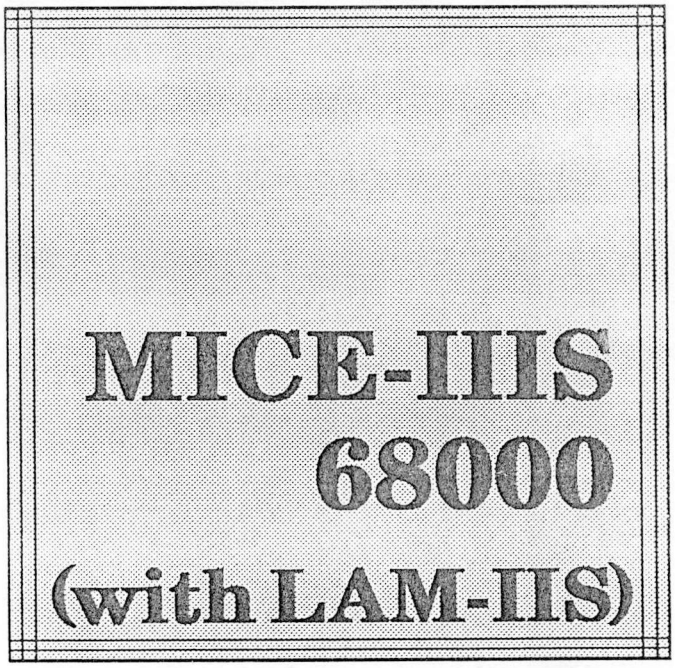
MICE-IIIS 68000 with LAM-IIS

User's Manual Addendum

MICROTEK

USER'S MANUAL ADDENDUM



Doc No. 149-000688
First Edition
October 1992

READ ME FIRST!!

This addendum contains unique information that applies to MICE equipped with LAM-IIS only. For other MICE information not mentioned in this addendum, please refer to the main manual or *"MICE-16 68000 User's Manual"*. MICE equipped with LAM-IIS is referred to as "MICE-IIIS 68000.

Trademarks Acknowledgement

IBM,PC,XT, AT and PS/2 are trademarks of International Business Machines.
MS-DOS and MS-WINDOWS are trademarks of Microsoft Corporation
Sun Microsystems is a trademark of Sun Microsystems, Inc.
UNIX is a trademark of AT&T Bell Laboratories.
MRI is a trademark of Microtec Research Inc.
NEC is a trademark of NEC Corporation.

This manual is subject to change without notice. Nothing herein shall be construed as a recommendation to use any product in violation of existing patents or other rights of third parties.

MICE-IIIS 68000 (WITH LAM-IIS) AND THIS ADDENDUM ARE COVERED BY THE LIMITED WARRANTY SUPPLIED WITH MICE-IIIS 68000.

Printed in Taiwan, ROC 10/92

CONTENTS

Addendum 1
LAM-IIS INTRODUCTION

When MICE-16 68000 is equipped with the upgraded Logic Analyzer Module (LAM-IIS), it is designated as "**MICE-IIIS 68000**". It offers several new and improved features designed to make the MICE a more efficient emulation machine. The following table summarizes the areas where improvements has been implemented:

Features	With LAM	With LAM-IIS
Event	Two Bus Breakpoints with: - Bit wildcard address - Bit wildcard data - Multiple bus status - Up to 64K event count (EV1 only)	Four Bus Breakpoints with: - Range or bit wildcard address - Bit wildcard data - Multiple bus status - Up to 64K event count - 8 external trace bits
Trigger	And/Or/Then DELay/BACKward/FOrward/CENter	Provides up to 8 of the following Trigger level: - And/Or/Not - Trace {On|Off}/Timer {On/Off} The following Trigger settings are available: - If...Else/Then/REset/DELay - Trace {On/Off} Timer {On/Off}
Qualify Trace	One Qualify trace with: - Bit wildcard address - Multiple bus status	Two Qualify trace with: - Range or bit wildcard address - Multiple bus status - 8 external trace bits
Break on Read before Write	None	1Mbytes (4 independent banks at 256Kbytes each) range address.
Code Coverage	None	1Mbytes (4 independent banks at 256Kbytes each) range address.

Trace Buffer	2K frames with: - 32 channels to address - 32 channels to data - 8 channels to status - 8 channels to external trace bits - 24 bits timer	32K frames with: - 32 channels to address - 32 channels to data - 16 channels to status - 8 channels to external trace bits - 8 channels to state and event - 32 bits timer
External Trigger Inputs	Buffer interface	Latch interface may be specified to high or low level trigger. Signal valid during low state STROBE.
Timebase	Unit of measurement: - from 1μs to 1000μs Timer length: - max: 44 hrs. - min : 16 sec.	Unit of measurement: - from 0.1μs to 1000μs Timer length: - max: 49 days 17 hrs. - min : 7 min. 9 sec.

Table-1 LAM and LAM-IIS Features Comparison

All features of LAM version are supported by LAM-IIS. Hence only features unique to LAM-IIS are discussed in this addendum.

1.1 KEY FEATURES OF MICE-IIIS 68000

1) **Real-Time Trace:**

The trace buffer is 32K frames deep and 128 bits wide. Signals monitored by the trace feature are:

EP address bus (32 bits)
EP data bus (32 bits)
EP status signals (16 bits)
External trace bits (8 bits)
Trigger state (8 bits)
Timer stamp (32 bits)

2) **Trace with Cycle Qualifiers:**

Qualifiers can be set so that only cycles with the specified range or bit wildcard address and/or external trace bits are recorded in the trace buffer. The address may include range address, wildcard bits or a wildcard nibble; any combination of processor states may be specified.

3) **Multiple (Seven) Real-Time Breakpoints:**

The MICE-IIIS 68000 provides three execution breakpoints; four bus breakpoints (with options for range/wildcard address, data, status, external trace bits and count qualify); and one external hardware breakpoint (with conditional qualify). All execution breakpoints are hardware breakpoints for programs located in either RAM or ROM, but bus breakpoints may be located in either program or data memory in RAM or ROM.

4) **Powerful Trigger Constructs:**

The trigger conditions is composed of up to 8 "trigger-level" where each level specifies an event combination. Each level may be a single event or a logic combination of EV1~ EV5 with **ANd**, **OR** and **Not** as operators. A cycle delay feature is also included allowing trace/emulation to continue 0-65535 cycles after all trigger conditions have been met.

5) **Initialize and Code Coverage**

LAM-IIS supports Initialize and Code Coverage RAM of 1Mbytes range address. The range consisted of 4 independent banks at 256Kbytes each. When using Initialize, program will break on Read-Before-Write at the specified address. When using Code Coverage, user will be able to monitor the performance program execution.

1.2 HOW TO UPGRADE FROM LAM TO LAM-IIS

LAM-IIS module when shipped to update a LAM equipped MICE-16 68000, is delivered consisting of the following items:

One LAM-IIS board
Four EPROMs U11, U12, U14 and U15 with LAM-IIS Firmware (Version 5.0 or above)
One MICE-16 68000 User's Manual with Addendum for LAM-IIS

Follow these steps to upgrade to LAM-IIS equipped MICE:

1. Replace existing LAM with the LAM-IIS board
2. Replace EPROMs on CPM board with the new EPROMs provided.

NOTE

The CPM board should be of Revision-G or above in order to be compatible with LAM-IIS firmware.

Use USD-III Version 2.0 to fully utilize LAM-IIS features under USD.

Addendum 2
INITIALIZATION

This addendum explains the unique power up screen display and default setup of MICE-IIIS 68000 with LAM-IIS.

2.1 POWER UP SELF-TEST

MICE-IIIS 68000 power up self-test arrangement is the same for both LAM and LAM-IIS. Use >S<CR> in lieu of >M<CR> to skip self-test.

2.2 POWER UP SCREEN DISPLAY

With LAM-IIS, the resulting power up self-test screen display when proper communications are established after keying >M<CR>, are as illustrated in the following Figures 2-1a/1b. Figure 2-2 shows power up screen display with self-test skipped:

```
*** MICE-16 68000/68010 (W/LAM2S), Firmware V5.0 ***
        MICROTEK International Inc., 1992

The emulation processor is 68000 ( 16-bit )

Self-test in progress...
CPM ROM    OK
CPM RAM    OK
EPM RAM    OK
LAM2S      OK
EMM RAM    OK
Self-test completed
Emulation memory is EMM
Four independent 64K-byte memory banks
Mapping resolution is 16K-byte

No Target VCC; BERR, BR, INTR (IPL0-2) disabled
>
```

Figure 2-1a LAM-IIS Self-Test Power Up Status Screen with EMM

```
*** MICE-16 68000/68010 (W/LAM2S), Firmware V5.0 ***
        MICROTEK International Inc., 1992

The emulation processor is 68000 ( 16-bit )

Self-test in progress...
CPM ROM    OK
CPM RAM    OK
EPM RAM    OK
LAM2S      OK
HEMM RAM BANK1  OK
HEMM RAM BANK2  OK
HEMM RAM BANK3  OK
HEMM RAM BANK4  OK
Self-test completed

Emulation memory is HEMM
Four independent 256K-byte memory banks
Mapping resolution is 4K-byte

No Target VCC; BERR, BR, INTR (IPL0-2) disabled
>
```

Figure 2-1b LAM-IIS Self-Test Power Up Status Screen with HEMM

```
*** MICE-16 68000/68010 (W/LAM2S), Firmware V5.0 ***
        MICROTEK International Inc., 1992

The emulation processor is 68000 ( 16-bit )

LAM2S Initializing   ...

No Target VCC; BERR, BR, INTR (IPL0-2) disabled

>
```

Figure 2-2 LAM-IIS Power Up Status Screen with Self-Test Skipped

Refer to Section 3.2 of the main manual for other status of power up screen displays.

2.3 MICE-IIIS 68000 DEFAULT SETUP PARAMETERS

The LAM-IIS setup defaults are practically the same as those stated in Chapter 3, Section 3.3 for LAM except for the following items:

1) Events 1 to 8 - Events 1, 2, 3, 4, 6, 7 and 8 Clear; Event 5 Low, no conditional

2) Trigger Condition - Trigger A
Trigger Timer On Trace On
Trigger Level A EV1 OR EV2 OR EV3 OR EV4 Trace On
TImer On
Trigger Level B to H, Clear

3) Timebase Selection - Recall from NOVRAM
(factory NOVRAM setting:
Timebase is 0.1 μs)

4) Cycle Qualify - All machine cycles recorded

Addendum 3
COMMAND SUMMARY

3.1 COMMAND FUNCTIONAL GROUPS OVERVIEW

The following is an overview of LAM-IIS equipped MICE-IIIS 68000 commands, grouped according to their functions.

1) Setup Commands

BAud	IDentify	RECall	SETup
CLock	INTerval	ROute	SIze
CONtrol	MAp	SAve	Verify
HANdshake	Prompt	SELect	WAit
Help	REAdy		

2) Memory Commands

Assemble	COMpare	Fill	SEarch
Byte	COpy	LOng	TEst
CHecksum	Disassemble	Memory	Word

3) Port Commands

Input	Output

4) Emulation Commands

Cycle	Register	RESet	Step
Jump			

5) Trace Commands

COVerage	HAlt	Qualify	TRAce
Event	INItialize	SYnc	Trigger
Go	List	TImebase	

3.2 COMMAND FUNCTIONAL GROUPS SYNTAX LISTING

3.2.1 Setup Commands

FUNCTION	COMMAND SYNTAX	
	KEYWORD	PARAMETER(S)
Port B Baud Rate Setting	BAud	[baud-rate [parity] [data-bits]]
Display Clock Source	CLock	
Control Signal Enable/Disable	CONtrol	[[Berr] [Intr] [BR] {Enable|Disable}]
Set Handshaking Code	HANdshake	[code]
Command Help	Help	[GRoup {Emulation|Memory|Port|Setup|Trace}|command|ALl]
Identify Emulator	IDentify	
Timer Interval Switch	INTerval	[On|OFf]
Memory Mapping	MAp	[space-adr adr {{I|iR} {1|2|3|4} |E|ER|G}|ALl {E|G}]
Change Command Prompt	Prompt	[string]
Ready Signal Selection	REAdy	[Internal|External]
Recall Status from NOVRAM	RECall	
Change Operation Mode	ROute	[System]
Save Status to NOVRAM	SAve	
Select Leading Code	SELect	[{Route|DEVice} [code]]
Display Setup Parameters	SETup	
Memory Access Size	SIze	[Byte|Word]
Memory Verification Switch	Verify	[On|OFf]
Insert Wait State	WAit	[On|OFf]

The Setup group of commands are used to configure MICE for the target before emulation or debugging could start.

3.2.2 Memory Commands

FUNCTION	COMMAND SYNTAX	
	KEYWORD	PARAMETER(S)
Line Assembly	Assemble	[space-adr]
Memory Modify	Byte	[space-adr [data]]
Memory Checksum	CHecksum	space-adr {adr|Length length}
Memory Compare	COMpare	space-adr1 {adr|Length length} space-adr2
Memory Copy	COpy	space-adr1 {adr|Length length} space-adr2
Disassembly	Disassemble	[space-adr [adr|Length length]]
Memory Fill	Fill	space-adr {adr|Length length} data
Memory Modify	LOng	[space-adr [data]]
Memory Dump	Memory	[space-adr [adr|Length length]]
Memory Search	SEarch	space-adr {adr|Length length} data
Memory Test	TEst	space-adr {adr|Length length}
Memory Modify	Word	[space-adr [data]]

The Memory Group commands provide access to the contents of designated memory locations. The MICE-IIIS 68000 supports four memory types: Supervisor Program (SP), Supervisor Data (SD), User Program (UP), and User Data (UD). For all of the commands described in this section, the memory type that it desired to access, should be specified. If memory type is not specified, the MICE will automatically treat the accessed memory as supervisor program space). The MICE-IIIS 68000 also supports three basic data formats: Byte, Word and Long Word. For the Memory Dump, CHecksum, Test, and Copy commands, the desired format should be specified using the Memory Access Size command (Section 5.3.16 of the main manual). The total amount of addressable memory is 16M bytes.

Because the 68000 is word-oriented, even-numbered addresses must be assigned for the program counter (PC) in the Assemble and Disassemble commands (Sections 5.4.1 and 5.4.6 of the main manual respectively) and for the supervisor stack pointer (SSP) and user stack pointer (USP). If an odd value is entered, the system will drop the least significant bit to modify it to a legal even value.

The MICE always makes sure that the memory and port addresses specified by the user are valid before performing an operation. Any reference to an invalid address will result in display of an error message and termination of the command. If Memory Verification (Section 5.3.17 of the main manual) is on, the MICE also verifies all memory write operations, and if data is not written correctly to the specified address, an error message will be displayed and the command terminated. The MICE, however, does not verify that there is memory or anything else connected to a port when performing a read operation.

Note that when the external ready signal is selected and read/write functions are being performed, and if the external circuitry does not respond with the $\overline{\text{DTACK}}$ signal,

```
"Address xxxxxxxx is not ready"
```

will appear on the screen with the contents of memory remaining unchanged.

3.2.3 Port Commands

FUNCTION	COMMAND SYNTAX	
	KEYWORD	PARAMETER(S)
Port Input	Input	space-adr [Length in-length]
Port Output	Output	space-adr data

These commands provide access to the contents of I/O ports. The MICE-IIIS 68000 supports three basic data formats: Byte, Word and Long Word. For the Port Input and Port Output commands, the desired data format must be specified with the Memory Access Size command (Section 5.3.16 of the main manual). I/O is memory mapped, while the total amount of addressable memory is 16M bytes and the address range is user-defined.

