





ICE/FIRE Analyzer Programming Dialog

TRACE32 Online Help

TRACE32 Directory

TRACE32 Index

TRACE32 Documents	
FIRE In-Circuit Emulator	
ICE Analyzer System	
FIRE Analyzer Programming	
ICE/FIRE Analyzer Programming Dialog	1
Introduction	2
How to Start	2
Programming	3
Elements in the Dialog Box	3
Actions	4
Conditions	6
Levels	7
Load, Save and Clear	7
Definitions	9
Address/Range Definition	9
Data Definition	12
Counter Definition	15
Examples	18
Example 1: Selective Trace on a Single Address	18
Example 2: Selective Trace on a Symbol	19
Example 3: Selective Trace on an Address Range	21
Example 4: Selective Trace on an Address Range defined through a Symbol	22
Example 5: Selective Trace on Write Access to a Symbol	24
Example 6: Selective Trace on Value on Data Bus	26
Example 7: Trace all and mark with Label A beginning from 2 μ s	28
Example 8: Trace all, mark the Cycles executed in Function Sieve with Label B and mark the Entrance of Function Sieve with Label A	30
Example 9: Trace all, when Function Sieve is reached goto Level 1 and stop Sampling and Debugging after 5 Cycles	31

Introduction

How to Start

To start the Analyzer Programming Dialog select under the menu **Trace** the option **Trigger Dialog...**

An empty dialog box will appear. Please go on reading this manual to get an overview on the functionality of the dialog.

Elements in the Dialog Box

Remark

- The intension of the Analyzer Programming Dialog Window is to provide an intuitive and easy to use interface to program the Analyzer Trigger Unit.
- The Analyzer Programming Dialog Window does not represent the full functionality of the Analyzer Trigger Unit.

Overview

The screenshot shows the 'Analyzer Programming' dialog box with the following sections:

- Address definition:** Three 'Address' input fields with 'or' separators and dropdown arrows. The first field contains 'Var.RANGE(sieve)'.
- Data definition:** Two rows for 'DATA0' and 'DATA1'. Each row has a 'Value' input field and a 'Type' dropdown menu (set to 'B0').
- Count definition:** Two rows for 'COUNT0' and 'COUNT1'. Each row has a 'Value' input field and a 'Unit' dropdown menu.
- Condition Table:** A table with columns: Action, Address/Range, RD/WR..., Data, and Count. The first row is highlighted with a red box and contains: 'Sample' (Action), 'A-Range' (Address/Range), '&' (RD/WR...), '&' (Data), and 'Condition' (Count). Below this are sections for 'Level1' and 'Level2', each containing three rows of empty condition entries.
- Buttons:** CLEAR, LOAD, VIEW, SAVE, Program, Program & Save, Program & Save & Close, and About.
- Actual used file:** A text input field with a dropdown arrow.

Panel to control the programming

The screenshot shows the "Analyzer Programming" dialog box. The "Definition" section contains fields for A-Range, B-Range, and C-Range, each with three address input fields and a "V" button. Below this are fields for DATA0 and DATA1, each with a "Value" input, a "Type" dropdown (set to "B0"), and "COUNT0" and "COUNT1" fields with "Value" and "Unit" dropdowns.

The "Level0" section is a table with columns: Action, Address/Range, RD/WR..., Data, and Count. The "Action" column has a dropdown menu open, listing the following actions: Sample, Stop Analyzer, Stop CPU&Analyzer, Goto Level1, Goto Level2, Inc COUNT0, Inc COUNT1, Restart COUNT0, Restart COUNT1, Mark A, Mark B, and Mark C. The word "Actions" is written in large black text over the menu.

At the bottom of the dialog, there are buttons for CLEAR, LOAD, VIEW, SAVE, Program, Program & Save, Program & Save & Close, and About. An "Actual used file:" field with a "V" button is also present.

An Analyzer Trigger Program executes **ACTIONS** depending on the current situation of the user program/target hardware.

Sample	Samples only the specified CPU cycles (Selective Tracing)
Stop Analyzer Stop CPU&Analyzer	Stops tracing Stops CPU and tracing
Mark A Mark B Mark C	Put a marker with label A, B or C to the line in the trace output
Inc Count0 Inc Count1	Increment the counter identified by the label COUNT0 or COUNT1 .
Restart Count0 Restart Count1	Set the counter identified by the label COUNT0 or COUNT1 to zero.
Goto Level0 Goto Level1 Goto Level2	Go on tracing according to the actions defined under Level0 , Level1 or Level2 .

Conditions

One situation of the user program/target system is described by different characteristics. Different characteristics combined through **LOGICAL AND** form a definite condition. An action will only be carried out if the program/target system reaches the situation defined through the condition. The characteristics are:

Address/Range	Address or address range To be defined in the address definition fields A-Range , B-Range or C-Range
RD/WR	CPU cycle type READ , WRITE or ACCESS (READ or WRITE)
Data	CPU data To be defined in the data definition fields DATA0 or DATA1
Count	Counter or Time Counter To be defined in the count definition fields COUNT0 or COUNT1

If no characteristic is chosen the action is carried out at **every** CPU cycle.

Levels

Three different levels can be used to define a set of actions. The levels can be referenced by the logical names **Level0**, **Level1** and **Level2**. At any time only one of this sets of actions can be carried out. Use the action **Goto Level1** to let the system carry out only the actions under Level1 beginning from a definite condition.

Action	Address/Range	RD/WR...	Data	Count
Mark A	IF	&	&	&
Goto Level1	IF A-Range	&	&	&
	IF	&	&	&
	IF	&	&	&
- Level1				
Mark B	IF	&	&	&
Sample	IF	&	&	&
	IF	&	&	&

For this example all CPU cycles are collected and marked with label A in the output until the address A-Range is reached. From that time on all CPU cycles are collected and marked with label B in the output.

Load, Save and Clear

At the bottom of the dialog you find the buttons to control where your dialog will be stored.

CLEAR LOAD VIEW SAVE Program Program & Save Program & Save & Close
Actual used file: [text box] V About

Use the button **CLEAR** to reset the dialog window if you want to begin from the scratch.

