EMS 8000
PHASE I

Zilog

USER'S GUIDE

January 1982

Zilog

ADVANCE COPY

Errata for EMS 8000 Phase I

(Document Number: 03-0123-01)

DISCREPANCY

The current EMS systems have a bug which will be corrected in the near future:

The General-Purpose counter can be used as either an Event Counter or a Timer. either of these modes, it can count to a number higher than FFFF (hex); the numb can contain up to 12 hex digits--therefore, the counter can count for over a we without overflowing.

It is possible to count the number the events or the amount of time in an inte which occurs repeatedly. Every time a new interval begins, the counter is rese zero. Phase I software does not directly support the ability to see each of counts for each of the intervals, but the count at the end of the emulati supposed to indicate the correct count for the last interval which occurred.

The bug occurs when a new interval begins within 50 microseconds or so of th that a carry occurred into the 5th (from the bottom) hex digit. In this cas high-order digits may or may not be reset to zero. In all other case, eve should work properly.

Here are some suggested ways to avoid this bug:

1. For timer mode, specify either Cumulative Mode (which computes the t for all intervals) or specify that Restart Allowed after First Finish i that only the first interval's time is recorded.
2. For event count mode, specify either Cumulative Mode or make the Trigger non-reentrant, so that the count interval will only occur once
3. Ensure that the time between each of several multiple intervals will 50 microseconds.
4. Ensure that the count for each interval is less than FFFF (hex).

Any of the techniques listed above should ensure proper operation.

EMS 8000

Phase I Software

User's Manual

TABLE OF CONTENTS

SECTION 1

INTRODUCTION

1.0 INTRODUCTION

The EMS 8000 is a state-of-the-art emulation subsystem which supports the hardware and software design engineer in developing systems using the Zilog family of microprocessors and peripheral components. Features such as sophisticated triggering, a large real-time trace buffer and a large mappable memory space provide the user with powerful debugging tools during the development cycle. In addition to these features, the EMS 8000 is also capable of performance measurements on benchmarks and critical timing loops.

The EMS 8000 is an intelligent peripheral whose software is downloaded during initial power up. This feature allows the EMS software to be easily upgraded. Additionally, custom programming of EMS for specific test floor operations is accomplished using this download link.

This manual describes the commands and their operations for Phase I of the EMS 8000 emulation susbsystem. The subject matter covered in this manual is likely to change both in form and content as the product is upgraded or enhanced. ALL FIRST TIME USERS should fully complete the familiarization session in Appendix E. This session introduces EMS's capabilities and provides useful examples. The following conventions are used throughout this manual:

1. EMS 8000 Phase I command prompts:

-	Main command level
t-	Trigger command sub-level
m-	Mapper command sub-level
mdef-	Macro Definition

2. Backspace and line delete conventions:
   The character 7F (hex) is used for "Delete Line".
   The character 08 (hex) is used for "Backspace".

3. Command syntax:
   A single set of square brackets [n] means that n is optional
   A set of parentheses (Opt1|Opt2|Opt3) means choose one option from within the parentheses.

1.1 GENERAL FEATURES

EMS 8000 is an in-circuit emulator. This means:

- An emulation cable replaces the CPU chip in the user's prototype target system. Hookup is simple and immediate.

- The user can access the target's registers, memory, and I/O with a convenient user interface on a CRT. Procedure files from a host computer can be transferred to and from the target's memory.

- The user can start, stop, and step execution of code. When started, code executes in real time.

- Logic state analysis is provided: 64 bits of target information are monitored during each machine cycle for recording in a trace memory and meeting specified conditions. The specification of conditions is quite versatile. Cycles meeting the conditions can affect the recording process or can stop emulation.

- Fast static RAM in EMS can be placed at any point in the user memory space and function as if it were in the target system. A mapper allocates this RAM and provides protection features.

- The user can evaluate performance of execution of real-time software using a counter.

- The terminal can function as if it were hooked directly to the host or to EMS. (EMS does not require an extra serial port on the host.) This is called transparent mode.

- A modular architecture allows support of future Zilog microprocessors.

- System and user definable macros are provided to ease utilization of the EMS 8000 features.

SECTION II
HARDWARE OVERVIEW

The EMS 8000 Hardware consists of:

- CCU Module
- Sample Bus
- Trace Module
- Trigger Module
- Emulator Module
- Mappable Memory

See Figure 2-1 for a block diagram of the EMS 8000 system configuration.

2.0 CCU MODULE

The CCU module contains a 4 MHz Z80 microprocessor with up to 16K bytes of EPROM and up to 256K bytes of RAM. This memory is used exclusively for the software of the EMS system. In addition, I/O ports are provided for a terminal and a host interface.

The terminal can be used to converse with EMS software while setting up and using the emulator, or to converse directly with the host (transparent mode) without affecting the emulation process. The host port is used for downloading the EMS monitor software to the Z80 RAM, and for downloading and uploading of user programs. In addition files shared within the host file management system can be accessed via the host port for executing "canned" sequences of EMS 8000 commands.

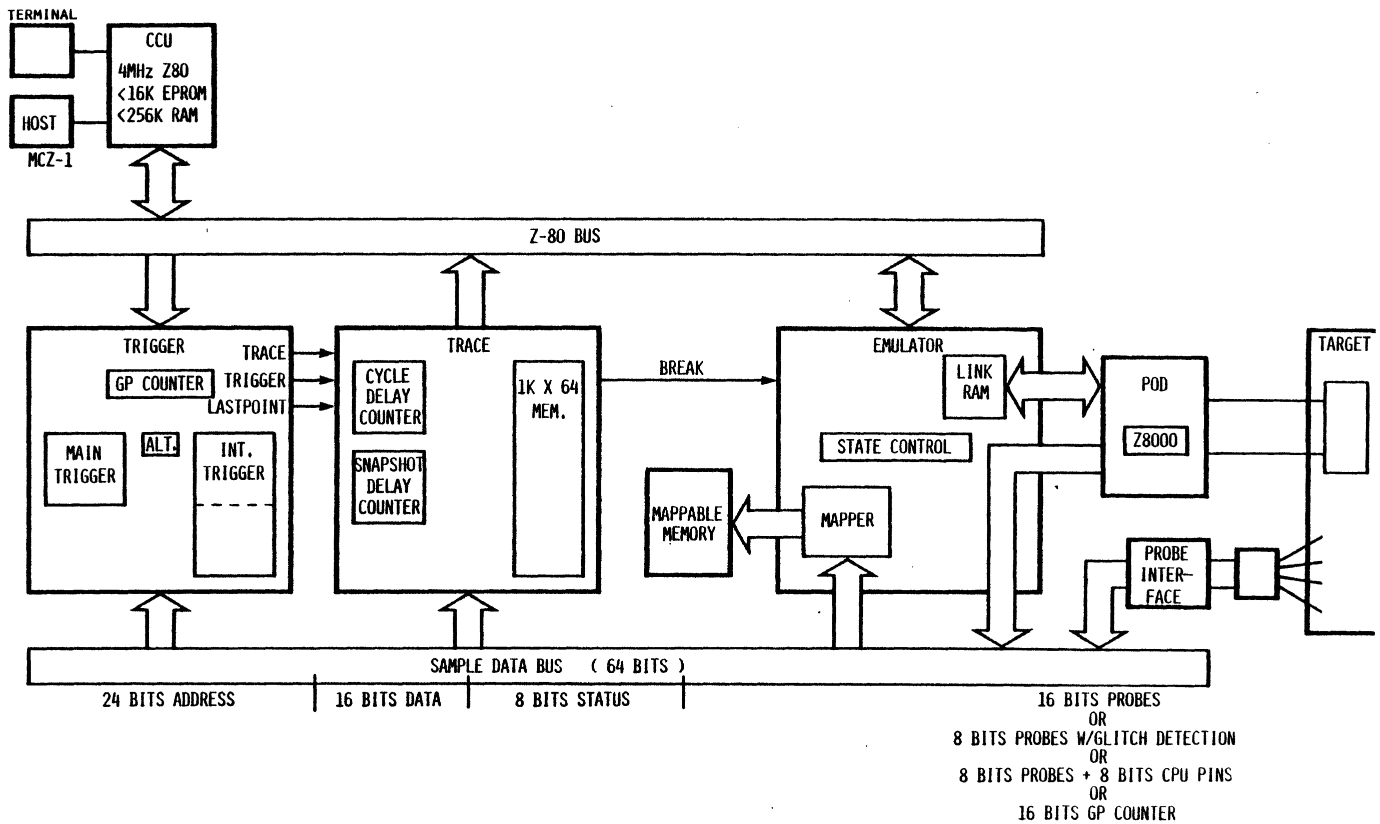


Figure 2-1. EMS 8000 System Configuration

2.1 SAMPLE BUS

EMS 8000 uses a 64-bit wide sample bus.

With the Z8000 these bits are assigned as follows:

16 bits	Address
8 bits	Segment
16 bits	Data
8 bits	Microprocessor Control Lines (ST_0-ST_3, S/N, R/W, B/W, BUSACK)
16 bits	External Probes or General Purpose Counter

2.2 TRACE MODULE

The Trace Module is used to record states on the sample bus. It has very selective trace features. It can trace all cycles of the Z8000 or trace only qualified cycles. It has one feature, unique to EMS, called a segmented trace. This feature allows the user to take "snapshots" of short sequences of execution. The trace module consists of:

- A trace memory which is 1024 entries deep and 64 bits wide.

  The trace memory can be partitioned into <u>SNAPSHOTS</u> of various sizes. Each of these snapshots is capable of recording bus information centered around a specified trigger point. The number of snapshots and the corresponding snapshot size is listed below.

  1 snapshot of 1024 bus cycles
  2 snapshots of 512 bus cycles
  4 snapshots of 256 bus cycles
  .
  .
  256 snapshots of 4 bus cycles.

- A CYCLE DELAY COUNTER which gives the user the ability to position cycles within a snapshot relative to a trigger (POST, CENTER, and PRETRIGGER recording).

- A SNAPSHOT DELAY COUNTER, which gives the user the ability to stop emulation a specified number of triggers after a specified final trigger.

2.3 TRIGGER MODULE

The Trigger Module compares the sampled microprocessor data against a set of user-specified patterns for the purpose of stopping emulation and/or controlling certain actions of the Trace Module. The Trigger Module consists of:

- 3 parallel trigger comparators used as building blocks in configuring the trigger, trace, and timing functions.

  1. Main Trigger: Can detect a sequence of 5 patterns (supports ranging and 2 address logical operations).

  2. Alternate Trigger: Can detect one pattern.

  3. Interval Trigger: Can detect a sequence of 8 patterns. Can be programmed to repeatedly arm and disarm either the main or alternate trigger.

- A general purpose counter allowing a wide variety of counting and timing functions.

2.4 EMULATOR MODULE

- Interface between target system and EMS. Starts and stops emulation.

- Functions custom to each processor are concentrated on the emulator card. (Z8001 and Z8002 microprocessors can use the same emulator card.)

- Complete buffering from user system. EMS can regain control during any portion of the emulation cycle.

- Transparent emulation. The whole memory and the whole I/O space are available to the target system.

2.5 MAPPABLE MEMORY

2 options:

64Kb static or
126Kb static

- The mappable memory can be mapped with a resolution of 1K words anywhere in the Z8000 memory space. The Mappable Memory module is supplied with fast static RAM and introduces no additional wait states on user access.

- Each block of 1K words can be mapped with a variety of memory access protections.

- Mapping on logical or physical address (physical address mapping requires the external probe option).

- A block of Z8000 mapped memory may have special attributes (Normal|System, Code|Data|Stack).

SECTION III
SOFTWARE OVERVIEW

EMS 8000 Phase I software is divided into 3 domains for entering commands.