The MICE always checks to make sure that the port addresses specified by the user are valid before an operation is performed. Any reference to an invalid address will result in an error message and termination of the command. If the Memory Verification Switch is on, the MICE will also verify all memory write operations (Section 5.3.17 of the main manual), and if data has not been written correctly to the specified port address, an error message will be displayed and the command will be terminated. The MICE, however, does not verify that there is memory or anything else connected to a port when executing a read operation.

NOTE

When the external ready signal is selected and read/write functions are being performed, if external circuitry does not respond with the $\overline{DTACK}$ signal,

"Address xxxxxxxx is not ready"

will appear on the console with the contents of memory remaining unchanged.

3.2.4 Emulation Commands

FUNCTION	COMMAND SYNTAX	
	KEYWORD	PARAMETER(S)
Cycle Step	Cycle	[Wait|count]
Jump	Jump	adr
Register Display/Modify	Register	[register]
Reset Emulation Processor	RESet	[pc [ssp]]
Instruction Step	Step	[adr1 adr2 [CALI]|count|CALI]

The total amount of addressable memory for the 68000 is 16M bytes. Because the 68000 is word-oriented, even-numbered addresses must be assigned for the program counter (PC) in the Jump and RESet commands (Sections 5.6.2. and 5.6.4 of the main manual respectively) as well as for the supervisor stack pointer (SSP) and user stack pointer (USP) in the Register command (Section 5.6.3 of the main manual). If an odd value is entered, the system will drop the least significant bit to convert it into an even value.

Before carrying out a command, the MICE always checks any memory and/or port addresses input by the user to make sure they are valid. A reference to an invalid address will result in display of an error message-

"Command syntax error"

and termination of the command.

When executing the Cycle Step and Instruction Step commands (Section 5.6.1 and 5.6.5 of the main manual respectively), refer to Table 5-2 for a description of the displayed status information. Note that breakpoints (Section 5.7.1 of the main manual) are not active during execution of these commands.

During execution of these commands, if an attempt is made to refer to the memories which are defined as either guided (G) or write protected (ER/IR),

"Write protected memory trespassed"
or
"Guarded memory trespassed"

will display to warn user of such violation. Emulation processing however, will continue after <CR> is entered. the illegally entered data for external /internal write protected (ER/IR) or non-existence (G) memory block will be indecipherable.

When performing read/write function with the external ready signal being selected and the external circuitry does not respond to the $\overline{\text{DTACK}}$ signal, the message-

"Address xxxxxxxx is not ready"

will display with the contents of memory remaining unchanged.

3.2.5 Trace Commands

FUNCTION	COMMAND SYNTAX	
	KEYWORD	PARAMETER(S)
Set Coverage	COVerage	[adr1 adr2[CLear]|[ENable|DIsable| CLear|Reset]]
	COVerage	List [adr]
	COVerage	SP|UP|SD|UD
Display Events	Event	
Set Bus Event 1	Ev[ent]	1 {Range adr1 adr2|adx} [datum] [status] [External trace-bits] [COunt count]
Set Bus Event 2	Ev[ent]	2 {Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Set Bus Event 3	Ev[ent]	3 {Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]

Set Bus Event 4	Ev[ent]	4 {Range adr1 adr2 adx} [datum] [status] [EXternal trace-bits] [COunt count]
Set External hardware breakpoint	Ev[ent]	5 [COnditional] {High|Low}
Set Execution Event 1	Ev[ent]	6 adr [fetch-state]
Set Execution Event 2	Ev[ent]	7 adr [fetch-state]
Set Execution Event 3	Ev[ent]	8 adr [fetch-state]
Clear Events	Ev[ent]	[1|2|3|4|5|6|7|8] [CLear]
Execute User Program	Go	[Run] [adr]
Halt Emulation Processor	HAlt	
Set Initialize	INItialize	[adr1 adr2[CLear]|[ENable|DIsable| CLear|Reset]]
	INItialize	List [adr]
	INItialize	SP|UP|SD|UD
List Trace Buffer	List	[frame [adr1 adr2] [status] [EXternal trace-bits]]
List with Source Code	List	{Source|Instruction} [frame]
List Trace Frame Total	List	Number
Set Trace Cycle Qualifier	Qualify	{1|2} [{Range adr1 adr2|adx} [status] [EXternal trace-bits]]
Display/Clear/En/Disable Qualifier	Qualify	{1|2} [CLear|DIsable|ENable]
Set Sync Input/Output	SYnc	[{Input|OUtput} {On|OFf}]
Timebase Selection	TImebase	[0.1|1|10|100|1000]
Display Trace Condition	TRAce	
Set Trigger Level	Trigger	Level level# [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev#]]]] [Trace {On|OFf}] [TImer {On|OFf}]
Display/Clear Trigger Level	Trigger	Level level# [CLear]
Set Trigger	Trigger	[RUn] {level# [<THen level#>] [Reset | If state# <ELse state#>] | If state# <ELse state#>} [FOrward|BAckward| CEnter|DElay count]
Set Initial Timer & Trace	Trigger	[TImer {On|OFf}] [Trace {On|OFf}].
Display/Clear Trigger	Trigger	[CLear]

These commands are used to trace program execution in real-time, with the emulation processor free running. Four bus breakpoints, three execution breakpoints and an external signal event can be set singly or in combination, using flexible trigger constructs, as conditions for termination of tracing and/or free-running emulation. An optional delay allows emulation/tracing to continue for up to FFFFH machine cycles after all other trigger conditions have been met. Timer On/Off or Trace On/Off may also be specified under any trigger conditions.

System activity is recorded in a 32K trace buffer with a capacity of up to 32768 machine cycles. A qualifier may be set to specify that only cycles meeting certain address bus, processor state conditions and/or external trace-bits be recorded. State and events are also recorded in the trace buffer.

A synchronization command is provided to allow simultaneous start of emulation/tracing on any number of MICE-IIIS 68000 units connected to a multiprocessor system. Operation is completely independent after activation by the sync signal.

The contents of the trace buffer may be listed after termination of the trace. The listing can be set to begin at any recorded cycle and to include either all subsequent cycles or only those cycles meeting certain bus, external and/or processor state conditions. Machine code recorded from the data bus may be displayed in disassembled form with the use of Source option.

If it is desired to use an external signal as a trigger condition, a signal I/O cable (provided with your MICE unit) must be connected between the signal source and the EXT TRIGGER INPUT port on the MICE-IIIS 68000 front panel. To record external signals in the trace buffer, the 9-miniprobe trace cable supplied with the MICE must be attached to the source points and plugged into the TRACE BITS port on the MICE. External trace-bits may also be used as a breakpoint condition. Signals input at all of these ports must be TTL compatible.

Tracing and free-run emulation are in real-time. Information monitored during tracing and emulation are displayed in the trace buffer listing and at the Cycle and Step commands. These are defined under the "COLUMN HEADING" in Table 3-1 below.

Table 3-1 shows the types of processor activities which are to be specified as conditions for trace-buffer recording/listing and bus breakpoints 1 to 4. The codes shown at the left column of the table are used to specify that only machine cycles with certain types of processor activity can be recorded in the trace buffer and displayed in the trace buffer listing. The codes are also used in Event 1 to Event 4 commands where a breakpoint is encountered at the specified address only when the indicated type of processor activity is matched.

Note that there is no limitation on the number of status types that may be specified. If no status is specified, all machine cycles will default.

COLUMN HEADING	DEFINITION
FRAME	is the sequential position of the displayed execution cycle, the range is 0-7FFF (0-32767).
ADDRESS	is the hexadecimal value on the address bus.
DATA	is the hexadecimal value on the data bus.
STATUS	is the type of processor activity.
TRACE BITS	indicates signal level input from miniprobes connected to the external TRACE-BITS port.
STATE	indicates trigger states.
EVENT	indicates the "Event encounter-flag" of EV1~EV5.

Table 3-1 Information Recorded in the Trace Buffer

Addendum 4
LAM-IIS COMMAND DESCRIPTION AND EXAMPLES

The following pages will describe the command syntax which are unique to the LAM-IIS equipped MICE-IIIS 68000 under system operating configurations described in Chapter 1, Section 1.3 of the main manual. These commands are arranged in alphabetical order and each command is provided with one or more typical examples. Details and examples of other commands which are common to both LAM and LAM-IIS are provided in Chapter 5 of the main manual.

For greater convenience in interpreting the symbols and notations used in each command syntax and in the given examples, user should first become familiar with the Conventions of Notation explained in Chapter 4, Section 4.1 of the main manual.

Using the Assemble command (Section 5.4.1 of the main manual), the following programs were derived and utilized to obtain most of the command execution examples given in this chapter. User should use these programs to duplicate the examples.

1. Writing data 55AA to memory 1000-1FFFH.

```
LOC         OBJ              SOURCE CODE
000400      327C 1000        MOVEA.W   #1000,A1
000404      32FC 55AA        MOVE.W    #55AA,(A1)+
000408      B2FC 2000        CMPA.W    #2000,A1
00040C      66F6             BNE.B     000404
00040E      4EF8 040E        JMP       00040E
000412      4E71             NOP
000414      4E71             NOP
000416      4E71             NOP
```

2. Checking memory 1000-1FFFH. If it does not equal to 55AA, then go to 518.

```
LOC         OBJ              SOURCE CODE
000500      327C 1000        MOVEA.W   #1000,A1
000504      3019             MOVE.W    (A1)+,D0
000506      B07C 55AA        CMP.W     #55AA,D0
00050A      660C             BNE.B     000518
00050C      B2FC 2000        CMPA.W    #2000,A1
000510      66F2             BNE.B     000504
000512      4EF8 0512        JMP       000512
000516      4E71             NOP
000518      4EF8 0518        JMP       000518
00051C      4E71             NOP
00051E      4E71             NOP
000520      4E71             NOP
```

4.1 CODE-COVERAGE TEST - COVerage

```
COVerage [adr1 adr2[ CLear]|[ENable|DIsable|CLear|Reset]]
COVerage List [adr]
COVerage SP|UP|SD|UD
```

COVerage is the command to set or clear coverage settings. COVerage alone without any parameter will display the current coverage memory type and address ranges settings.

adr1 is the hexadecimal address specifying the starting point of a memory block.

adr2 is the hexadecimal address indicating the ending point of a memory block.

CLear is used to clear a particular range of address or to clear all existing address ranges.

ENable is a global control to enable all of the address settings.

NOTE

If the INItialize command (Break on Read-Before-Write function, Section 4.6 of this Addendum) is active at the time when COVerage command is enabled, the INItialize command will automatically switch to inactive condition. Likewise, COVerage function is disabled when INItialize command is enabled while COVerage is active. Hence only one command (COVerage or INItialize) is active at a time.

DIsable is a global control to disable the settings.

Reset re-initializes the code-coverage test without clearing the address ranges setting.

List lists the address ranges and the percentage of memory which were accessed for a coverage test.

adr is the hexadecimal address specifying the starting point of the address ranges to be listed.

SP|UP|SD|UD are the four codes specifying memory type to be executed with the code coverage test.

Where: SP is the supervisor program
SD is the supervisor data
UP is the user program
UD is the user data

If none is specified, all four memory types will default.

Code-coverage test checks the execution efficiency of a users program and displays the total rate of accessed memory ranges (actual accessed ranges / user specified ranges). The address ranges actually covered or accessed are also displayed.

A maximum of 1M-bytes in four independent 256K-byte ranges of memory (0~3FFFFH, 40000H~7FFFFH, etc.) can be set for code-coverage test. Up to 20 coverage address ranges can be set within these four independent 256K-byte memory banks of the 16M memory map. The locations of each of the coverage address must be kept within the boundaries of these four independent 256K-byte ranges. No coverage address range should straddle across any other range other than the specified four banks. If a non-specified range is included in a coverage address settings, the following error message will display:

```
"At most 20 address sets."
```

or

```
"At most 4 bank sets."
```

Code-coverage test is effective only in real-time emulation. It will not work under Cycle Step or Instruction Step commands. Note that the prefetched cycle during free-running is also in effect when performing code-coverage test.

If COVerage is invoked while INItialize is active, the former is disabled and the following message will display:

```
"Coverage test is globally disabled"
```

User should therefore enable the COVerage command first, before invoking COVerage. INItialize is automatically disabled whenever COVerage is enabled if the former is active. Both commands are disabled by default on MICE power up.

Examples:

1. Using the program provided at the beginning of this chapter, enable code-coverage test, set memory ranges to be covered, run emulation and then list the program execution efficiency and the address ranges actually covered:

```
>RESet<CR>
>COVerage ENable<CR>
>COVerage 400 430<CR>
>COVerage 1000 2000<CR>
>COVerage 3000 4000<CR>
>COVerage<CR>
 The ranges for performing coverage test :
 Memory type : SP,SD,UP,UD
  000400 000430
  001000 002000
  003000 004000
>Go 400<CR>
 Trace in progress...
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 02.048 093 7
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 49.90%
 The list of address ranges are covered :
 000400 000411
 001000 001FFF
>
```

2. Using the program provided at the beginning of this chapter, clear one of the set ranges and list the program execution efficiency and the address ranges actually covered.

```
>COVerage<CR>
 The ranges for performing coverage test :
 Memory type : SP,SD,UP,UD
  000400 000430
  001000 002000
  003000 004000
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 49.90%
 The list of address ranges are covered :
 000400 000411
 001000 001FFF
>COVerage 400 430 CLear<CR>
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 49.98%
 The list of address ranges are covered :
 001000 001FFF
>
```

3. Using the program provided at the beginning of this chapter, reset the coverage, qualify the memory type "UP", run emulation and list the program execution efficiency and the address ranges actually covered.

```
>COVerage RESet<CR>
>Register S<CR>
S      1 0
I2     1 <ESC>
>COVerage UP<CR>
>COVerage 400 430<CR>
>COVerage<CR>
 The ranges for performing coverage test :
 Memory type : UP
  000400 000430
  001000 002000
  003000 004000
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 00.00%
>Go 400<CR>
 Trace in progress...
 Emulation processor stopped by user
```

```
 Last frame = 7FFF,timer = 01.406 338 9
>
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 00.21%
 The list of address ranges are covered :
 000400 000411
>
```

4. Using the program provided at the beginning of this chapter, reset and disable the coverage test, then run emulation and list the program execution efficiency and the address ranges actually covered.

```
>COVerage REset<CR>
>COVerage DIsable<CR>
>COVerage<CR>
 Coverage test globally disabled.
>COVerage List<CR>
 Coverage test globally disabled.
>Go 400<CR>
 Trace in progress...
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 01.223 123 1
>COVerage List<CR>
 Coverage test globally disabled.
>COVerage Enable<CR>
>COVerage List<CR>
 Percentage of memory accessed in the program ranges :
 00.00%
>
```

5. Clear all the coverage settings.

```
>COVerage CLear<CR>
>COVerage<CR>
 No address range setting for performing coverage test.
>
```

4.2 DISPLAY/SET/CLEAR BREAKPOINTS - Event

Display Breakpoint Settings	Event
Set Bus Breakpoint 1	Ev[ent]1 {Range adr1 adr2|adx} [datum] [status] [EXternaltrace-bits] [COunt count]
Set Bus Breakpoint 2	Ev[ent]2 {Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Set Bus Breakpoint 3	Ev[ent]3 {Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Set Bus Breakpoint 4	Ev[ent]4 {Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Set External Hardware Breakpoint	Ev[ent]5 [COnditional] {High|Low}
Set Execution Breakpoint 1	Ev[ent]6 adr [fetch-state]
Set Execution Breakpoint 1	Ev[ent]6 adr [fetch-state] [Vector-Base value][1]
Set Execution Breakpoint 2	Ev[ent]7 adr [fetch-state]
Set Execution Breakpoint 2	Ev[ent]7 adr [fetch-state] [Vector-Base value]
Set Execution Breakpoint 3	Ev[ent]8 adr [fetch-state]
Set Execution Breakpoint 3	Ev[ent]8 adr [fetch-state] [Vector-Base value]
Clear Events	Ev[ent] [[1|2|3|4|5|6|7|8] CLear]

Event is the command to display, set or clear breakpoints.

