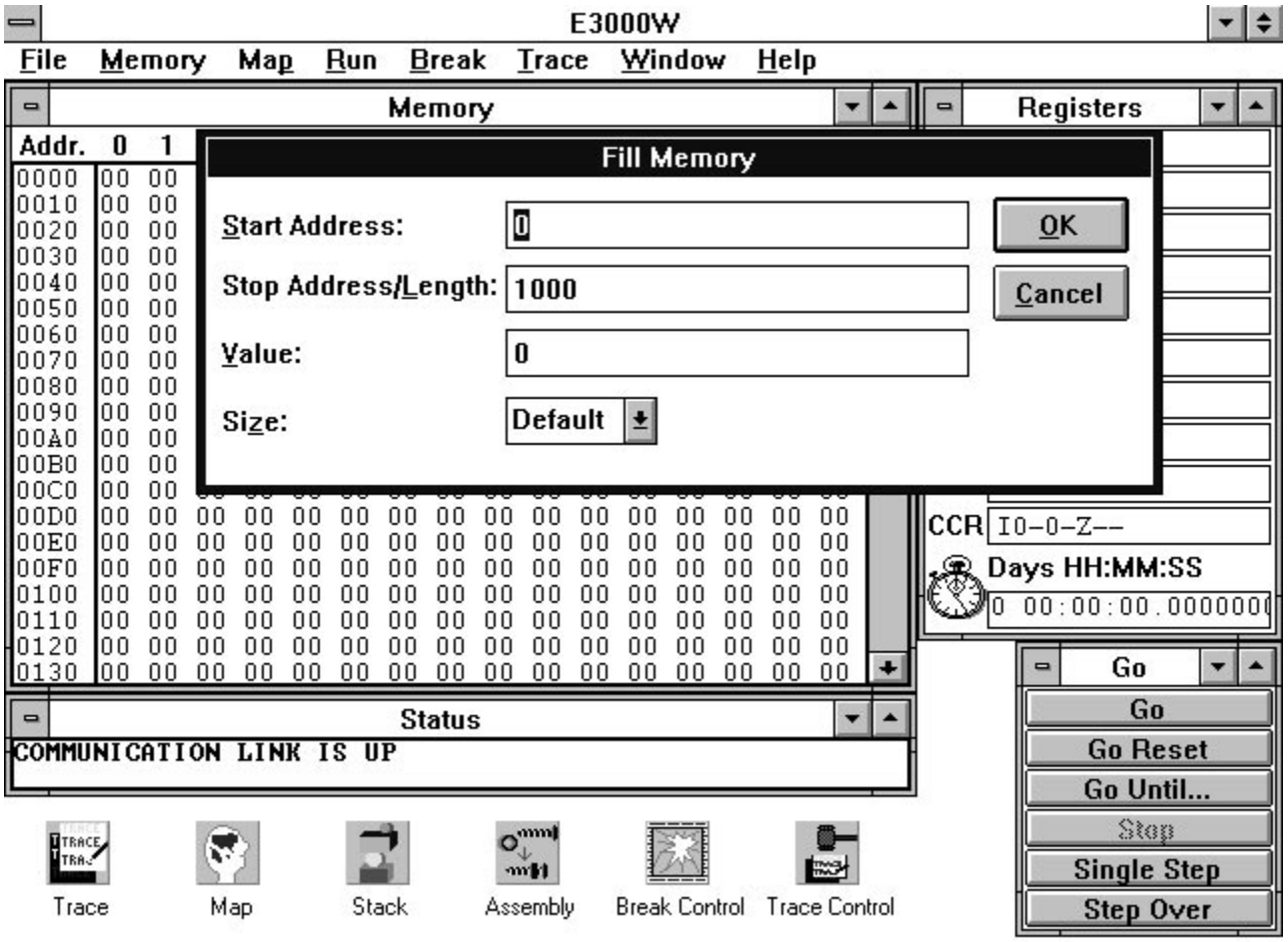
   b. Set the memory map: *Map* file menu - *Edit Memory Map* - *Start Address* (type 0) - *Stop Address* (type 1000) - choose *Emulator R/W* - click *Set* and *Done.*

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



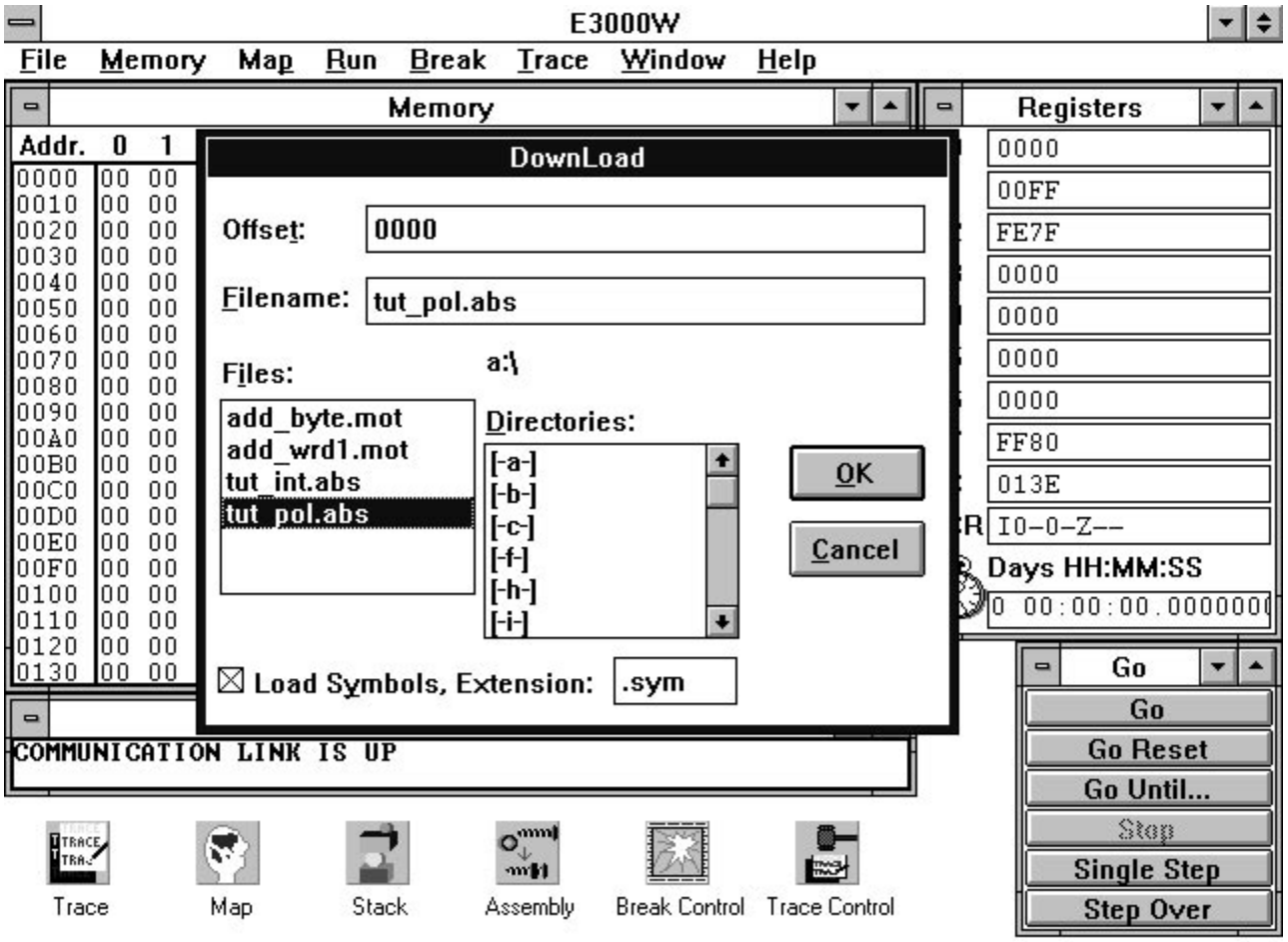
c. Fill the memory with NOPs: *Memory* file menu - *Fill* - *Start Address* (type 0) - *Stop Address* (type 1000) - *Value* (type 0) - click *OK*.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



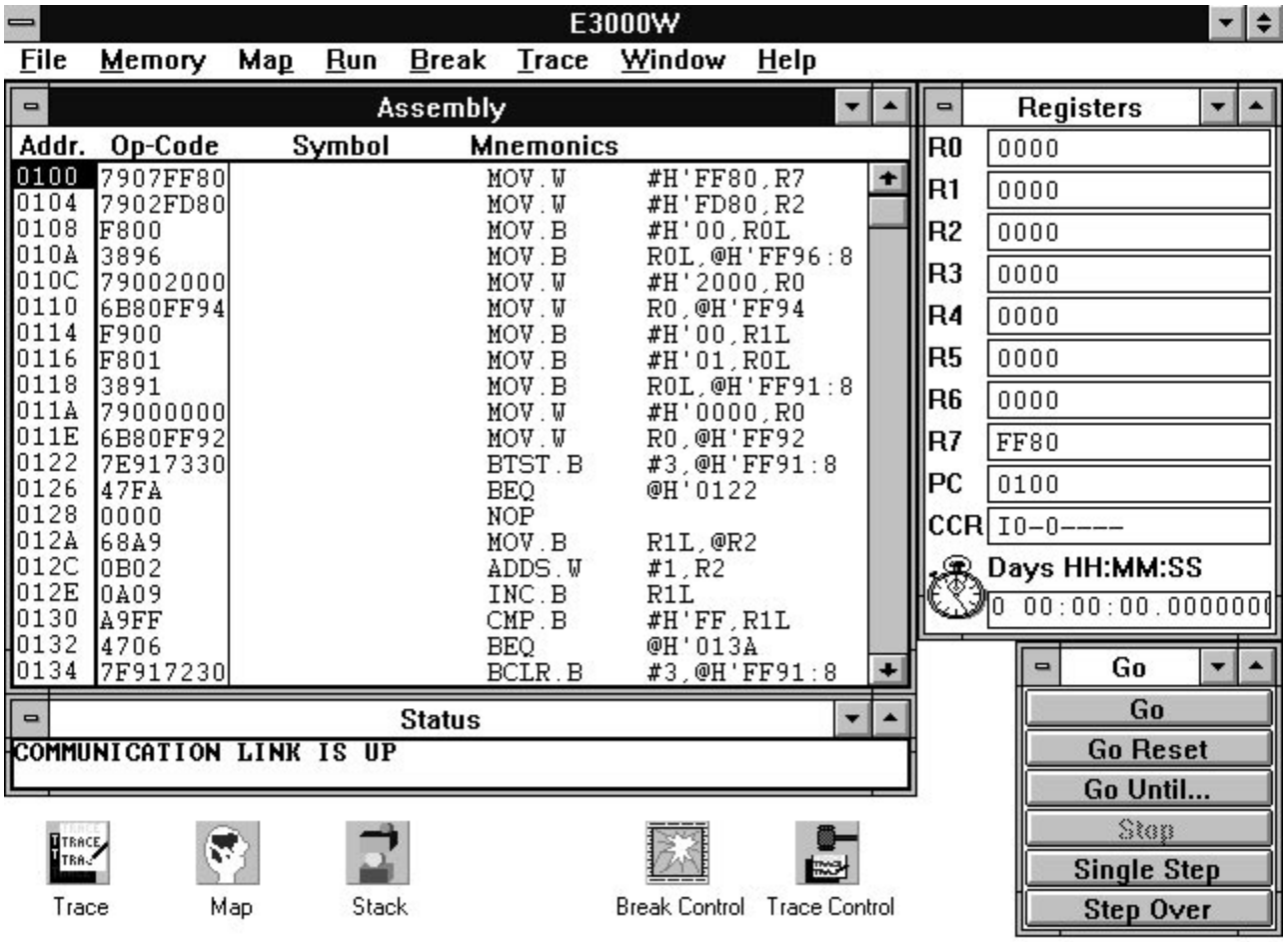
2. **Download program 1 (tut_pol.abs):**
   a. Download the s-record file of program 1 (tut_pol.abs) into the emulator R/W memory: *File* menu - *Download* - *Filename* (type tut_pol.abs) - *Directories* (type directory) - click *OK*.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



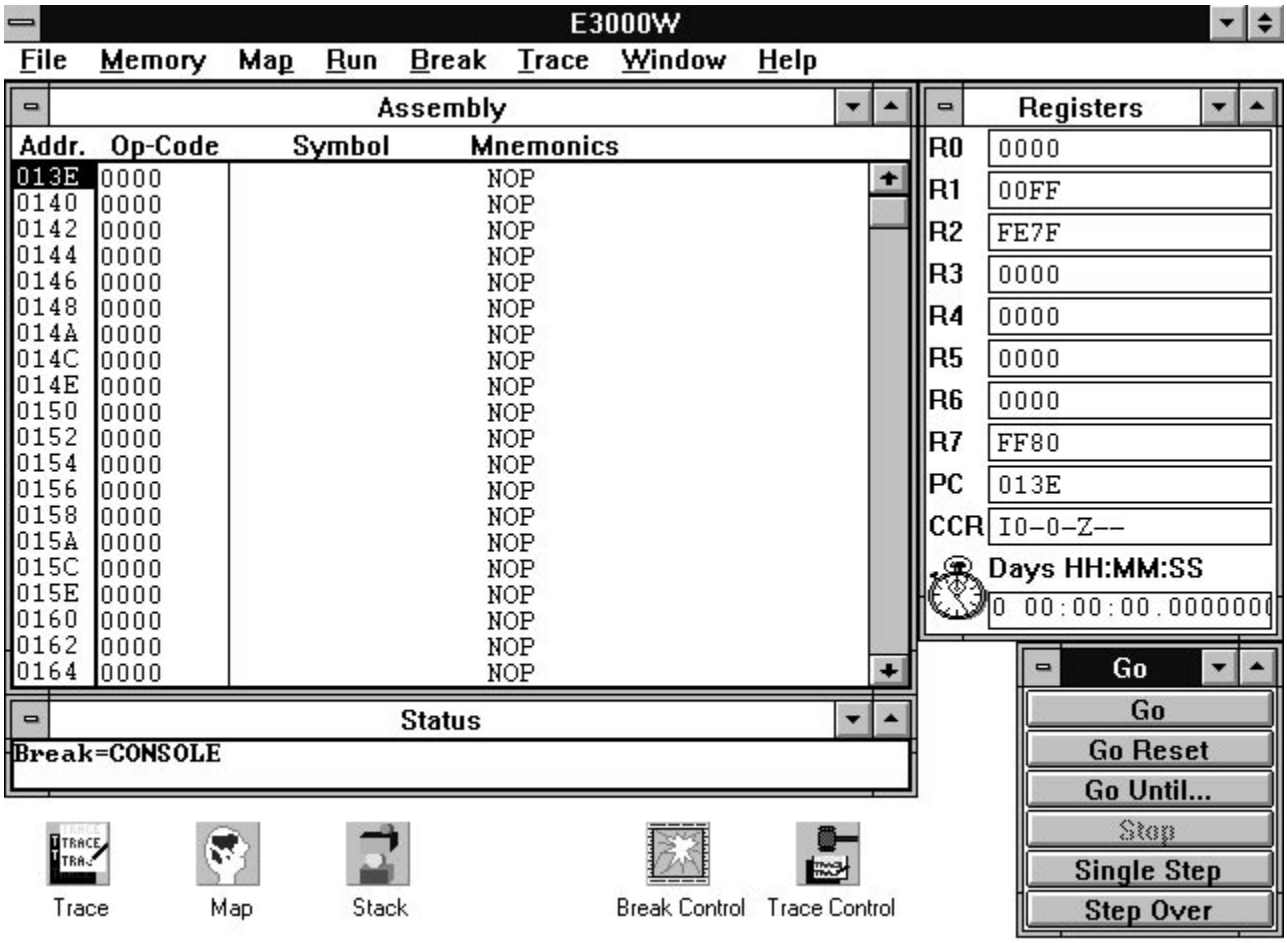
3. **Run program 1 (tut_pol.abs):**
   a. Bring up the *Assembly* window: click on the *Assembly* icon.
   b. Set PC at start program address: click on the *Addr* column - *Goto Memory Address* window (type 0100 in the *Address* field) - highlight *PC* field in the *Registers* window and type 0100. Now the program is ready to be run.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