1. The Emulation domain is for entering emulation, trace, host and macro commands. It also allows the user to enter the other two domains, Trigger and Memory Mapping. Emulation commands allow the user to modify and monitor the target system (i.e. examine memory, write to a port).

2. The Trigger domain controls the 3 parallel resources which can be used to trigger, qualify the trace memory, and stop emulation. In addition, a General Purpose counter is provided to count machine cycles.

3. The Memory Mapping domain controls the use of EMS memory for the target system.

The "ESC" key on the terminal is used to break emulation. "Type-ahead" is allowed at any time even though these characters will not appear on the terminal until emulation has broken and the previous command is finished.

A good method to start using the system with a minimal amount of knowledge is using System Macros. Using ".C" to trace all and ".B" to set breakpoints, simple debugging can be done. Using ".S", ".N", and ".G", to step through code is relatively easy.

A more detailed command description can be found in the following sections

Emulation domain commands	IV
Trigger domain commands	VII
Memory Mapping Domain	VI
Macros	V
Command summary	Appendix B
Familiarization Session	Appendix E

SECTION IV
EMULATION COMMANDS

4.0 EMULATOR DESCRIPTION

The Emulator module is the interface between the target system and EMS. The emulator has complete buffering from the target system so that EMS can regain control from any unusual behavior caused by the target system. The electrical interface with the user's system always looks like a valid microprocessor state. The only difference between EMS and a real microprocessor is the slight change in access and setup times introduced by high-speed EMS buffering and control. The entire memory and I/O space are available to the user's target.

4.1 EMULATION CONVENTIONS

In all of these commands, the following conventions are used:

<addr> For the Z8002, this is a word entered in hexadecimal.

For the Z8001, this address may be preceded by a segment number enclosed in brackets, such as <3F> 6A3E. If the segment number is not specified, the previously specified segment is used.

<range> This may be in either of two forms:

<beginning address> , <end address>

<beginning address> <length>

Both of the addresses specified here use the rules for <addr> given above. <length> is expressed in Hex.

DEBUG COMMANDS

These commands allow the user to modify the target system's registers, memory and I/O. <Esc> (1B hex) aborts the execution of the D (display) and C (compare) commands.

Compare

CB Compare Byte
CW or C Compare Word

(CB|CW|C) <addr> <range>

The memory block specified by <range> is compared with the block beginning at <addr> with equal length. The comparison is done on a byte-by-byte or word-by-word basis depending on the command. All unequal items are printed on the screen.

Display

DB Display Byte
DW or D Display Word
DL Display Longword

(DB|DW|D|DL) <range>

This displays a range of memory as bytes, words or long words.

Memory is displayed in hex and ASCII.

NOTE:

If a length is typed it is always interpreted as a length in bytes, even if word or long words are being displayed. If word or long word is selected, the number of bytes specified is rounded to the nearest word or long word.

(DB|DW|D) <addr>

This opens memory up for display and modification of one location at a time. The first location to be modified is the <addr> typed. The address and data at each location is displayed, and the user is prompted for input. Any value may be entered, or no value may be entered causing the location to remain unchanged.

If the carriage return is typed immediately after the value, or if only a carriage return is typed, the next location will be opened for possible modification.

If one of the three characters shown below is typed after the value displayed, or if only the character is typed, the location opened for possible modification after the current one will be as shown:

∧ (5E Hex)	Location before current one
. (period)	Current location is shown again
Q	Will return to main command level

Each of these characters must be separated from any entered value by a space and followed by a carriage return.

Fill

FB Fill Byte

FW or F Fill Word

(FB|FW|F) <range> <string of values>

The range of memory specified is filled with the value string specified. If the range is larger than the length of the string (as is normal), the string is replicated as many times as necessary.

NOTE:

AN LDIR instruction is used, so that working read/write memory must exist over the entire range of the fill.

Input

IB Input byte

IW or I Input word

(IB|IW|I) <port addr>

An input is performed to the specified port and the value read is displayed. (See also "O" for output.) The type of I/O Access (Special I/O or Normal I/O) is specified by the IT (I/O Type) command.

Move

M Move Memory

M <addr> <range>

The block of memory described by <range> is copied to the block of equal length beginning at <addr>.

Output

OB Output Byte

OW or O Output Word

(OB|OW|O) <port addr> <value>

The value specified is output to the specified port. No reading of the port is performed. The type of I/O access (Special I/O or Normal I/O) is specified in the IT (I/O Type) command.

Register

R Registers

R

This version of the command causes all registers to be displayed.

R <reg name>

This version causes each register beginning with <reg name> to be opened for display and possible modification. The register name must be typed exactly as shown in the register display.

NOTE

These register values are only written into the target CPU when emulation begins (Go command) or when a register test (RT command) is performed.

The $\wedge$ may be used as in the Display command to return the previous register value.

NOTE

Only word registers may be displayed and altered.

On the Z8001, the PC and PSAP are assumed to contain addresses. Therefore they are displayed and entered with the syntax for <addr> as specified at the beginning of Section 4.2.

Register Test

RT Register Test

This command causes the registers to be written to the actual CPU and read back. It provides a good indication that most of the emulator is working. It needs only a clock from the target system or test socket. Use the R (register) command to set register values and examine results of RT.

Reset

RE Reset

This command resets the emulator CPU. The CPU must be reset after every interruption in clock or power. The CPU is automatically reset when the EMS monitor (ZAP8001 or ZAP8002) is initialized.

Store

SB Store Byte

SW or S Store Word

(SB|SW|S) <addr> <value string>

The string of values given is stored in memory beginning at the given address.

4.2 EMS SYSTEM COMMANDS

These commands modify the state of EMS resources.

AD System Addresses

AD (L|P)

AD L selects Logical addresses for mapping, triggering, and tracing. The default value for AD is L. AD P selects physical addresses for mapping, triggering, and tracing. The External Probe board (optional) is required to use physical addresses.

CT Cycles Traced

CT (A|D)

If CT A is typed, all cycles are traced including Internal Operation and Refresh. If CT D is typed, only cycles with data transactions (data cycles) are traced. The default parameter is D (Data cycles only)--this parameter is set when the system is initialized.

DI Disable

EN Enable

(DI|EN) (NMI|NVI|VI|SEGT|STOP|RESET|BUSRQ|WAIT|BREAK|MAP|WB|MB)

Any of the control inputs or features listed may be enabled or disabled using these commands. The current state of any of these may be displayed with the Setup command (SE). The options WB (write break) and MB (mode violation break) are used with the mappable memory feature of EMS and will only be available if the Map feature is enabled. EMS is initialized with NMI, NVI, VI, SEGT, STOP, RESET, BUSREQ, WAIT and MAP enabled. BREAK is initialized as Independent (for single EMS configurations) and both WB and MB are initialized as disabled.

ER Emulator Reset

The Emulator control logic is reset by this command.

E2 Field E2 (External Probe 2 on front panel)

E2 (P|I|G|C)

This command is used to select whether the data on field E2 of the Sample Data Bus comes from the external probe (E2 P) or from the CPU control inputs (E2 I) or the Glitch detector (E2 G) for probe E1; or from the General Purpose Counter (E2 C).

E2 I: Field E2 of the External Probes monitors internal CPU signals (NMI, NVI, VI, SEGT, reserved, STOP, M_1, M_0). These signals are internally monitored by EMS and do not require an External Probe Interface board.

E2 P: Field E2 monitors signals connected to External Probe 2. These signals may be Port I/O lines or other TTL signals which will be sampled with an internal strobe based on AS from the CPU.

E2 G: External Probe 2 is not physically connected. Instead EXT probe 2 is used to monitor the signals on EXT probe 1 for "glitches". Glitches are defined as transitions which occur between AS (address strobe) sample points.

E2 C: Neither External Probe is connected to the Sample Data Bus. Instead, the outputs of the general-purpose counter are put on both field E1 and E2. This is useful for time stamping. The numbers displayed are the 1's complements of the actual counts.

G or GO Go

The emulator enters User Emulation mode. The starting address is the PC register. A mappable memory violation, trace completed condition, or typing the <esc> key on the terminal will break emulation, print the CPU registers, print any relevant messages, and return to the normal EMS monitor.

IT I/O Type

IT (S|N)

Sets up the I/O accesses type to Special (S) or Normal (N).

MM Mappable Memory

MM

This command enters the Mappable Memory sub-command level. The commands available at this level are given in the Mappable Memory section. Type "Q" followed by "return" to exit from the mappable memory sub-command level.

MT Memory Type

MT (N|S) (C|D|S)

This command sets the type of memory access made when target memory is accessed. This command selects from Normal or System mode, and Code, Data, or Stack Accesses.

PM Print Macro

PM <char>

This command prints the macro definition string stored internally. See Macros.

PN Print Names

This command prints the names (single character) of all the macros currently defined. Macros currently supplied (which can be changed) are S, N, B, Q, G, J, C, M, W, U, H.

QU Quit

Restarts the monitor, clears the triggers and memory map, and resets all system parameters to their default state.

WA WAIT

WA [n] U

This command causes n Wait states to be inserted in every memory access. The U must be typed. The value of n may range from 0 to 7. If no value is entered for n the number of wait states is unchanged.

SE Setup display command

Displays the current setup of the following:

Description	**Symbols**	**Mode List**
CPU pins	NMI, NVI, VI, SEGT, STOP, RESET, BUSRQ, WAIT	(ENABLED|DISABLED)
Breakpoint Enable/Disable	BREAK	(Independent (enabled)| Disabled)
General Trace qualifier	Cycles Traced	(Data cycles only| all cycles)
External Probe configuration (E2 command)	External Field Mode	(E1: Probes E2: Inputs Probes|Glitches|Counter)
Debug command Memory access type	Memory Type	(Norm|sys) (code|data| stack)
Debug command I/O access type	I/O Type	(Normal|Special)
Mappable Memory	Mem MAP	(ENABLED|DISABLED)
Trigger|Trace address source	System Addresses	(Logical|Physical)

4.3 TRACE DESCRIPTION

This section describes many of the details of the EMS 8000 trace mechanism. A good understanding of the trace mechanism and its relationship to the trigger mechanism is desirable for effective use of the EMS 8000 emulator. The features of the trace mechanism working in conjunction with the trigger mechanism provide the user with a very comprehensive and powerful set of debugging capabilities, some of which are unique to EMS 8000. Both this section and Section VII (Trigger Domain) should be read in detail.

4.3.1 Basic Concepts

The EMS 8000 trace mechanism is an extension of a conventional logic analyzer trace mechanism. In general, all bus transactions are recorded into a trace memory. They are recorded in a circular fashion: when the end of the memory is reached, new cycles overwrite the old ones at the beginning of the memory. This recording process continues until a trigger event stops it.

In EMS (as with all logic analyzers), cycles may be collected before the trigger event (pre-trigger recording), around the trigger event (center trigger recording), or after the trigger event (post-trigger recording). For pre-trigger trace, recording stops immediately on the trigger condition. For center-trigger trace, after the trigger occurs, there is a delay equal to half the size of the trace memory, and then recording is stopped. Then half of the memory will contain what happened before the trigger occurred, and the other half will contain what happened afterwards. For post-trigger trace, after the trigger occurs, there is a delay equal to the size of the trace memory, and then recording is stopped. The memory will contain all of the bus transactions beginning with the trigger event itself.

The only difference between pre-trigger, center-trigger, and post-trigger tracing is the amount of delay after the trigger event before recording is stopped. In EMS, this may be continuously adjusted: for instance, the trace memory, which is 1024 cycles long, can contain 924 cycles before the trigger, and 100 cycles after the trigger. It is also possible to make the delay much larger than the trace size. This is useful when target software seems to "get lost" a fixed amount of time after emulation begins. For example, suppose a delay of 9024 is used: the cycles traced would range from 8000 to 9024 after the trigger. If the program "gets lost" somewhere in this window, it can be found this way.