1,2,3,4,5,6,7 specifies the breakpoint to be set or cleared.

Range is the required prefix when specifying an address range.

adr1 is a hexadecimal starting address for range breakpoint.

adr2 is a hexadecimal ending address for range breakpoint.

adx is a hexadecimal address setting, with option to include wildcard nibbles/bits, e.g. 1234, 1XX4, 1(X1X0)X4, 1(X1X0X0XX)4.

[1] [Vector-Based value] option is applicable to 68010 CPU only. It does not work on 68000 CPU.

datum — is a byte or word[2] in hexadecimal with wildcard nibbles/bits option (e.g. 1234, 1XX4, 1(X1X0)X4, 1(X1X0X0XX)4), to be matched on the data bus; may include wildcard nibbles and/or bits, or is specified as "X" (don't care) if only 'status' and/or 'count' must be matched along with the breakpoint address.

status — is 1 to 7 processor status codes (SP SR SW UP UR UW AK) specifying processor state to be matched as part of the break-point condition.

EXternal — is the required prefix when specifying trace-bits.

trace-bits — is the 8 bits external signals from the external TRACE BITS port.

COunt — is the required prefix when specifying a "count" (see next paragraph).

count — is a hexadecimal value from 0H to 0FFFFH specifying the number of times the breakpoint condition must be matched to complete the specific event (Event 1 or 4). Default is for 01H.

COnditional — specifies the signal at the EXT TRIGGER INPUT port for Event 5. This signal is only valid during low state $\overline{\text{STROBE}}$ ($\overline{\text{LDS}}$ or $\overline{\text{UDS}}$) condition.

High|Low — specifies the signal level to be matched at the EXT TRIGGER INPUT port for Event 5. Default is for a TTL "low" signal.

adr — is a hexadecimal address setting in the program memory space of the emulation processor.

fetch-state — is 1 to 2 processor status codes (SP UP).

Vector-Base value — is a hexadecimal value of vector-base register for processing "exception" vector address on 68010 CPU only. New value is only assigned when user's program required such value to change. If omitted, MICE will then use the current vector-base register contents (set during Event command setting) as default value.

---

[2] "datum" can not be a "word" when MICE is emulating 68000 (8-bit).

If vector-base value is changed during program execution and the new value does not match with the default or previously set value, Events 4, 5 and 6 will not break at the set address. Also the resulting program execution will be indecipherable.

CLear is the parameter to clear a particular breakpoint setting or to clear all of them at once.

Breakpoints affect only real-time emulation/tracing, not operation during the Cycle Step or Instruction Step commands. Any breakpoints that have been set, therefore, become active only after input of the Go command (Section 4.3 of this Addendum). In addition, Events 1 to 5 affect emulation/tracing only as specified in the Trigger setting (Section 4.11 of this Addendum). The power-on default setting of the MICE-IIIS 68000 is:

- Events 1, 2, 3, 4, 6, 7 and 8 Clear
- Event 5 set for a "low" signal, not qualified by $\overline{\text{LDS}}$ or $\overline{\text{UDS}}$
- Trigger set for EV1 OR EV2 OR EV3 OR EV4

Events 6, 7 and 8, if set, are automatically combined with each other and the Trigger setting in a logical OR construct; The Trigger setting may be cleared in order to deactivate Events 1 to 5 and control emulation/tracing with Events 6, 7 and/or 8 alone.

4.2.1 DISPLAY BREAKPOINT SETTINGS - Event

Event

"Event" alone without any parameters, will display all current breakpoint settings

Example: Display the current settings for Event 1 to 8.

```
>Event<CR>
 EV1 (Bus) Range 112233 556677 9XAX SP External XD Count EEFF
 EV2 (Bus) 1234XX XX78 SR,SW External FF Count 9ABC
 EV3 (Bus) 123456 90AB  External XX Count 0001
 EV4 (Bus) XXXXXX XXXX  External XX Count 0001
 EV5 (External) Conditional High
 EV6 (Execution) 000450 SP
 EV7 (Execution) 000560 SP,UP
 EV8 (Execution) 000780 SP
>
```

4.2.2 SET BUS BREAKPOINTS - Event 1, Event 2, Event 3, Event 4

Ev[ent]1	{Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Ev[ent]2	{Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Ev[ent]3	{Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]
Ev[ent]4	{Range adr1 adr2|adx} [datum] [status] [EXternal trace-bits] [COunt count]

Events 1 to 4 are real-time bus breakpoints. They consist of up to five elements: address, data, status, trace-bits and count.

An address must be specified; all other parameters are optional. When specified, data-bus value, processor status and trace-bits value must be matched in the same bus cycle as the breakpoint address. If more than one status is included, any one of those specified will satisfy the breakpoint condition. If none is given, the break-point may be encountered during any type of processor activity.

Address setting can be specified in range or wildcard. When using the prefix "Range", "adr1" should be the start address and "adr2" the end address of the range breakpoint. When wildcard breakpoint is used, only one address is neces-sary. The prefix "External" is used only when specifying trace-bits signals of TRACE BITS port. The user may connect up to 8 target signals to LAM-IIS board through the TRACE BITS port. These signals are not only recorded in the trace buffer but may be also specified in each event setting by the user.

An optional count may be specified to indicate the number of times the breakpoint condition must be matched in order to complete the event. If no count is specified, the default is 01H and the event will be encountered at the first match.

Whenever Event 1 or 2 is matched, an external sync signal is output. A negative signal pulse of 200 ns duration is generated at the SYNC 1 OUTPUT port for Event 1 and at the SYNC 2 OUTPUT port for Event 2.

These signals can be sent to an external device such as an oscilloscope or a logic analyzer. For further details refer to Section 5.3, LAM-IIS TIMING NOTES. For in-structions on generating a continuous SYNC 2 output, refer to the description of the Set Trigger command (Section 4.11 of this Addendum).

If the Trigger setting is matched at a bus breakpoint, the MICE will stop tracing at the end of the current cycle or the next cycle, depending on CPU speed and its wait cycle.

NOTE

1) *When setting a bus breakpoint at a program fetch cycle, the breakpoint may be set at any byte address for an 8-bit-wide data bus, but must be set at a word boundary for a 16-bit-wide data bus.*

2) *Bus breakpoints match bus activity, not the program counter. If the desired breakpoint address immediately follows any jump or branch instruction, set the actual breakpoint 2 or 3 words beyond the program counter. This is not necessary for execution breakpoints (Events 6, 7 and 8) because they match execution activity in the microprocessor, breaking emulation only when the instruction at the specified address is actually to be executed and not when it is just prefetched.*

Examples:

1. Set Event 1 for address 8050H and a count of 5; set Event 2 for any address in the range of 10H to 1FH, using wildcard option, data of 41H and status of SW; Set Event 3 for any address in the range of 1083H to 1097H, External trace-bits 23H and a count of 94FDH; Set Event 4 for status of SR, UR; then display the current breakpoint settings. (Remember that Events 1 to 5 are not active unless specified in the Trigger setting, the default for which is EV1, EV2, EV3 or EV4.)

```
>Event 1 8050 COunt 5<CR>
>Event 2 1X 41 SW<CR>
>Event 3 Range 1083 1097 External 23 COunt 94FD<CR>
>Event 4 X SR UR<CR>
>Event<CR>
 EV1 (Bus) 008050 XXXX  External XX Count 0005
 EV2 (Bus) 00001X 0041 SW External XX Count 0001
 EV3 (Bus) Range 001083 001097 XXXX  External 23 Count 94FD
 EV4 (Bus) XXXXXX XXXX SR,UR External XX Count 0001
 EV5 (External) Low
 EV6 (Execution) Clear
 EV7 (Execution) Clear
 EV8 (Execution) Clear
>
```

2. Set Event 1 for range address 1000H to 2000H, data of 1234H and status of UW; Set Event 2 of range address 10H to 1FH (using "Range" prefix), data of 1200H to 12FFH, status of SP, external trace-bits 3XH and a count of 10H; Set Event 3 for any location X1200H in any 1M-byte memory block in the 68000 memory range; Set Event 4 for any location X1200H within the first 1M-byte block. Also set a count of 20 and a data value with multiple wildcard for Event 4.

```
>Event 1 Range 1000 2000 1234 UW<CR>
>Event 2 Range 10 1F 12XX SP EXternal 3X COunt 10<CR>
>Event 3 X1200<CR>
>Event 4 0X1200 (X01X)2X(XXX0) COunt 20<CR>
>Event<CR>
 EV1 (Bus) Range 001000 002000 1234 UW External XX Count 0001
 EV2 (Bus) Range 000010 00001F 12XX SP External 3X Count 0010
 EV3 (Bus) XX1200 XXXX  External XX Count 0001
 EV4 (Bus) 0X1200 (X01X)2X(XXX0)  External XX Count 0020
 EV5 (External) Low
 EV6 (Execution) Clear
 EV7 (Execution) Clear
 EV8 (Execution) Clear
>
```

4.2.3 SET EXTERNAL HARDWARE BREAKPOINTS - Event 5

Ev[ent]5 [COnditional] {High|Low}

The external hardware breakpoint can be set to match either high/floating or low signal input. Note, however, that a setting of high/floating will be matched immediately if the EXT TRIGGER INPUT port is not connected to an appropriate signal source. The system default for Event 5 after power-on or a hardware reset is low. Event 5 logic is TTL compatible; a match occurs when the specified logic state is detected. For further details refer to Appendix C of the main manual for Breakpoint Timing.

The optional parameter "COnditional" allows signal from EXT TRIGGER INPUT port to remain valid only when the bus cycle is valid ($\overline{\text{STROBE}}$ is low whenever the bus cycle is valid. For MICE-IIIS 68000, $\overline{\text{STROBE}}$ is generated from CPU $\overline{\text{LDS}}$ or $\overline{\text{UDS}}$ signal). The following timing diagram illustrates how "COnditional" option works with Event 5 is set for a high signal level at the EXT TRIGGER INPUT port.

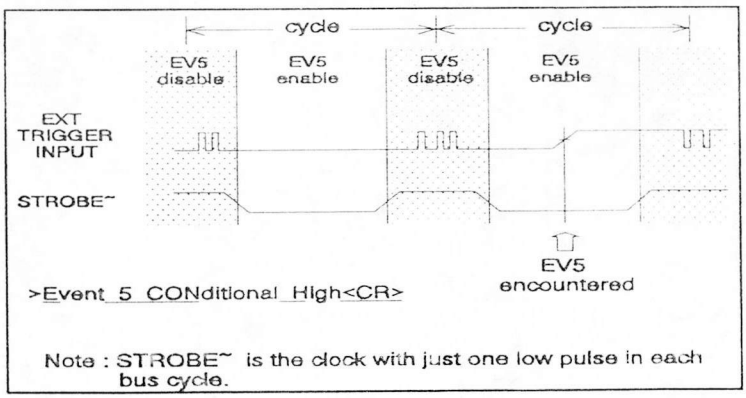


Figure 5-1 Event 5 COnditional High

NOTE

The input signal must meet the specifications for a 74LS14 Schmitt trigger IC.

Example: Set Event 5 for a low signal level at the EXT TRIGGER INPUT port and LDS or UDS low (remember that the setting does not become effective until Event 5 is specified in the Trigger setting).

```
>Event 5 COnditional Low<CR>
>
```

4.2.4 SET EXECUTION BREAKPOINTS - Event 6, Event 7, Event 8

Ev[ent]6 adr [fetch-state]
Ev[ent]6 adr [fetch-state] [Vector-Base value][3]
Ev[ent]7 adr [fetch-state]
Ev[ent]7 adr [fetch-state] [Vector-Base value]
Ev[ent]8 adr [fetch-state]
Ev[ent]8 adr [fetch-state] [Vector-Base value]

The LAM-IIS equipped MICE-IIIS 68000 provides three realtime execution breakpoints consisting of a required address. Tracing is always terminated and the emulation pro-cessor halted when an execution breakpoint is encountered.

When either Event 6, 7, or Event 8 is matched, the MICE halts the emulation processor at the cycle specified by the break condition. All machine cycles up to the breakpoint are recorded, including prefetch cycles.

Execution breakpoints remain effective. Unlike bus breakpoints, these breakpoints end the trace at a point in the program where the specified address is actually going to be executed.

The following program segment illustrates the difference between execution breakpoints and bus breakpoints:

```
>Disassemble 8000 801E<CR>
LOC         OBJ                 SOURCE CODE
008000      1A3C 0010           MOVE.B    #10,D5
008004      207C 0000 1000      MOVEA.L   #00001000,A0
00800A      227C 0000 2000      MOVEA.L   #00002000,A1
008010      3018                MOVE.W    (A0)+,D0
008012      6100 0FEC           BSR.W     009000
008016      32C1                MOVE.W    D1,(A1)+
008018      5305                SUBQ.B    #01,D5
00801A      66F4                BNE.B     008010
00801C      4E71                NOP
00801E      60FE                BRA.B     00801E
 Disassembly completed
>
```

[3] [Vector-Based value] option is applicable to 68010 CPU only. It does not work on 68000 CPU.

Setting a bus breakpoint at instruction NOP (801C) in the above example would cause the program to break at a bus prefetch cycle, even though program execu-tion has not reached the breakpoint. If the same breakpoint address is set using an execution breakpoint, the program will break only if execution reaches the breakpoint address.