With the button **LOAD** an analyzer program can be read into the dialog window.

	Only analyzer programs written with the Analyzer Dialog can be loaded into the dialog again.
--	--

Press the button **VIEW** and you will see the contents of the dialog written down in statements of the analyzer programming language. Use this button now and then to learn the basics of the analyzer programming language.

Program will write the contents of the dialog to the file which is specified under **Actual used file:** . After that the analyzer will be programmed. If no file name is given you will be asked to define one. The analyzer is successfully programmed if you get the message "Analyzer programmed successfully"

E::
Analyzer programmed successfully!
components trace Data Var List

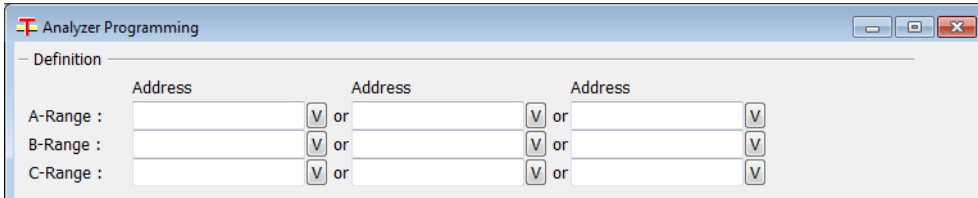
Otherwise a message box shows the error.

Special for C166

Before the analyzer is programmed you will be asked “For programming all current breakpoints will be deleted! Continue?”

Press **OK** if you want to go on programming the analyzer. But be aware that all breakpoints you set before will be deleted.

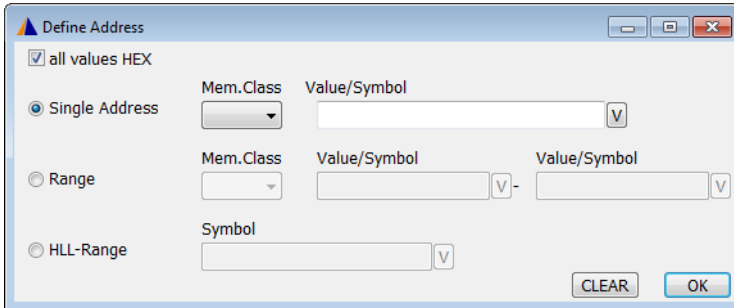
Address/Range Definition



Up to three different address areas can be defined in this part of the dialog. These areas are later referenced by the logical names **A-Range**, **B-Range** or **C-Range**. The logical name **AB-Range** defines an area declared through the area **A-Range** combined with the area **B-Range** by **LOGICAL AND**.

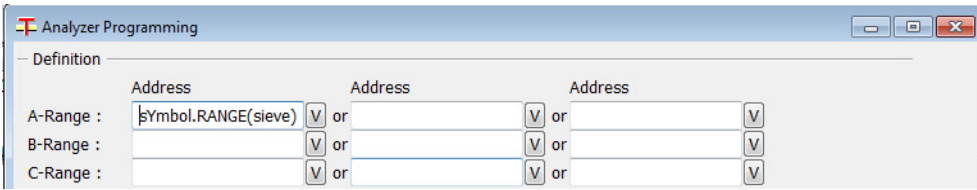
Each area e.g. **A-Range** allows to define the address space through three different singular address parts. One of these parts can hold a single address or an address range. All three singular address parts are combined through **LOGICAL OR** to form the characteristic named **A-Range**.

If you don't know how to specify an address or an address range just click the **V** button.

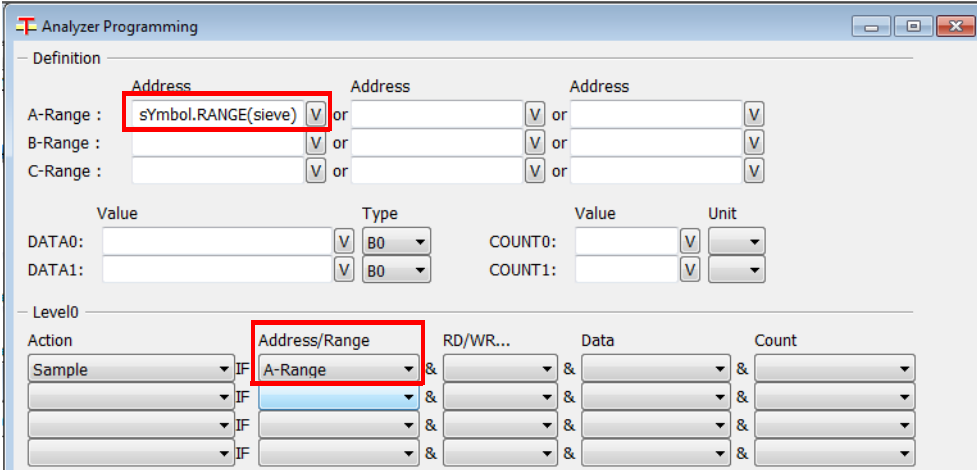


With the appearing sub-dialog a single address or any kind of range can be defined easily.

As you are familiar with the syntax of defining an address in TRACE32 just fill out the dialog.



The defined address area will be used to define the condition for carrying out an action in the main dialog.



If not address area is defined at the time you want to choose an address area name you will be asked to do so.