If the trigger condition never occurs, or no condition is set up, then recording will occur continuously until the user intervenes with a manual break. Tracing will be stopped immediately, even if a center- or post-trigger scheme was being used.

Cycles are numbered with respect to the trigger: the trigger is cycle number 0. The cycle which occurs 5 cycles before the trigger is cycle number -5. The cycle which occurs 7046 cycles after the trigger is cycle number 7046 (even though the trigger is overwritten).

4.3.2 Trace Qualifications

With EMS and many logic analyzers, it is possible to restrict the recording of bus activity to only those cycles meeting certain conditions. This is called trace qualification. With EMS many conditions may be used as qualifiers. A condition may be an address match, a range of addresses, a bus status (such as interrupt or I/O) or any reasonable combination of bus activities. A condition may also be an ordered occurrence of other conditions.

It is also possible with EMS to use an ENABLE and DISABLE feature to define "time slots" during which tracing occurs. Cycles will be recorded only after the enable point and before the disable point, if this mode is desired.

These two forms of qualification may also be combined, allowing only cycles which meet certain conditions AND occur in a certain time slot to be traced.

NOTE:

Trigger conditions (Trigger and Final Triggers) are ALWAYS traced, regardless of qualification.

When a trigger condition is used with a delay count for center- or post-trigger tracing, AND trace qualification is used, the delay will be a number of QUALIFIED cycles.

4.3.3 Multiple Snapshots

In a conventional logic analyzer, after the trigger condition has been met, tracing is stopped and the results are displayed with the trigger positioned where it is desired. If the user has a problem which only happens on the nth occurrence of the trigger, many logic analyzers (and EMS) allow the user to specify that only the nth occurrence is to stop the trace.

If the user has a problem which happens on SOME random occurrence of the trigger, it is desirable to collect, in one trial, execution data before or after each of several triggers. This capability is called "multiple snapshots" and is a unique feature of EMS.

To do this, the 1024-cycle trace memory is broken into a collection of smaller memory partitions. The number and size of the partitions is widely adjustable, because some problems will have a larger number of triggers, and other problems will need a larger amount of data around each trigger. The memory can be divided into 2 partitions of 512 cycles, 4 partitions of 256 cycles, and so on, down to 256 partitions of 4 cycles.

When emulation begins, instead of overwriting the entire trace memory, recording is restricted to the first partition. At some point, a trigger condition occurs. Then, the cycle delay occurs (for center- or post-trigger tracing). Then, recording in the first partition stops, and proceeds in the second partition. Each trigger condition which occurs thereafter will cause cvcles to be traced in the next partition. There will always be one partition per trigger. When the last partition has been filled, the first one will then be overwritten (as long as triggers keep occurring).

This process will continue until something occurs which stops emulation altogether; either:

1. A certain number of triggers have occurred (shown in Figure 4-1).
2. An Emulation Breakpoint or a manual break has occurred (shown in Figure 4-2). Emulation stops.
3. An Emulation Breakpoint has occurred, followed by a certain number of snapshots (shown in Figure 4-3).

Snapshots are numbered relative to the Emulation Breakpoint. If no Emulation Breakpoint is specified numbering is relative to the end of emulation; therefore, the oldest snapshot is -n, and the most recent snapshot is 0. Snapshot numbering is shown in figures 4-1 through 4-3.

If multiple snapshots are used in center- or post-trigger mode, and the trigger occurs AGAIN before the cycle delay is exhausted, recording in that snapshot will be stopped immediately, and will continue in the next one. This ensures that as much post-trigger information as possible is traced for each trigger. This is shown in Figure 4-4.

Any extra triggers which occur after the Emulation Breakpoint, but before the cycle delay is exhausted, will NOT cause recording in that partition to be stopped. This allows the user to gain full post-trigger information from the FIRST n triggers which occur. The extra triggers will appear in the trace display as "Triggers", and the numbering of the cycles will begin with the first trigger. This is shown in Figure 4-5.

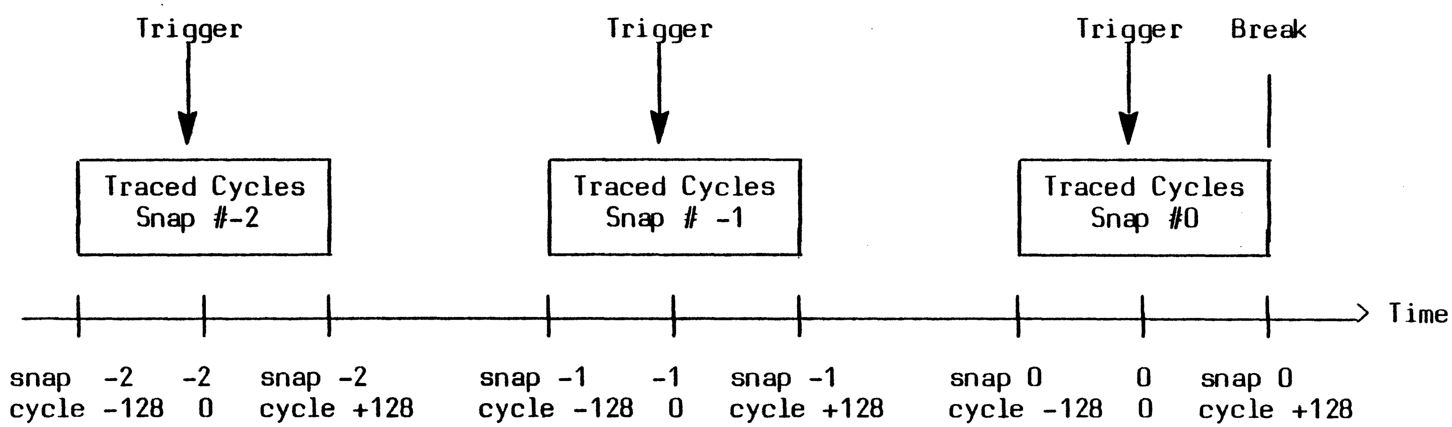
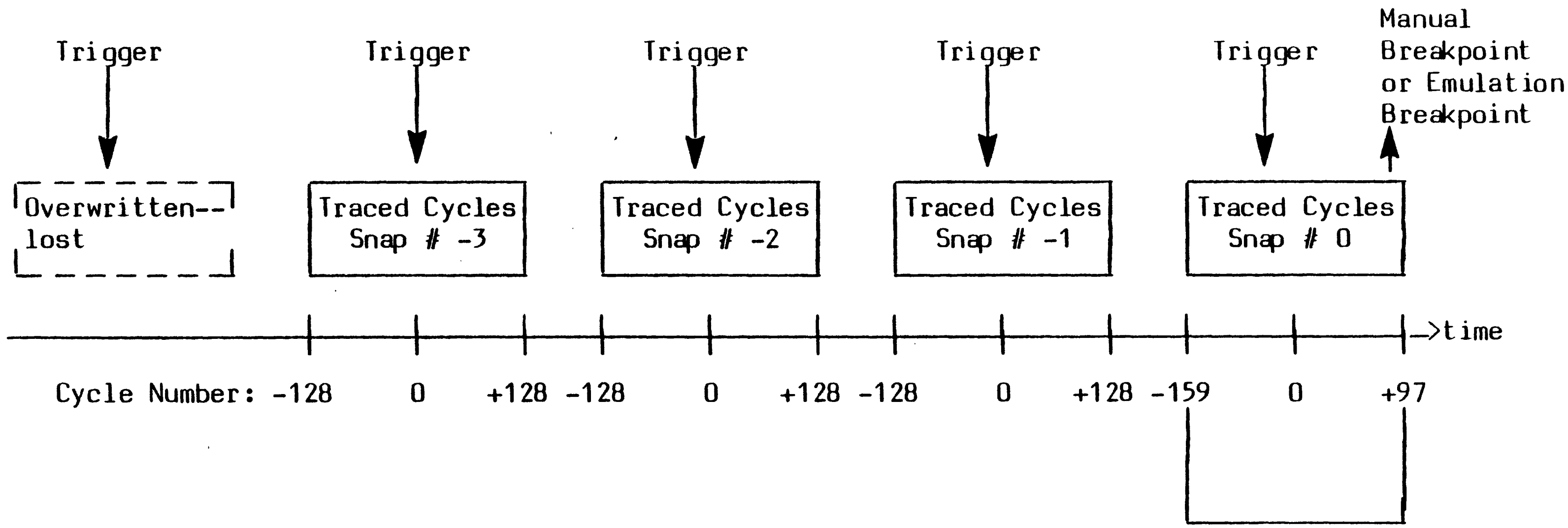


Figure 4-1. Multiple Snapshots, Breaking After a Certain Number of Triggers

(Break after 3)
(No qualification)
(Center trigger)
(4 partitions, 256 cycles)



More pre-trigger information and less post-trigger information because breakpoint occurred. Still 256 valid cycles "around" trigger.

Figure 4-2. Multiple Snapshots, Break with Breakpoint or Manual Break

(No qualification)
(Center Trigger)
(4 partitions, 256 cycles each)
(Breakpoint 97 cycles after a trigger)

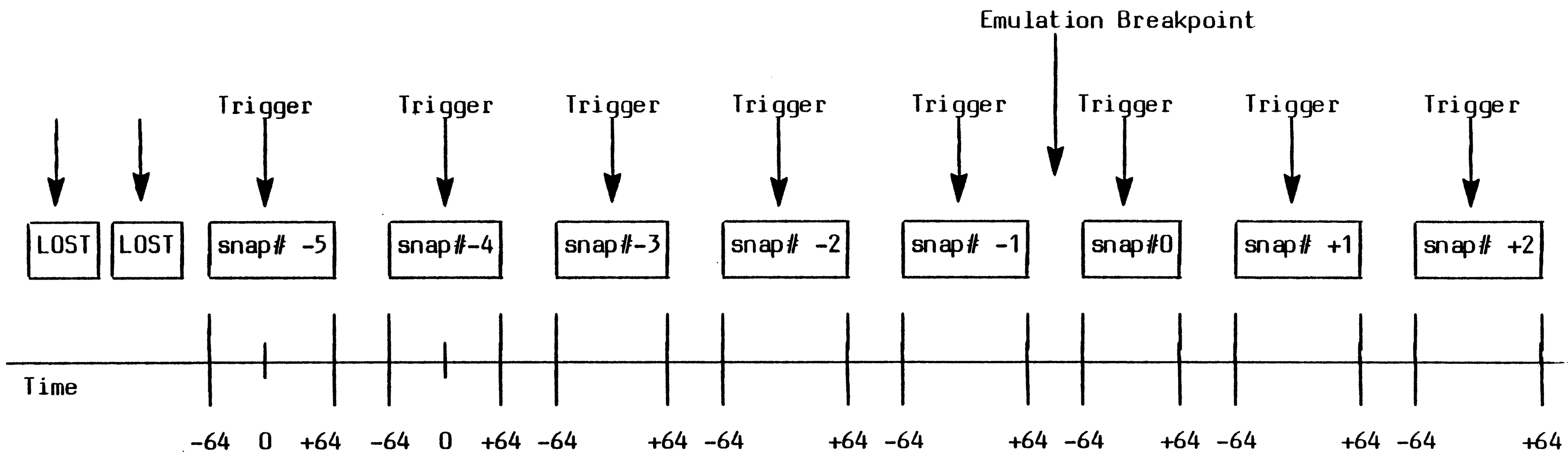


Figure 4-3. Multiple Snapshots, Emulation Breakpoint 3 Snapshots After Final Trigger

(No qualification)
(Center Trigger)
(8 Partitions 128 Cycles Each)