NOTE

1) *When the program stops at the breakpoint address, the user's program instruction at that location has not yet been executed. The same address cannot be specified again. Setting any breakpoint where the PC equals the specified address will halt tracing immediately.*

2) *Execution breakpoints must be set at the first word of an instruction.*

3) *For Events 6, 7 and 8, emulation is controlled with 68000 line emulator code 0A0H.*

4) *Stack contents from SP-1 thru SP-6 (SP-8 for 68010) will be modified if the breakpoint is executed; but the USP and SSP registers will not be affected.*

5) *The code for breakpoint addresses is modified to 0A0H during command execution; the List command will therefore display this data instead of the user's code, for instruction prefetch cycles. This has no effect on program execution.*

6) *0A0H instructions defined in user's program are executed correctly without any conflict with internal 0A0H codes generated by Event 6, 7 or 8.*

7) *If any supervisor data memory read cycle occurs at 028H (or a line emulator code 0A0H in the user's program is executed), a few non-real time cycles will be inserted to process this code.*

8) *The execution breakpoint can be set to RAM/ROM area.*

9) *If the execution breakpoint is set and the bus breakpoint is matched, the CPU will finish executing the current instruction cycle and stop at the opcode fetch of the next instruction.*

Example: Using the program provided at the beginning of this chapter, set the bus breakpoint 1 at address 40E and display the register, then clear the bus breakpoint. Set the execution breakpoint to the same address and display register again.

```
>EVent CLear<CR>
>Disassemble 400 414<CR>
LOC         OBJ              SOURCE CODE
000400      327C 1000        MOVEA.W  #1000,A1
000404      32FC 55AA        MOVE.W   #55AA,(A1)+
000408      B2FC 2000        CMPA.W   #2000,A1
00040C      66F6             BNE.B    000404
00040E      4EF8 040E        JMP      00040E
000412      4E71             NOP
000414      4E71             NOP
 Disassembly completed
>Event 1 40E<CR>
>Go 400<CR>
 EV1 (Bus) encountered
 Emulation processor stopped
 Last frame = 0008,timer = 00.000 002 7
>Register<CR>
       0        1        2        3        4        5        6        7
Dn  FFFFFFF6 FFFFFFFF 7FFDDFF7 FFFFBFF6 FFFFFFFF DFFFFFFF FFFFFFFF FFF7FFFF
An  FFFF7FFF 00001002 7FFEFFFF EFFFFFDF DFF7FFFD F7FBFFFF FFFFF2DF 0001FFF0
SSP=0001FFF0 USP=FFFFFFFE                SR  T S  III    XNZVC
PC=000404                               2719 0010011100011001
>Event 1 CLear<CR>
>Event 6 40E<CR>
>Go 400<CR>
 Trace in progress...
 EV6 (Execution) encountered
 Emulation processor stopped
 Last frame = 1FFB,timer = 00.005 463 8
>Register<CR>
       0        1        2        3        4        5        6        7
Dn  FFFFFFF6 FFFFFFFF 7FFDDFF7 FFFFBFF6 FFFFFFFF DFFFFFFF FFFFFFFF FFF7FFFF
An  FFFF7FFF 00002000 7FFEFFFF EFFFFFDF DFF7FFFD F7FBFFFF FFFFF2DF 0001FFF0
SSP=0001FFF0 USP=FFFFFFFE                SR  T S  III    XNZVC
PC=00040E                               2714 0010011100010100
>
```

4.2.5 CLEAR BREAKPOINTS - Event CLear

```
Ev[ent ]1 CLear
Ev[ent ]2 CLear
Ev[ent ]3 CLear
Ev[ent ]4 CLear
Ev[ent ]5 CLear
Ev[ent ]6 CLear
Ev[ent ]7 CLear
Ev[ent ]8 CLear
```

Keying "Event Clear" alone, without any other parameters, clears the settings for Events 1, 2, 3, 4, 6, 7 and 8, and sets Event 5 to low, not qualify $\overline{LDS}$ or $\overline{UDS}$. To clear Event n only, input "Event n Clear."

Examples:

1. Display all breakpoint settings, then clear bus breakpoint Event 4.

```
>Event<CR>
 EV1 (Bus) Range 001000 002000 1234 UW External XX Count 0001
 EV2 (Bus) Range 000010 00001F 12XX SP External 3X Count 0010
 EV3 (Bus) XX1200 XXXX  External XX Count 0001
 EV4 (Bus) 0X1200 (X01X)2X(XXX0)  External XX Count 0020
 EV5 (External) Conditional Low
 EV6 (Execution) 000500 UP
 EV7 (Execution) 008000 SP,UP
 EV8 (Execution) Clear
>
>Event 4 CLear<CR>
>Event<CR>
 EV1 (Bus) Range 001000 002000 1234 UW External XX Count0001
 EV2 (Bus) Range 000010 00001F 12XX SP External 3X Count 0010
 EV3 (Bus) XX1200 XXXX  External XX Count 0001
 EV4 (Bus) Clear
 EV5 (External) Conditional Low
 EV6 (Execution) 000500 UP  Count 0001
 EV7 (Execution) 008000 SP,UP
 EV8 (Execution) Clear
>
```

2. Display all breakpoint settings, then clear.

```
>Event<CR>
 EV1 (Bus) Range 001000 002000 1234 UW External XX Count 0001
 EV2 (Bus) Range 000010 00001F 12XX SP External 3X Count 0010
 EV3 (Bus) XX1200 XXXX  External XX Count 0001
 EV4 (Bus) 0X1200 (X01X)2X(XXX0)  External XX Count 0020
 EV5 (External) Conditional Low
 EV6 (Execution) 000500 UP
 EV7 (Execution) 008000 SP,UP
 EV8 (Execution) Clear
>
>Event CLear<CR>
>Event<CR>
 EV1 (Bus) Clear
 EV2 (Bus) Clear
 EV3 (Bus) Clear
 EV4 (Bus) Clear
 EV5 (External) Low
 EV6 (Execution) Clear
 EV7 (Execution) Clear
 EV8 (Execution) Clear
>
```

4.3 GO/EXECUTION - Go

Go [Run] [adr]

Go is the command for Go/Execution.

Run specifies free run. During free run all breakpoint settings are ignored, but target information is still recorded in the trace buffer. (Any command other than Halt, Clock, Memory, Port, COVerage, INItialize commands and emulation group of commands may be input without stopping the emulation processor.)

adr is the hexadecimal (logical) address setting in the emulation CPU's program memory where emulation is to begin. Default is to begin from current PC.

If an address is specified, the emulation processor's program counter is first set accordingly. The MICE-IIIS 68000 then starts real-time emulation of the emulation processor, and immediately begins recording system status information as specified in the Trace Cycle Qualify setting. If no address is specified, emulation starts at the current program counter. Note that the green "EP RUN" indicator on the front panel is lit whenever the emulation CPU is active. The recorded information can be viewed with List Trace Buffer commands.

During emulation,

```
"Trace in progress..."
```

is displayed as long as data is being recorded in the trace buffer. When the Trigger condition is matched, the message-

```
"EV# encountered"
```

is displayed. If an attempt is made to write into Read-only area (defined as "IR" or "ER" in the MAp command (Section 5.3.8 of the main manual), the message-

```
"...Write protected memory trespassed!"
```

will display. Likewise, if a non-existent area (defined as "G" in the MAp command is accessed, the following message will display-

```
...Guarded memory trespassed!"
```

If the Run option was not specified for either the Trigger setting or Event 6, the additional message-

```
"Emulation processor stopped"
```

will also display. If no breakpoint is encountered (i.e., `"Trace in progress..."` remains on the screen), the user may terminate emulation at any time by pressing <ESC>; the message-

```
"Emulation processor stopped by user"
```

will then be displayed.

Note, however, that if several breakpoints are matched nearly simultaneously, the order in which they are listed in the `"EV# encountered"` message may not reflect the actual order in which they were encountered during emulation. This is especial-ly true when both execution and bus/signal breakpoints are matched almost simul-taneously: because of the higher priority of execution breakpoints, Events 6, 7 and 8 will be displayed before Events 1 to 5 even if they were encountered slightly later.

Examples:

1. Using the program provided at the beginning of this chapter, display register contents and then start tracing all bus activity until trigger address 40C is accessed. Use a timebase of 1 μs, with the timer set to start when emulation begins.

```
>Register<CR>
        0        1        2        3        4        5        6        7
 Dn  FFFFFFF6 FFFFFFFF 7FFDDFF7 FFFFBFF6 FFFFFFFF DFFFFFFF FFFFFFFF FFF7FFFF
 An  FFFF7FFF 00002000 7FFEFFFF EFFFFFDF DFF7FFFD F7FBFFFF FFFFF2DF 0001FFF0
 SSP=0001FFF0 USP=FFFFFFFE                SR  T S  III    XNZVC
 PC=000400                                 2714 0010011100010100
>Event 1 40C<CR>
>TRigger Level A EV1<CR>
>TRigger A<CR>
>TTimebase 1<CR>
>Go<CR>
 EV1 (Bus)  encountered
 Emulation processor stopped
 Last frame = 0008,timer = 00.000 002 0
>
```

The time that elapses before the trace buffer is filled, depends on command specifications, and may be quite lengthy. If the trigger condition has not yet been matched, you may enter an <ESC> to terminate the trace, retaining the information already recorded in the trace buffer.

2. Using the program provided at the beginning of this chapter, set the trigger for a low signal at the external trigger input port, and when this is not matched enter an <ESC> to terminate tracing.

```
>Event 5 Low<CR>
>Trigger Level A EV5<CR>
>Trigger A<CR>
>Go 400<CR>
 Trace in progress...<ESC>
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 00.528 890 0
>
```

3. Using the program provided at the beginning of this chapter, set bus breakpoint EV1, address 40C and Trigger Level A EV1. Then free run by using "Run" option. Clear the Trigger, then execute "Go" command again.

```
>Event 1 40C<CR>
>TRigger<CR>
 TRigger A  Delay 0000
 Trigger Timer On Trace On
 Level A  EV5 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
>TRigger LEvel A EV1<CR>
>Go 400<CR>
 EV1 (Bus) encountered
 Emulation processor stopped
 Last frame = 0008,timer = 00.000 002 0
>Go Run 400<CR>
 Emulation processor free running, all breakpoints masked
Run>HAlt<CR>
 Emulation processor stopped by user
```

```
>TRigger CLear<CR>
>TRigger<CR>
 Trigger Clear
 Trigger Timer On Trace On
 Level A  EV1 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
>Go 400<CR>
 EV1 (Bus) encountered
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 01.420 365 0
>
```

4.4 HALT - HAlt

HAlt

HAlt is the command to halt the emulation processor.

This command may be used during free-run emulation to stop program execution at any time. Executing such commands as Register, Cycle, Step, Memory, Input, Output, COVerage and INItialize will also halt emulation. Note that the green "EP RUN" indicator on the front panel is lit whenever the emulation processor is active.

The Halt command may also be used after tracing has been stopped by the specified trigger condition, and the emulation processor is in free run state due to a "Run" qualifier in the trigger setting or in the breakpoint setting for Event 6.

Example: Using the program provided at the beginning of this chapter, input the Halt command during emulation to stop the emulation processor.

```
>Event 1 40C<CR>
>Go Run<CR>
 Emulation processor free running, all breakpoints masked
Run>HAlt<CR>
 Emulation processor stopped by user
>
```

4.5 COMMAND HELP - Help

```
Help [GRoup {Emulation|Memory|Port|Setup|Trace}|command|ALl]
```

Help is the keyword for Command Help. "Help" alone, without any para-meter, causes the MICE to display a list of command keywords arranged in functional groups.

GRoup is the prefix to use to display a specific functional group of commands. When followed by the desired group's name, the syntax for all the commands in that particular group will display. Definitions of typical parameters used in such group, will also appear on screen. MICE commands are divided into the following groups:

Setup Commands	(see Section 3.2.1 of this Addendum)
Memory Commands	(see Section 3.2.2 of this Addendum)
Port Commands	(see Section 3.2.3 of this Addendum)
Emulation Commands	(see Section 3.2.4 of this Addendum)
Trace Commands	(see Section 3.2.5 of this Addendum)

command is any of the specific MICE command keywords. Keying "Help" followed by a specific command keyword, will display the full syntax for that particular command and the definitions for the typical parameters used by such command.

ALl is the parameter to display full syntax and typical parameter definitions for all available MICE-IIIS 68000 commands.

In order to make efficient use of screen space, special conventions of notation are used in Command Help displays. These conventions are explained in Section 4.1 of the main manual.

Examples:

1. Display a command overview.

```
>Help<CR>
  Setup Group:
   BAud              IDentify          RECall            SETup
   CLock             INTerval          ROute             SIze
   CONtrol           MAp               SAve              Verify
   HANdshake         Prompt            SELect            WAit
   Help              REAdy
  Memory Group:
   Assemble          COMpare           Fill              SEarch
   Byte              COpy              LOng              TEst
   CHecksum          Disassemble       Memory            Word
  Port Group:
   Input             Output
  Emulation Group:
   Cycle             Register          RESet             Step
   Jump
  Trace Group:
   COVerage          HAlt              Qualify           TRAce
   Event             INItialize        SYnc              Trigger
   Go                List              TImebase
>
```

2. Display full syntax for all MICE-IIIS 68000 commands. (When the message "[More]" appears, press any key to scroll for the next screen.)

```
>Help ALl<CR>
Line Assembly                 Assemble [space-adr]
Port B Baud rate Setting      BAud [baud-rate [parity] [data-bits]]
Memory Modify                 Byte [space-adr [data]]
Memory Checksum               CHecksum space-adr {adr|Length length}
Display Clock Source          CLock
Memory Compare                COMpare space-adr1 {adr|Length length} space-adr2
Control Signal En/Disable     CONtrol [[Berr] [Intr] [BR] {Enable|Disable}]
Memory Copy                   COpy space-adr1 {adr|Length length} space-adr2
Set Coverage                  COVerage [adr1 adr2[ CLear]|[ENable|DIsable|CLear|
                                       Reset]]
                              COVerage List [adr]
                              COVerage SP|UP|SD|UD
Cycle Step                    Cycle [Wait|count]
Disassembly                   Disassemble [space-adr [adr|Length length]]
```

```
 Set Bus Event 1            Ev[ent ]1 {Range adr1 adr2|adx} [datum] [status]
                                      [EXternal trace-bits] [COunt count]
[ More ]
 Display/Clear Event 1      Ev[ent ]1 [CLear]
 Set Bus Event 2            Ev[ent ]2 {Range adr1 adr2|adx} [datum] [status]
                                      [EXternal trace-bits] [COunt count]
 Display/Clear Event 2      Ev[ent ]2 [CLear]
 Set Bus Event 3            Ev[ent ]3 {Range adr1 adr2|adx} [datum] [status]
                                      [EXternal trace-bits] [COunt count]
 Display/Clear Event 3      Ev[ent ]3 [CLear]
 Set Bus Event 4            Ev[ent ]4 {Range adr1 adr2|adx} [datum] [status]
                                      [EXternal trace-bits] [COunt count]
 Display/Clear Event 4      Ev[ent ]4 [CLear]
 Set External Trigger       Ev[ent ]5 [COnditional] {High|Low}
 Display/Clear Ext-Trigger  Ev[ent ]5 [CLear]
 Set Execution Event 1      Ev[ent ]6 adr [fetch-state]
 Display/Clear Exe-Event 1  Ev[ent ]6 [CLear]
 Set Execution Event 2      Ev[ent ]7 adr [fetch-state]
 Display/Clear Exe-Event 2  Ev[ent ]7 [CLear]
[ More ]
 Set Execution Event 3      Ev[ent ]8 adr [fetch-state]
 Display/Clear Exe-Event 3  Ev[ent ]8 [CLear]
 Memory Fill                Fill space-adr {adr|Length length} data
 Execution                  Go [Run] [adr]
 Halt Emulator              HAlt
 Set Handshaking Code       HANdshake [code]
 Command Help               Help [GRoup {Emulation|Memory|Port|Setup|Trace}
                                 |command|ALl]
 Identify Emulator          IDentify
 Set Initialize             INItialize [adr1 adr2[ CLear]|[ENable|DIsable|
                                       CLear|Reset]]
                            INItialize List [adr]
                            INItialize SP|UP|SD|UD
 Port Input                 Input space-adr [Length in-length]
 Timer Interval Switch      INTerval [On|OFf]
 Jump                       Jump adr
[ More ]
 List Trace                 List [frame [adr1 adr2] [status]
                                 [EXternal trace-bits]]
 List Source Code           List {Source|Instruction} [frame]
 List Frame Total           List Number
 Memory Modify              LOng [space-adr [data]]
 Memory Map                 MAp [space-adr adr {{I|IR} {1|2|3|4}|E|ER|G}
                                | ALl{E|G}]
 Memory Dump                Memory [space-adr [adr|Length length]]
 Port Output                Output space-adr data
```

```
Change Prompt                Prompt [string]
Set Trace Cycle Qualifier    Qualify {1|2} [{Range adr1 adr2|adx}
                                           [status [EXternal trace-bits]]
Display/Clear/Disable        Qualify [1|2] [CLear|DIsable|ENable]
Enable Qualifier
Ready Signal Selection       REAdy [Internal|External]
Recall Status from NOVRAM    RECall
[ More ]
Register Display/Modify      Register [register]
Reset Emulation Processor    RESet [pc [ssp]]
Change Operational Mode      ROute [System]
Save Status to NOVRAM        SAve
Memory Search                SEarch space-adr {adr|Length length} data
Select leading code          SELect [{Route|DEVice} [code]]
Display Setup Parameters     SETup
Memory Access Size           SIze [Byte|Word]
Instruction Step             Step [adr1 adr2 [CALl]|count|CALl]
Synchronization              SYnc [{Input|OUtput} {On|OFf}]
Memory Test                  TEst space-adr {adr|Length length}
Timebase Selection           TImebase [0.1|1|10|100|1000]
Display Trace Parameters     TRAce
Set Trigger Level            Trigger Level level# [Not] ev# [{OR|ANd} [Not] ev#
                                  [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev#
                                  [{OR|ANd} [Not] ev# ]]]]
[ More ]
                                  [Trace {On|OFf}] [TImer {On|OFf}]
Display Trigger Level/       Trigger Level level# [CLear]
Clear Trigger Level
Set Trigger                  Trigger [RUn] {level# [<THen level#>] [Reset |
                             If state# <ELse state#>]|If state# <ELse state#>}
                             [FOreward|BACKward|CENter|DELay count]
Set Initial Timer & Trace    Trigger [TImer {On|OFf}] [Trace {On|OFf}]
Display/Clear Trigger        Trigger [CLear]
Memory Verification          Verify [On|OFf]
Insert Wait State            WAit [On|OFf]
Memory Modify                Word [space-adr [data]]
```