Analyzer Programming

- Definition

Address Address Address

A-Range : V or V or V

B-Range : V or V or V

C-Range : V or V or V

Value Type Value Unit

DATA0: V B0 ▾ COUNT0: V ▾

DATA1: V B0 ▾ COUNT1: V ▾

- Level0

Action	Address/Range	RD/WR...	Data	Count
Sample ▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
- Level1				
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
- Level2				
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&
▾	IF ▾	&	&	&

Actual used file:

CLEAR LOAD VIEW

Define Address

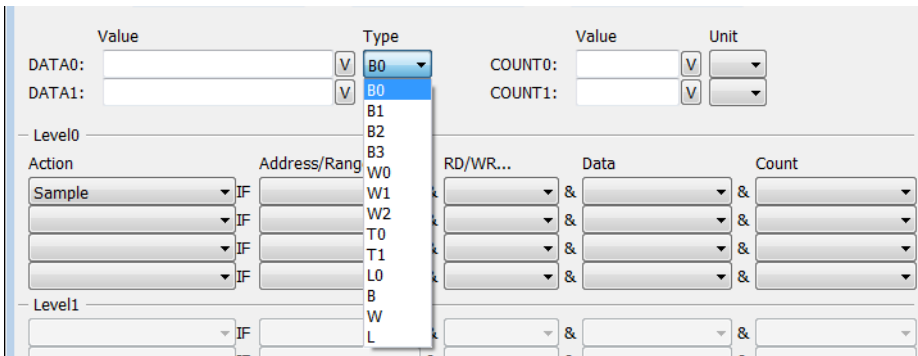
all values HEX

Single Address Mem.Class ▾ Value/Symbol V

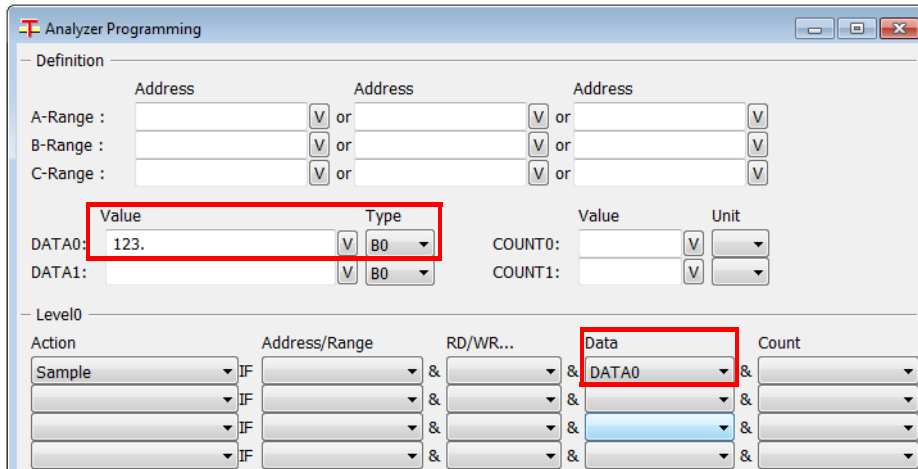
Range Mem.Class ▾ Value/Symbol V - V

HLL-Range Symbol V

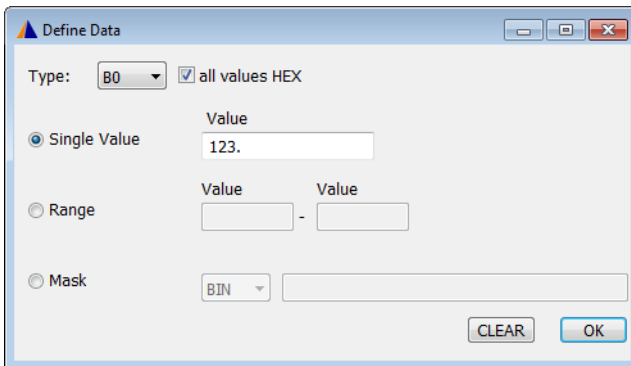
CLEAR OK



Two different data values can be used to observe the data on the data bus. The data values are later referenced by the labels **DATA0** and **DATA1**.



The data is defined through the value and the type. If you don't know how to specify a data value just click the **V** button.



With the appearing sub-dialog a single data, a data range or a mask can be defined easily.

With the type you specify the width of the value and the position on the databus. Depending on your CPU you have different types available.

Databus 8 bit

B	Byte size and whole databus
----------	-----------------------------

Databus 16 bit

B	Byte size and anywhere on the databus
B0	Byte size and at the lower byte of the data bus
B1	Byte size and at the upper byte of the data bus
W0	Word size and lower word on data bus
W	Word size and any position on data bus

Databus 32 bit

B	Byte size and anywhere on the databus
B0	Byte size and at the lower byte of the lower word on data bus
B1	Byte size and at the upper byte of the lower word on data bus
B2	Byte size and at the lower byte of the upper word on data bus
B3	Byte size and at the upper byte of the upper word on data bus
W0	Word size and lower word on data bus
W1	Word size and middle word on data bus
W2	Word size and upper word on data bus
W	Word size and any position on data bus
L0	Double word size and lower double word on data bus
L	Double word size and any position on data bus

For details on other data types see [Analyzer Programming Guide](#) chapter [Data Selectors](#).

If there is no data defined at the time you want to use a data name you will be asked to do so.

The image shows the 'Analyzer Programming' dialog box with the 'Define Data' sub-dialog open. The main dialog has sections for 'Definition', 'Level0', 'Level1', and 'Level2'. The 'Define Data' sub-dialog is currently active, showing options for 'Type' (set to B0), 'all values HEX' (checked), and three radio buttons for 'Single Value', 'Range', and 'Mask'. The 'Single Value' option is selected, and there is a text input field for the value. The 'Range' option has two input fields separated by a minus sign. The 'Mask' option has a 'BIN' dropdown and a text input field. Buttons for 'CLEAR' and 'OK' are at the bottom of the sub-dialog.

Counter Definition

Value	Type	Value	Unit
DATA0: <input type="text"/>	<input type="button" value="V"/> B0	COUNT0: <input type="text"/>	<input type="button" value="V"/> [dropdown]
DATA1: <input type="text"/>	<input type="button" value="V"/> B0	COUNT1: <input type="text"/>	<input type="button" value="V"/> [dropdown]

Level0: _____

Action	Address/Range	RD/WR...	Data	Count
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]

For controlling the flow two different counters can be defined. For each counter you can define the maximum value and if you want to define the counter as time counter the unit. The counters are referenced in the condition of an action by the labels **COUNT0** and **COUNT1**.