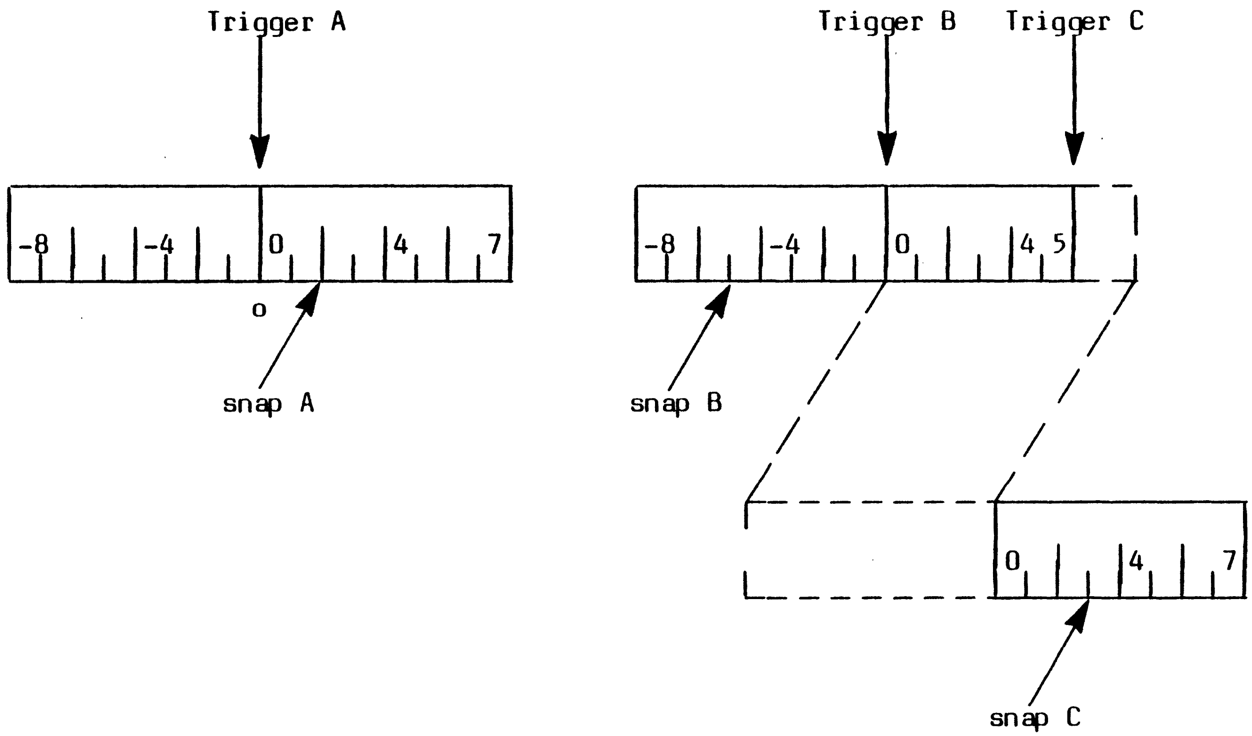


Figure 4-4. Triggers Occurring During a Cycle Delay

(partition size = 16)
(Snaps A, B and C will appear in the trace as shown above by the solid lines.)

(Dotted lines show that 8 cycles are still traced before and after each trigger.)

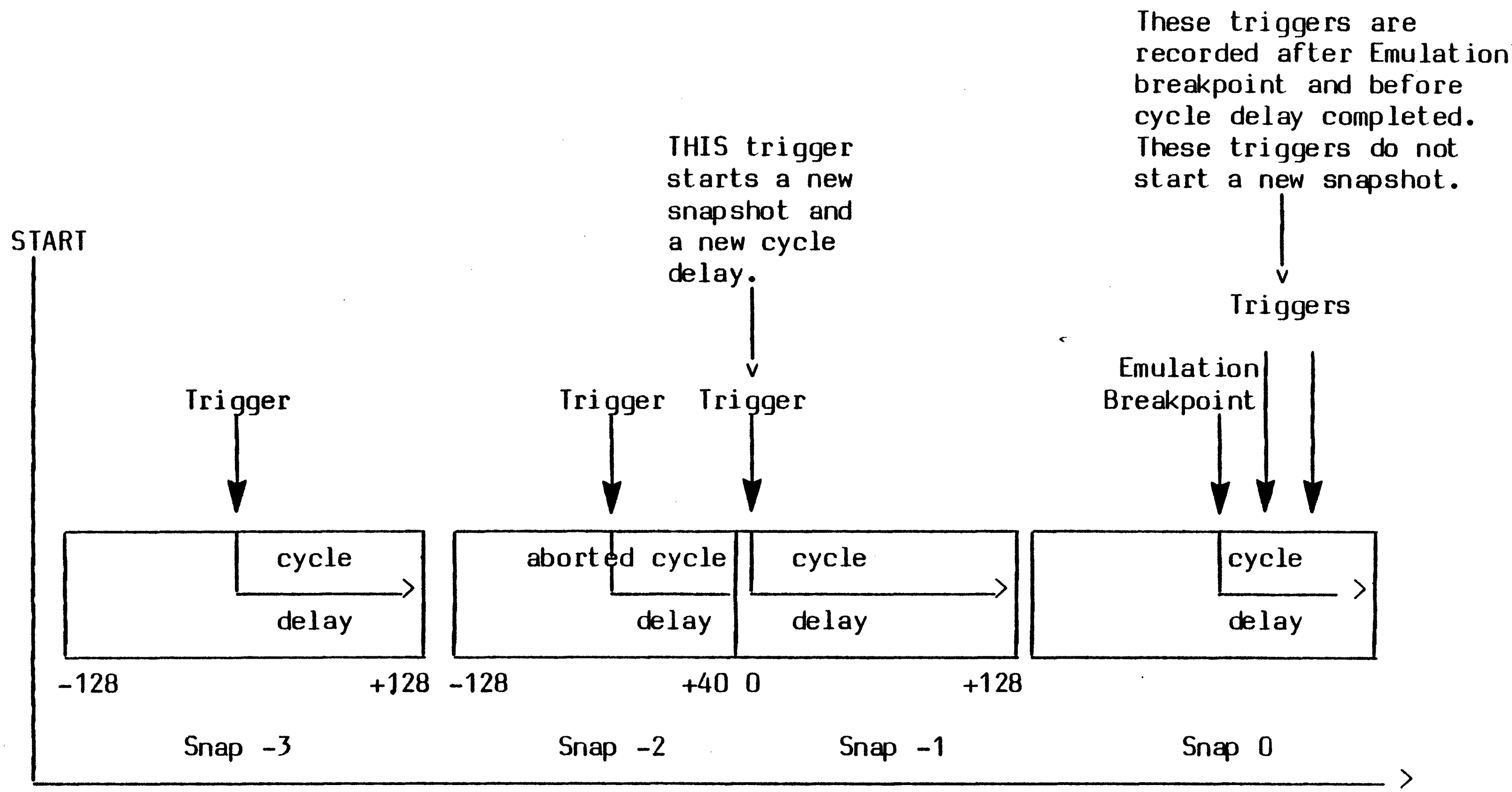


Figure 4-5. Triggers Occurring During the LAST Cycle Delay

(4 Partitions, 256 Cycles Each)
(Center Trigger (cycle delay =128)
(BREAK after 4 triggers)

4.4 TRACE COMMANDS

The command used to display the trace is:

HD [[<snapshot#>]<cycle#>]

This causes traced cycles to be displayed beginning with the one specified, and ending with the last valid cycle. If no cycles have been traced, nothing will be displayed. Cycles are displayed one screenful at a time: the user can terminate the trace display after any screenful simply by typing "Q".

"Snapshot#" specifies which snapshot to begin with, and "cycle#" specifies which cycle within the snapshot to begin with. Snapshots are numbered with respect to the Emulation Breakpoint as described previously. Cycles are numbered with respect to the trigger in each snapshot as described previously.

"Snapshot#" and "cycle#" are entered in hex with an optional minus sign. If "snapshot#" is omitted, the oldest valid snapshot is used as the default. If both Snapshot# and Cycle# are omitted, the oldest valid cycle and oldest valid snapshot is used as the default.

HP COMMAND

The HP (History Parameters) command is used to partition the trace memory, allocate trigger resources, qualify trace entries, and specify Emulation Breakpoints. A number of parameters (with previous values as default) are requested for input. If the user types <CR> in response to an HP prompt, the previous values will be used. Otherwise, the values input by the user will be used for HP setup and will be displayed when HP is typed again at the command level. Always the latest values will be displayed. To enter the HP screen from the command level, type HP<CR>. Errors are indicated by repeating the prompt with "***" in the first three columns.

-HP <CR>

1. "Number of Snapshots (nnnH)"

Initially, the number of snapshots is set to 1. Any value typed will be displayed immediately after <CR> to the nearest higher power of 2 along with the computed snapshot size. Note that the product of the number of snapshots and snapshot size will equal 1K. Valid entries are 1,2,4,8,10,20,40,80,100. Other entries are rounded to the nearest valid entry.

2. "Trace Trigger (M,A,I,C,N)"

Initially the Trace Trigger is set off ("x"). The options available to serve as a snapshot trigger are:

M = MAIN resource
A = ALTERNATE resource
I = INTERVAL resource
C = GENERAL PURPOSE COUNTER resource
N = No Trace Trigger

M,A,I and C may be logically OR-ed (i.e., M+A) in any combination.

3. "Cycle after Trigger (nnnnH)"

Initially, the default value is 0 (no cycles will be traced after the snapshot trigger). Any value typed by the user in response to this prompt will be taken as the number of cycles to be traced after the occurrence of the snapshot trigger.

4. "Trace Qualifier (M,A,I,X)"

Initially the Trace Qualifier is set off ("X"). All cycles are traced without qualification. Valid Trace Qualifiers include:

M = MAIN resource
A = ALTERNATE resource
I = INTERVAL resource
X = turn Trace Qualifier "off"

M, A and I may be logically OR-ed (i.e., A + I + M) in any combination.

5. "Emulation Breakpoint (M,A,I,C,S,X)"

Initially the Emulation Breakpoint is set off ("X"). The Emulation Breakpoint stops target system emulation and returns the user to the EMS monitor. The valid entries for Emulation Breakpoint are:

M = MAIN resource
A = ALTERNATE resource
I = INTERVAL resource
C = GENERAL PURPOSE COUNTER
S = After a number of snapshots
X = Turn Emulation Breakpoint "off"

A manual break (type <esc> on the terminal) will also cause an immediate Emulation Breakpoint. Valid entries (with the exception of "X") may be OR-ed (i.e., S+A would indicate an emulation is to be terminated after n snapshots OR when the ALTERNATE resource has become "True"). Values for the number of snapshots will be requested if the "S" option is selected. The request will be of form:

"Total number of snapshots (nnH)"

Initially the total number of snapshots is set to 1.

Valid entries for nnH are (1<nn<100H). If one or more of the valid options (M,A,I,C) is selected and the "S" option is not part of the selection, emulation will terminate when the selected resource goes true. In this case a further option is made available which allows the user to specify how many snapshots are to be recorded after the selected resource goes true. The form of this request is:

"Snapshots after Breakpoint (nnH)"

Initially nnH is set to 0, which causes the Emulation Breakpoint to occur 0 snapshots after the selected resource has gone true (i.e., right away). The value nnH must be less than 100H.

4.5 HOST INTERACTION

A Host may be attached to EMS 8000 for uploading or downloading. Phase I EMS software supports Zilog's MCZ-1 and MCZ-2 software development system as a host. EMS/Host communication baud rate is selectable by using a DIP switch on the rear panel. The host MCZ must contain (on floppy) the program "HOST." HOST must be invoked before doing a load or save. HOST can be invoked through Transparent mode.

EMS Host - Interaction commands include:

TR — Transparent Mode
Terminal talks directly with the connected MCZ-1
(At this point the user may wish to execute the program "HOST" on the MCZ if it is not already active.)
To exit, press <Break> key on the EMS terminal.
Terminal talks directly with the EMS monitor

L <filename> [<seg.#>]
Load (download) to Target System in segment <seg.#>.