---Definition----------

adr	is a hexadecimal address setting.
adr1	is a hexadecimal address indicating the starting point of the memory block.
[More]	
adr2	is a hexadecimal address indicating the end point of the memory block.
adx	is a hexadecimal address setting, with option to include wildcard nibbles/bits, e.g. 1234, 1XX4, 1(X1X0)X4, 1(X1X0X0XX)4.

ALl — is the parameter to Display full syntax for all available commands.
BACKward — is the delay count = 0.
baud-rate — is baud rate of serial channel B (110 150 300 600 1200 2400 4800 9600 19200).
CALl — is the keyword for skipping subroutines.
CENter — is the delay count = FFFH/3FFFH for 8K/32K trace buffer.
CLear — is the parameter to clear a particular setting or clear all of them at once.
code — is a one-byte hexadecimal Value.
command — is a specific MICE command.
COnditional — the signal is only valid during the condition of low state
[More]
STROBE.
count — is a hexadecimal Value (from 0H to 0FFFFH).
data — is 1 to 32 bytes of data in hex and/or ASCII.
data-bits — is either 7 or 8 data bits of serial channel B.
datum — is a byte or word in hexadecimal, with wildcard nibbles/bits option, e.g. 1234, 1XX4, 1(X1X0)X4, 1(X1X0X0XX)4.
DIsable — is the parameter to disable the settings.
ENable — is the parameter to enable the settings.
ev# — is EV1, EV2, EV3, EV4 or EV5 (do not specify same event twice in one command).
fetch-state — is 1 to 2 processor status codes (SP UP).
FOrward — is the delay count = 1FFFH/7FFFH for 8K/32K trace buffer.
frame — is a hexadecimal Value (from 0H to 1FFFH/7FFFH for 8K/32K) indicating the frameat which the listing is to begin.
Instruction — is the parameter to list the trace buffer containing frame, address, data and assembly source-code, but do not include
[More]
status, trace-bits, timer and read/write cycle.
in-length — is a hexadecimal Value (from 01H to 0FFFFFFH).
length — is a hexadecimal Value (from 01H to 0FFFFFFH) specifying the number of location to access.
level# — is one of level identifiers -- A, B, C, D, E, F, G or H.(do not specify same level twice in one command.
List — is the parameter to list the test result.
parity — is E (even), O (odd) or N (none) parity of serial channel B.
pc — is a hexadecimal address to load in place of the current PC.
Range — is the required prefix when specifying a address range.
register — is the standard Motorola designation for the register, the contents of which are to be displayed or modified.
Reset — initializes the code-coverage test or Read-Before-Write flag, but the range setting will not be cleared.
Source — is the parameter to list the trace buffer containing frame, address, data, status, trace-bits, timer and assembly

```
[ More ]
                source-code.
 space-adr      is a hexadecimal address setting, with option to specify
                memory type: "[UP|UD|SP|SD] adr".
 SP|SD|UP|UD    is one to four two-letter codes specifying memory type.
                Where : SP is the supervisor program
                        SD is the supervisor data
                        UP is the user program
                        UD is the user data
                If none of the memory type is specified in this command,all four
                memory types will default.
 ssp            is a hexadecimal address to load in place of the current SSP.
 state#         level# [RESet] | (level# {<Then level#> [ RESet |
                If state# <ELse state#> ] | If state# <ELse state#> } )
 <    >         Recursive (Minimum 1 time).
 (    )         Reserved Symbols.
 status         is 1 to 7 processor status codes (SP SR SW UP UR UW AK).
[ More ]
 string         is a 1-8 character command prompt, input in ASCII.
 System         is the path of operational mode from Terminal to System mode.
 trace-bits     is a one-byte hexadecimal setting indicating signal levels for
                trace bits input from the external port. Includes option to
                specify wildcard nibbles/bits.
 Wait           stops the CPU in a wait state.

 Note : Uppercase character(s) in the command keyword and command parameters
        represent the minimum abbreviated (shorthand) input that would remain
        transparent to MICE. Lowercase character(s) are optional. A space
        should be entered between a command keyword and a subsequent parameter,
        as well as between parameters.
>
```

3. Display a single command syntax.

```
>Help BAud<CR>
 Port B Baud rate Setting   BAud [baud-rate [parity] [data-bits]]

 --- Definition ----------------------------------------------------
 baud-rate    is baud rate of serial channel B (110 150 300 600 1200
              2400 4800 9600 19200).
 data-bits    is either 7 or 8 data bits of serial channel B.
 parity       is E (even), O (odd) or N (none) parity of serial
              channel B.
>
```

4. Display the "Setup" group command syntax.

```
>Help GRoup Setup<CR>
 Port B Baud rate Setting   BAud [baud-rate [parity] [data-bits]]
 Display Clock Source       CLock
 Control Signal En/Disable  CONtrol [[Berr] [Intr] [BR] {Enable
                                 |Disable}]
 Set Handshaking Code       HANdshake [code]
 Command Help               Help [GRoup {Emulation|Memory|Port
                            |Setup|Trace}|command|ALl]
 Identify Emulator          IDentify
 Timer Interval Switch      INTerval [On|OFf]
 Memory Map                 MAp [space-adr adr {{I|IR} {1|2|3|4|}
                                 |E|ER|G}|ALl {E|G}]
 Change Prompt              Prompt string
 Ready Signal Selection     REAdy [Internal|External]
 Recall Status from NOVRAM  RECall
 Change Operational Mode    ROute [System]
 Save Status to NOVRAM      SAve
 Select leading code        SELect [{Route|DEVice} [code]]
[ More ]
 Display Setup Parameters   SETup
 Memory Access Size         SIze [Byte|Word]
 Memory Verification        Verify [On|OFf]
 Insert Wait State          WAit [On|OFf]

 --- Definition ----------------------------------------------------
 ALl          is the parameter to display full syntax for all
              available commands.
 baud-rate    is baud rate of serial channel B (110 150 300 600 1200
              2400 4800 9600 19200).
 code         is a one-byte hexadecimal value.
```

```
 command      is a specific MICE command.
[ More ]
 data-bits    is either 7 or 8 data bits of serial channel B.
 parity       is E (even), O (odd) or N (none) parity of serial
              channel B.
 string       is a 1-8 character command prompt, input in ASCII.
 System       is the path of operational mode from Terminal to
              System mode.
>
```

5. Display the "Register" command syntax.

```
>Help Register<CR>
 Register Display/Modify    Register [register]

 --- Definition ----------------------------------------------------
 register     is the standard Motorola designation for the
              register, the contents of which are to be displayed
              or modified.
>
```

4.6 BREAK ON READ BEFORE WRITE - INItialize

```
INItialize [adr1 adr2[ CLear]][[ENable|DIsable|CLear|Reset]]
INItialize List [adr]
INItialize SP|UP|SD|UD
```

INItialize is the command to set, display or clear "INItialize" or "Read-Before-Write" function.

adr1 is the hexadecimal address specifying the starting point of a memory block.

adr2 is the hexadecimal address indicating the last point of a memory block.

CLear is used to clear a particular range of address or to clear all existing address ranges.

ENable is a global control to enable all of the address settings.

NOTE

If the COVerage (Section 4.1 of this Addendum) command is active at the time when the INItialize command is enabled, the COVerage command is automatically switch to inactive condition. Likewise, INItialize function is disabled when COVerage command is enabled while INItialize is active. Hence only one command (INItialize or COVerage) is active at a time.

DIsable is a global control to disable, but without clearing the settings.

Reset re-initializes the Read-Before-Write function without clearing the address ranges setting.

List lists the address ranges which were actually written after running emulation.

Adr is the hexadecimal address specifying the starting point of the address ranges to be listed.

SP|UP|SD|UD are the four codes specifying memory type to be executed with the Read-Before-Write function.

Where: SP is the supervisor program
SD is the supervisor data
UP is the user program
UD is the user data

If none is specified, all four memory types will default.

INItialize command is used to ensure that only memory range which were previously written, can be read during emulation, otherwise the "Read-Before-Write" break will be encountered. This function is most helpful when emulating such areas as vector space, input/output, RAM, etc. The written areas may be displayed with the INItialized List command. Whenever processor tries to read an unwritten area, emulation is aborted and the following message will display-

```
"Read-Before-Write Encountered"
```

If INItialize is invoked while COVerage is active, INItialize is ignored and the following message will display:

```
"Read-Before-Write is globally disabled"
```

User should therefore enable the INItialize command first, before invoking INItialize. COVerage is automatically disabled whenever INItialize is enabled if the former is active. Both commands are disabled by default on MICE power up.

Up to 20 initialized ranges of memory may be set in four independent 256K-byte of memory segments for this function.

NOTE

The command is effective only in real-time emulation. It will not work under Cycle Step or Instruction Step commands.

Examples:

1. Using the program provided at the beginning of this chapter, enable "INItialize" function. Set ranges and try to read an unwritten address. List the written areas after running.

```
>INItialize Enable<CR>
>INItialize 1000 1200<CR>
>INItialize 2000 2100<CR>
>INItialize 3000 3100<CR>
>INItialize<CR>
 The ranges for Read-Before-Write breakpoint :
 Memory type : SP,SD,UP,UD
  001000 001200
  002000 002100
  003000 003100
>Go 500<CR>
 Read-Before-Write encountered
 Emulation processor stopped
 Last frame = 0005,timer = 00.000 000 7
```

```
>List<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE   EVENT TIMER
 0000 000500  327C   SP     00000000   111    11111 00.000 000 0
 0001 000502  1000   SP     00000000   000    00000 00.000 000 0
 0002 000504  3019   SP     00000000   000    00000 00.000 000 2
 0003 000506  B07C   SP     00000000   000    00000 00.000 000 5
 0004 001000  EFEE   SR     00000000   000    00000 00.000 000 7
 0005 000508  55AA   SP     00000000   000    00000 00.000 000 7
>INItialize List<CR>
 The memory ranges of Read-Before-Write have been written :
 None
>
```

2. Using the program provided at the beginning of this chapter, try to write an address before reading an "INItialized" ranges. List the written areas after running.

```
>INItialize<CR>
 The ranges for Read-Before-Write breakpoint :
 Memory type : SP,SD,UP,UD
  001000 0010FF
  002000 0020FF
  003000 003100
>Go 400<CR>
 Trace in progress...
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 05.018 640 2
>Event 6 512
>Go 500<CR>
 Trace in progress...
 EV6 (Execution) encountered
 Emulation processor stopped
 Last frame = 4802,timer = 00.005 633 5
>INItialize List<CR>
 The memory ranges of Read-Before-Write have been written :
 001000 0010FF
 >
```

3. Using the program provided at the beginning of this chapter, clear a range of the "INItialized" settings and list the written area.

```
>INItialize<CR>
 The ranges for Read-Before-Write breakpoint :
 Memory type : SP,SD,UP,UD
  001000 0010FF
  002000 0020FF
  003000 003100
>INItialize_List<CR>
 The memory ranges of Read-Before-Write have been written :
 001000 0010FF
>INItialize_1000 10FF CLear<CR>
>INItialize_List<CR>
 The memory ranges of Read-Before-Write have been written :
 None
>
```

4. Using the program provided at the beginning of this chapter, reset the "INItialized" sets, qualify the space SD and list the written areas before and after emulation.

```
>INItialize_Reset<CR>
>INItialize<CR>
 The ranges for Read-Before-Write breakpoint :
 Memory type : SP,SD,UP,UD
  001000 001200
  002000 0020FF
  003000 003100
>INItialize_SD<CR>
>INItialize_List<CR>
 The memory ranges of Read-Before-Write have been written :
 None
>Go_400<CR>
 Trace in progress...
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 04.055 596 4
>Go_500<CR>
 Trace in progress...
 EV6 (Execution) encountered
 Emulation processor stopped
 Last frame = 4802,timer = 00.005 633 6
>INItialize_List<CR>
 The memory ranges of Read-Before-Write have been written :
 001000 001200
>
```

5. Using the program provided at the beginning of this chapter, reset and disable the "INItialize" function. Then list the written areas after emulation.

```
>INItialize Reset<CR>
>INItialize DIsable<CR>
>Go 400<CR>
 Trace in progress...
 Emulation processor stopped by user
 Last frame = 7FFF,timer = 07.024 674 8
>Go 500<CR>
 Trace in progress...
 EV6 (Execution) encountered
 Emulation processor stopped
 Last frame = 4802,timer = 00.005 633 5
>INItialize List<CR>
 Read-Before-Write breakpoint globally disabled.
>INItialize ENable<CR>
>INItialize List<CR>
 The memory ranges of Read-Before-Write have been written :
 None
>
```

6. Clear the "INItialize" function.

```
>INItialize CLear<CR>
>
```

4.7 LIST TRACE BUFFER - List

List Trace Results	List [frame [adr1 adr2] [status] [External trace-bits]]
List Trace Results with Source/Instruction Code	List {Source|Instruction} [frame]
List Total Frames and Time	List Number

List is the command to display information recorded in the trace buffer. This information is organized into frames, each representing one execution bus cycle, and displayed one page (screenful) at a time. Address and data bus activity, processor status, trace bit levels and trigger information (STATE and EVENT column) are displayed for each frame. "List" alone, or "List Source" with no other parameters, displays all frames in the last page of the buffer.

frame is a hexadecimal value (from 0H to 7FFFH with 32K trace buffer) indicating the frame at which the listing is to begin. The default is to begin at the first frame in the last page of the buffer.

adr1, adr2 are hexadecimal values specifying a particular range of memory, and indicating that only those cycles with address bus activity within that range, are to be listed. "adr1" indicates the starting address, "adr2" the end address. Default is 0H to FFFFFFH.

status is 1 to 7 processor status codes (SP SR SW UP UR UW AK) representing specific type of processor activity (see Table 5-3 of the main manual) and indicating that only cycles in which the indicated type of activity take place are to be listed. The default is for all cycles regardless of processor status.