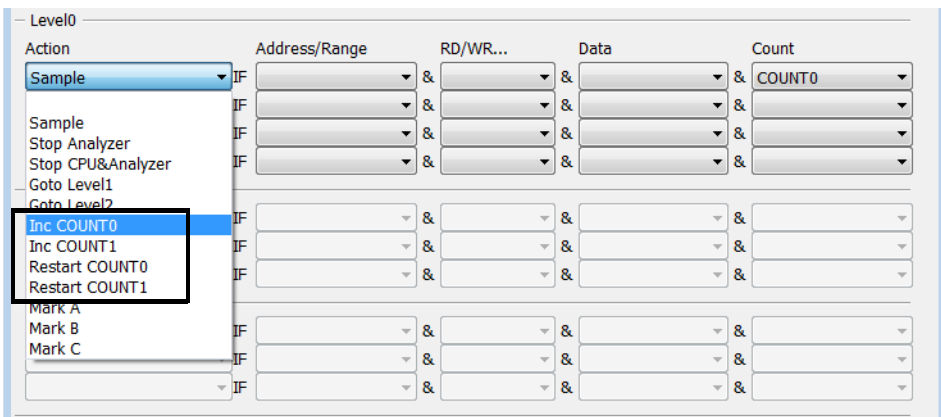
Value	Type	Value	Unit
DATA0: <input type="text"/>	<input type="button" value="V"/> B0	COUNT0: <input type="text" value="12"/>	<input type="button" value="V"/> [dropdown]
DATA1: <input type="text"/>	<input type="button" value="V"/> B0	COUNT1: <input type="text"/>	<input type="button" value="V"/> [dropdown]

Level0: _____

Action	Address/Range	RD/WR...	Data	Count
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> COUNT0
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]
<input type="button" value="Sample"/>	IF <input type="text"/>	& <input type="text"/>	& <input type="text"/>	<input type="button" value="V"/> [dropdown]

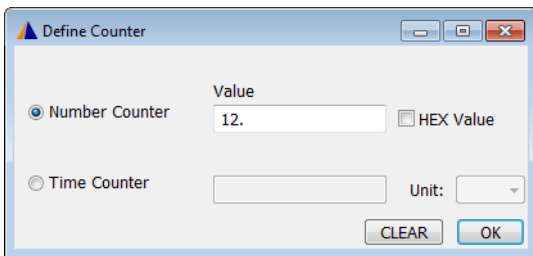
If you use the COUNT0/1 in the way showed above the sampling begins when the counter COUNT0 has reached the value of 12 ks. But as there are no actions defined to set the counter COUNT0 this will never happen.

To control the value of the counters the actions **Inc Count0/1** and **Restart Count0/1** are available.



Inc COUNT0/1 adds 1 to the value of the counter COUNT0/1. **Restart COUNT0/1** sets the value of the counter COUNT0/1 to zero.

If you don't know how to define the value of a counter just click the **V** button.



As you are familiar with the syntax of defining counters in TRACE32 just fill out the dialog.

If there is no counter defined at the time you want to use a counter name in a condition or an action you will be asked to define it.

Analyzer Programming

Definition

A-Range : Address or Address or Address

B-Range : Address or Address or Address

C-Range : Address or Address or Address

Value Type Value Unit

DATA0: B0 COUNT0: 12.

DATA1: B0 COUNT1:

Level0

Action	IF	Address/Range	&	RD/WR...	&	Data	&	Count
Inc COUNT1	IF	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>	&	COUNT0
<input type="text"/>	IF	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>
<input type="text"/>	IF	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>
<input type="text"/>	IF	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>	&	<input type="text"/>

Level1

IF

IF

IF

Level2

IF

IF

IF

Actual used file:

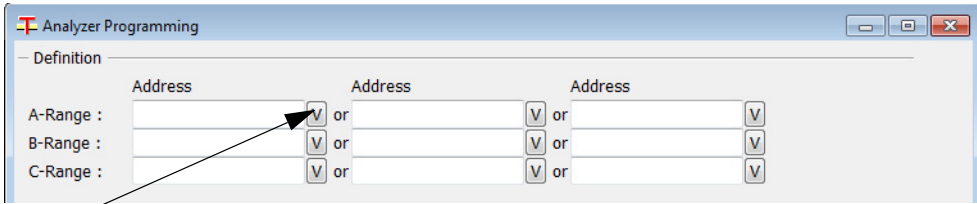
Define Counter

Number Counter Value: HEX Value

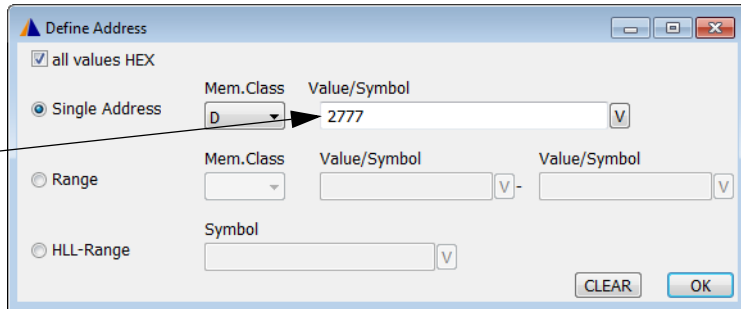
Time Counter Value: Unit:

Examples

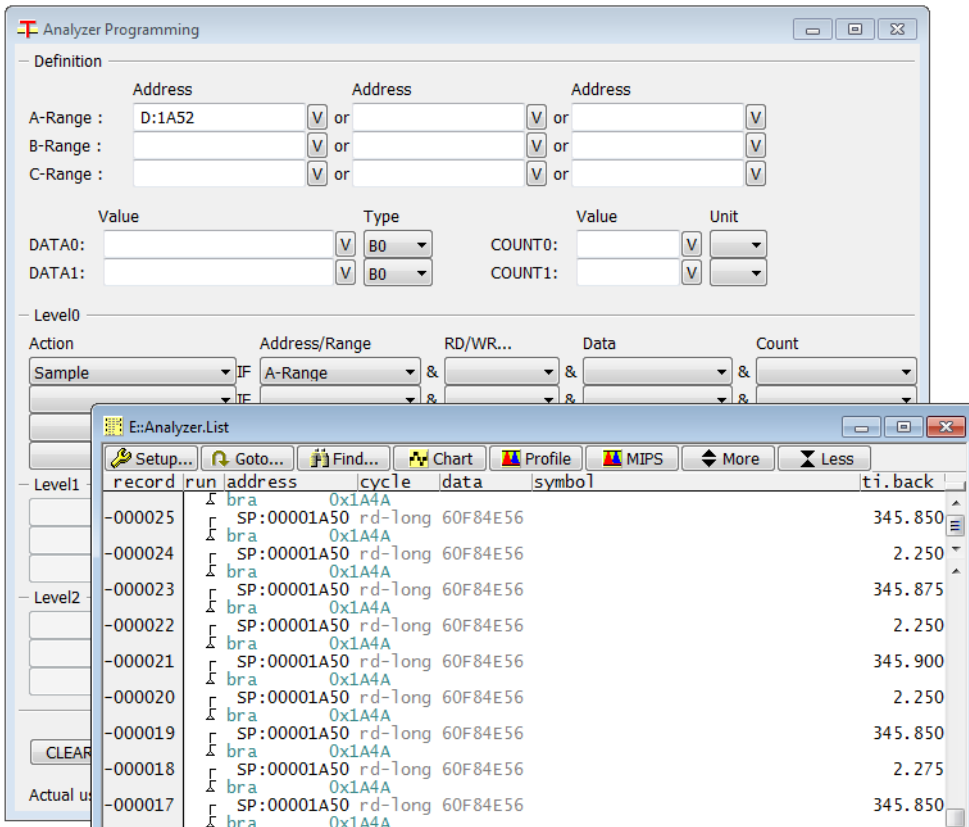
Example 1: Selective Trace on a Single Address



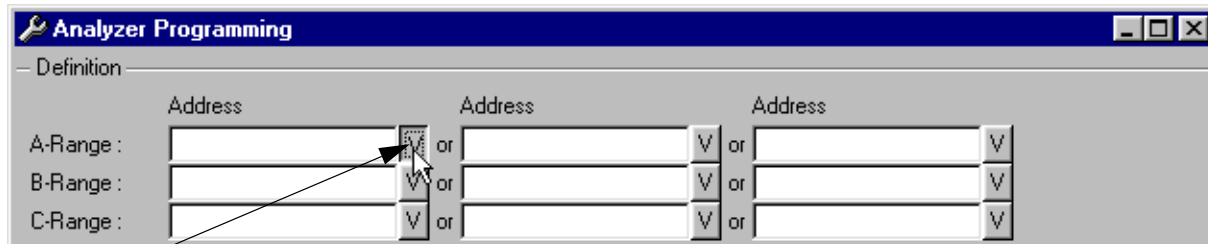
Click here to open the Define Address Window



Enter the address and confirm with OK.

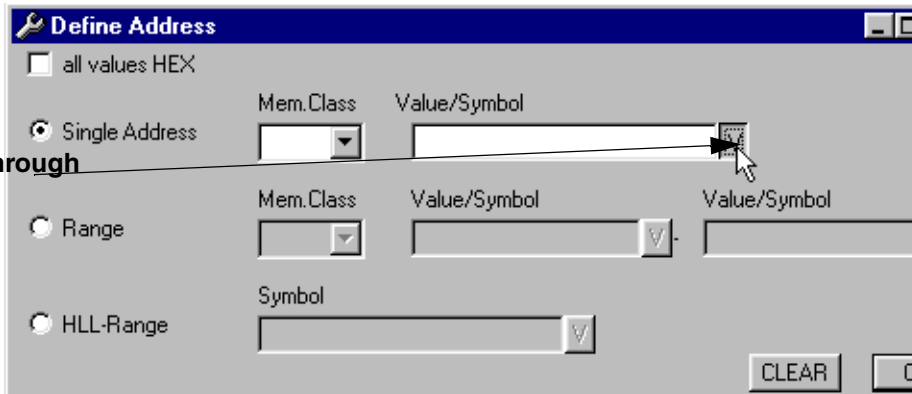


Example 2: Selective Trace on a Symbol

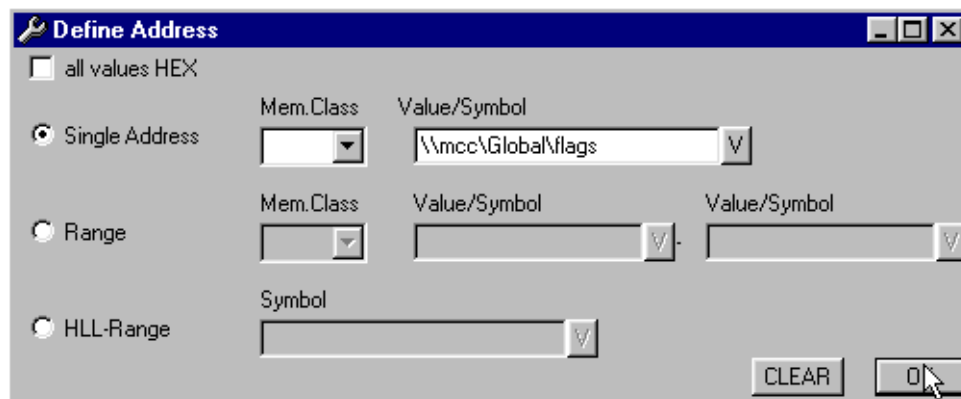
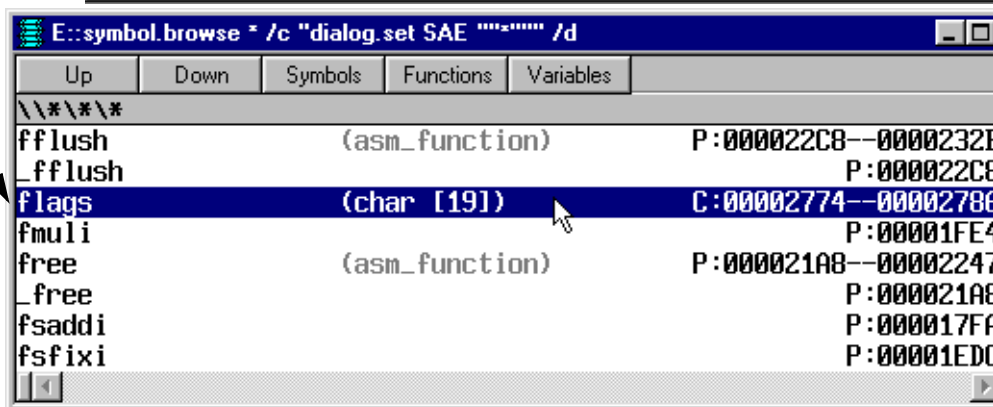