SA <filename>

Save (upload) from Target System

Command will prompt with:

Entry point addr:

Low addr:

High addr:

"Low addr:" and "High addr:" will prompt repeatedly until the user types "0" or carriage return in response to "High addr:" In which case the command will prompt with:

Segment number:

Some of this data will be encrypted into the filename.

U <filename> Use Command File

The file <filename> will be interpreted as command input. The character sequence in the command file has the same effect as an identical keystroke sequence. The <Escape> character cannot be included in command files and functions normally. Macros and macro definitions may be included.

P Pause

This command in a command file will cause execution of the file to pause until any character except Q is typed on the console. Typing "Q" will abort the command file.

SECTION V
MACROS

5.0 MACRO DESCRIPTION

Simple macros are provided by EMS 8000. There are both system (pre-defined) and user defined macros. Parameters are allowed but are unnumbered (unlike RIO). A macro call within a macro acts like a "GO TO". This way, a macro can chain to another macro, or can chain to itself, causing everything to be done repeatedly. This is especially useful for test loops. A loop can be terminated by pressing the MONITOR button on the EMS front panel. Recursive macros will not work because of this method.

NOTE

> Macros can only be called from the emulator sub-command level (prompt "-").

Macros are executed by entering ".n p1;p2;...." where "n" is the macro name and "p1", "p2", ... are sequential parameters to the macro. **Semicolons must be used to separate parameters.** Macro names can be any character. System macros are macros already defined in the software. These will be listed later. All macros, including system macros, can be redefined (One exception is the the macro "D" which is used to define macros).

Macros are defined using the macro "D" with one parameter, the new macro name. Its prompt "mdef-" indicates macro definition mode. Commands are typed in exactly as they would be during normal operation. It is suggested you type in the commands before defining a macro to insure correct responses. Parameters entered on the macro invocation line replace the character "$" in the macro definition. The first "$" is replaced by the first parameter, the second "$" is replaced by the second parameter, and so on. If fewer parameters are typed in than are requested by the macro definition, the remaining "$"s will be replaced by a null string. To end a macro, type Control C (%03 ASCII) and a carriage return (%0D ASCII). Control will then return to the main loop (prompt "-").

5.1 SYSTEM MACROS

System macros are described below:

.B<addr> — Breakpoint Macro
Selects the main resource as the Emulation Breakpoint.
Also programs the Main resource with instruction fetch at address <addr>. Additional trigger conditions can be added after the address.

.C — Set Action Ram Macro
Initializes EMS with trace qualifies "off," Emulation Breakpoint set to "off," Emulation Breakpoint set to "off," and limits trace recording to the first snapshot.

.D — Define Macro
This is an internal macro. It is used for defining macros. It cannot be redefined. See above.

.G — Go Macro
Starts emulation (G) and on completion displays trace memory (HD).
Used mainly for Single Step and Next commands.

.J<addr> — Jump Macro
Loads the user's PC with "addr" and starts emulation (G).

.N<numb> — Next Macro
Selects the Alternate resource as the Emulation Breakpoint.
Programs the ALTERNATE resource with status = IF1. Sets the General Purpose Counter to <numb> where number > 1. Will step <numb> instructions.

.Q<trig. param.> Qualify Trace

Selects the ALTERNATE resource as the trace qualifier. Specific trigger parameters are entered with <trig. param>.

.S Single Step Macro

Selects the ALTERNATE resource as the Emulation Breakpoint. Programs the ALTERNATE resource with status = IF1. Used with .G.

SECTION VI

MEMORY MAPPER DOMAIN

6.0 INTRODUCTION AND CAPABILITIES

The EMS memory mapper may be used to substitute target system memory with high speed EMS static RAM. This feature is useful when developing programs destined for ROM and for substituting blocks of faulty or missing RAM. The mapper resolution is 2K bytes. Each 2K block can be declared unprotected, write-protected, data memory only, or nonexistent.

Many target systems distinguish between the various address spaces available on the Z8000. These address spaces include System/Normal, and Code/Data/Stack. The EMS memory mapper can be programmed to distinguish between these various address spaces so that an address which corresponds to one space (for instance, System Code space) can be mapped differently from another (for instance, Normal Data space).

6.1 THEORY OF OPERATION

The EMS Memory Mapper operates as a two level mechanism (see Figure 6-1). The first level is the segment-map. The segment-map contains 80H entries, one for each possible segment in the domain of the Z8000 memory space. Each entry in the segment-map contains information about how the FMS 8000 mapping logic will process the segment.

The user has the choice of either protecting the segment with one of six unique choices of EMS 8000 memory protection (which implies no static RAM substitution for this segment or referring to a second level of mapper programming called the Block Map.

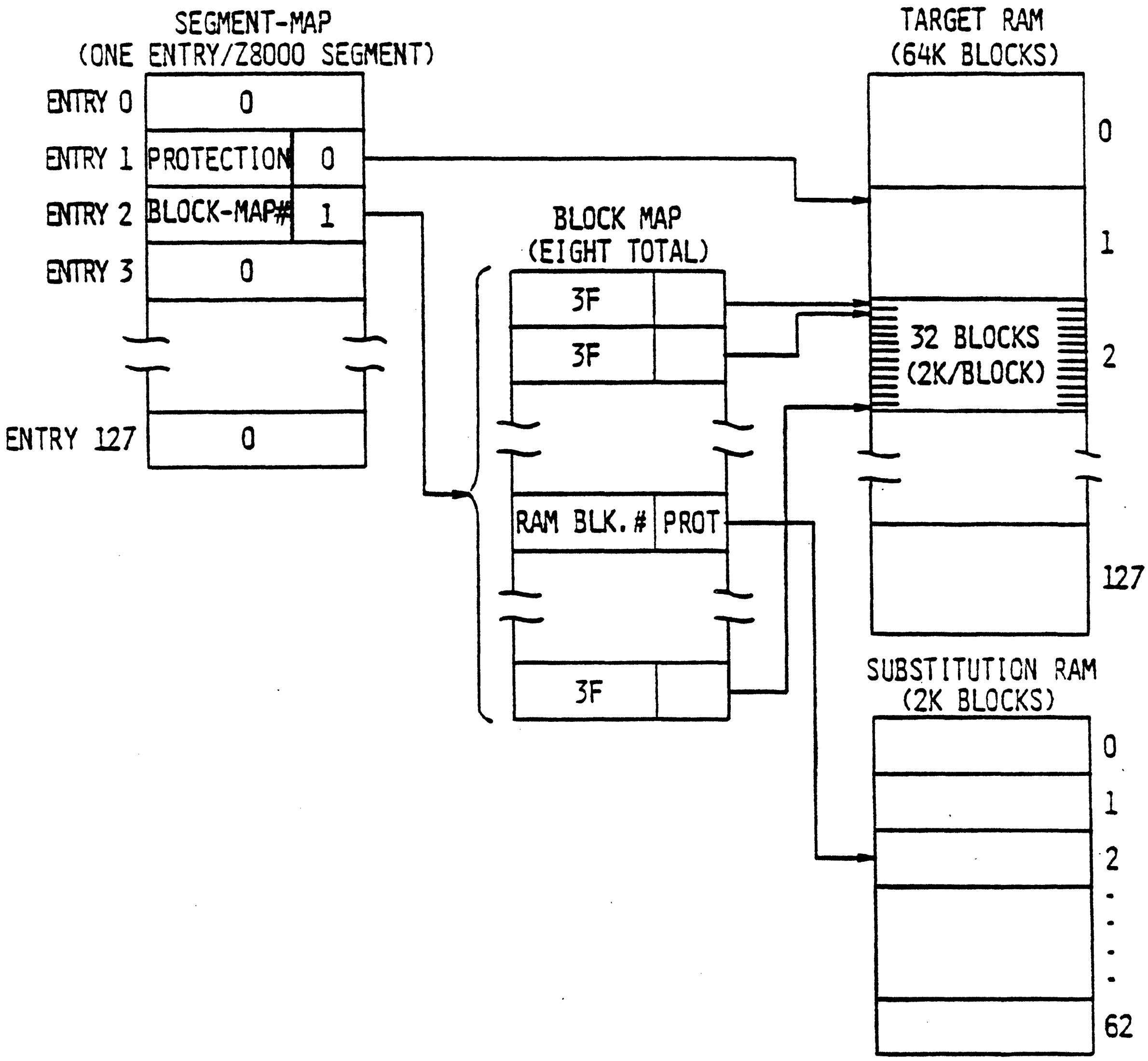


Figure 6-1. EMS 8000 Memory Mapping

The Block-Map allows substitution of EMS static RAM for target system memory. The EMS 8000 contains 8 block-maps. Each Block-Map contains 20H entries. Each entry represents a consecutive 2k byte address space within an associated segment. Each entry can substitute one 2K byte block of static RAM for each 2K bytes of user address space and provide the block with four different types of memory protection.

Commands within the Memory Mapper domain provide the user with the ability to use the features of the segment-map and block-map in a very flexible manner.

Another feature of the EMS 8000 memory mapper is the address distinction register. This register is programmed by the address space distinction mapper commands (6.2). With this feature the various address spaces of the Z8000 (System/Normal, and Code/Data/Stack) can be mapped differently from one another. Use of this feature, however, restricts the number of segments that may be processed by the EMS 8000 memory mapper.

When no distinctions are made, the EMS 8000 can provide block substitution for up to eight Z8000 segments (by virtue of the 8 Block-Maps). When a distinction between Normal and System accesses is required, the EMS 8000 can provide block substitution for up to four Z8000 segments. Likewise, when a distinction between Code and Data is required, the EMS 8000 can provide block substitution for up to four Z8000 segments. When a distinction between Code, Data, and Stack is required, the EMS 8000 can only provide block substitution for two Z8000 segments. If a distinction between Normal/System Code, Data, and Stack is required the EMS 8000 can only provide block substitution for one Z8000 segment.

To enter the Memory Mapper Domain (from the Main Command Domain only), type "MM". EMS will respond with the Memory Mapper Domain prompt, "m-". To exit from the Memory Mapper Domain (to the Main Command Domain) type "Q".

6.2 ADDRESS SPACE DISTINCTION MAPPER COMMANDS

NS Normal and System accesses are to be treated as different address spaces by the EMS mapper. The user must be careful to fully specify mapper commands which ask for a {N|S} input (i.e., select one). To eliminate the distinction between Normal and System, type the command "NS X".

CD Code and Data accesses are to be treated as different memory spaces by the EMS mapper. The user must be careful to fully specify mapper commands which have the format {C|D|S} (i.e., select Code or Data). To eliminate the distinction between Code and Data, type the mapper command "CD X".

CDS Code, Data, and Stack accesses will all be treated as different memory spaces by the EMS mapper. The user must be careful to fully specify mapper commands which have the format {C|D|S} (i.e., select one). To eliminate the distinction between Code, Data, and Stack; type the mapper command "CD X".

6.3 MEMORY MAPPER DOMAIN COMMAND SUMMARY

Command	Parameters	Description
MM		Enter the Mappable Memory Domain from the Main command level.
Q		Exit the Mappable Memory Domain and return to the Main command level.
DS		Display Segment-map (Z8001 only). Display available (AV) segments for mapping. Display the current distinctions assigned to the various Z8000 address spaces.

D NIS CIDIS <<seg #>>

Display the Block-Map of the segment number specified in seg# (this field is ignored for Z8002). If a distinction has been programmed for Normal and System then one of the option {N|S} must be selected. If there is no distinction between Normal and System then the {N|S} field must be ignored. If a distinction has been programmed for Code, Data, (or stack) then one of the options {C|D|S} must be selected. If there is no distinction between Code, Data, (or stack) then the {C|D|S} field must be ignored.

ES <seg #>.

Enters edit mode for the segment-map beginning at segment "seg #". The characters ▲ , <CR>, and Q provide scroll and exit capability. This command is available for the Z8001 only. In response to this command EMS will prompt for mapping and protection. Refer to the MS command syntax.