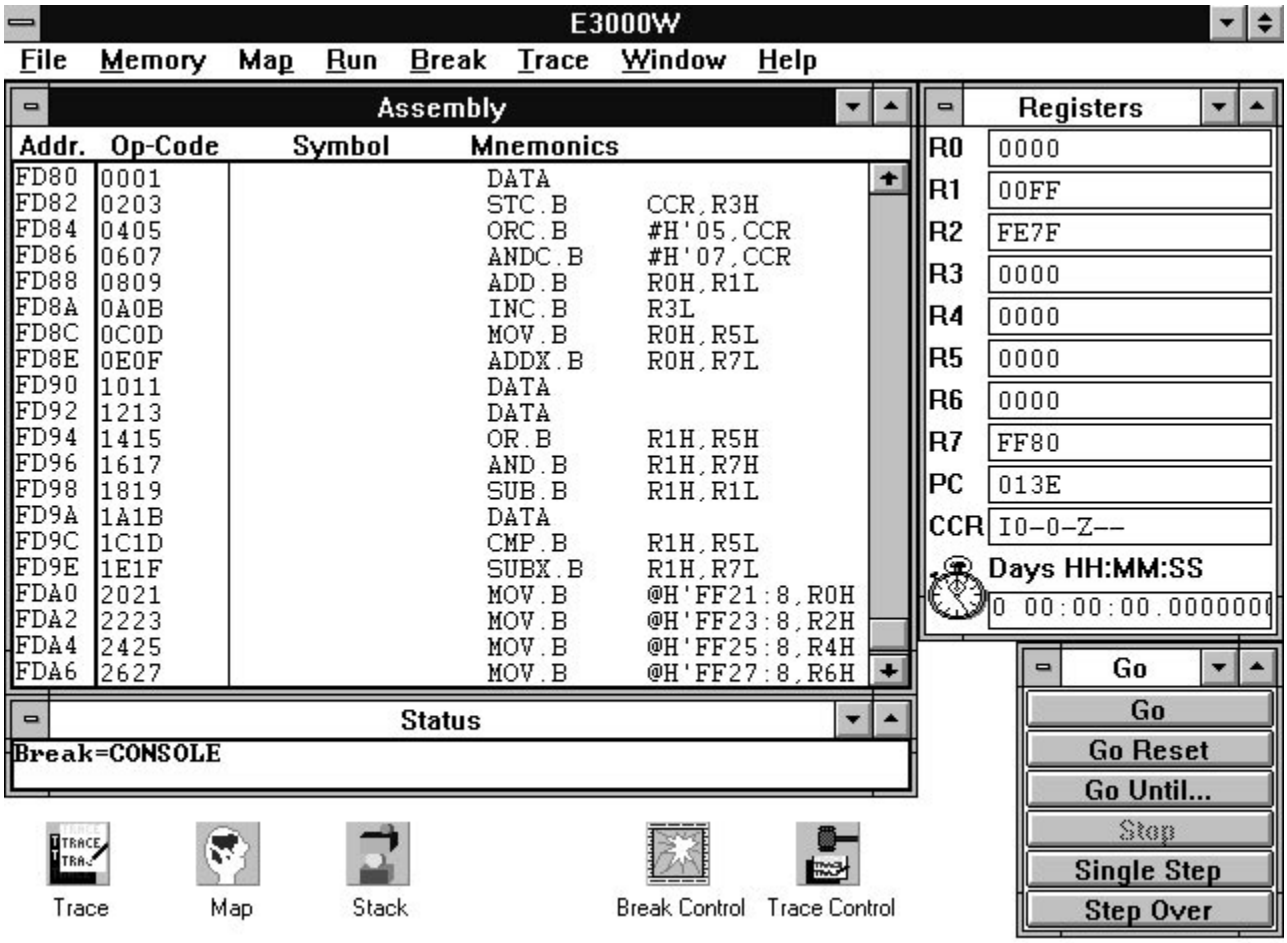
c. Click on the *Go* switch and the program will stop at address 013e (upon the SLEEP instruction) after fillling up the assigned memory area with the incremented data (0 - ff). By pressing the *Stop* switch, we can observe the contents of the CPU registers: R1 contains h'ff, R2 is loaded with the memory address containing the last data in the memory block (h'fe), and the PC points at the next instruction following the SLEEP command.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



4. **Display memory contents:**
   To display the data contents loaded by the program in the emulator memory, double-click with the cursor placed anywhere inside the address column, and the *Go to memory address* dialog box will appear. Fill the *Start Address* box with the beginning address of the memory block in which the data has been loaded. The memory block between H'FD80 - H'FE7F has been filled with incrementing data from H'00 to H'FE.

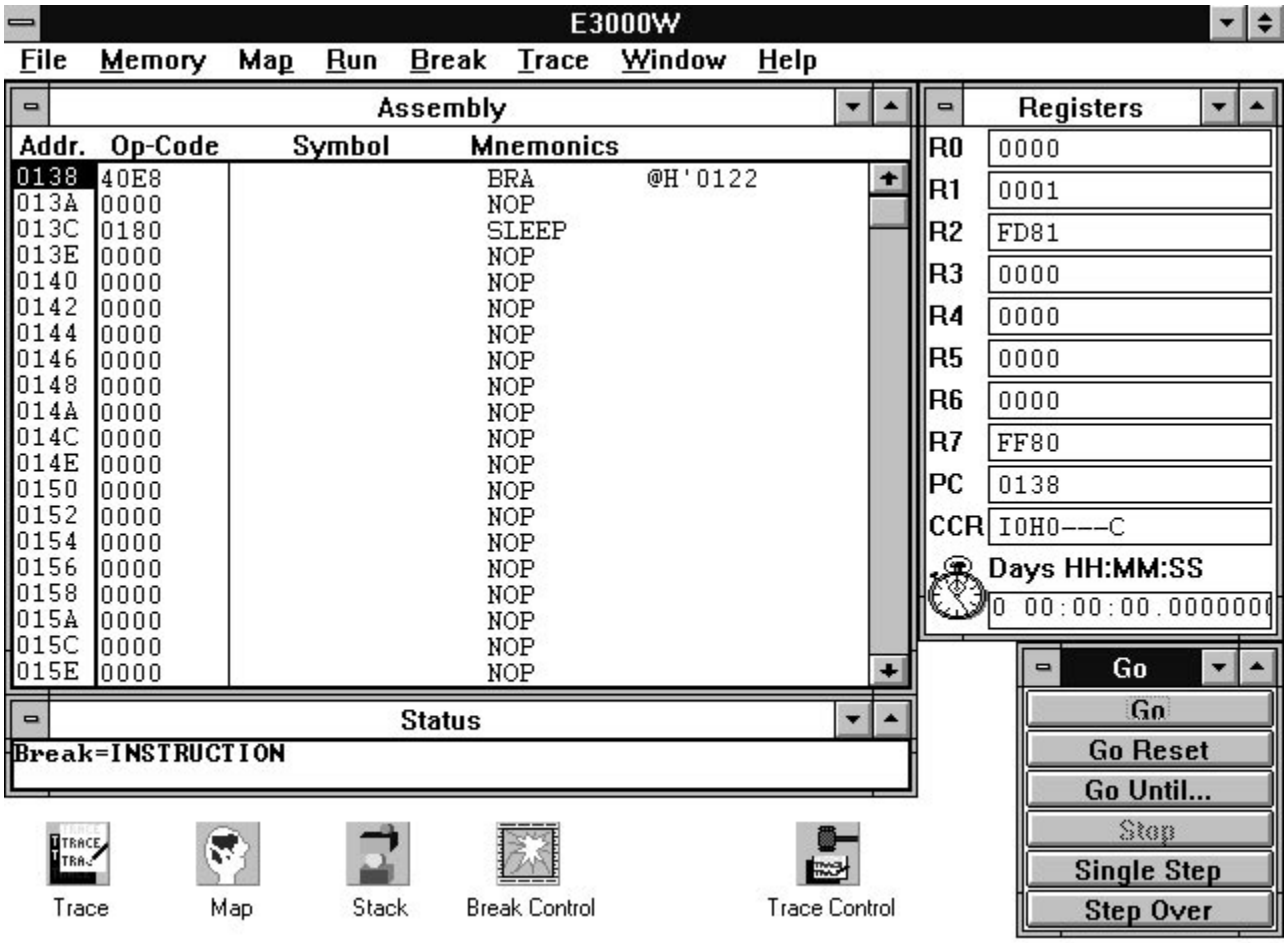
HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



5. **Re-initialize the MCU:**
   Press *Re-initialize CPU* option in the *Run* file menu (thus forcing a hardware reset on the CPU). The timer and I/O port registers are initialized, but not the CPU registers and the emulator memory. Clear the contents of R1, R2, and initialize the CCR contents by highlighting the register contents and press the BACKSPACE key on the keyboard. Press the TAB key to highlight the next register and perform the same operation. Load the PC with the program start address.

6. **Set a software breakpoint and run the program:**
   Click on the *Break Control* icon, and then click on the *Instruction Breaks* switch inside the *Break Control* window. Enter address 0138 in the *Expression* field, turn the *Status* switch on, and click on the *OK* button. Close the *Break Control* window, and run the program by pressing the *Go* button. The program execution will stop at address 0138, R1 will contain the next data (H'01), and R2 will point to the next memory location (H'FD81).

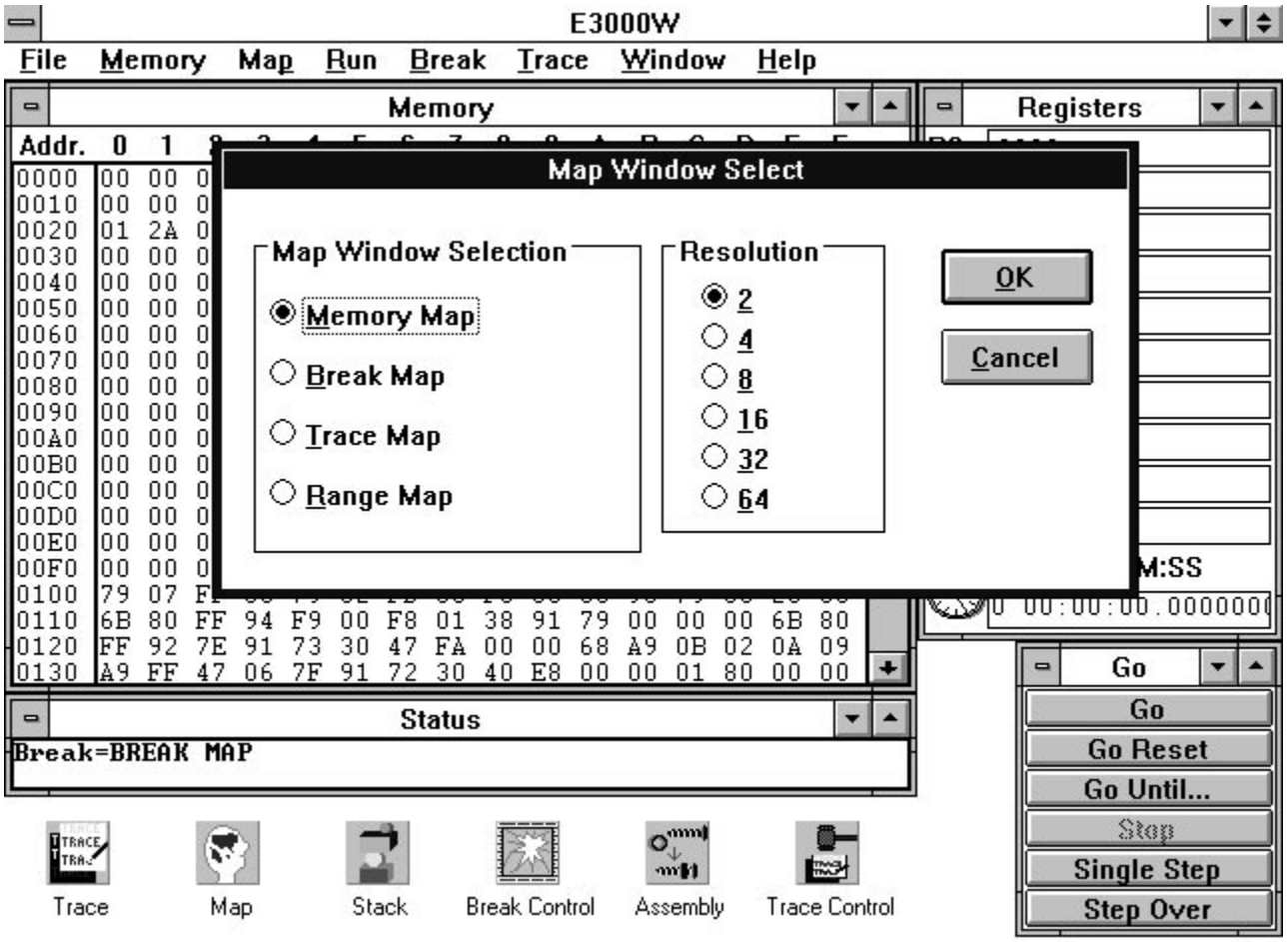
HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



By pressing the *Go* button again, the program will stop at the 0138 address after going though the loop. The contents of R1 are now H'02 and R2 is loaded with the next memory location (H'FD82). By repeating this process, the values in R1 and R2 are incremented on each execution pass. Note that software breakpoints are not counted, that is a break cannot be inserted after a certain number of loop executions.