Click here to open the Define Address Window

Click here to browse through the symbol data base



Select the symbol by a double click



Analyzer Programming [] [] [X]

— Definition —

Address Address Address

A-Range : \mcc\Global\flags [V] or [] [V] or [] [V]

B-Range : [] [V] or [] [V] or [] [V]

C-Range : [] [V] or [] [V] or [] [V]

Value Type Value Unit

DATA0: [] [V] B0 [] COUNT0: [] [V] [] []

DATA1: [] [V] B0 [] COUNT1: [] [V] [] []

— Level0 —

Action Address/Range RD/WR... Data Count

Sample [] IF A-Range [] & [] [] & [] [] & [] []

[] IF [] & [] [] & [] [] & [] []

[] IF [] & [] [] & [] [] & [] []

E::Analyzer.List

Goto...	Find...	Set Ref	Set Zero	View	More	Less	
record	run	address	cycle	d.w	symbol		ti.back
000009	f r	SD:00002774	rd-byte	01	\\mcc\Global\flags		86.1:
000008	f r	SD:00002774	wr-byte	01	\\mcc\Global\flags		201.2:
000007	f r	SD:00002774	rd-byte	01	\\mcc\Global\flags		86.1:
000006	f r	SD:00002774	wr-byte	01	\\mcc\Global\flags		201.2:
000005	f r	SD:00002774	rd-byte	01	\\mcc\Global\flags		86.1:
000004	f r	SD:00002774	wr-byte	01	\\mcc\Global\flags		201.2:

Example 3: Selective Trace on an Address Range

Click here to open the Define Address Window

1. Click here to select the radio button for Range

2. Click here to choose a memory class out of the list of available classes

3. Type in the base address of the range

4. Type in the end address of the range

5. Press OK button

Definition

A-Range: P:1056-105F

Value Type

DATA0: [] B0

DATA1: [] B0

COUNT0: [] []

COUNT1: [] []

Action	Address/Range	RD/WR...	Data	Count
Sample	IF A-Range	&	&	&
	IF	&	&	&
	IF	&	&	&

E::Analyzer.List ALL

record	run	address	cycle	t	d.w	symbol
000004	f	SD:00001058	wr-word	2F2F		\\mcc\mcc\func1
000003	f	SD:0000105A	wr-word	2F2F		\\mcc\mcc\func1+2
000002	f	SD:0000105C	wr-word	2F2F		\\mcc\mcc\func1+4
000001	f	SD:0000105E	wr-word	2F2F		\\mcc\mcc\func1+6

Example 4: Selective Trace on an Address Range defined through a Symbol

The image illustrates the steps to define an HLL-Range in the Analyzer Programming Dialog:

- Analyzer Programming Dialog:** Shows the "Definition" section with three rows for "A-Range", "B-Range", and "C-Range". Each row has three "Address" input fields. An arrow points to the "V" button in the first "Address" field of the "A-Range" row, with the text: "Click here to open the Define Address Window".
- Define Address Dialog:** Shows the "HLL-Range" radio button selected. An arrow points to it with the text: "Click here to select the radio button for HLL-Range". Below it, the "Symbol" field contains a dropdown menu. An arrow points to this dropdown with the text: "Select the symbol by a double click". To the right, the text "Press the V button" has an arrow pointing to the "V" button next to the Symbol field.
- Symbol Browser:** A window titled "E::symbol.browse * /c "dialog.set SE ***** /d" showing a list of symbols. The "func1" entry is highlighted in blue. An arrow points to this entry with the text: "Select the symbol by a double click".
- Define Address Dialog (Final):** Shows the "Symbol" field now containing the path "\\mcc\mcc\func1". The "HLL-Range" radio button remains selected. Buttons for "CLEAR" and "OK" are visible at the bottom.

Definition

A-Range: or or

B-Range: or or

C-Range: or or

Value Type Value Unit

DATA0: B0 COUNT0:

DATA1: B0 COUNT1:

Level0

Action Address/Range RD/WR... Data Count

Sample IF A-Range & & & &

IF & & & &

E::Analyzer.List ALL

record	run	address	cycle	t	d.w	symbol	padc
000004	f	SD:00001058	wr-word	2F2F		\\mcc\mcc\func1	
000003	f	SD:0000105A	wr-word	2F2F		\\mcc\mcc\func1+2	
000002	f	SD:0000105C	wr-word	2F2F		\\mcc\mcc\func1+4	
000001	f	SD:0000105E	wr-word	2F2F		\\mcc\mcc\func1+6	

Definition

Address Address Address

A-Range : \cosz80\cosz80\flags V or V or V

B-Range : V or V or V

C-Range : V or V or V

Value Type Value Unit

DATA0: V B COUNT0: V

DATA1: V B COUNT1: V

Level0

Action Address/Range RD/WR... Data Count

Sample IF A-Range & WRITE & &

IF & & &

IF & & &

Level1

E::Analyzer.List

Setup... Goto... Find... Set Ref Set Zero View More Less

record	run	address	cycle	.b	symbol
000011	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000010	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000009	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000008	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000007	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000006	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000005	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000004	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000003	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000002	f r	D:00E032	wr-mem	01	\cosz80\cosz80\flags
000001	BRK				