E N|S C|D|S <seg#>addr

Edits segment-block specified in <seg#> (for Z8002 <seg#> is ignored). If a distinction has been programmed for System and Normal memory accesses then one of the options {N|S} must be selected. If there is no distinction between Normal and System, then the {N|S} field must be ignored. If a distinction has been programmed for Code, Data, or Stack then one of the options {C|D|S} must be selected. If there is no

distinction between Code, Data, or Stack memory accesses then this field must be ignored. The mapper will respond with:
"<seg#>0000 -"
Mapping and protection features may be entered at this time in the format used by the M command (see description below).

MS <1st seg.#>[,<last seg.#>] (mapping [protection])

Sets a segment or group of segments in the segment-map beginning at "1st seg.#" and continuing through "last seg#" (if specified). Mapping and protection are specified as in the responses to the prompts in the ES command, i.e,:

mapping is one of (U|M|N)
where: U = User
M = Mapped
N = Nonexistent

For the User option only, protection may be provided. The format is:

protection is one of [S|N|C|D|W|D W]
where: S = System access only
N = Normal access only
C = Code access only
D = Data access only
W = Write protect: all accesses
DW = Write protect: Data accesses only

NOTE:

Only 8 segments can be mapped with no distinctions.

M {N|S} {C|D|S} <<seg.#>><1st addr.>[,<last addr.>]
(U|S|<BLOCK#>)[PROTECTION]

Sets a block or group of blocks in the segment-block map. "1st ADDR." and "last ADDR." are memory addresses in the specified segment <seg.#>. The block entry or entries are computed to the nearest 2K blocks.

EMS static RAM is substituted for target memory starting at "BLOCK#" if specified. If the "S" option is specified then EMS will assign static, RAM starting at an arbitrary 2K block boundary. If the "U" option is specified then no static RAM is assigned to the specified entries in the segment-block map:

Protection may be specified as follows

PROTECTION is one of [M|W|M W]

where: M = data memory only
W = write protected
M W = nonexistent

6.4 MISCELLANEOUS MEMORY MAP COMMANDS

X Clear all maps and deallocate all blocks;
Set the entire map, hardware and memory to User, Unprotected.

SECTION VII

TRIGGER DOMAIN

7.0 TRIGGER DESCRIPTION

There are four main parts to the Trigger Module; the 3 trigger resources (Main, Alternate, and Interval), and the General Purpose Counter.

The 3 trigger resources allow the user to generate events on certain patterns or sequences of patterns. The trigger resources can be operated independently or qualify one another, i.e.

o The Alternate resource can reset and arm the Main resource.

o The Interval resource can reset and/or arm/disarm the Alternate resource.

The Main resource can recognize a sequence of 5 patterns where each pattern supports either address ranging or 2 address logical operations.

The Alternate resource can recognize 1 pattern. The Interval resource can recognize a sequence of 8 patterns. These 8 patterns can be further subdivided into separate ENABLE and DISABLE sequence.

The General Purpose Counter (GPC) can be used for counting or timing. It has 3 major modes of operation that will be described in detail in the "GM" command. In one of the counting modes (Pass Count) the GPC can generate an event.

The events generated by the 3 resources, and by the GPC can be allocated to the Trace Module Control functions by the "GM" and "HP" commands. Section 4.4 describes the command and the GM command is described in Section 7.6.

The events generated by the 3 resources can also be counted by the General Purpose Counter. (See "GM" command.)

The Trigger Domain is entered by typing "T" at the main command level. This domain contains all the commands necessary to set up the three resources and the General Purpose counter. It also allows the user to display the current setup. Typing a "Q" exits the user from the Trigger Domain. Typing an "X" resets the trigger domain to an initial state.

7.1 TRIGGER CONVENTIONS

The symbols used by the trigger commands AT, MT, IT are listed below:

n is the pattern number

@ is (>|<|=|#) meaning >=, <=, =, ≠. It specifies the relationship between the field and the compare value.

% is (hex digit |X|(BBBB)) where X is a don't carehex digit and B is (0|1|X) where X is a don't care binary digit

<SS> is segment Number. SS is (|%|%%) where % is defined above

The form of address parameters is <SS>%%%%. The segment <SS> can be explicitly specified. In an alternate form, a 5 or 6 hex digit address can be used. The low order 4 digits will be interpreted as the offset. The high offset 1 or 2 digits will be interpreted as the segment number.

Example:

12345 equivalent to <1>2345

& means A1 and A2 fields are ANDed (only for MT command)
OR means A1 and A2 fields are ORed (only for MT command)

A1 is an Address field (only for MT command)

A2 is an Address field (only for MT command)

AD is an Address field

DA is Data Field

EE is the external fields concatenated

E1 is the first external field

E2 is the second external field

ST is Z8000 cycle status and sym is [IF1 IFN INT REF IO SIO NMI NVI VIA SGT DAT STA] these agree with the first 3 letters in the trace output (see 7.2.1 SYMBOLS FOR STATUS FIELD).

CP specifies CPU cycles only

DM specifies DMA cycles only

RE specifies READ

WR "WRITE""

BY "BYTE""

WO "WORD""

NO "NORMAL""

SY "SYSTEM""

XC resets the whole control field to don't care (ST, CP or DM, RE or WR, BY or WO, NO or SY)

XX resets the whole pattern n

7.1.1 Symbols for Status Field

The first 3 characters agree with the TRACE display.

INT	0	0000
REF	1	0001
IO	2	0010
SIO	3	0011
SGT	4	0100
NMI	5	0101
NVI	6	0110
VIA	7	0111
DAT	8	1000
STA	9	1001
IFN	C	1100
IF1	D	1101
IA		01XX
IF		110X

7.2 MAIN TRIGGER

MAIN STATUS

MS

This command displays the status of all information relevant to the MAIN resource. "Don't care" bits are indicated by showing the main status field logically "AND-ed" with the entered value. For example the value CDF (111x) is shown CDFF&FFFE.

MAIN RESET

MT XX

This command resets the entire pattern array for the MAIN resource. The 'xx' must be included for it to take place.

MAIN TRIGGER

MT n [A1@<SS>%%%%] [OR|&|] [A2@<SS>%%%%]
[DA@%%%%] [EE@%%%%] [E1@%%] [E2@%%]
[ST=(%|sym)] [CP|DM]
[RE|WR] [BY|WO] [NO|SY] [XC] [XX]

This command edits the pattern n (for Main Trigger, 1 <= n <= 5) of the MAIN resource.

The MAIN resource can specify "in-range" or "out-of-range" for addresses. In-range is selected with A1 > <addr 1> & A2 < <addr 2> where <addr 1> < <addr 2>.

Out-of-range may be selected with A1 < <addr 1> + A2 > <addr 2> where <addr 1> < <addr 2>.

For example:

MT 1 A1 > 1234 & A2 < 5678 ST = DAT
will trigger on all DATA accesses in the memory range 1234 to 5678.

The OR option may be used even if ranging is not desired. In this case, the A1 and A2 address fields will act like independent fields.

MAIN LENGTH

ML = n

$(1 \leq n \leq 5)$

This command sets the length of the main sequence.

This length should agree with the number of patterns programmed with the MT command.

Default = 1.

MAIN COUNTER

MC = n

$(0 \leq n \leq 255)$

If n = 0 the MAIN COUNTER is disabled.

If the MAIN COUNTER is enabled $(1 \leq n \leq 255)$ the entire main sequence must occur within n cycles or else the sequence will be reset to look for the first pattern.

Default = disabled.

7.3 ALTERNATE RESOURCE

ALTERNATE STATUS

AS

This command prints out the ALTERNATE resource status. If the ALTERNATE resource qualifies the MAIN resource this information is also displayed.

ALTERNATE RESOURCE

AT

[AD@%%%%][SE@%%][DA@%%%%][E1@%%][E2@%%][EE@%%%%]
[ST=(%|sym)][CP|DM][RE|WR][BY|WO]
[NO|SY][XC][XX]

Sets the pattern for the ALTERNATE Trigger. The symbols are the same as they are for the MAIN resource EXCEPT for XX which resets the pattern to the initial value.

AD	Address field	Initial value = '=XXXX'
SE	Segment field	Initial value = '=XX'
DA	Data field	Initial value = '=XXXX'
E1	Probe E1 field	Initial value = '=XX'
E2	Probe E2 field	Initial value = '=XX'
ST	Status field	Initial value = '=X'

Addresses may be entered alternately as AD@<SS>%%%%.

7.4 INTERVAL RESOURCE

INTERVAL STATUS

IS

This command displays the INTERVAL resource status. The Interval resource may be programmed to rearm itself (reset), and may qualify other resources. This information is also displayed with the IS command.

INTERVAL RESET

IT XX

This command resets the entire pattern array for the INTERVAL resource. The 'XX'' must be included for it to take place.

INTERVAL RESOURCE

IT n

[AD@%%%%][SE@%%][DA@%%%%][E1@%%][E2@%%][EE@%%%%]
[ST=(%|sym)][CP|DM][RE|WR][BY|WO]
[NO|SY][XC][XX]

This command edits the pattern 'n' of the INTERVAL Trigger. Symbols are the same as for MAIN resource EXCEPT for 'XX' which resets the pattern 'n' to the initial value.

n	Pattern number to be modified ($1 \le n \le 8$)	
AD	Address field.	Initial value = '=XXXX'
SE	Segment field.	Initial value = '=XX'
DA	Data field.	Initial value = '=XXXX'
E1	Probe E1 field.	Initial value = '=XX'
E2	Probe E2 field.	Initial value = '=XX'
ST	Status field.	Initial value = '=X'

Addresses may be entered alternately as AD@<ss>%%%%.

INTERVAL LENGTH

IL

[E = len1][D = len2]

This command sets the length of the ENABLE and DISABLE interval.

len1 = Length of the ENABLE pattern.
Initial value = '1'
len2 = Length of the DISABLE pattern.
Initial value = '1'

The following must hold: $0 \leq$ len1, len2; and $1 \leq$ (len1 + len2) ≤ 8. If the sum of the lengths is too large, the two lengths are reset to lengths of '1'.

The lengths entered (len1 and len2) should agree with the number of patterns programmed with the IT command.

If len2 = 0 then no DISABLE sequence exists.

INTERVAL REENTRANT

IR (Y|N)

This command decides if the INTERVAL resource will be reentrant.

Y The INTERVAL resource will be reentrant.
N The INTERVAL resource will not be reentrant.

Initial value = 'Y'

7.5 RELATION COMMANDS

ALTERNATE QUALIFIES

AQ (Y|N)

This command decides if the ALTERNATE resource resets and arms the MAIN resource.

Y The ALTERNATE resource resets and qualifies the MAIN resource.

N The MAIN resource is independent of the ALTERNATE resource.

Initial value = 'N'

INTERVAL QUALIFIES

IQ (O|M|A|2)

This command decides which trigger the INTERVAL arms.

O ALTERNATE and MAIN are independent of INTERVAL.

A INTERVAL qualifies ALTERNATE, MAIN is independent.

M INTERVAL qualifies MAIN, ALTERNATE is independent.

2 INTERVAL qualifies ALTERNATE and MAIN.

Initial value = 'O'

INTERVAL (ENABLE RESETS) MAIN

IM (Y|N)

This command allows the ENABLE portion of the Interval resource to reset the MAIN resource.

Y Each ENABLE sequence restarts the MAIN resource.
N MAIN resource is independent of the ENABLE point.

Initial value = 'N'

7.6 GENERAL PURPOSE COUNTER (GPC)

The General Purpose Counter can function in 3 modes:

- o In Stopwatch Mode, the GPC can count sample-bus cycles or T-states. This provides the function of timing.

- o In Event Counter Mode, the GPC can count events generated by the 3 resources.