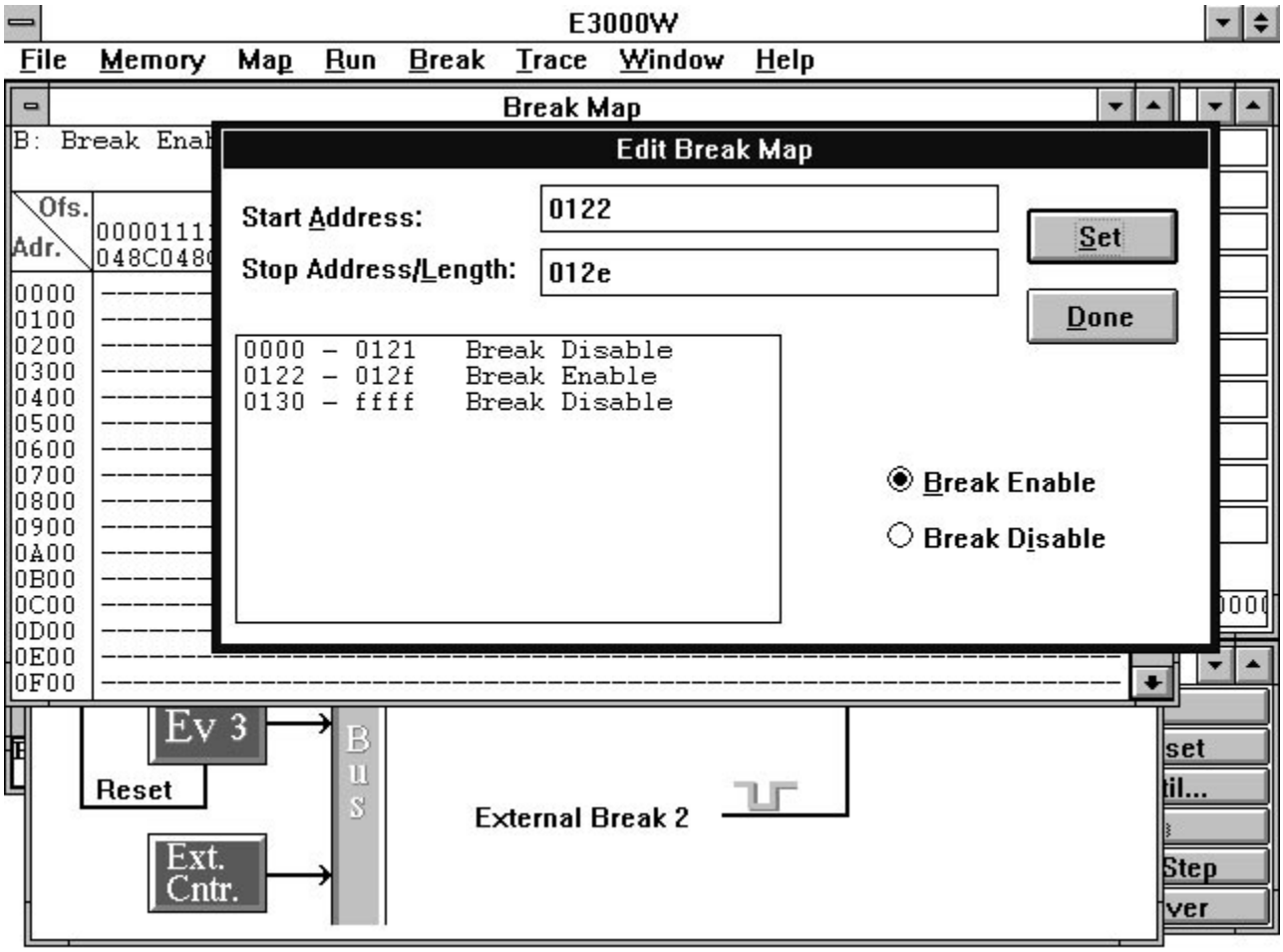
7. **<u>Set a break range and run the program:</u>**
   First, re-initialize the CPU, clear the CPU registers, and load the PC with H'0100. Then, click on the *Map* file menu, and click on the *Map Window Select* option. Choose the *2* option in the *Resolution* column, and click on the *OK* button. This action will display any memory map in byte resolution.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



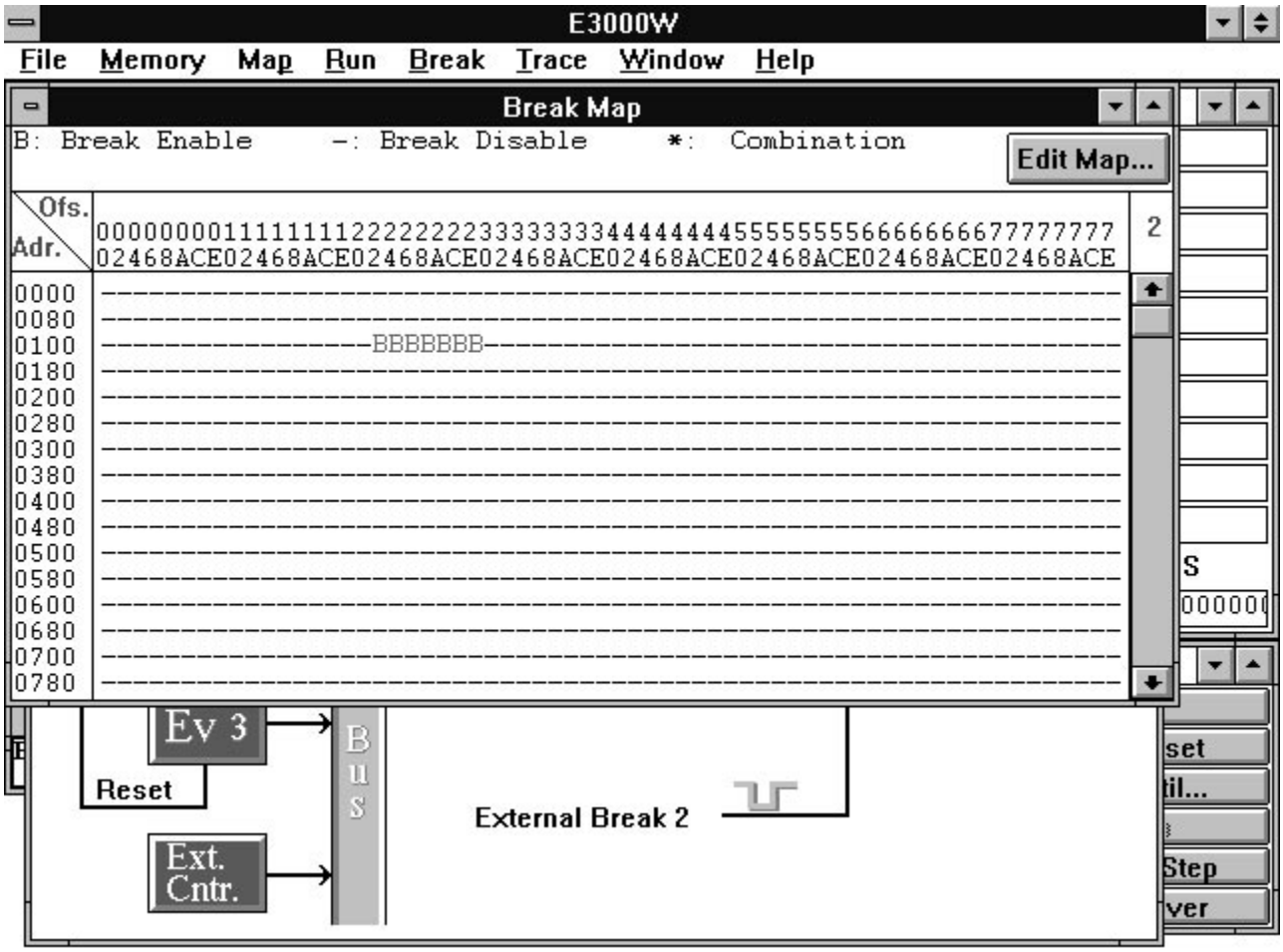
Add a breakpoint range by bringing up the *Break Control* window, clicking on the *Break Map* option, and clicking on the *Edit Map* switch. Enter H'0122 in the *Start Address* field and H'012e in the *Stop Address/Length* field, and click on the *Break Enable, Set, and Done* switches.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



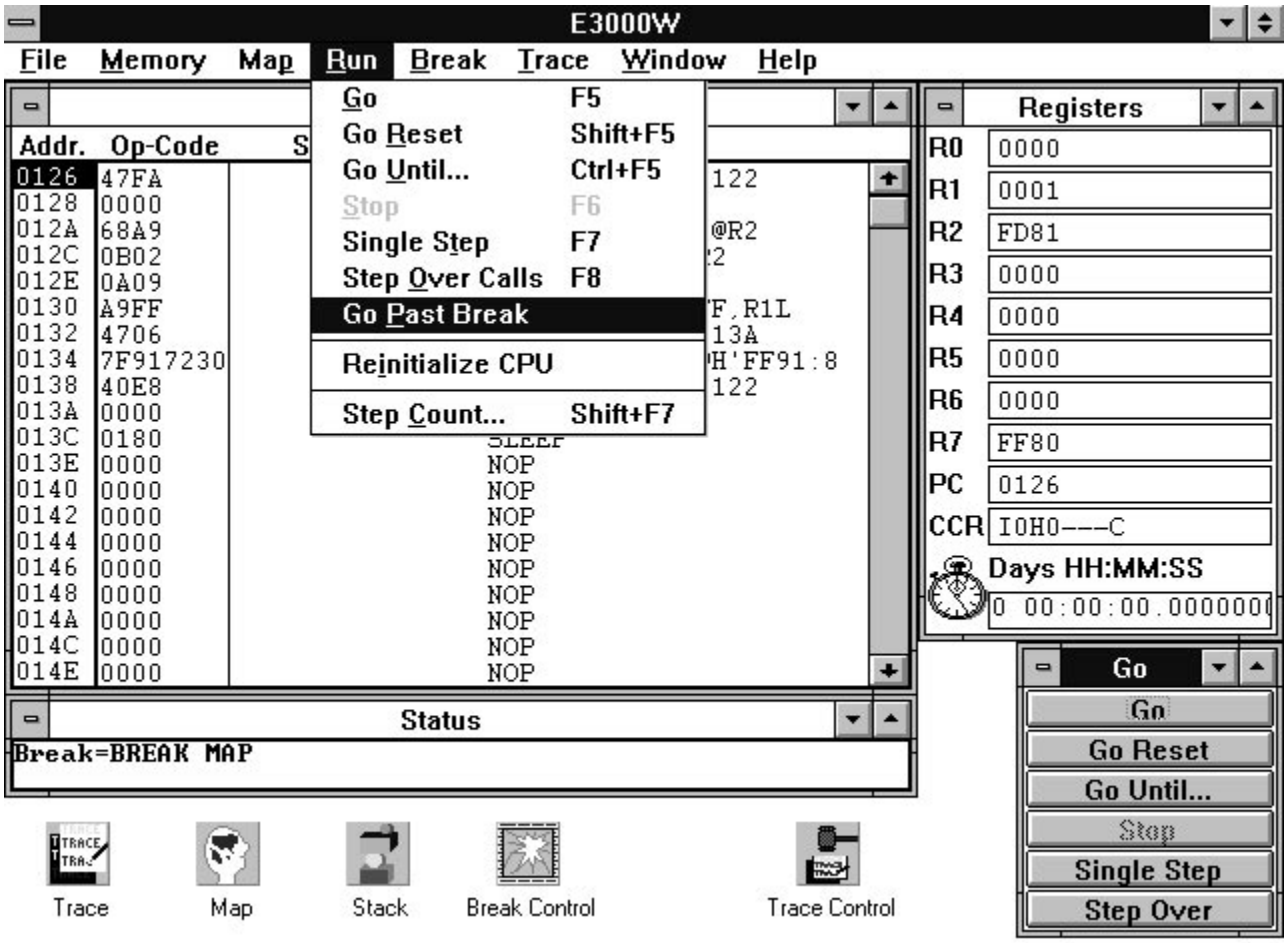
The *Break Map* window will display a *B* under locations H'0122 - H'012e.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



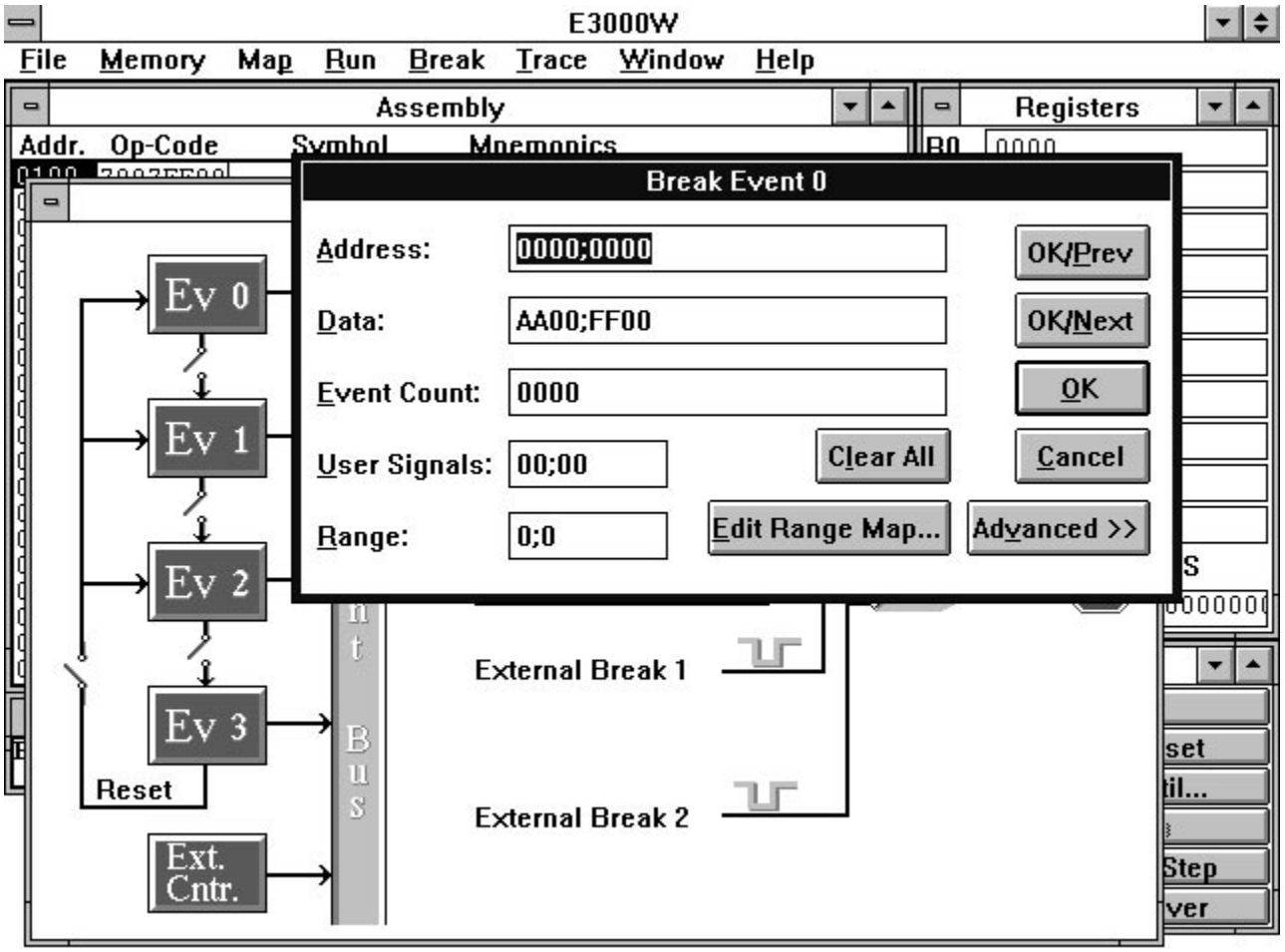
Close the *Break Map* and *Break Control* windows, and run the program by clicking on the *Go* button. The program will break at location H'0122. Press *Go* again, and the PC will point at location H'0128, skipping H'0126. Press *Go* again and the PC will point at H'012C, skipping H'012A. Press *Go* again and the PC will point at H'0130, skipping H'012e. Run the program again, and the PC will stop at H'0138, which has been earlier configured as a software breakpoint. The reason the PC skips every other location within the break range is that the *Go Past Break* switch is checked on, meaning that the PC will stop at the next address past the breakpoint address. The program will break at each consecutive memory location within the break range **only if the *Go Past Break* switch is checked off.**