EXternal is the required prefix when specifying trace-bits.

trace-bits is a one-byte hexadecimal setting indicating signal levels for trace bits input from the external TRACE BITS port specifying that only cycles in which the indicated levels are matched, are to be listed. This setting includes option to specify wildcard nibbles/bits e.g., "XX" "FX" "F(10XX)" etc., where "X" is a nibble or bit indicating 'don't care'. The default setting is for all cycles regardless of trace bit levels.

Source is the parameter to list the trace buffer containing frame, address, data, status, trace-bits, state, event and assembly source code. Note that the source code listing must start at an instruction fetch cycle, otherwise incorrect code will be displayed for the first few machine cycle.

Instruction is same with "Source" parameter above. Only the trace buffer contents to be listed will not include status, trace-bits, state, event and read/write cycle.

Number displays the number of the last frame recorded in the trace buffer and the overall elapsed time since timer start.

Trace information recorded during emulation is displayed using the List command. If a frame number is specified, the MICE lists the trace buffer starting at the specified frame. If no frame number is specified, the MICE lists all frames in the last page of the buffer. Qualifiers for address range, processor activity and/or external signal levels can optionally be entered as described above so that only cycles in which the specified condition is matched are listed.

One page (i.e., one screenful) of trace information is displayed at a time; press <CR> to display the next page and <SP> for continues scrolling. To terminate the List operation press <ESC>. Note that the command also ends if the trace buffer is empty or a frame number higher than that of the last recorded frame is specified.

Example: Using the program provided at the beginning of this chapter, run a trace up to address 40E and list the last page in the trace buffer.

```
>Disassemble 400 40E<CR>
LOC         OBJ              SOURCE CODE
000400      327C 1000        MOVEA.W   #1000,A1
000404      32FC 55AA        MOVE.W    #55AA,(A1)+
000408      B2FC 2000        CMPA.W    #2000,A1
00040C      66F6             BNE.B     000404
00040E      4EF8 040E        JMP       00040E
 Disassembly completed
>Event 6 40E<CR>
>Go 400<CR>
 Trace in progress...
 EV6 (Execution) encountered
 Emulation processor stopped
 Last frame = 3802,timer = 00.004 097 5
>List<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE   EVENT TIMER
 37F3 00040E  A0F8   SP     00000000   000    00000 00.004 091 7
 37F4 000404  32FC   SP     00000000   000    00000 00.004 091 9
 37F5 000406  55AA   SP     00000000   000    00000 00.004 092 4
 37F6 000408  B2FC   SP     00000000   000    00000 00.004 092 7
 37F7 001FFC  55AA   SW     00000000   000    00000 00.004 092 9
 37F8 00040A  2000   SP     00000000   000    00000 00.004 093 2
 37F9 00040C  66F6   SP     00000000   000    00000 00.004 093 4
 37FA 00040E  A0F8   SP     00000000   000    00000 00.004 093 7
 37FB 000404  32FC   SP     00000000   000    00000 00.004 093 9
 37FC 000406  55AA   SP     00000000   000    00000 00.004 094 4
 37FD 000408  B2FC   SP     00000000   000    00000 00.004 094 7
 37FE 001FFE  55AA   SW     00000000   000    00000 00.004 094 9
 37FF 00040A  2000   SP     00000000   000    00000 00.004 095 2
 3800 00040C  66F6   SP     00000000   000    00000 00.004 095 4
 3801 00040E  A0F8   SP     00000000   000    00000 00.004 095 7
 3802 000410  040E   SP     00000000   000    00000 00.004 095 9
>
```

4.7.1 List Trace Results - List

List [frame [adr1 adr2] [status] [EXternal trace-bits]]

Examples:

1. After the trace operation in the previous example, list all frames in the first page of the trace buffer.

```
>List 0<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE    EVENT TIMER
 0000 000400   3204   SP     00000000   111    11111 00.000 000 0
 0001 000402   1204   SP     00000010   000    00010 00.000 000 0
 0002 000404   3604   SP     00000100   000    00100 00.000 000 2
 0003 000406   570E   SP     00000110   000    00110 00.000 000 4
 0004 000408   3A04   SP     00001000   000    01000 00.000 000 7
 0005 001000   551A   SW     00000000   000    00000 00.000 000 9
 0006 00040A   2A04   SP     00001010   000    01010 00.000 001 2
 0007 00040C   6E06   SP     00001100   000    01100 00.000 001 4
 0008 00040E   AE04   SP     00001110   000    01110 00.000 001 7
 0009 000404   3604   SP     00000100   000    00100 00.000 001 9
 000A 000406   570E   SP     00000110   000    00110 00.000 002 4
 000B 000408   B2FC   SP     00000000   000    00000 00.000 002 7
 000C 001002   55AA   SW     00000000   000    00000 00.000 002 9
 000D 00040A   2000   SP     00000000   000    00000 00.000 003 2
 000E 00040C   66F6   SP     00000000   000    00000 00.000 003 4
 000F 00040E   A0F8   SP     00000000   000    00000 00.000 003 7
[ More ]<ESC>
>
```

2. List the trace buffer starting at frame 0, and limit the listing to supervisor data memory write (SW) cycles in the memory range of 1200H to 120FH.

```
>List 0 1200 120F SW<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE    EVENT TIMER
 0705 001200   55AA   SW     00000000   000    00000 00.000 512 9
 070C 001202   55AA   SW     00000000   000    00000 00.000 514 9
 0713 001204   55AA   SW     00000000   000    00000 00.000 516 9
 071A 001206   55AA   SW     00000000   000    00000 00.000 518 9
 0721 001208   55AA   SW     00000000   000    00000 00.000 520 9
 0728 00120A   55AA   SW     00000000   000    00000 00.000 522 9
 072F 00120C   55AA   SW     00000000   000    00000 00.000 524 9
 0736 00120E   55AA   SW     00000000   000    00000 00.000 526 9
>
```

3. List all frames in the trace buffer after the execution operation in Go command (Section 4.3, Example 7 in this Addendum).

```
>Disassemble 400 418
LOC         OBJ              SOURCE CODE
000400      303C 0002        MOVE.W    #0002,D0
000404      907C 0001        SUB.W     #0001,D0
000408      0C40 0000        CMPI.W    #0000,D0
00040C      6706             BEQ.B     000414
00040E      4EB8 0510        JSR       000510
000412      60F0             BRA.B     000404
000414      4EB8 0520        JSR       000520
000418      4E71             NOP
 Disassembly completed
>Fill 500 5FF 4E 75
>Event CLear
>Event 1 404
>Event 2 510
>Event 3 520
>Event 4 418
>Trigger TImer On
>Trigger Level A EV1 TImer OFf
>Trigger Level B EV2 TImer On
>Trigger Level C EV3 TImer On
>Trigger Level D EV4 TImer On
>Trigger A If B Reset ELse ( C THen D )
>Go 400
 EV4 (Bus) encountered
 Emulation processor stopped
 Last frame = 0020,timer = 00.000 004 0
>List 0<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE   EVENT TIMER
 0000 000400  303C   SP     00000000   011    10000 00.000 000 0
 0001 000402  0002   SP     00000000   000    00000 00.000 000 0
 0002 000404  907C   SP     00000000   000    00000 00.000 000 2
 0003 000406  0001   SP     00000000   000    00001 00.000 000 4
 0004 000408  0C40   SP     00000000   001    00000 00.000 000 4
 0005 00040A  0000   SP     00000000   001    00000 00.000 000 4
 0006 00040C  6706   SP     00000000   001    00000 00.000 000 4
 0007 00040E  4EB8   SP     00000000   001    00000 00.000 000 4
 0008 000410  0510   SP     00000000   001    00000 00.000 000 4
 0009 000510  4E75   SP     00000000   001    00000 00.000 000 4
 000A 01FFEC  0000   SW     00000000   001    00010 00.000 000 4
 000B 01FFEE  0412   SW     00000000   000    00000 00.000 000 7
 000C 000512  4E75   SP     00000000   000    00000 00.000 000 9
 000D 01FFEC  0000   SR     00000000   000    00000 00.000 001 2
 000E 01FFEE  0412   SR     00000000   000    00000 00.000 001 4
 000F 000412  60F0   SP     00000000   000    00000 00.000 001 7
[ More ]
```

```
 0010 000414   4EB8    SP     00000000    000     00000 00.000 001 9
 0011 000404   907C    SP     00000000    000     00000 00.000 002 2
 0012 000406   0001    SP     00000000    000     00001 00.000 002 5
 0013 000408   0C40    SP     00000000    001     00000 00.000 002 5
 0014 00040A   0000    SP     00000000    001     00000 00.000 002 5
 0015 00040C   6706    SP     00000000    001     00000 00.000 002 5
 0016 00040E   4EB8    SP     00000000    001     00000 00.000 002 5
 0017 000414   4EB8    SP     00000000    001     00000 00.000 002 5
 0018 000416   0520    SP     00000000    001     00000 00.000 002 5
 0019 000520   4E75    SP     00000000    001     00000 00.000 002 5
 001A 01FFEC   0000    SW     00000000    001     00100 00.000 002 5
 001B 01FFEE   0418    SW     00000000    010     00000 00.000 002 8
 001C 000522   4E75    SP     00000000    010     00000 00.000 003 0
 001D 01FFEC   0000    SR     00000000    010     00000 00.000 003 3
 001E 01FFEE   0418    SR     00000000    010     00000 00.000 003 5
 001F 000418   4E71    SP     00000000    010     00000 00.000 003 8
[ More ]
 0020 00041A   4E71    SP     00000000    010     01000 00.000 004 0
>
```

The columns "STATE" and "EVENT" denotes the trigger state and event information respectively. Each bit is defined as follows:

STATE Bits 2 - 0 is the trigger state 0 - 7

EVENT Bit 4 is active when Event 5 is matched ("Event encounter-flag" of EV5)
Bit 3 is active when Event 4 is matched ("Event encounter-flag" of EV4).
Bit 2 is active when Event 3 is matched ("Event encounter-flag" of EV3).
Bit 1 is active when Event 2 is matched ("Event encounter-flag" of EV2).
Bit 0 is active when Event 1 is matched ("Event encounter-flag" of EV1).

Note that bits 4-0 remain in effect until the state changes and the state value of frame 0 is in random. In the above example, the sequence occurs- as follows:

Event 1 (address 404H) is matched, it changes to State 1 and timer turned OFF.
Event 2 (address 510H) is matched, it changes to State 0 and timer turned ON.
Event 1 (address 404H) is matched, it changes to State 1 and timer turned OFF.
Event 3 (address 520H) is matched, it changes to State 2 and timer turned ON.
Event 4 (address 418H) is matched, emulation will then finally stop.

4.7.2 LIST TRACE RESULTS WITH SOURCE/INSTRUCTION CODE - List

List {Source|Instruction} [frame]

Examples:

1. Starting at frame 00H, list the address, traced data, CPU status, trace bits, state and event, displaying mnemonic code together with machine activity.

```
>List Source 0<CR>
FRAME ADDRESS DATA STATUS TRACE BITS STATE   EVENT TIMER
 0000 000400  303C   SP     00000000   011    10000 00.000 000 0
          MOVE.W   #0002,D0
 0001 000402  0002   SP     00000000   000    00000 00.000 000 0
 0002 000404  907C   SP     00000000   000    00000 00.000 000 2
          SUB.W    #0001,D0
 0003 000406  0001   SP     00000000   000    00001 00.000 000 4
 0004 000408  0C40   SP     00000000   001    00000 00.000 000 4
          CMPI.W   #0000,D0
 0005 00040A  0000   SP     00000000   001    00000 00.000 000 4
 0006 00040C  6706   SP     00000000   001    00000 00.000 000 4
          BEQ.B    000414
 0007 00040E  4EB8   SP     00000000   001    00000 00.000 000 4
          JSR      000510
 0008 000410  0510   SP     00000000   001    00000 00.000 000 4
 0009 000510  4E75   SP     00000000   001    00000 00.000 000 4
          RTS
[ More ]<ESC>
>
```

2. Starting at frame 00H, list the address, traced data and assembly source code, displaying mnemonic code together with machine activity.

```
>List_Instruction_0<CR>
FRAME   ADDRESS      DATA      SOURCE-CODE
 0000   00000B00     2E7C      MOVEA.L   #00006000,A7
 0002   00000B06     207C      MOVEA.L   0002FFF0,AO
 0006   00000B0C     4E60      MOVE.L    00,USP
 0007   00000B0E     4EF9      JMP       00020412
 000A   00000410     4E71      NOP
 000B   00000412     207C      MOVEA.L   #00020700,AO
 000E   00000418     227C      MOVEA.L   #00020600,A1
 0010   0000041E     2018      MOVE.L    (AO)+,DO
 0012   00000420     2280      MOVE.L    DO,(A1)
 0013   00000422     2219      MOVE.L    (A1)+D1
 0016   00000424     B280      CMP.L     D0,D1
 0017   00000426     6100      BSR,WL    00000C00
 001E   00000C00     670A      BEQ,B     00000C0C
 0024   00000C0C     4E75      RTS
 002A   00000428     07D8      BSET.B    D3,(AO)+
[ More ]<ESC>
>
```

NOTE

1. *When an instruction is fulfilled while executing a series of program control instructions, only the first instruction (NOT necessarily the fulfilled instruction) will display.*

2. *When displaying mnemonic code with the List Source command, the following conditions may cause incorrect information to be listed for the initial frame:*

   a) *The machine cycle recorded in the initial frame is not a fetch cycle for the beginning of an instruction.*

   b) *The specified starting frame contains prefetch instructions that were not executed due to a program branch.*

4.7.3 LIST TOTAL FRAMES AND TIME - List Number

List Number

The MICE-IIIS 68000 displays "Last frame = " followed by the number of the last frame recorded in the trace buffer, and "Timer = " followed by the total elapsed time since timer started. (Note that the frame count begins at zero, so the total number of frames is one more than the value displayed. If

"Last frame = 7FFFH" (for 32K trace buffer)

is displayed, the trace buffer is full.)

Example:

1. Display the number of the last frame in the trace buffer and the total elapsed time since timer start.

```
>List Number<CR>
 Last frame = 0018, Timer = 00.000 006 7
>
```

4.8 CYCLE QUALIFY - Qualify

```
Qualify {1|2} [{Range adr1 adr2|adx} [status] [EXternal trace-bits]]
Qualify [1|2] [CLear|DIsable|ENable]
```

Quality displays, sets, enables, disables or clears the cycle qualifier, which specifies which machine cycles are to be recorded in the trace buffer. "Qualify" alone, without any parameters, displays the current Cycle Qualify setting.

1|2 are the two available Qualifiers under LAM-IIS which perform "OR" function.