- o In Pass Mode, the GPC can generate an event similar to the main, Alternate, and Interval resources discussed above. This event can be used to control Trace Module functions with the HP command.

GPC Mode

GM = (O|P|S|C)

This command sets the operating mode of the General Purpose Counter.

Each of the modes have a separate set of prompts. "GM = O" will disable the counter. The long count is cleared each time emulation begins. The modes "P", "S", and "C" stand for "Pass Count", "Stopwatch", and "Event Counter".

GPC Status

GS

Displays the current mode of operation for the counter.

SYMBOLS

The following symbols are used for the "Mode" examples:

"#1="	Shorthand for manual examples. Refer to command format for corresponding question.
"E"	is the ENABLE EVENT of the INTERVAL resource
"D"	is the DISABLE event of the INTERVAL resource.
"P"	is the PASS COUNT event
"M"	is the Main Trigger event
"A"	is the Alternate Trigger event
"S"	START event
"F"	FINISH event

7.6.1 Mode "P" (Pass Counter)

In this mode the output of the GPC will become true after the nth occurrence of a complex event; or become true only after the nth occurrence of a complex event AFTER the mth occurrence of the ENABLE sequence in the Interval resource. The complex event can be the event generated by the Main resource, Alternate resource, or the logical "OR" of both.

Command Format:

1. ENABLE PASS COUNT ($0 \leq n < 64K$)?
   (This is the value m in the description above.)

2. M/A PASS COUNT ($0 \leq n \leq 64K$)?

   (This is the value n in the description above. M/A is the complex event described above.

3. Event to be Pass Counted?
   (The responses can be M, A, or M + A where M stands for MAIN resource, A stands for ALTERNATE resource, AND M + A is the logical OR of the two resources.
   (if (EN Pass = 0) and (M/A Pass $\neq$ 0) then)

4. SHOULD M/A PASS COUNT BE WITHIN ONE INTERVAL (Y/N)?
   (In the case where counting only begins.)

Examples:

1. IF answer to command format questions are:
   #1 = 0, #2 = 5, #3 = M, #4 = N

   ```
   M  M  M  M  M
               P
   ```

   The pass count will go true after 5 counts of the MAIN resource.

2. If answer to command format questions are:

   #1 = 0, #2 = 5, #3 = M + A, #4 = N

   ```
   M  A  A  M  A
               P
   ```

   The pass count will go true after 5 counts of either the MAIN resource or the ALTERNATE resource.

3. #1 = 3, #2 = 5, #3 = A,

   ```
   E  E  A  E  A  A  A  A  A
                           P
   ```

   The pass count will go true after 5 counts of the ALTERNATE resource, but only after 3 ENABLE sequences of the INTERVAL resource.

4. #1 = 1, #2 = 5, #3 = A, #4 = Y

   ```
   E  A  A  A  D  A  E  A  A  D
   ```

   (no pass count event)

   Every ENABLE resets the GP counter to look for 5 occurrences of the ALTERNATE resource. In this example 5 occurrences have not occurred.

5. #1 = 3, #2 = 5, #3 = A

   ```
   E  E  E  A
   ```

The ALTERNATE resource pass count has never reached 5. The GP counter will not go true.

7.6.2 Mode "S" (STOPWATCH)

In this mode the GPC operates as a stopwatch measuring the time between events generated by the Main and Alternate Triggers. These can be single or disjoint intervals, i.e., the stopwatch can measure time elapsed between the START event to the FINISH event, reload on every START event, or restart and reload if the START event occurs again after the FINISH event. Time can be measured in sample-bus cycles or T-states.

Command Format:

1. What is the Start Event?
   (response is M for Main resource, A for ALTERNATE resource, M + A for MAIN or ALTERNATE resource)
2. Finish ALTERNATE, MAIN or None (A|M|N)?
3. Time Units in T States or Cycles (T/C)?
4. Cumulative Time (Y/N)?
   (if N then)
5. Reload on New Begin (Y/N)?
6. Restart Allowed After First Finish (Y/N)?

Examples:

1. #1 = M, #2 = A, #3 = T, #4 = Y

   M M A M M A M M A A

   Elapse time is cumulative between Start Event = M and Finish = A. The counter pauses after each Finish = A, and resumes upon each Start Event = M.

2. #1 = M, #2 = A, #3 = T, #7 = N, #5 = N, #6 = N

M M A   M M A   M M A

Elapse time is measured between first occurrence of Start Event = M and first occurrence of Finish = A.

3. #1 = A, #2 = M, #3 = T, #4 = N, #5 = Y, #6 = N

A A M   A A M   A A M

Elapse time begins on first Start Event = A. Elapse time is reset (reloaded) on new Start Event = A (New Begin). Elapse time stops at first Finish = M.

4. #1 = A, #2 = M, #3 = C, #4 = N, #5 = Y, #6 = Y

A A M   A A M   A A M

Elapse time (measured in cycles) begins on first Start Event = A. Elapse time is reset (reloaded) on new Start Event = A New Begin . Elapse time stops at first Finish = M. A new count begins on next Start Event = A. If the previous count is not stored in the Trace memory (see HP command), this count is lost and only the last count is stored in the GP counter.

5. #1 = A, #2 = M, #3 = T, #4 = N, #5 = N, #6 = Y

```
A A M   A A M   A A M
```

Elapse time begins on first Start Event = A. Elapse time stops at first Finish = M. A new count begins on next Start Event = A. If the previous count is not stored in the trace memory (see HP command), this count is lost and only the last count is stored in the GP counter.

7.6.3 Mode "C" (EVENT COUNTER)

In this mode the GPC can count the number of complex events across several intervals defined by the ENABLE|DISABLE sequences of the INTERVAL resource, or the GPC can count only the number of complex events in the last interval before an EMULATION BREAKPOINT. The complex event may be generated by the MAIN resource, the ALTERNATE resource, or the logical "OR" of both.

Command Format:

1. Event to be counted?
   (response is M for MAIN resource event, A for ALTERNATE resource event, M + A for MAIN or ALTERNATE resource)
2. Cumulative Count (Y/N)?

Examples:

1. #1 = A, #2 = Y

```
E A A A D E A A A D
  1 2 3     4 5 6
```

In example 1, the count begins on the first occurrence of the event to be counted (ALTERNATE resource) after the ENABLE sequence is recognized. The counter pauses after the first DISABLE sequence is recognized. The counter resumes after the next ENABLE sequence and stops after the subsequent DISABLE sequence.

2. #1 = A, #2 = N

```
E A A A D E A A A D
  1 2 3     1 2 3
```

Count begins on the first occurrence of the event to be counted (ALTERNATE resource) after the ENABLE sequence is recognized. The counter stops after the first Disable sequence. The counter is reset to 1 on the first occurrence of the event to be counted in the next interval (ENABLE sequence recognized). The previous count is lost if it is not stored in Trace Memory (see HP command).

APPENDIX A

GLOSSARY

APPENDIX A
GLOSSARY

BEGIN	The condition on which the counter will start counting.
BREAK	Action of suspending execution of the TARGET processor by EMS. A BREAKPOINT refers to point where BREAK occurs. The term halt is not used to avoid confusion with the HALT instruction. BREAK is performed by the OP code substitution method (JR$ is forced on the bus during OP code fetch).
complex event	Can be any of the following: MAIN resource, ALTERNATE resource, or MAIN resource ORed with ALTERNATE EVENT.
concurrent pass count	Pass count on both Enable and either MAIN or ALTERNATE.
cycles	CPU cycles of the target system.
EMULATION	A portion of real time execution by the target microprocessor where the sample bus is being monitored. The EMULATION starts on GO by the user and can stop when - a break point is encountered. - Tracing is done. - <esc> is typed. - A memory protection violation is detected.
ENABLE count	See load values.

EVENT	A sequence of recognitions or matches on the sample bus involving only one trigger.
event count	The number of times the input to the counter GPE has become true.
FINISH	The condition at which the counter will stop counting.
IDLE STATE	State of the target system after execution of the processor has been suspended by EMS. Target system's registers, memory and I/O ports can be accessed only while the processor is in the IDLE STATE.
load values	The values that are loaded into the counter. There are two load values, one when INTERVAL is true (Main load value), one when INTERVAL is false (ENABLE load value).
MAIN LOAD VALUE	Same as Main load value. See load values.
Main Count	See load values.
occurrences	See event count.
pass count	Allows a certain number of complex events to occur before a Point is generated.
POINT	Refers to a sample where a certain real time decision is being made.
qualifies	As it relates to triggers, Triggers recognize patterns only when qualified.
reloaded	The value of the MAIN or ENABLE is loaded into the counter.

reset by	The counter stops counting.
restarted	The counter is enabled to continue counting. This does not mean that a new value is reloaded.
Snapshot	A certain number of trace samples defined in relation with one point only. These samples are consecutive ones if there is no Trace Qualification.
Snapshot Trigger	The trigger allocated to define the snapshot point.
start condition	Now is called BEGIN.
stop condition	Now is called FINISH.
T states	Clock cycles of the target system.
Target system	The user's system being developed with EMS used in place of the Z8000 CPU. Reference is also made to TARGET processor, TARGET memory, etc...
time	Time in cycles or T states between two points.
Trace Qualifier	This (ANDed with INTERVAL if selected) is used to tell the trace when cycles should be recorded.

APPENDIX B

COMMAND SUMMARY

B.0 SESSION SETUP AND DEBUG COMMANDS

%HOST	(4.6 RIO command to start HOST program)

NOTE:
Section numbers are given along with brief description.

X	(4.6 exit HOST driver while in transparent mode)
-TR	(4.6 talk to RIO HOST driver)
-QU	(4.3 restarts EMS session)
-U <RIO filename>	(4.6 use a file-HOST must be active)
-P	(4.6 pause in a command file till key pressed)
-L <RIO filename> <seg>	(4.6 save target memory--HOST must be active)
-SA <new RIO filename>	(4.6 save target memory--HOST must be active)
-PM	(4.3 print macro)
-PN	(4.3 print macro character names)
-.D <name>	(5.1 define a macro)
-R	(4.2 display registers)
-R (R0...R15:PSAP:REFR:PC:FCW:N14:N15:)	(4.2 display/modify registers)

NOTE:

Enter	to display previous register values
Enter **.**	to re-display current register values
Enter **Q**	to stop prompting

-RT	(4.2 force registers to target)
-(IB:I:IW) <port address>	(4.2 input from target port)
-(OB:O:OW) <port address> <value>	(4.2 output to target port)
-(SB:S:SW) <address><value string>	(4.2 set target memory)

NOTE:
<address> is <<seg>><offset> of Z8001 e.g. <4E>45A0

-M <to address> <from range>	(4.2 move target memory)

NOTE:
<range> is either <begin address>, <end address> or
<begin address> <length>

-(FB:F:FW) <range><value string> (4.2 fill target memory)

-(DB:D:DW:DL) <address> (4.2 display/modify target memory)
-(DB:D:DW:DL) <range> (4.2 display a block of target memory)
-(CB:C:CW) <address><range> (4.2 compare target memory)

B.1 EMULATION SETUP DISPLAY AND COMMANDS (includes mappable memory)

-SE (4.3 display emulator setup)

```
NMI      Enabled          Break                  Independent
NVI      Enabled          Cycles Traced          Data Cycles Only
VI       Enabled          External Field Mode    Probes
SEGT     Enabled          Memory Type            Sys Data
STOP     Enabled          I/O Type               Normal
RESET    Enabled          Mem MAP                Enabled
BUSREQ   Enabled          System Addresses       Logical
WAIT     Enabled

   WB  Write Protection Break          Disabled
   MB  Mode Protection Break           Disabled

Wait States:      System: 0  User: 0  Mapped: 0
```

Figure B-1. Sample Display

-WA [0...7] (4.3 set number of wait states)

-(DI:EN) NMI:NVI:VI:SEGT:STOP:RESET:BUSREQ:WAIT:BREAK:WAIT:MAP:WB:MB
(4.3 disable/enable interrupts)
-CT (A:D) (4.3 trace all/data strobe cycles)
-E2 (I:P:G) (4.3 define external probe)
-MT (N:S) (C:D:S) (4.3 default target memory decode)

NOTE:

The MT command effects commands which access target memory, e.g., the load, save, compare, display, move, and fill commands.