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



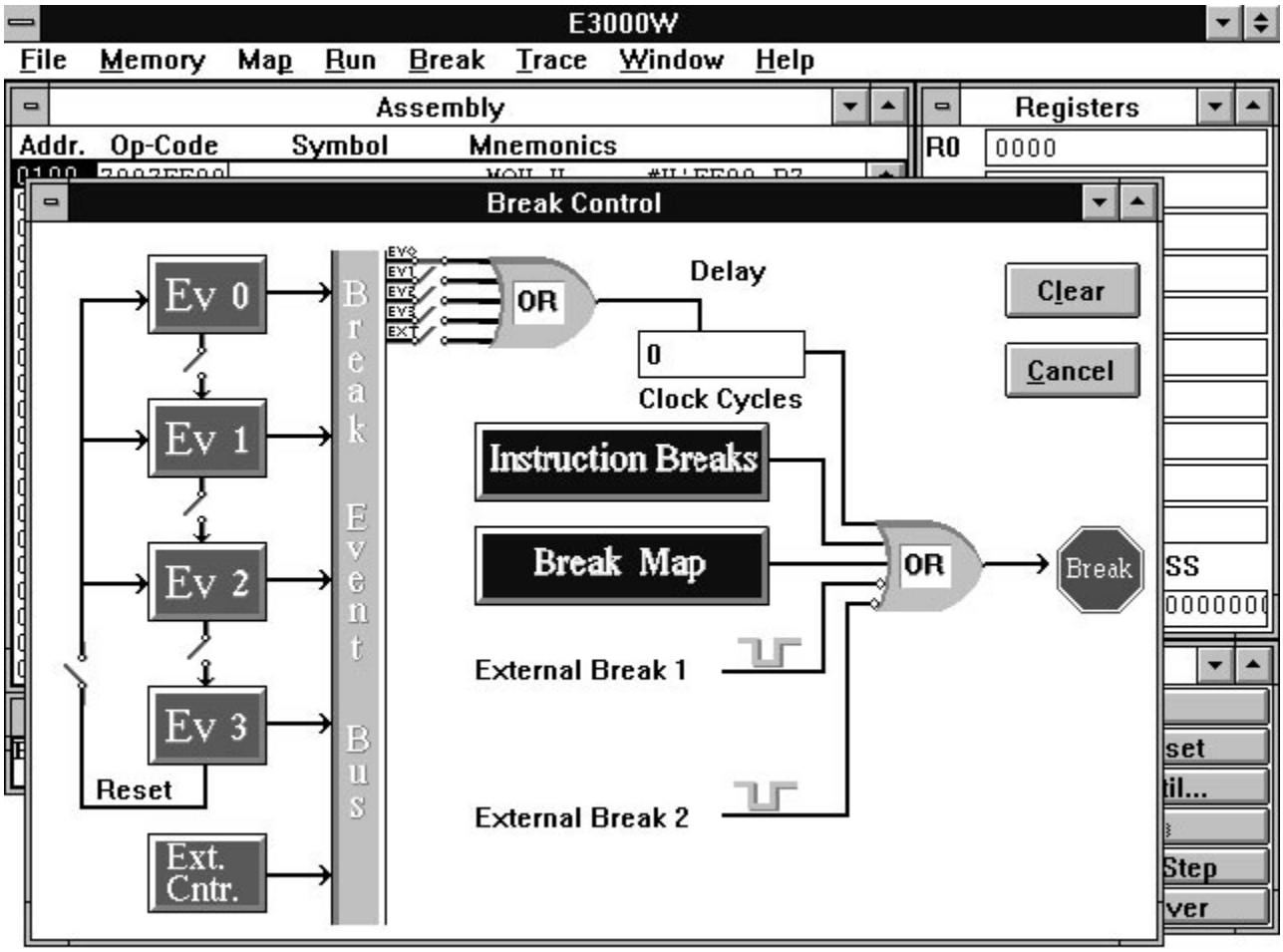
8. **Set a hardware breakpoint:**
First, clear the previous software breakpoint by accessing the *Break Control* window, clicking on the *Instruction Breaks* switch, and turning of the *Status* switch. Next, clear the range breakpoints by clicking on the *Break Map* switch in the *Break Control* window, click on the *Edit Map* button, highlight the H'0122 - H'012e break range, and click on the *Break disable, Set,* and *Done* switches. To set a hardware break point, open the *Break Control* window, and click on the *EV0* button to display the *Break Event* window. Highlight the *Data* field, type AA00:FF00, and click on the *OK* button. This means that a breakpoint will occur when a H'AA byte data is detected on the data bus. Note that only the upper 8 bits of the data bus are used and that the lower 8 bits must be masked. Typing F's allow the bits to be recognized, while typing 0's mask the bits.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



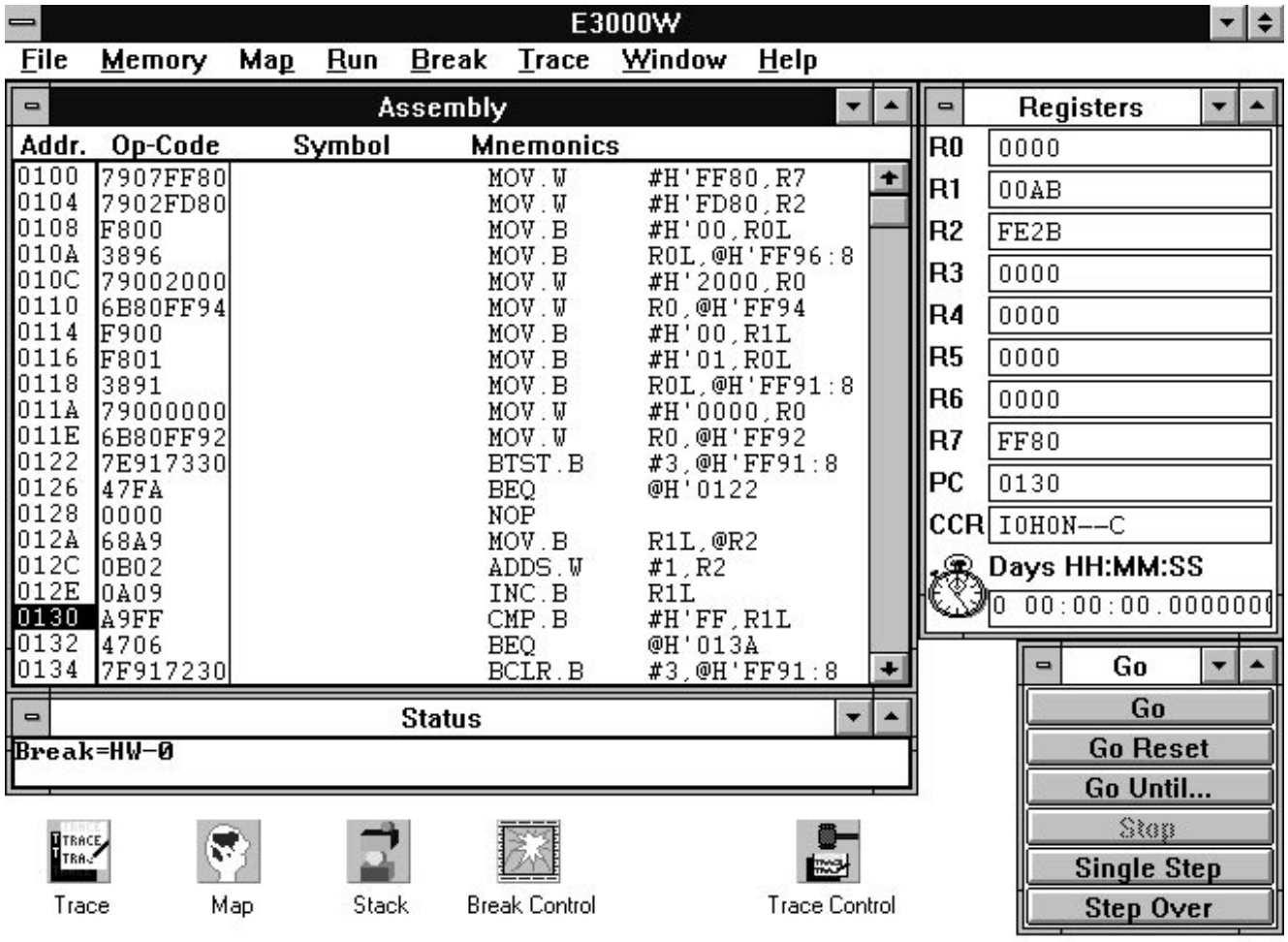
To enable the breakpoint setting, close the *EV0* switch to the OR-gate input in the *Break Control* window, and close the window.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



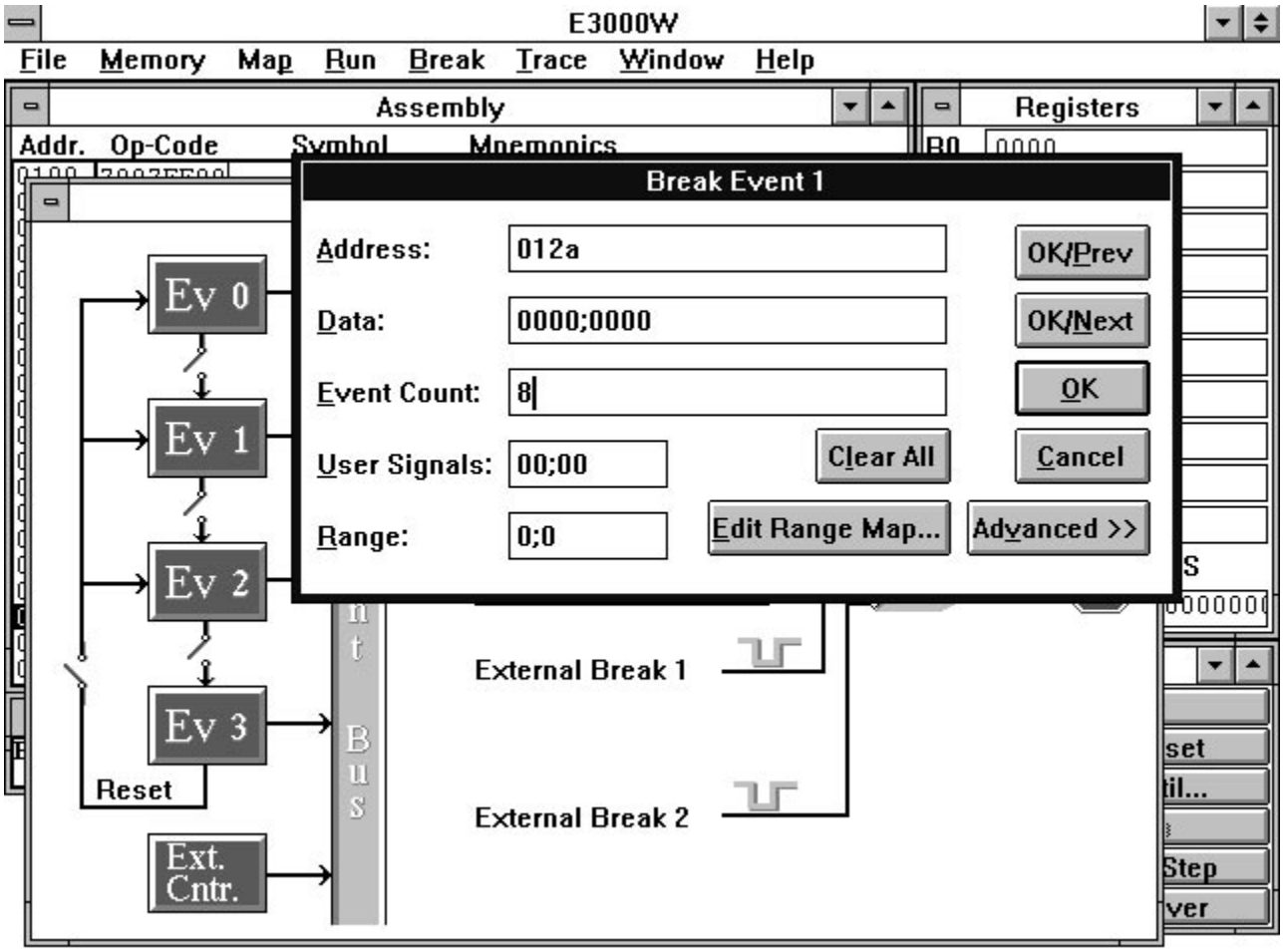
Run the program by pressing the *go* button. The program execution will stop at location H'0130, after the emulator detects a H'AA byte data onto the data bus. Observe that R1 contains the next byte data (H'AB) to be transferred into memory, since R1 has already been incremented by the time the emulator firmware issues a data break.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