Range is the required prefix when specifying an address range.

adr1 is a hexadecimal address indicating the starting address.

adr2 is a hexadecimal address indicating the end address.

adx is a hexadecimal address setting with option to include wildcard nibbles/bits, e.g. 1234, 1XX4, 1(X1X0)X4, 1(X1X0X0XX)4.

status is 1 to 7 processor status codes (SP SR SW UP UR UW AK AW) indicating the type of processor activity to be recorded.

EXternal is the required prefix when specifying trace-bits.

trace-bits is a 1-byte hexadecimal setting for trace bits.

CLear is the parameter to clear the Cycle Qualify setting.

The power-on default setting of the MICE-IIIS 68000 is to record all machine cycles during a trace operation for either Qualify 1 or Qualify 2. Cycle Qualify is used to limit recording to cycles in which certain areas of program memory are accessed and/or certain types of processor activity take place. This makes it possible to record more useful information in the trace buffer with fewer trace commands.

Examples:

1. Set Cycle Qualify 1 for all SP and SR cycles at address 82000, 82010...82070; and cycle Qualify 2 for SW cycles between 92000...92700 and trace bits with wildcard; then display the current setting.

```
>Qualify 1 820(0XXX)0 SP SR<CR>
>Qualify 2 Range 92000 92700 SW EXternal 4(XX10)<CR>
>Qualify<CR>
Cycle qualifier 1 : 0820(0XXX)0 SP,SR External XX  Enable
Cycle qualifier 2 : Range 092000 092700 SW External 4(XX10) Enable
>
```

2. Disable cycle Qualify 2, set cycle Qualify 1 for UW cycle at all address, then display the setting.

```
>Qualify ENable<CR>
>Qualify 2 Disable<CR>
>Qualify 1 X UW<CR>
>Qualify<CR>
 Cycle qualifier 1 : XXXXXX UW External XX Enable
 Cycle qualifier 2 : Range 092000 092700 SW EXternal 4(XX10) Disable
>
```

3. Clear the current Cycle Qualify setting.

```
>Qualify CLear<CR>
>Qualify<CR>
 Cycle qualifier 1 : All address and status Enable
 Cycle qualifier 2 : All address and status Disable
>
```

4.9 TIMEBASE SELECTION - TImebase

TImebase [0.1|1|10|100|1000]

TImebase is the command for timebase selection.

0.1...1000 specifies the Timebase selection as indicated in Table 4-1 below.

A timer is provided for trace mode. Execution time is displayed in the Go command (Section 4.3 of this Addendum), with the unit of measurement determined by the Timebase selection. The default is for a timebase of 0.1 μs, or the most recent user setting. The timer is activated when emulation starts, unless the timer OFf option is specified in the Trigger command.

Timebase Number	1	2	3	4	5
Unit of Measurement	0.1μs	1μs	10μs	100μs	1000μs
Maximum Timer Length	7min 9sec	1hr 11min	11hrs 55min	4days 23hrs	49days 17hrs

Table 4-1 LAM-IIS Timebase Selections

Example: Set the timebase to 0.1 μs when emulation starts; then display the current Timebase setting to verify input.

```
>TImebase 0.1<CR>
>TImebase<CR>
 Timebase = 0.1 us
>
```

The format for timing data displayed at the last ("Timer =") line of Go command execution (see Example 1 of Section 4.3 of this Addendum) is-

"dd hh:mm:ss.mmm uuu n"

where:
- dd = days
- hh = hours
- mm = minutes
- ss = seconds
- mmm = milliseconds
- uuu = microseconds
- n = 100 nano seconds

4.10 DISPLAY CURRENT TRACE PARAMETERS - TRAce

TRAce

TRAce is the command to display the following settings:

Breakpoints EV1 to EV7
Trigger condition (Trigger setting and Trigger Level settings)
Cycle qualifier
Timebase
Synchronization setting

Example: Display the current MICE-IIIS 68000 trace control settings.

```
>TRAce<CR>
 EV1 (Bus) 000400 XXXX SR,SW External XX Count 0020
 EV2 (Bus) 000480 4E71  External XX Count 0001
 EV3 (Bus) Range 000500 0005FF XXXX  External 34 Count 0001
 EV4 (Bus) XX34XX X6X7 UP External 9X Count AABB
 EV5 (External) High
 EV6 (Execution) 000500 SP,UP
 EV7 (Execution) 000520 SP,UP
 EV8 (Execution) clear
 Trigger A Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
 Cycle qualifier 1 : XXXXXX UW XX  Enable
 Cycle qualifier 2 : 092000 092700  SW 4(XX10)  Disable
 Timebase = 1 ms
 Sync start in Off
 Start slave out Off
>
```

4.11 DISPLAY/SET/CLEAR TRIGGER - Trigger

Set Trigger Level	**Trigger Level** level# [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev# [{OR|ANd} [Not] ev#]]]] [Trace {On|OFf}] [TImer {On|OFf}]
Display/Clear Trigger Level	**Trigger Level** level# [CLear]
Set Trigger	**Trigger** [RUn] {level# [<THen level#>] [Reset | If state# <ELse state#>] | If state# <ELse state#>} [FOrward|BAckward |CEnter|DElay count]
Set Initial Timer & Trace	**Trigger** [TImer {On|OFf}] [Trace {On|OFf}]
Display/Clear Trigger	**Trigger** [CLear]

Trigger specifies Trigger display/set/clear. "Trigger" alone, without any para-meters, displays the current Trigger condition.

Level is the required prefix when specifying a level identifier.

level# is one of level identifiers- A, B, C, D, E, F, G or H.

Ev# is one of the Events 1, 2, 3, 4 or 5.

NOTE

Do not specify the same level and Event twice in one command.

Trace {On|OFf} Enable/disables trace program execution in the trace buffer.

Timer {ON|OFf} Enable/disables time measurement of user's program execution.

Run is an optional parameter which allows the emulation processor to free run after the trace ends.

state# level# [RESet] | (level# {<Then level#> [RESet | If state# <ELse state#>] | If state# <ELse state#> })

< > Recursive (Minimum 1 time).

( ) Reserved Symbols.

FOrward is the delay count = 7FFFH for 32K trace buffer.

BAckward is the delay count = 0.

CEnter is the delay count = 3FFFH for 32K trace buffer.

count is a hexadecimal value (from 0H to 0FFFFH).

CLear is the command to clear the current Trigger setting.

The Trigger setting is composed of up to 8 Trigger Levels. Each level may be a single event or a logical construct of up to five breakpoints with **OR**, **ANd**, and **Not** (i.e., the Event never occured). Only Events 1 to 5 can be used in the Trigger Level definition; execution breakpoints are automatically joined to the specified construct by a logical **OR**. EV1~EV5 can only be used once in a Trigger Level setting. "Trace On|OFf" or "Timer On|OFf" may also be specified at any Trigger Level or at the beginning of free-running emulation. The default at power-on and after a hardware reset is the construct "Level A EV1 OR EV2 OR EV3 OR EV4 Trace On Timer On". Refer to Figure 4-2a (next page) for the "Trigger Level" setting flow chart.

LAM-IIS supports a very powerful Trigger condition by setting "Trigger Level" and "Trigger". **THen** connectives and **If...ELse...** could be flexibly used in the Trigger setting. In order to be familiar with the complex command structure, it is recommended that user establishes a tree structure first as shown in Examples II-4 and II-5 below before writing a Trigger setting syntax. The "Trigger" setting syntax flow chart is shown in Figure 4-2b.

As illustrated in the Trigger command flow chart (Figure 4-2a/b), the Trigger setting only specifies the break condition. It is not effective until the Go command is input to start the emulation. Note also that the Trigger setting has no effect during the Cycle Step and Instruction Step commands.

When emulation begins, the MICE-IIIS 68000 immediately starts recording target system and emulation processor status (in accordance with the Cycle Qualify setting) in real-time. Data are recorded until the Trigger condition, including any cycle-count delay, is matched.

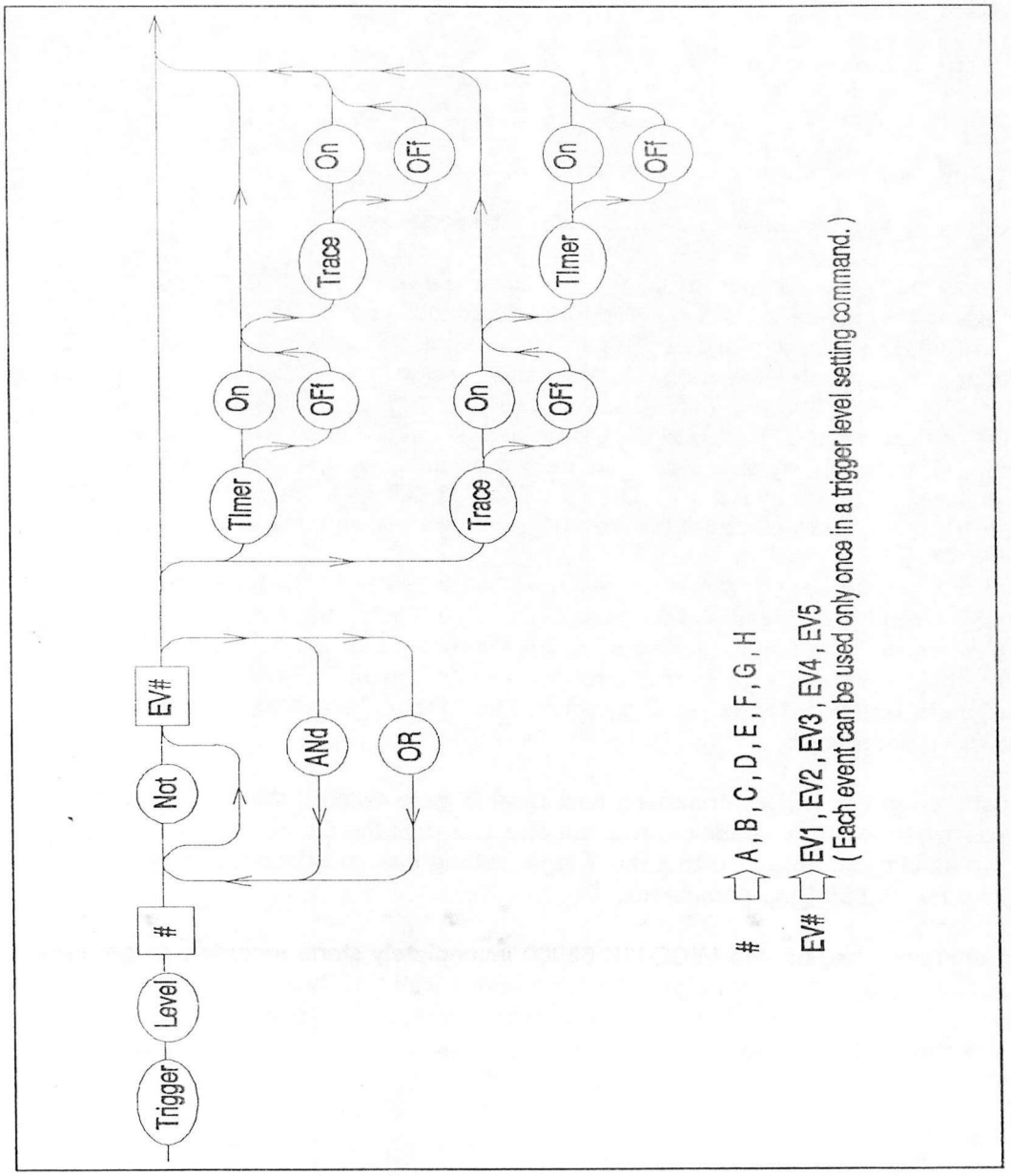


Figure 4-2a LAM-IIS Trigger Level Setting Syntax Rules and Flow Chart

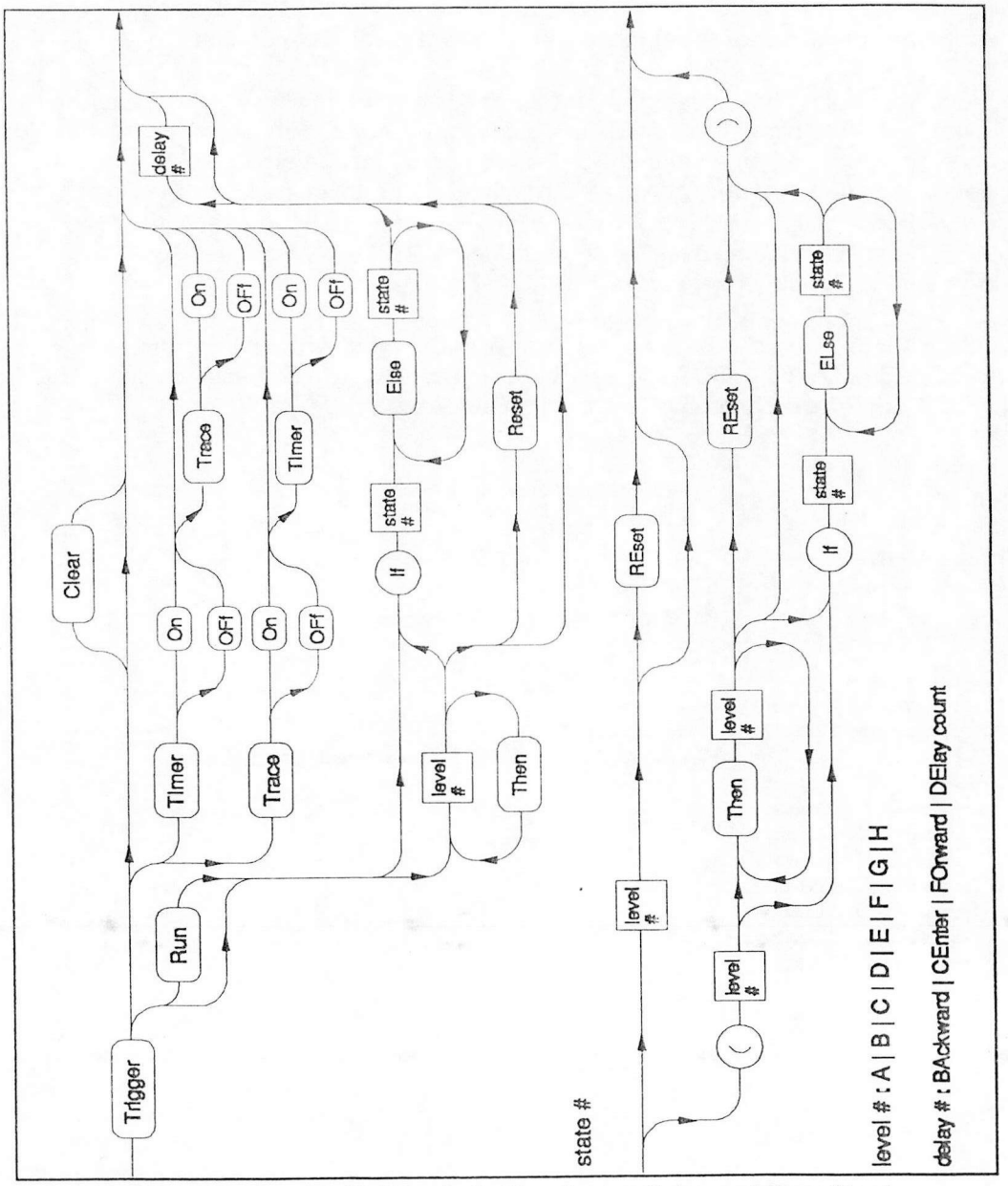


Figure 4-2b LAM-IIS Trigger Setting Syntax Rules and Flow Chart

Up to 32768 machine cycles can be recorded. If more than 32768 cycles had elapsed before tracing ends, only the last 32768 cycles will be recorded.