Example 6: Selective Trace on Value on Data Bus

Analyzer Programming

Definition

A-Range : [] V or [] V or [] V
B-Range : [] V or [] V or [] V
C-Range : [] V or [] V or [] V

Value Type Value Unit
DATA0: [] V B0 [] V []
DATA1: [] V B0 [] V []
COUNT0: [] V []
COUNT1: [] V []

Click here to open the Define Data Window

Click here to select the width of the value

Click here to select a single value

Define Data

Type: W [] all values HEX

Single Value
 Range
 Mask

Value: 5282

Value Value: [] - []

BIN []

CLEAR OK

Press OK

Definition

A-Range : V or V or V

B-Range : V or V or V

C-Range : V or V or V

Value Type Value Unit

DATA0: W COUNT0:

DATA1: B0 COUNT1:

Level0

Action Address/Range RD/WR... Data Count

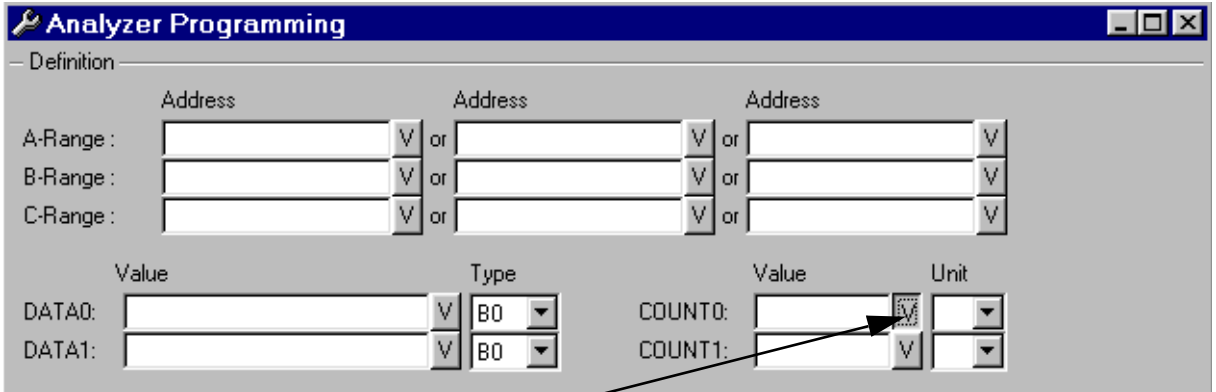
Sample IF & & DATA0 &

IF & & &

E::Analyzer.List ALL

record	run	address	cycle	t	d.w	symbol
000011	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000010	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000009	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000008	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000007	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000006	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000005	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000004	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000003	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000002	f r	SP:000017CC	rd-word	5282		\\mcc\mcc\sieve+44
000001	BRK					

Example 7: Trace all and mark with Label A beginning from 2 μ s



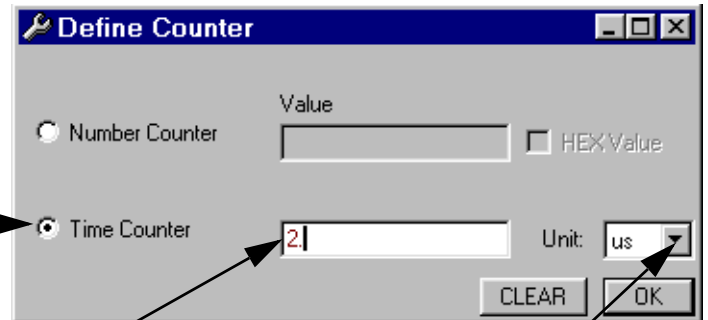
The **Analyzer Programming** dialog box is shown with the **Definition** tab selected. It contains three columns of **Address** input fields for **A-Range**, **B-Range**, and **C-Range**. Below these are two rows of **Value** and **Type** settings: **DATA0** and **DATA1** (both set to **B0**), and **COUNT0** and **COUNT1** (both set to **us**).

Click here to open the Define Counter Window

Click here to select a Time Counter

Type in the value

Click here to select time unit



The **Define Counter** dialog box is shown with the **Time Counter** radio button selected. The **Value** field contains **2.** and the **Unit** dropdown is set to **us**. There is also a **HEX Value** checkbox and **CLEAR** and **OK** buttons.

Definition

A-Range : V or V or V
 B-Range : V or V or V
 C-Range : V or V or V

Value Type Value Unit
 DATA0: V B0
 DATA1: V B0
 COUNT0: V us
 COUNT1: V

Level0

Action	Address/Range	RD/WR...	Data	Count
Inc COUNT0	IF <input type="text"/> & <input type="text"/> & <input type="text"/>			
Mark A	IF <input type="text"/> & <input type="text"/> & <input type="text"/>			COUNT0
Sample	IF <input type="text"/> & <input type="text"/> & <input type="text"/>			

```

E::a.l mark run address cycle t.a d.w symbol ti.back level l.mix
Setup... Goto... Find... Set Ref Set Zero View More Less
record mark run address cycle t.a d.w symbol
000659 --- G0 SP:00001788 rd-word 00000000 48E7 \\mcc\mcc\sieve
000658 --- f SP:0000178A rd-word 00000000 3830 \\mcc\mcc\sieve+2
000657 --- f SP:0000178C rd-word 00000000 227C \\mcc\mcc\sieve+4
550
551
552 char flags[SIZE+1];
553
554 int sieve() /* sieve of erathost
555 {
movem.l [d2-d4,a2-a3],-(a7)
000656 --- f SD:0000FECA wr-word 00000000 2540
000655 A-- f SD:0000FEC8 wr-word 00000000 0000
000654 A-- f SD:0000FEC6 wr-word 00000000 2550
000653 A-- f SD:0000FEC4 wr-word 00000000 0000
000652 A-- f SD:0000FEC2 wr-word 00000000 0000
000651 A-- f SD:0000FEC0 wr-word 00000000 0000
000650 A-- f SD:0000FEBE wr-word 00000000 0000
000649 A-- f SD:0000FEB8 wr-word 00000000 0000
000648 A-- f SD:0000FEBA wr-word 00000000 000B
000647 A-- f SD:0000FEB8 wr-word 00000000 0000
000646 A-- f SP:0000178E rd-word 00000000 0000 \\mcc\mcc\sieve+f
movea.l #2774,a1
000645 A-- f SP:00001790 rd-word 00000000 2774 \\mcc\mcc\sieve+f
000644 A-- f SP:00001792 rd-word 00000000 7200 \\mcc\mcc\sieve+f
    
```