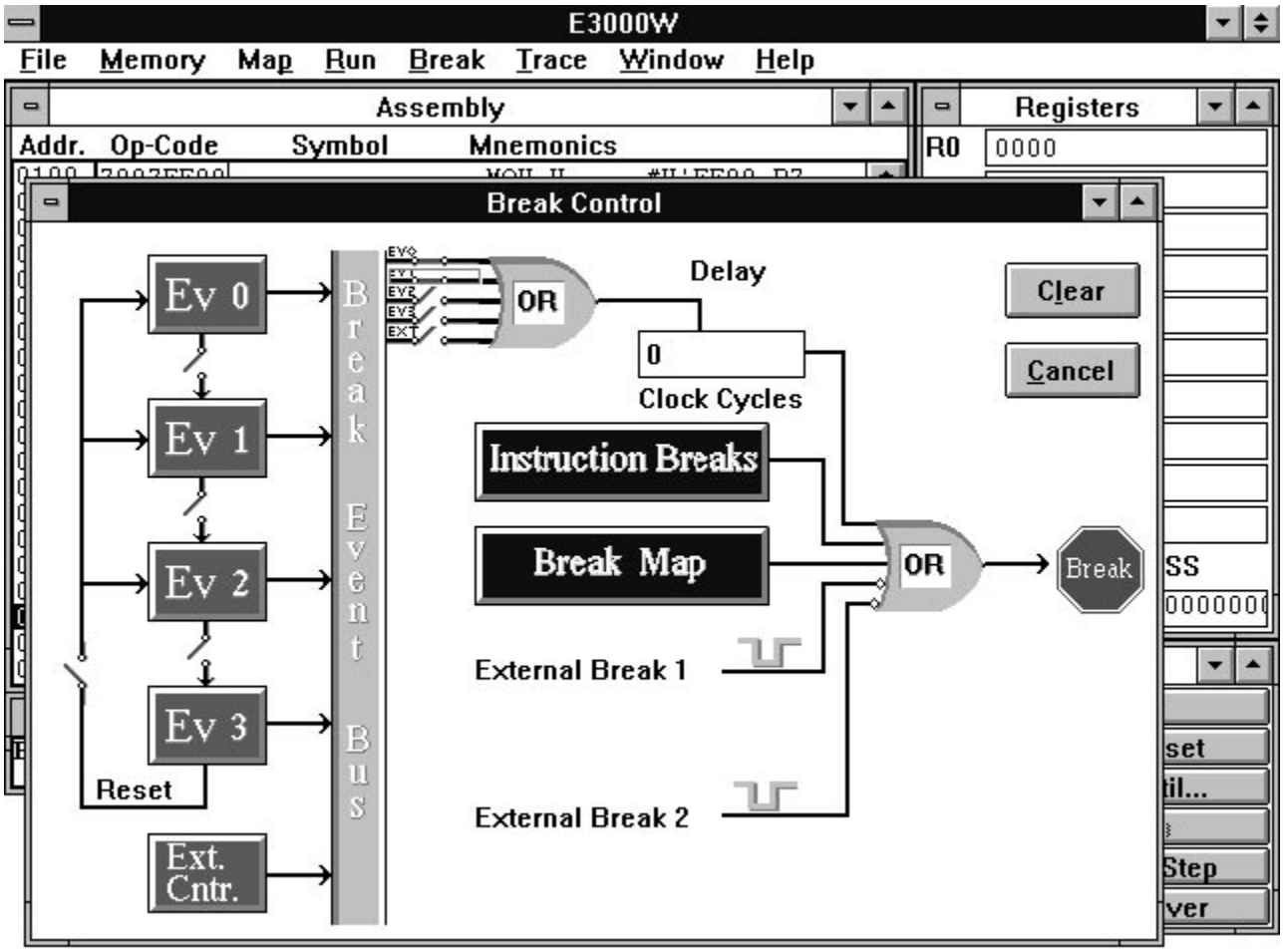
Next, a hardware breakpoint will be triggered after a break event has occured a number of times. Specifically, a break will occur after the address H'012A is present on the address bus 8 consecutive times. Open the *Break Control* window and click on the *EV1* box. Highlight the *Address* field entry, and type 012A. Next, highlight the contents of the *Event Count* field, type 8, and click on the *OK switch.*

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



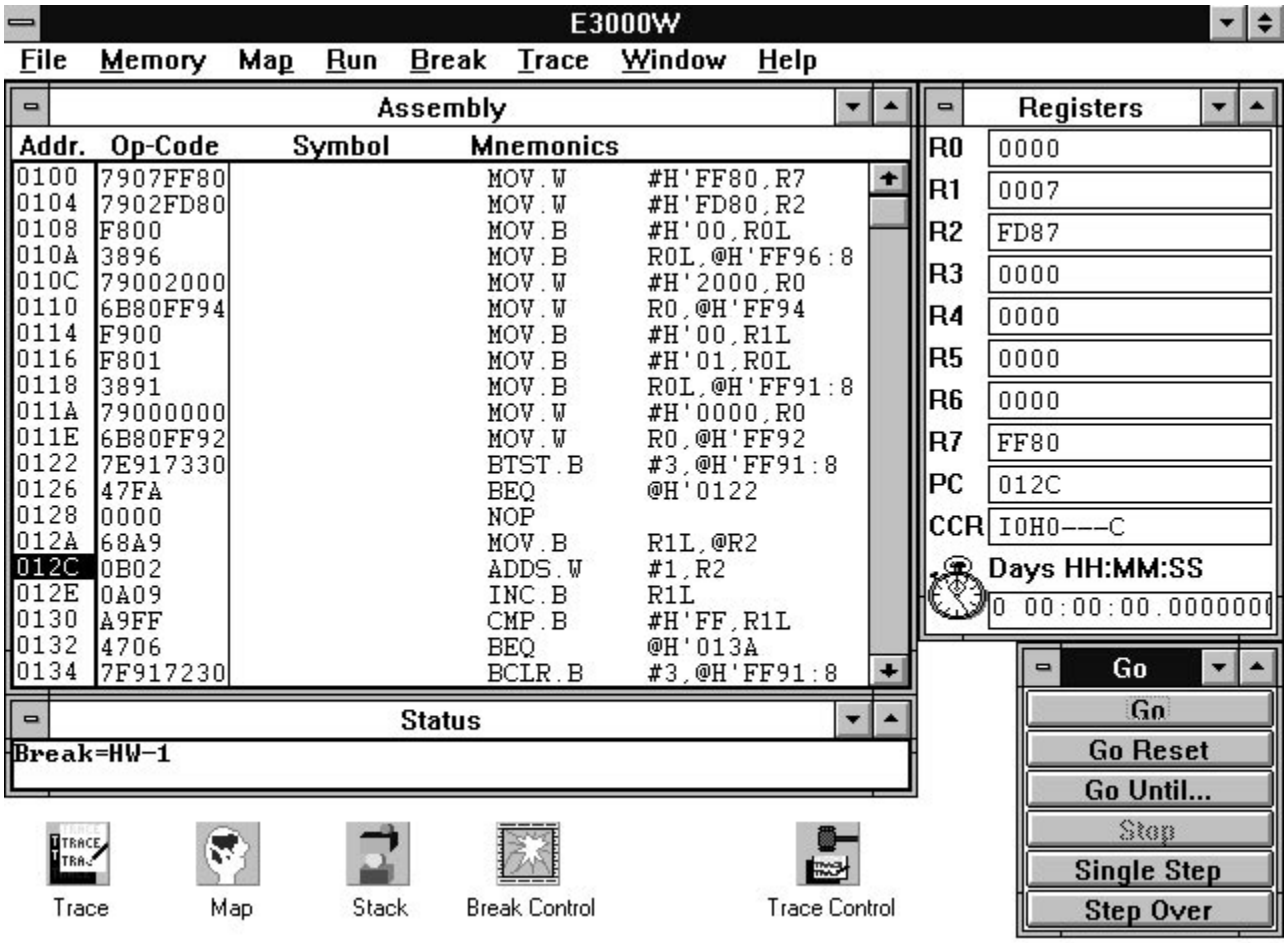
Next, close the *EV1* switch at the OR-gate input and close the *Break Control* window.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



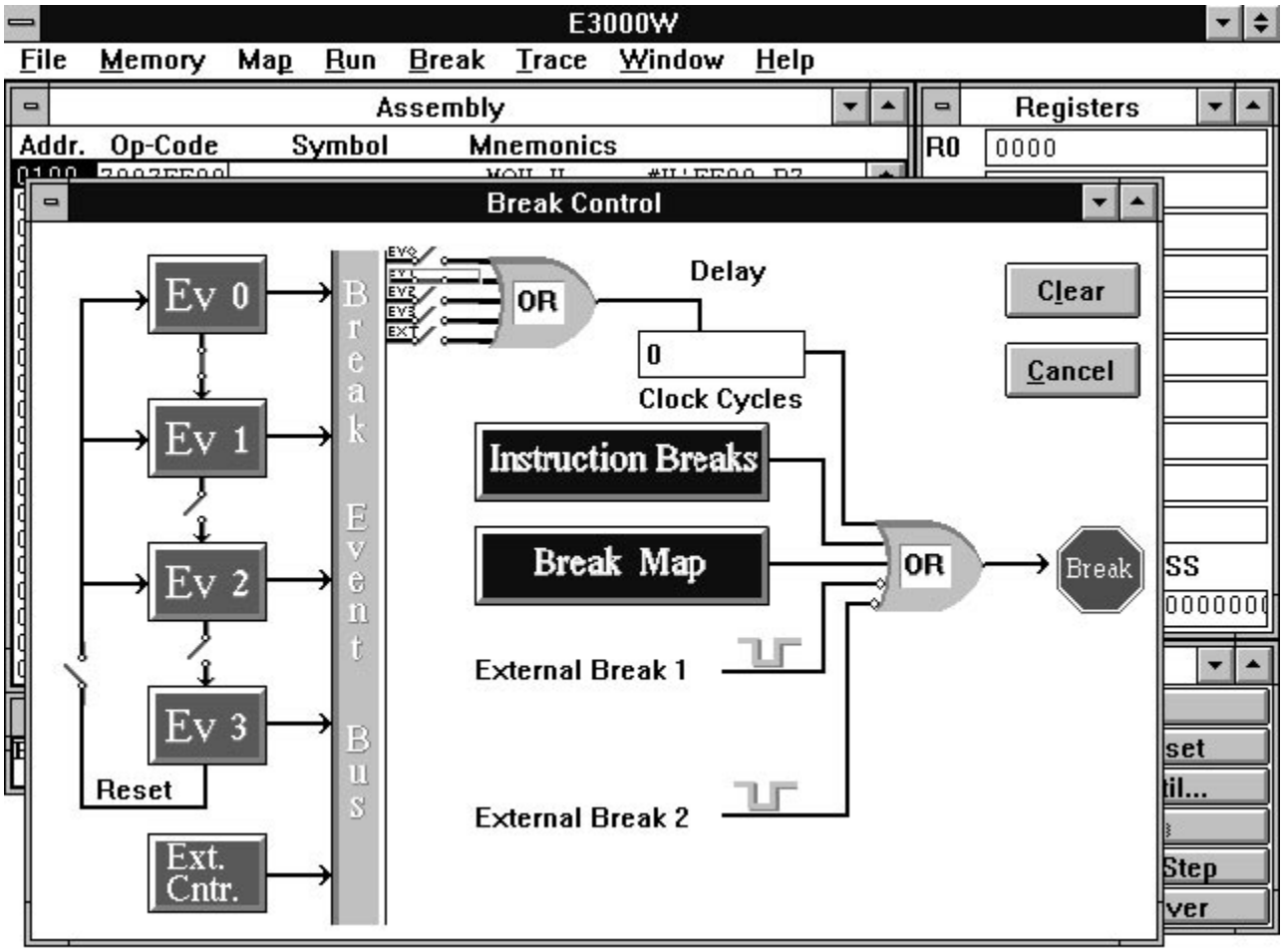
Load the PC with the program start address H'0100 and press the *Go* button. The program will break after the H'012A address has been on the address bus 8 times. Notice that the content of R1 is 07, and that the PC points at the next location past the breakpoint (H'012C).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



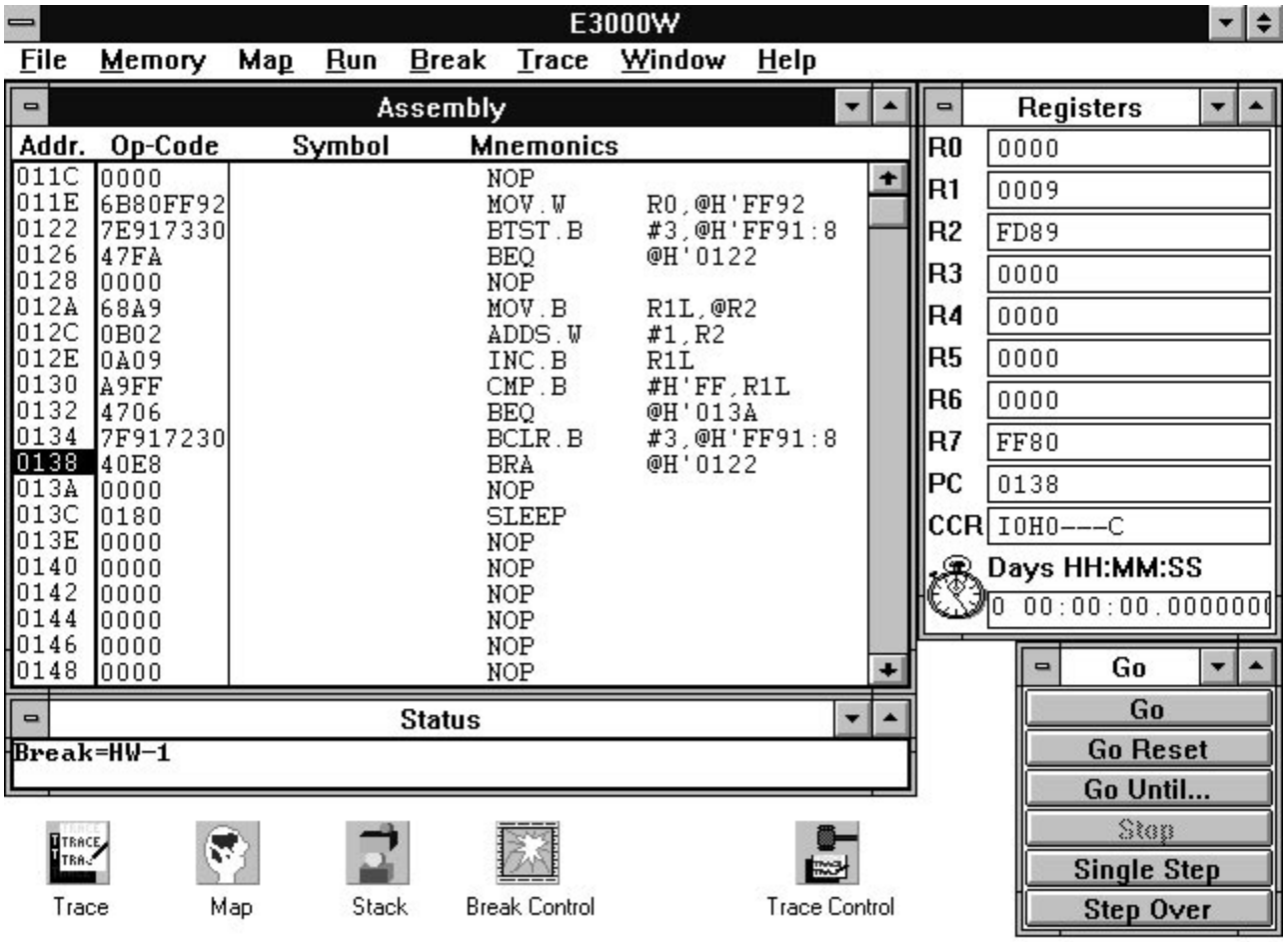
9. **Set a sequence break:**
Re-initialize the CPU and clear all registers in the *Registers* window. To set a sequence break, at least 2 break events must be specified and occuring in a certain order. In this example, we specify 2 break events. First, open the *Event 0* dialog box, enter address H'012a in the *Address* field, type 2 in the *Event Counter* field, and click on the *OK* switch. This means that the first break condition is satisfied when address H'012a is on the address bus 2 times. Second, open the *Event 1* dialog box, enter address H'0134 in the *Address* field, type 8 in the *Event Counter* field, and click on the *OK* switch. This means that the second break condition will occur after address H'0134 has been on the address bus 8 times. Next, close the switch connecting the *EV0* to the *EV1* box, and also close the *EV1* switch at the OR-gate input. This means that the emulator will stop program execution when both the *EV0* and *EV1* break conditions have occured in order.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