When the trace ends, the event(s) matched are displayed and (unless the Run option was used) the MICE stops the emulation processor one cycle (or two cycles, depending on the CPU speed and its wait cycle) beyond the location where the Trigger condition was matched. The recorded information can be examined with the List Trace Buffer command. The "STATE" column in the List command execution denotes the change of state. Watch "Event encounter flag" of EV1~EV5 in the "EVENT" column resets to 00000 whenever the "STATE" changes.

Note that whenever the processor is running, external hardware sync signals are generated, at the SYNC 1 OUTPUT port the first time Event 1 is matched and at the SYNC 2 OUTPUT port each time Event 2 is matched.

Examples:

i. Trigger Display

1) Display the current Trigger and Triger Level settings.

```
>Trigger<CR>
 Trigger A Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
>
```

2) Display the current Trigger Level setting.

```
>Trigger Level<CR>
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
>
```

II. Setting Trigger Condition

1) Specify some combinations of Trigger constructs in Trigger Levels.

```
>Trigger Level A EV1 OR EV2 OR EV3 OR EV4<CR>
>Trigger Level B EV1 ANd EV3 ANd EV5<CR>
>Trigger Level C Not EV2<CR>
>Trigger Level D EV1 OR EV4 OR EV5<CR>
>Trigger Level E EV2 ANd EV3 OR Not EV4<CR>
>Trigger Level F EV1 OR EV2 OR EV3 ANd EV4 TImer OFf<CR>
>Trigger Level G EV1 ANd EV5 TRace OFf<CR>
>Trigger Level H EV3 Trace OFf TImer OFf<CR>
>Trigger<CR>
 Trigger If (A If B ELse C ELse D Reset) ELse (E If F ELse G ELse H Reset)
 Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Not EV2 Trace On Timer On
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

Note that the precedence of the logical factors is **NOT > AND > OR** and each level may specify Trace On|OFf and Timer On|OFf while the trigger level is matched.

2) Using the Trigger Level setting in the preceding example, set a single level trigger with 10FFH delay and then display result.

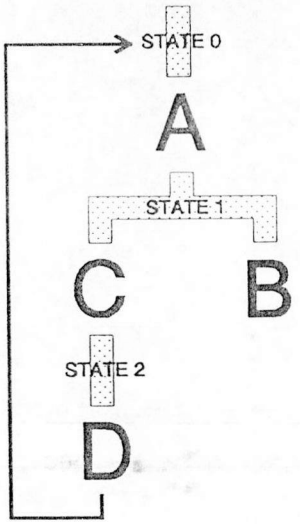
```
>Trigger B DElay 10FF<CR>
>Trigger<CR>
 Trigger B DElay 10FF
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Not EV2 Trace On Timer On
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

3) Using the Trigger Level setting in Example 1, set a "Then" trigger connectives and continue emulation after the trace stops.

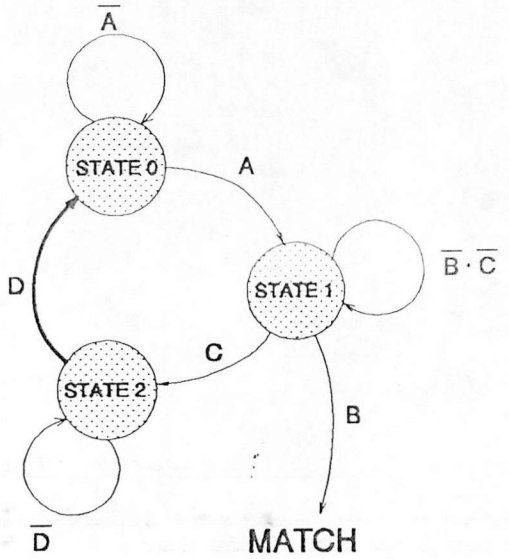
```
>Trigger RUn A THen D THen F<CR>
>Trigger<CR>
 Trigger RUn A THen D THen F Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Not EV2 Trace On Timer On
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

4) Construct a tree structure of the following Trigger condition:

```
Trigger A IF ( C THen D Reset ) ELse B
Trigger Level A EV1 OR EV4
Trigger Level B EV3
Trigger Level C EV2
Trigger Level D EV5
Event 1 8008
Event 2 8010
Event 5 High
Event 4 8018
Event 3 8020
```

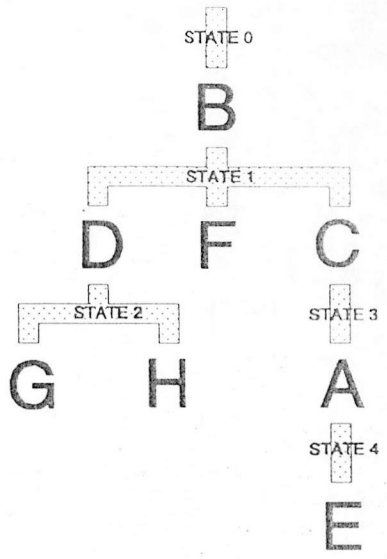


Tree

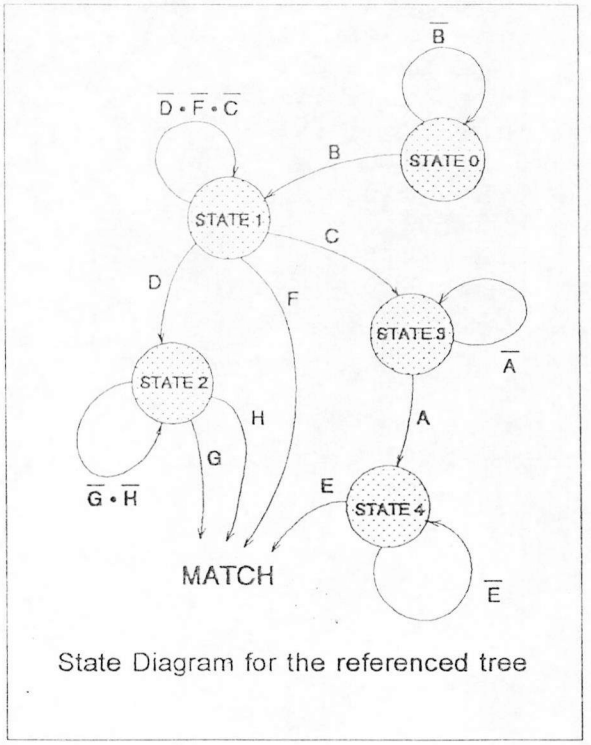


State Diagram for the constructed tree

5) Specify a Trigger setting in reference to the following tree structure.



Tree



State Diagram for the referenced tree

```
>Trigger B If ( D If G ELse H ) ELse F ELse ( C THen A THen E )<CR>
>Trigger<CR>
 Trigger B If ( D If G ELse H ) ELse F ELse ( C THen A THen E) Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Not EV2 Trace On Timer On
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

6) Set Timer On and Trace OFf when emulation begins to run in Trigger condition.

```
>Trigger TImer On Trace OFf<CR>
>Trigger<CR>
 Trigger A Delay 0000
 Trigger Timer On Trace Off
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  Clear
 Level C  Clear
 Level D  Clear
 Level E  Clear
 Level F  Clear
 Level G  Clear
 Level H  Clear
>
```

III. Trigger Clear

1) Clear the current Trigger Level C setting.

```
>TRigger Level C CLear <CR>
>TRigger<CR>
 Trigger B If ( D If G ELse H ) ELse F ELse ( C THen A THen E ) Delay 0000
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Clear
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

2) Clear the current Trigger setting.

```
>Trigger CLear<CR>
>Trigger<CR>
 Trigger Clear
 Trigger Timer On Trace On
 Level A  EV1 Or EV2 Or EV3 Or EV4 Trace On Timer On
 Level B  EV1 And EV3 And EV5 Trace On Timer On
 Level C  Clear
 Level D  EV1 Or EV4 Or EV5 Trace On Timer On
 Level E  EV2 And EV3 Or Not EV4 Trace On Timer On
 Level F  EV1 Or EV2 Or EV3 And EV4 Trace On Timer Off
 Level G  EV1 And EV5 Trace Off Timer On
 Level H  EV3 Trace Off Timer Off
>
```

Addendum 5
APPLICATION NOTES

5.1 LAM-IIS APPLICATION NOTES

1) Under LAM version, EV1 & EV2 were bus breakpoints; EV3 was an external hardware breakpoint; EV4, EV5 & EV6 were execution breakpoints. Under LAM-IIS version, EV1 to EV4 are bus breakpoints; EV5 is an external hardware breakpoint; EV6, EV7 & EV8 are execution breakpoints.

2) Whenever EV1 and EV2 are matched, a low pulse will occur at SYNC 1 OUTPUT and SYNC 2 OUTPUT ports respectively. For more information on timing, see Section 5.2 below.

3) If the trace buffer is not full (i.e., last frame not equal to 7FFFH for 32K trace buffer) when it stops recording, the data under STATE and EVENT fields of Frame 0 are inaccurate and should be ignored.

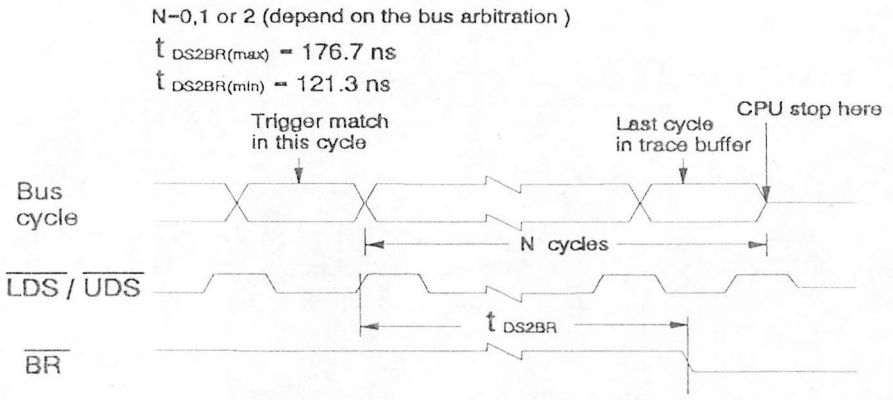
4) When upgrading MICE-16 68000 (with LAM) to MICE-IIIS 68000 (with LAM-IIS version), the CPM board on which the firmware (EPROM U11, U12, U14, U15) are to be replaced, must be of REV-G or later.

5.2 LAM-IIS TIMING NOTES

The following timing diagrams only applies to MICE-IIIS 68000 equipped with LAM-IIS:

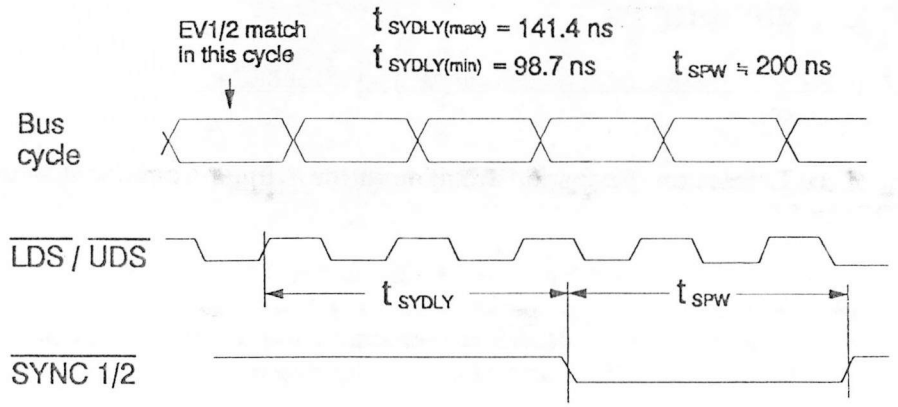
1) **Timing of the Emulation Processor Breaking at the Trigger Command Setting with "DELay 0".**

   Literally, whenever the emulation stops, MICE asserts $\overline{BR}$ signal to stop running the program. From this point, Emulation Processor ceases to be the bus master. The time between the point $\overline{LDS}/\overline{UDS}$ is negated during a "Trigger matched bus cycle", to the point when BR is asserted, is equal to t_{DS2BR}.



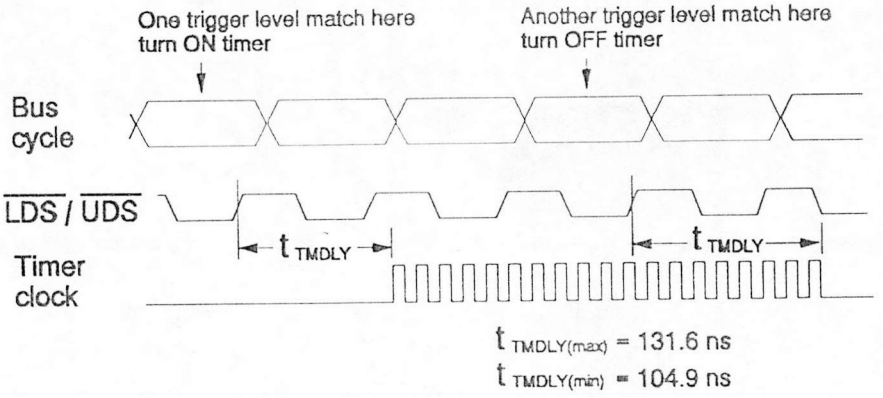
2) SYNC 1/2 OUTPUT Signals Timing

Although the "Event encounter-flag" of EV1 to EV5 is cleared when the STATE changes during free running, the $\overline{\text{SYNC1}}$ or $\overline{\text{SYNC2}}$ output signal will be asserted for about 200ns, whenever the EV1 or EV2 is matched and sets the "Event encounter- flag". The time between the point when $\overline{\text{LDS}}$/$\overline{\text{UDS}}$ is negated during the bus cycle where EV1 or EV2 is matched, to the point when $\overline{\text{SYNC1}}$ or $\overline{\text{SYNC2}}$ is asserted, is equal to t_{SYDLY}.



3) Timer On/OFf Timing

The timer can be turned On or turned Off at any bus cycle by specifying timer On/Off locations at the end of each "Trigger Level". The timer is then turned On/Off accor-dingly whenever the "Trigger Level" is matched. But in reality, there is a time delay between the time "Trigger Level" is actually matched and the time the timer clock is actually On (or Off). The time between the point when $\overline{LDS}/\overline{UDS}$ is negated during the bus cycle in which "Trigger Level" is matched, to the point when timer clock is put On (or Off), is equal to t_{TMDLY}.



4) Trace On/OFf Timing

The trace buffer can also be enabled/disabled at any bus cycle by specifying trace On/Off locations at the end of each "Trigger Level". The trace buffer is then turned On/Off whenever the "Trigger Level" is matched. But in reality, there is a time delay between the actual time "Trigger Level" is matched and the actual time the trace buffer is enabled or disabled. Hence, when one "Trigger Level" is matched in the current cycle and the trace buffer is turned Off, tracing will actually stop in the next and subsequent cycles (not in the current cycle). Tracing is resumed when another "Trigger Level" is matched and trace buffer is turned On.