Example 8: Trace all, mark the Cycles executed in Function Sieve with Label B and mark the Entrance of Function Sieve with Label A

Analyzer Programming

Definition

Define the begin of function sieve as A-Range

A-Range : or or

B-Range : or or **Define the end of function sieve as B-Range**

C-Range : or or

Analyzer Programming

Definition

A-Range : or or

B-Range : or or

C-Range : or or

Value Type Value Unit

DATA0: B0 COUNT0:

DATA1: B0 COUNT1:

Level0

Action	IF	Address/Range
Mark A	IF	A-Range
Goto Level1	IF	A-Range
Sample	IF	

Level1

Mark B	IF	
Goto Level0	IF	B-Range
Sample	IF	

Display was cut here

Setup...	Goto...	Find...	Set Ref	Set Zero	View
record	mark	level	run	address	cycle
000012	---	LEVEL0	f	SP:0000177E	rd-wor
000011	---	LEVEL0	f	SP:00001776	rd-wor
000010	---	LEVEL0	f	SP:00001778	rd-wor
000009	A--	LEVEL0	f	SP:00001788	rd-wor
				breakpoint	
542					
543				func1b(&vint);	/* fc
544					
545				while (TRUE)	
546				{	
547				sieve();	
				jsr 1788(pc)	
000008	-B-	LEVEL1	f	SD:0000FECC	wr-wor
000007	-B-	LEVEL1	f	SD:0000FECE	wr-wor
000006	-B-	LEVEL1	f	SP:0000178A	rd-wor
000005	-B-	LEVEL1	f	SD:0000FECA	wr-wor
000004	-B-	LEVEL1	f	SD:0000FEC6	wr-wor
000035	-B-	LEVEL1	f	addq.l #1,d2	SP:000017CE rd-wor
000034	-B-	LEVEL1	f	moveq #12,d0	SP:000017D0 rd-wor
000033	-B-	LEVEL1	f	cmp.l d2,d0	SP:000017D2 rd-wor
000032	-B-	LEVEL1	f	bge 17AA	SP:000017D4 rd-wor

Example 9: Trace all, when Function Sieve is reached goto Level 1 and stop Sampling and Debugging after 5 Cycles

Define the begin of function sieve in A-Range

Analyzer Programming

— Definition —

	Address	or	Address	or	Address
A-Range :	sieve	V		V	
B-Range :		V		V	
C-Range :		V		V	

	Value	Type	Value	Unit
DATA0:		B0	COUNT0: 5	
DATA1:		B0	COUNT1:	

— Level0 —

Action	IF	Address/Range	&	RD/wR...	&	Data	&
Sample	IF		&		&		&
Goto Level1	IF	A-Range	&		&		&
Restart COUNT0	IF		&		&		&

Level1

Action	IF	Address/Range	&	RD/wR...	&	Data	&
Sample	IF		&		&		&
Inc COUNT0	IF		&		&		&
Stop CPU&Analyzer	IF		&		&		&

Change level if sieve is reached

Always set COUNT0 to zero

Add one to COUNT0 every cycle

Stop sampling and debugging when COUNT0 reaches 5 (decimal)

E::d.l							
Step	Step Over	Go Next	Go Return	Go Up	Go	Break	Mode
addr/line	code	label	mnemonic	comment			
	int sieve()			/* sieve of erathostenes			
555	{						
SP:00001788	48E73830	sieve:	movem.l	[d2-d4,a2-a3],-(a7)			
SP:0000178C	227C00002774		movea.l	#2774,a1	; #flags,a1		
	register int i, prime, k;				Debugger stops here		
	int count;						
559	count = 0;						
SP:00001792	7200		moveq	#0,d1	; #0,count		
561	for (i = 0 ; i <= SIZE ; flags[i++] = TRUE) ;						
SP:00001794	7400		moveq	#0,d2	; #0,i		
SP:00001796	2042		movea.l	d2,a0	; i,a0		
SP:00001798	5282		addq.l	#1,d2	; #1,i		
SP:0000179A	13BC00018800		move.b	#1,0(a1,a0.l)	; #1,0(a1,a0.l)		
SP:000017A0	7012		moveq	#12,d0	; #18,d0		
SP:000017A2	B082		cmp.l	d2,d0	; i,d0		

E::a.l level default						
Setup...	Goto...	Find...	Set Ref	Set Zero	View	More
record	level	run	address	cycle	d.w	symbol
-000009	LEVEL0	f	SD:00002742	wr-word	0001	\\mcc\G
-000008	LEVEL0	f	SD:00002740	wr-word	0000	\\mcc\G
-000007	LEVEL0	f	SP:00001788	rd-word	48E7	\\mcc\m
542						
543			func1b(&vint);			/* forward re
544						
545			while (TRUE)			
546			{			
547			sieve();			
			jsr 1788(pc)			; ;
-000006	LEVEL1	f	SD:0000FECC	wr-word	0000	
-000005	LEVEL1	f	SD:0000FECE	wr-word	177A	
-000004	LEVEL1	f	SP:0000178A	rd-word	3830	\\mcc\m
550						
551						
552			char flags[SIZE+1];			
553						
554			int sieve()			/* sie
555			{			
			movem.l [d2-d4,a2-a3],-(a7)			
-000003	LEVEL1	f	SP:0000178C	rd-word	227C	\\mcc\m
-000002	LEVEL1	f	SD:0000FECA	wr-word	2540	
-000001	LEVEL1	-f	SD:0000FEC8	wr-word	0000	
T000000	LEVEL1	ft	SD:0000FEC2	wr-word	0000	

Level is changed →

Sampling stops here →