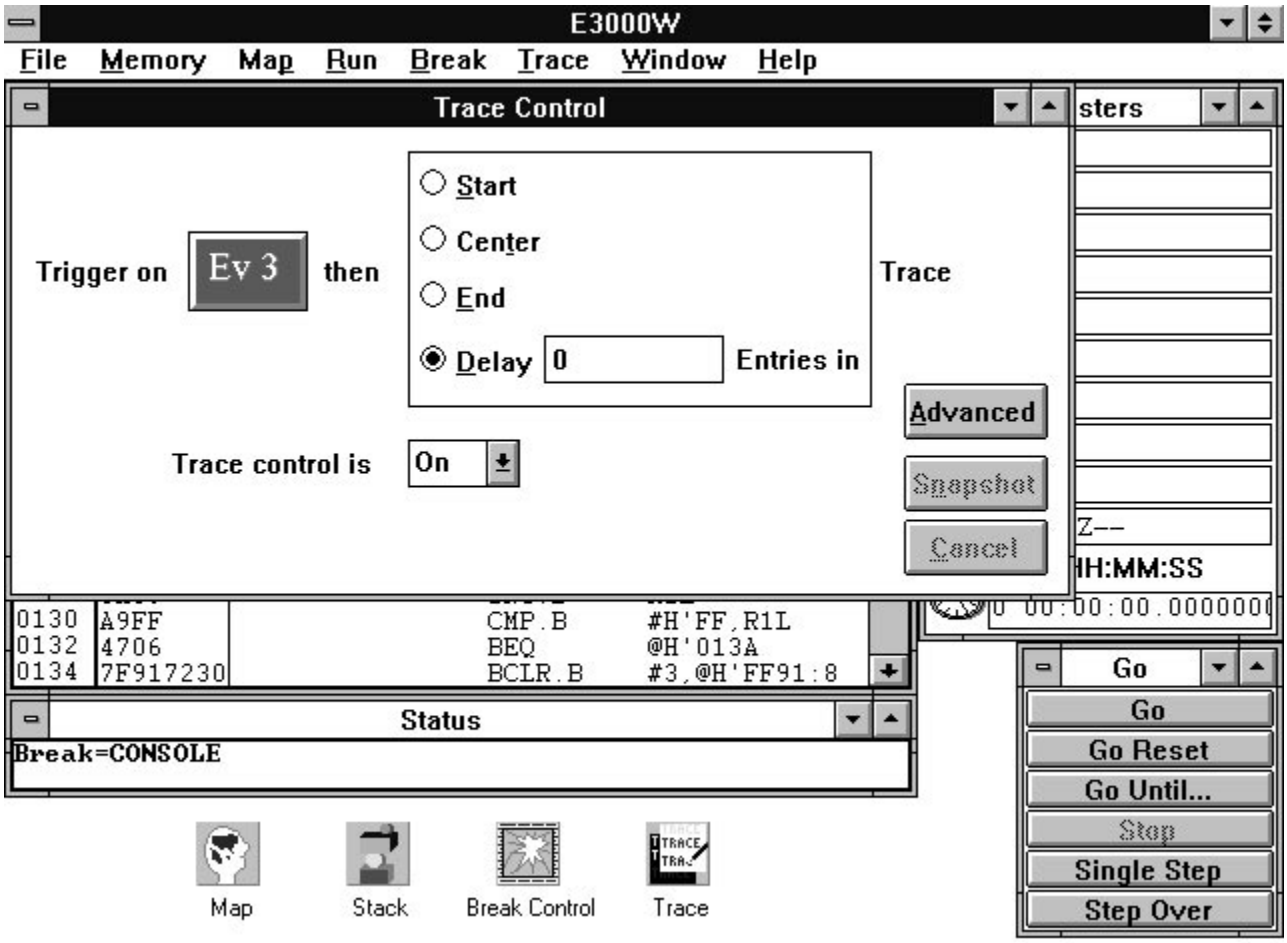
Close the *Break Control* window, load the PC with H'0100, and press the *Go* button. The program will stop after the *EV1* condition when both *EV0* and *EV1* are satisfied. Note that R1 contains the next data byte H'09, and the PC points at the next location past the last break address (H'0138).

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



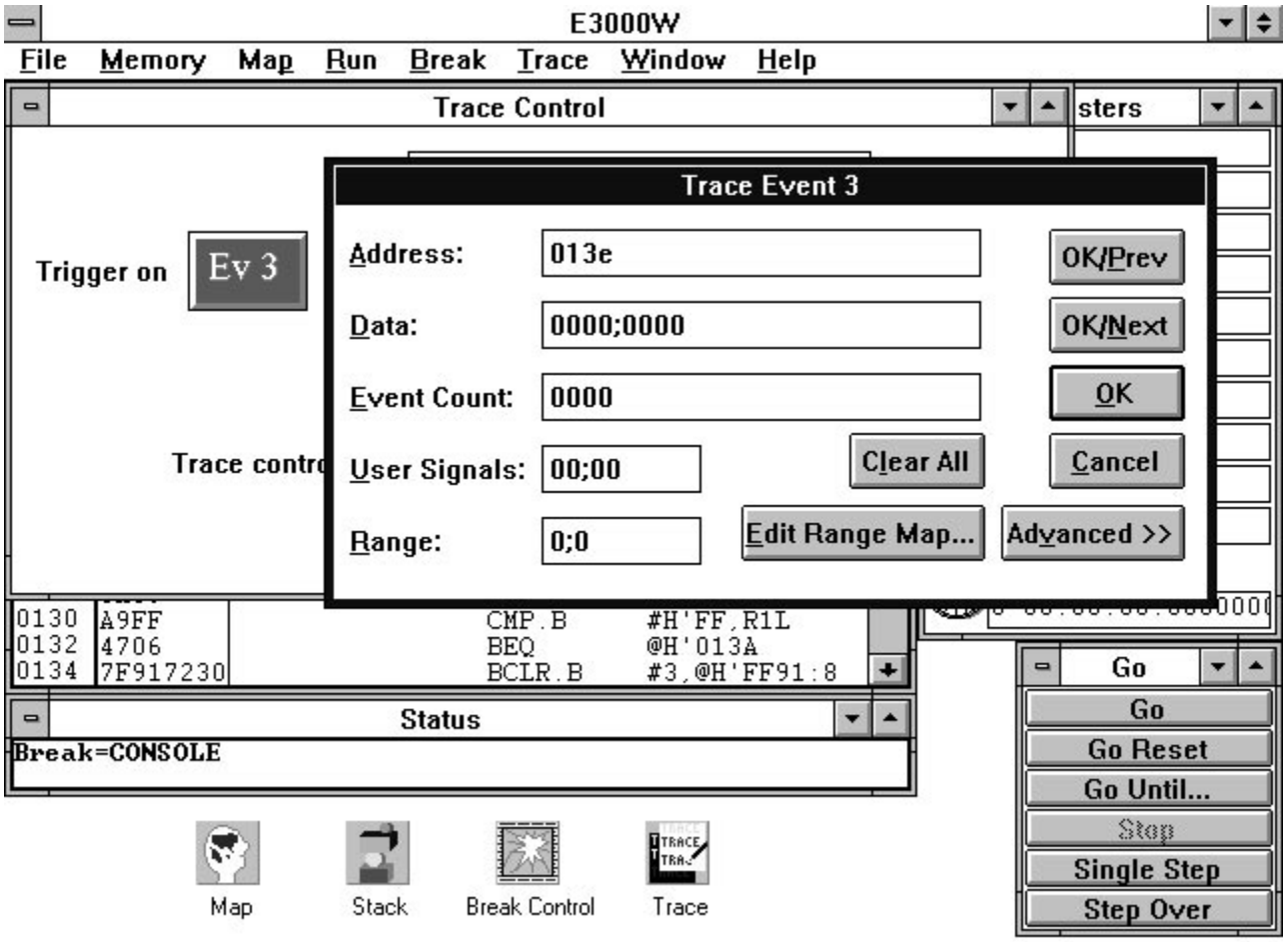
10. **Run a trace using the simple trace control:**
First, re-initialize the CPU and clear all breakpoints. Access the *Trace Control* window by double-clicking on the *Trace Control* icon. The simple trace control dialog box is displayed, which allows trace triggering only upon a condition(s) set on *EV3*. Enable the trace control by turining on the *Trace control is* switch.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



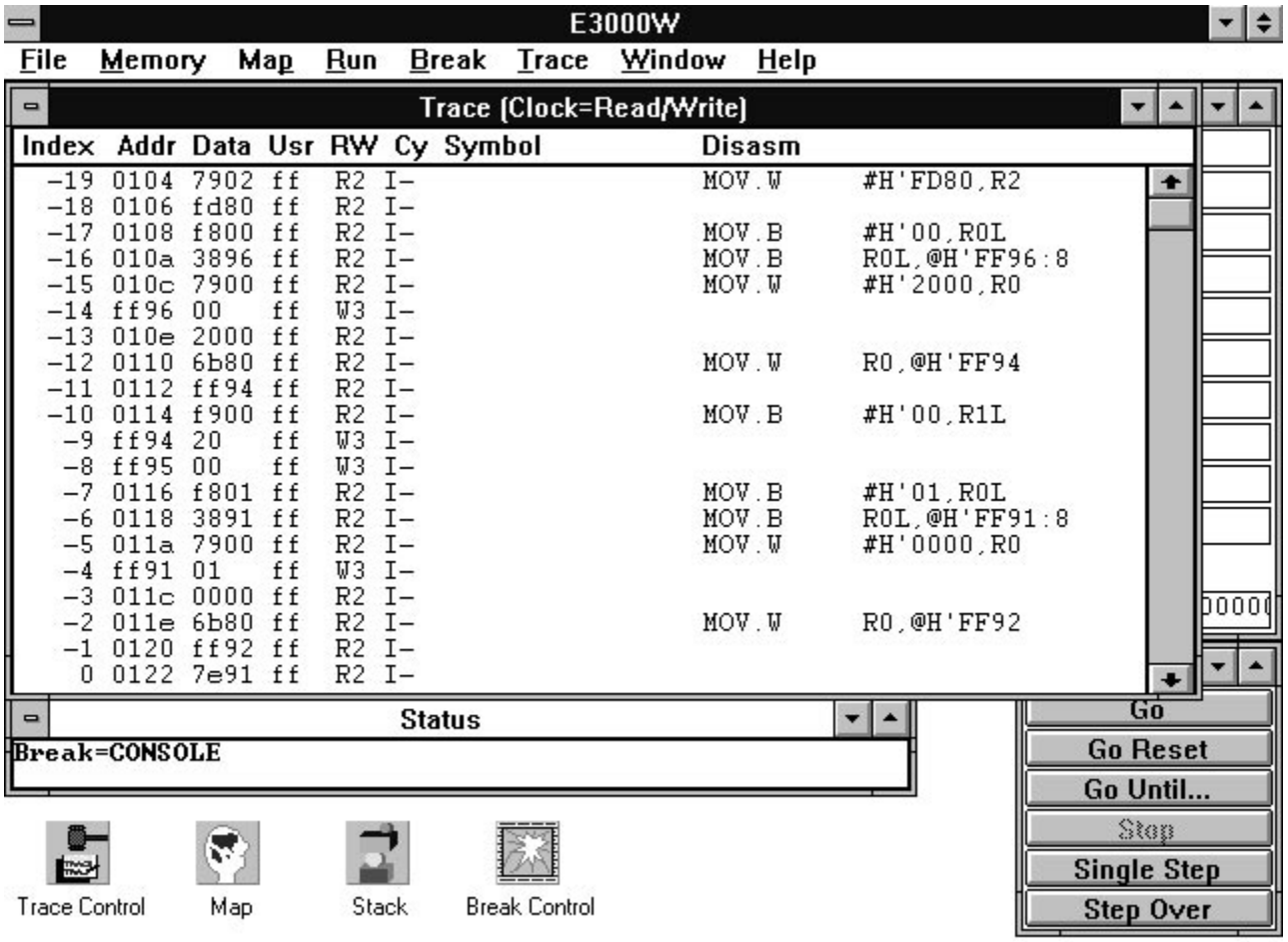
Next, click on the *EV3* box, and the *Trace Event 3* window will be displayed. Highlight the *Address* field, enter H'0122, and click on the *OK* button. This means that a trace should be recorded in the trace buffer starting from the beginning of the program up to address H'0122.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



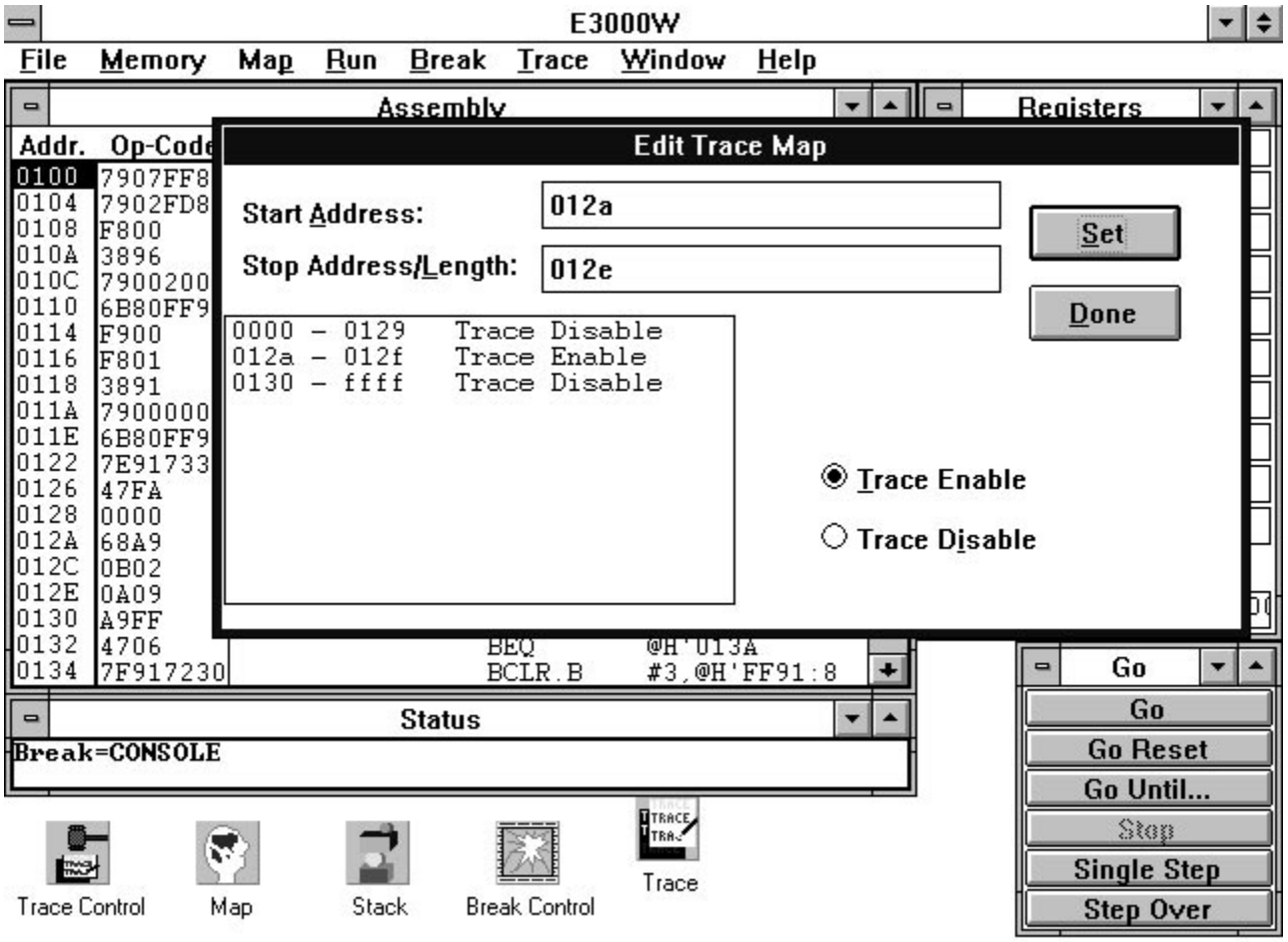
Close the *Trace Control* window, load the PC with the start address of the program (H'0100), and click on the *Go* switch. The program will run until it reaches the Sleep instruction. The message TRACE = COMPLETE will be displayed in the status box of the application window, implying that the trace condition is satisfied. Click on the *Trace* icon, and the trace will be displayed.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



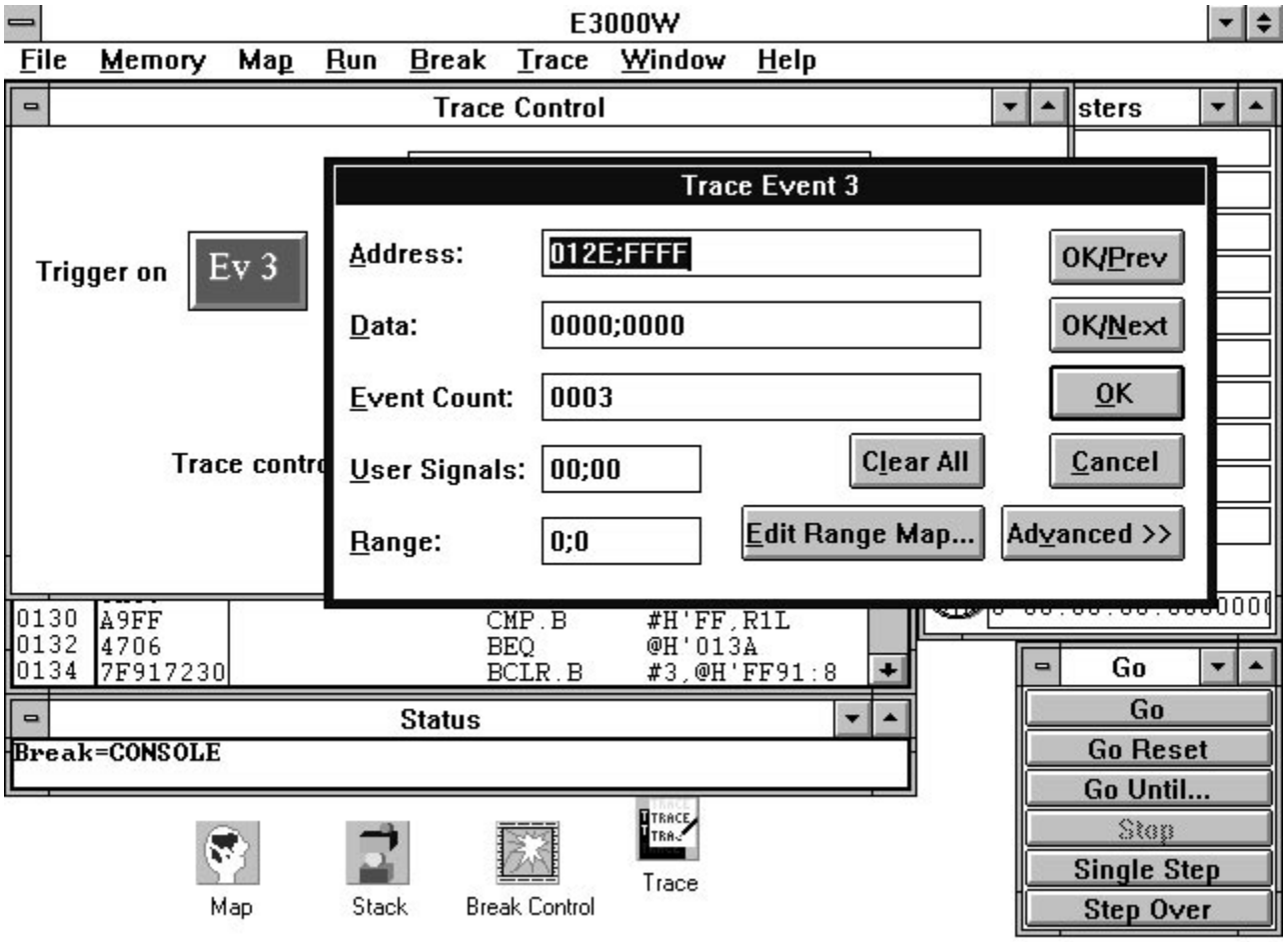
11. **Run a trace triggered on an event count:**
Re-initialize the CPU, clear all CPU registers, and load the PC with the start program address. Click on the *Edit Trace Map* option under the *Map* file menu. The default trace map allows tracing to occur throughout the entire memory range. Modify the trace map so that only the portion between H'012a - H'012e will be enabled, with the rest of the memory disabled for tracing.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



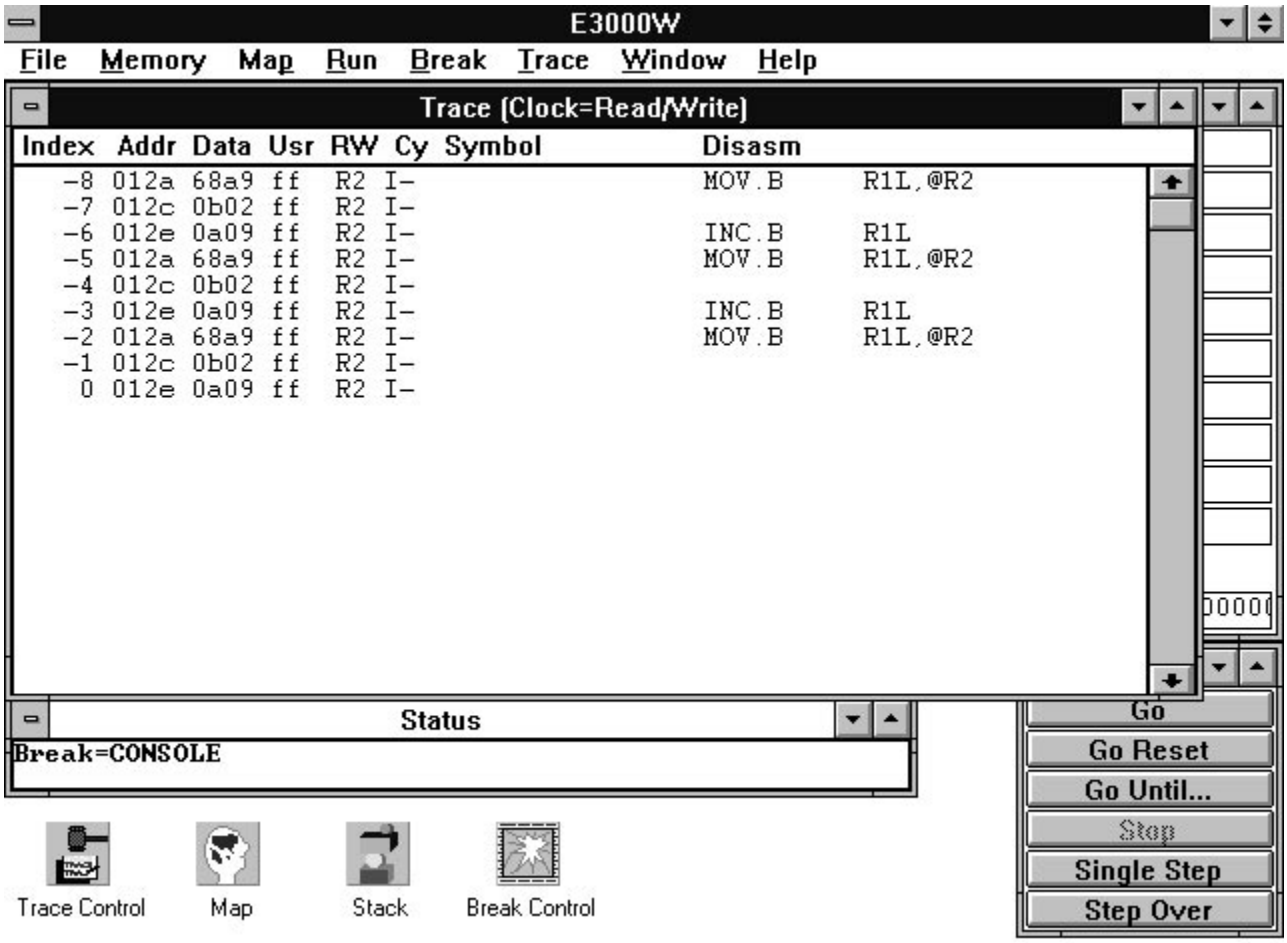
After clicking on the *Set* and *Done* switches, the *Edit Trace Map* dialog box will close. Access the *Trace Control* window, and click on the *EV3* box. In the *Trace event 3* dialog box, enter H'012e in the *Address* field, and 3 in the *Event count* field.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



This means that a trace should be collected only during instruction execution between memory locations H'012a - H'012e as the programs loops 3 times through these locations, and up to the end address H'012e. Click on the *OK* button, and close the *Trace Control* window. Run the program by pressing the *Go* switch, and display the trace buffer contents by clicking on the *Trace* icon.

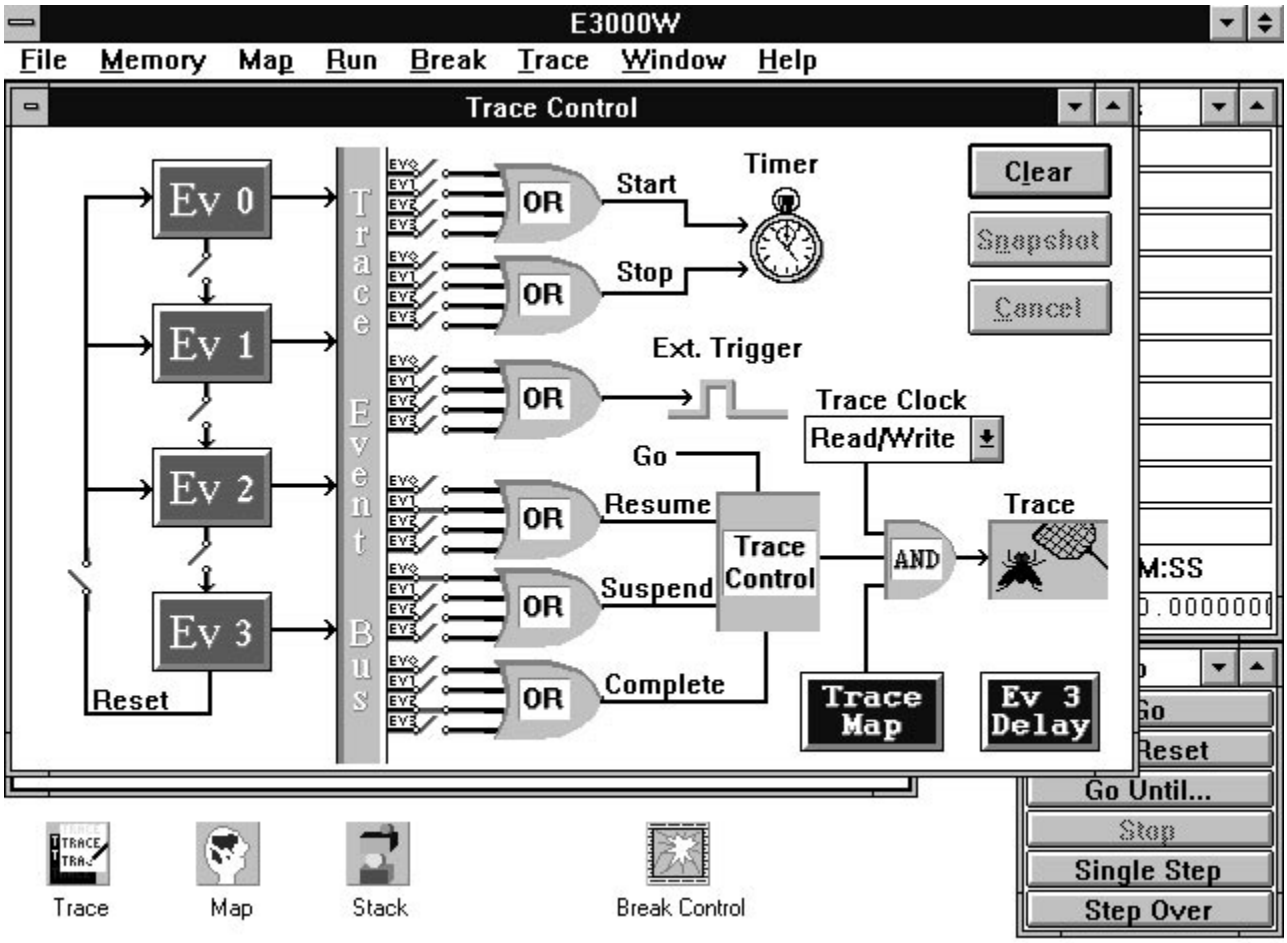
HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



12. **Run a trace using the advanced trace control**

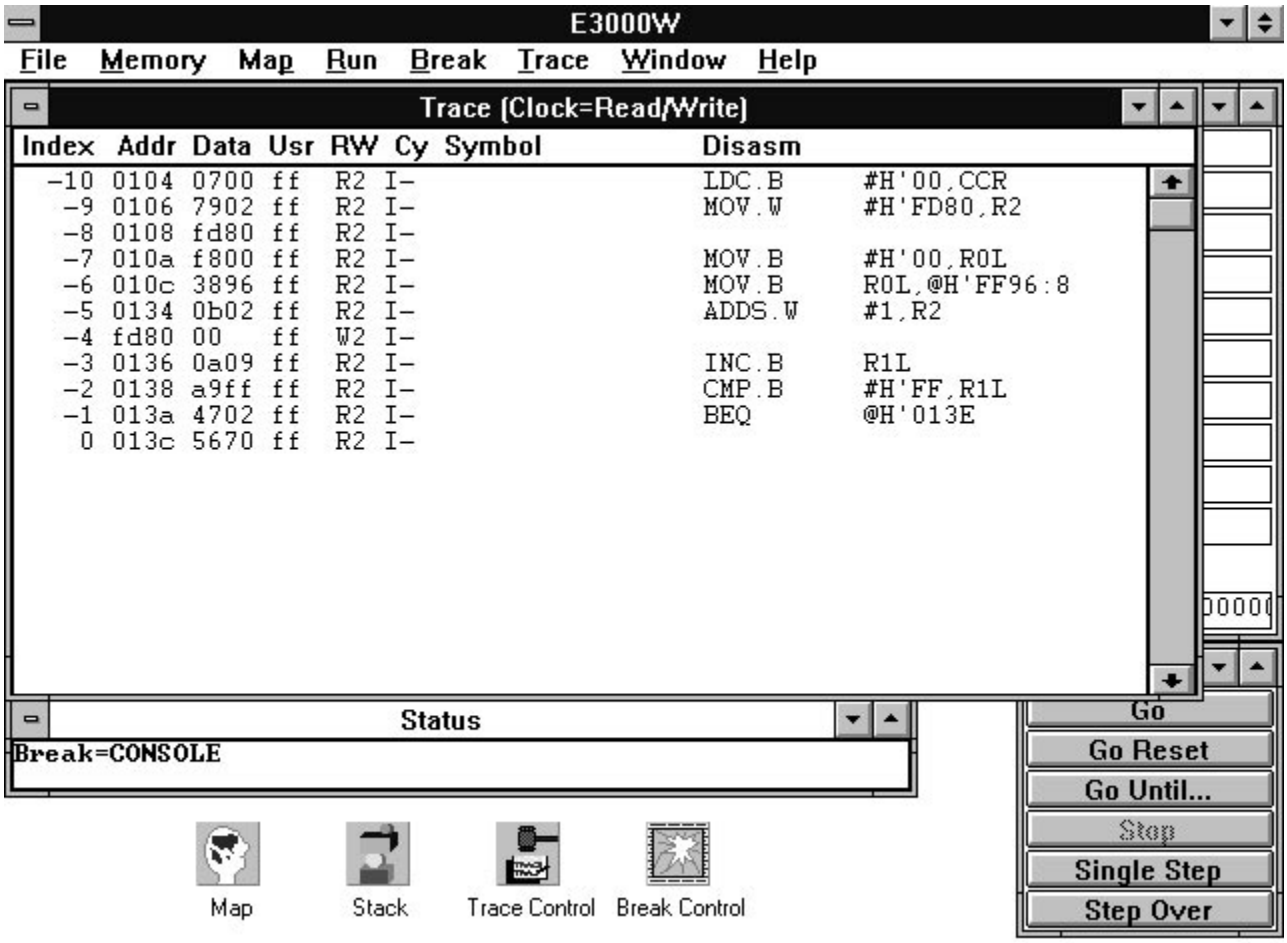
The second test program (tut_int.abs) will be used for these examples. First, however, clear the memory contents where the first program is loaded, and download the second program, as explained in the previous sections. Access the *Trace Control* window, click on the *Advanced* switch, and the *Advanced Trace Control* window will be displayed. A trace is to be captured starting from the beginning of the program, suspend the tracing at address H'010c, resume tracing at the start of the interrupt service routine H'0132, and complete the tracing upon the last instruction in the interrupt service routine at H'013c. Click on the first event box, *EV0*, and enter H'010c in the address field as previously explained. Click on the second event box, *EV1*, and enter H'0132 inn the address field. Similarly, click on *EV3* and enter H'013c in the address field. Go back to the *Advanced Trace Control* window, and close the EV1-labeled switch to the input of the *Resume* OR-gate, the EV0-labeled switch to the input of the *Suspend* OR-gate, and the EV2-labeled switch to the input of the *Complete* OR-gate.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



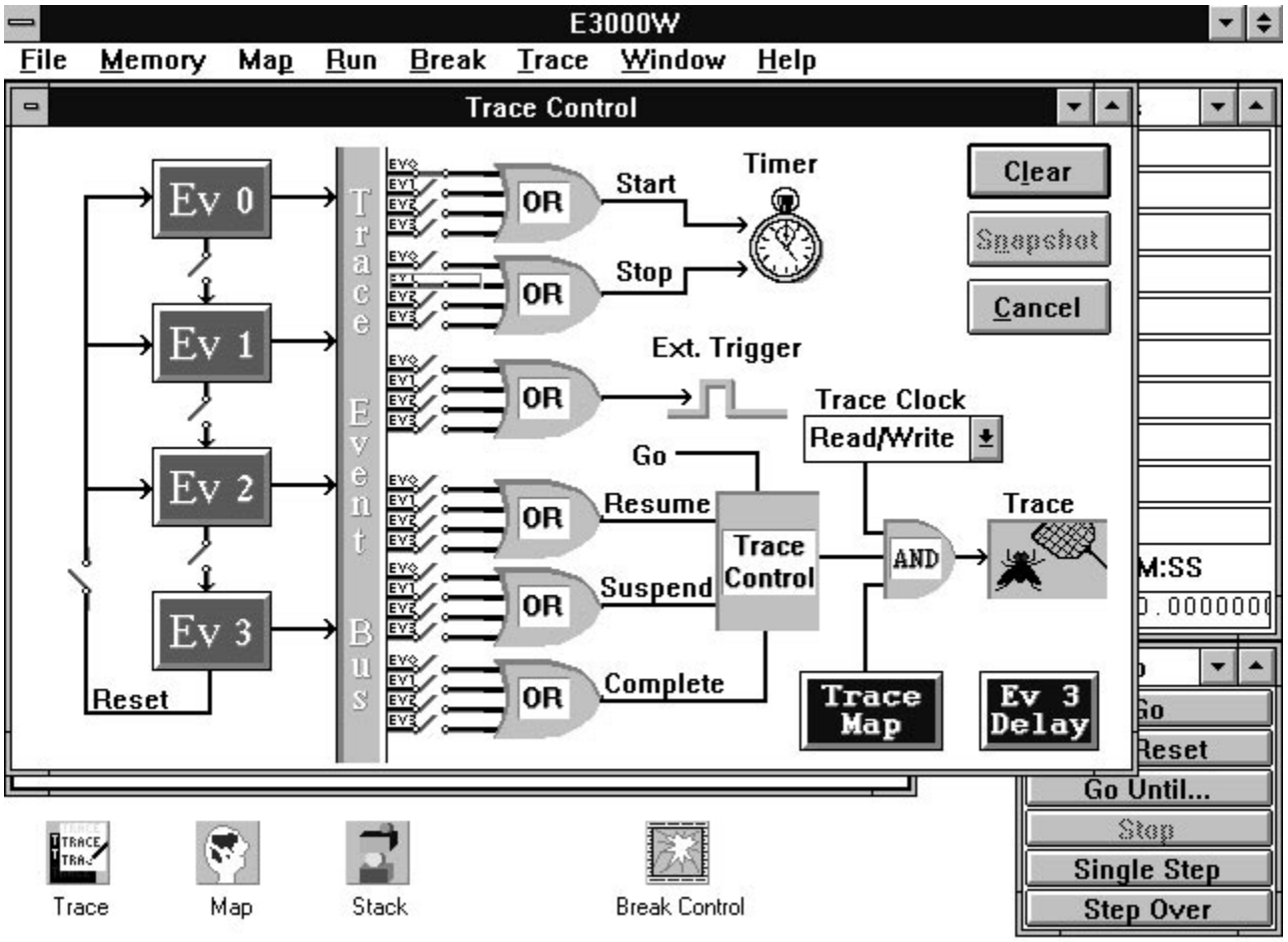
Close the *Trace Control* window, and run the program. To view the captured trace, double-click on the *Trace* icon.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



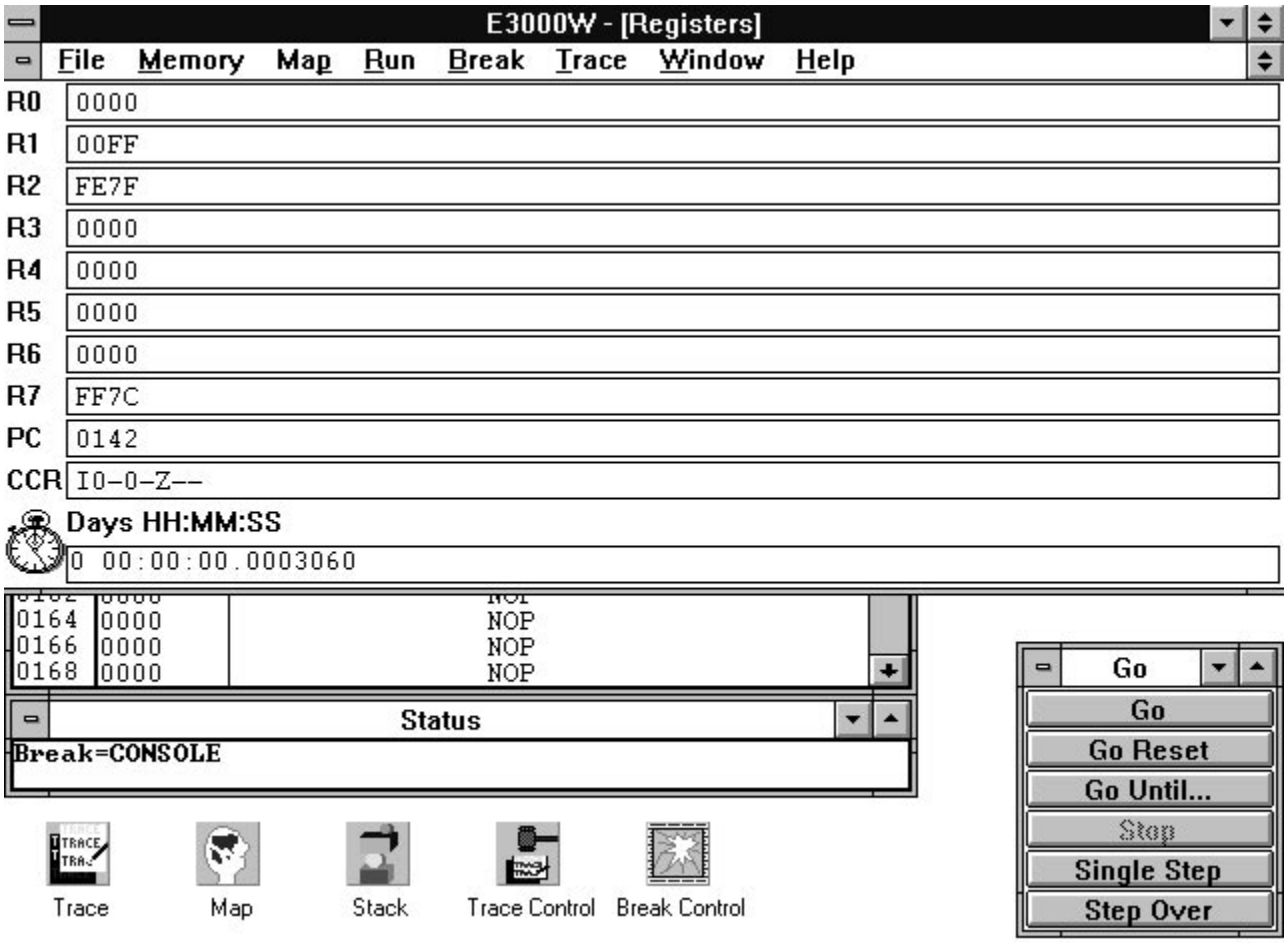
13. **Time 2 trace events with user clock frequency.**
First, disable the tracing conditions set above by clicking on the closed switches in the *Advanced Trace Control* window. Then, click on the *EV0* box, and enter H'0132 in the address field. Next, click on the *EV1* box and enter H'013c in the address field. Close the *Trace Event 1* dialog box, close the EV0-labeled switch to the input of the *Start* OR-gate, and close the EV1-labeled switch to the input of the *Stop* OR-gate.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



Run the program again by clicking on the *Go* switch. The internal clock time it took to execute the interrupt service routine portion of the program will be indicated in the *Clock* field on the bottom of the *Registers* dialog box in the main applications window.

HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300



HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

APPENDIX A - PROGRAM LISTINGS

Program 1:

```
frc_tier        .equ    h'ff90
frc_tcsr        .equ    h'ff91
frc_frc         .equ    h'ff92
frc_ocra        .equ    h'ff94
frc_tcr         .equ    h'ff96
frc_tocr        .equ    h'ff97

;       Initialize interrupt vector address

        .org    h'0020
        .data.w EXECUTE

;       Start program

        .org    h'100
        mov.w   #h'ff80,r7      ;initialize stackpointer
        mov.w   #h'fd80,r2      ;set memory buffer start pointer
        mov.b   #0,r0l
        mov.b   r0l,@frc_tcr    ;set clock /2 mode
        mov.w   #h'2000,r0
        mov.w   r0,@frc_ocra    ;set count
        mov.b   #0,r1l          ;set initial data
        mov.b   #h'01,r0l
        mov.b   r0l,@frc_tcsr   ;counter cleared at compare-match
        mov.w   #0,r0
        mov.w   r0,@frc_frc     ;reset free-running counter
WAIT_LOOP:
        btst.b  #3,@frc_tcsr    ;compare match?
        beq     WAIT_LOOP       ;if not, re-check
        nop
EXECUTE:
        mov.b   r1l,@r2         ;load data into buffer memory
        adds    #1,r2           ;point to next memory location
        inc.b   r1l
        cmp.b   #h'ff,r1l       ;last data?
        beq     END             ;if yes, end program
        bclr.b  #3,@frc_tcsr    ;if not, clear compare-match flag
        bra     WAIT_LOOP       ;do it again
END:    nop
        sleep
        .end
```

Program 2:

```
frc_tier        .equ    h'ff90
frc_tcsr        .equ    h'ff91
frc_frc         .equ    h'ff92
frc_ocra        .equ    h'ff94
frc_tcr         .equ    h'ff96
frc_tocr        .equ    h'ff97

;       Initialize interrupt vector address

        .org    h'0020
        .data.w EXECUTE

;       Start program

        .org    h'100
        mov.w   #h'ff80,r7      ;initialize stackpointer
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300

```
        ldc     #0,ccr          ;unmask all interrupts
        mov.w   #h'fd80,r2      ;set memory buffer start pointer
        mov.b   #0,r0l
        mov.b   r0l,@frc_tcr    ;set clock /2 mode
        mov.w   #h'2000,r0
        mov.w   r0,@frc_ocra    ;set count
        mov.b   #0,r1l          ;set initial data
        mov.b   #h'01,r0l
        mov.b   r0l,@frc_tcsr   ;counter cleared at compare-match
        bset.b  #3,@frc_tier    ;enable compare-match interrupt
        mov.w   #0,r0
        mov.w   r0,@frc_frc     ;reset free-running counter
WAIT_LOOP:
        nop
        bra     WAIT_LOOP       ;wait for compare-match interrupt
        nop
        sleep
        nop

;       Compare-match interrupt service routine

EXECUTE:
        mov.b   r1l,@r2         ;load data into buffer memory
        adds    #1,r2           ;point to next memory location
        inc.b   r1l
        cmp.b   #h'ff,r1l       ;last data?
        beq     END             ;if yes, end program
        rte
END:    nop
        sleep
        .end
```


HITACHI
Hitachi America, Ltd. • San Francisco Center • 2000 Sierra Point Parkway • Brisbane, CA 94005-1819 • (415) 589-8300