-(IO:IT) S:N (4.3 default I/O decode--special/normal)

NOTE:
The I/O command effects the IB, IW, OB, and OW commands.

```
-MM                                         (6.3 enter mappable memory domain)
m-X                                         (6.4 don't decode on C:D:S or N:S)
m-NS                                        (6.2 decode target N:S pin)
m-NS X                                      (6.2 don't decode target N:S pin)
m-CD                                        (6.2 decode target code:data status)
m-CDS                                       (6.2 decode target
                                            code:data:stack:status)
m-CD X                                      (6.2 don't decode target code:data:stack
                                            status)
```

NOTE:
The NS, CD, and CDS commands can change the syntax of the DS, MS, D, and M commands. Also, the inputs to the ES and E commands are changed. These changes are denoted by the notation.

```
m-MS <1st seg#>[,<last seg#>]               (6.3 target segment nonexistent)

m-MS <1st seg#>[,<last seg#>] M [00...07]   (6.3 map target segment to EMS)
m-MS <1st seg#>[,<last seg#>] U [C:D:N:S:W:D W]
                                            (6.3 give unmapped memory a mode)
m-ES <seg#>                                 (6.3 interactive form of m-MS)
m-DS                                        (6.3 display segment map, and N:S C;DLS
                                            decodes)
m-M  N:S   D:D:S   <<seg#>> <1st addr>[,<last addr>] U [M:M W:W]
                                            (6.3 block is unmapped with optional
                                            mode)
```

NOTE:
The <<seg#>> notation indicates that the outer angle brackets are typed while the inner brackets are part of the meta notation.

NOTE:
The notation is equivalent to the [ ] notation if the corresponding distinction is in effect. Otherwise, the field disappears.

```
m-M  N:S   C:D:S   <<seg#>> <1st addr>[,<last blk>] S [M:M W:W]
                                            (6.3 block is mapped with optional mode)
m-M  N:S   C:D:S   <<seg#>> <1st addr>[,<last addr>] <block id> [M:M W:W]
                                            (6.3 an identifiable EMS block is mapped
                                            to target)
m-E  N:S   C:D:S   <<seg>>                  (6.3 interactive form of m-M)
m-D  N:S   C:D:S   <<seg>>                  (6.3 display block mapping for segment)
m-Q                                         (6.3 quit memory mapping domain)
```

B.2 MONITORING EMULATION

-TS (4.3 trigger and state status)

Main Pattern: 0006 Interval Pattern: 0001

Event	Occurred
Main Counter Overflowed	N
Main Sequence Recognized	N
Enabled First	N
Interval Has Been Entered	N
Alternate FIRST	N
Snapshot and Address Counter:	00 0000
Trace Address:	00 0000
Cycle Delay Counter:	01C0
Segment Delay Counter:	00

Figure B-2. Example of the EMS Display During Emulation

-HD [[<snapshot number>] <cycle number>] (4.5 display trace)

NOTE:
These numbers may be negative.

GO	(4.3 start emulation)
.G	(5.1 go macro)
.J <address>	(5.1 jump and go macro)
.N n	(5.1 next instruction macro. 1 < n < 65535)
.S	(5.1 step 1 instruction macro)

NOTE:
.G will execute using the conditions set up by .N or .S, whichever was last entered. For example, the sequence .S, .G, .G, .G will single cycle three instructions.

.B <address> (5.1 set breakpoint macro)

B.3 TRACE MEMORY SETUP

-HS (4.5 trace status)

Segment and Address Counter:	00 0000
Ten Bit Address:	0000
Cycle Count:	01C0
Number of Segments	0001
Cycle Delay:	0000
Trace Mode Register:	000B
Finish Instruction:	Y
Segment Defined By Event	N
Override Break	Y

Figure B-3. Trace Status Display--Sample

-HP (4.5 trace setup)

	Old	New
		--
Number of Snapshots (hex: nnn)	1 (size = 400)	--
Trace Trigger (M, A, I, C, N)	N	--
Trace Qualifier (M, A, I, X)	X	--
Trace in Interval Only (Y, N)	N	--
Emulation Breakpoint (M, A, I, C, S, N)	N	--

Figure B-4. Prompts Given in Response to HP Command-Sample

-.C

-.C (clear HP screen)

-.Q <trigger parameters> (5.2 trace only specified cycles)

Notes on HP parameters:

1. Snapshot size can be from 4H to 400H cycles (4 to 1024 decimal). The number of snapshots is 100H to 1H (256 to 1 decimal). The actual number of cycles stored in a snapshot will depend on the number of cycles between snap points.

2. The position of the snap point in the snapshot is controlled by specifying the number of cycles to be traced after the snap point.

3. The Emulation Breakpoint may cause an immediate break.

4. All cycles of the last instruction traced can be stored, even if the trace parameters specify to stop before an instruction boundary.

5. The enable-disable interval is useful for restricting traced cycles to tested subroutines. Another application would be in sampling.

B.4 TRIGGER COMMANDS

-T (7.0 enter trigger domain)

t-GS (7.6 general purpose counter status)

```
GP Counter Status

          Counter Mode:  Disabled
          Long Count:  FFFFFFFFFFFF
          Main Count:  FFFF Enable Count:  FFFF


          Feature                     Enabled

          Event Mode                     Y
          GP Non-Retriggerable           Y
```

Figure B-5. General-Purpose Register Display--Sample

t-MS (7.2 main trigger status)

```
M:    Address 1  :  Address 2  :   Data  :   Cycle  :Rw:Bw:Sw:Ext 1:Ext 2
```

Figure B-6. Main Trigger Status Display--Sample

t-AS (7.3 alternate trigger status)

```
Status for the Alternate Trigger

       ADDR    DATA    CTL    E1    E2    E1E2
<XX>   =XXXX   =XXXX   =XX                =XXXX    A
```

Figure B-7. Alternate Trigger Display--Sample

t-IS (7.4 enable-disable status)

```
Status for the Interval Trigger

       ADDR    DATA    CTL    E1    E2    E1E2
  <XX> =XXXX   =XXXX   =XX                =XXXX    1  E

=XX    =XXXX   =XXXX   =XX                =XXXX    2  D

Trigger Resets on Disable        Y
Enable Resets Main               N
Interval Qualifies:
Interval Reentrant               Y
```

Figure B-8. Internal Trigger Display--Sample

t-GM=(P:S:C) (7.6 select GP counter mode)

NOTE

P--delay effect of trigger till nth pass
S--time between main and alternate triggers
C--count main/alternate triggers that occur within interval

t-AQ (Y:N) (7.5 alternate arms main trigger)
t-IQ (O:M:A:2) (7.5 interval arms alternate/main)
t-IM (Y:N) (7.5 enable resets main trigger)
t-MT=XX (7.2 initializes main trigger)

t-MT 1:2:3:4:5 [A1@<seg1><offset1>] [OR:&:] [A2@<seg2><offset2>] [DA@<word>]
[EE@<word>] [E1@<byte>] [E2@<byte>]
[CP:DM] [RE:WR] [BY:WO] [NO:SY] [XC] [XX]
[ST=0...15:INT:REF:IO:SIO:SGT:NMI:NVI:VIA:DAT:STA:IFN:IF1:IA:IF]
(7.2 specify main trigger)

NOTE:

See 7.1: @ is = or <or>

CP--CPU cycles only
DM--DMA cycles only
RE--read
WR--write
BY--byte
WO--word
NO--normal
SY--system

```
t-ML=1:2:3:4:5                               (7.2 depth of main trigger sequence)
t-MC = 1...255                               (7.2 main sequence cycle limit)
t-AT XX                                      (7.3 initialize alternate trigger)
t-AT [SE@<seg>] [AD@<offset>] [DA@<word>]
     [EE@<word>] [E1@<byte>] [E2@<byte>]
     [CP:DM] [RE:WR] [WO:BY] [SY:NO] [XC] [XX]
     [ST=0...15:INT:REF:IO:SIO:SGT:NMI:NVI:VIA:DAT:STA:IFN:IF1:IA:IF]
                                             (7.3 specify alternate trigger)
t-IT XX                                      (7.4 initialize interval trigger)
t-IT 1...8 [SE@<seg>] [AD@<offset>] [DA@<word>]
    [EE@<word>] [E1@<byte>] [E2@<byte>]
    [CP:DM] [RE:WR] [BY:WO] [NO:SY] [XC] [XX]
    [ST=0..15:INT:REF:IO:SIO:SGT:NMI:NVI:VIA:DAT:STA:IFN:IF1:IA:IF]
                                             (7.4 specify interval trigger)
t-IL [E=<length1>] [D=length2>]              (7.4 length 1 + length2 LE 8)
t-IR (Y:N)                                   (7.4 interval one time only or
                                             multi-pass)
```

B.5 BUILT IN MACROS (See Section 5.1)

```
.B <address>                                 (breakpoint macro)
          T                                  (trigger domain)
          MT 1 XX AD=$                       (1st main trigger at address "$")
          Q                                  (quit trigger domain)
          HP                                 (setup trace)
          1                                  (select 1 snapshot)
          N                                  (no Trace Trigger required)
          <CR>                               (trace Qualifier doesn't matter)
          <CR>                               (trace in Interval doesn't matter)
          M                                  (set Emulation Breakpoint to MAIN)

.C                                           (trace everything until manual break)
          HP                                 (enter HP mode)
          1                                  (select 1 snapshot)
          N                                  (no trace trigger)
          X                                  (trace qualifier off)
          N                                  (trace everywhere)
          N                                  (no emulation breakpoint)

.D <macro name>                              (define a macro)

.G                                           (go macro)
          G                                  (go)
          HD                                 (display trace after emulation
                                             breakpoint)

.J <address>                                 (jump macro)
          R PC                               (change program counter)
          $ Q                                (specify address and quit)
          G                                  (go)
```

```
.M                                  (move memory from target to mappable
                                    memory macro)
        MM                          (enter memory map domain)
        MS 0 M                      (map segment 0)
        M <0>4000,7FFF 0            (place to copy target memory)
        Q                           (quit memory map domain)
        M <0>4000 <0>0, 3FFF        (move target contents)
MM                                  (enter memory map domain)
        M <0>0, 3FFF 0              (replace target memory with mapped
                                    memory)
        M <0>4000,7FFF U            (unused part of segment mapped to user)
        Q                           (quit mapper domain)
        EN WB                       (allow write protect breaks)

.N                                  (emulate for next n instruction)

        T                           (enter trigger domain)
        GM=P                        (GP counter set to pass mode)
        0                           Do not use pass count on ENABLE)
        $                           (Select pass count for MAIN resource)
        M                           (Use MAIN resource to count IF1's)
        N                           (Do not restrict count to 1 interval)
        MT 1 XX  ST = IF1           (Set up MAIN resource for IF1)
        Q                           (leave trigger domain)
        HP                          (enter HP screen)
        <CR>                        (ignore number of snapshots)
        N                           (turn trace trigger off)
        <CR>                        (ignore number of snapshots)
        <CR>                        (ignore trace in interval)
        C                           (select counter for emulation breakpoint)

.Q <trigger parameters>             (trace particular cycles macro)
        T                           (enter trigger domain)
        AT XX $                     (set up alternate trigger)
        RQ = A                      (trace only when alternate trigger
                                    matches)
        Q                           (quit trigger domain)

.S                                  (single-step macro)
        T                           (enter trigger domain)
        MT 1 XX  ST=IF1             (MAIN resource watches IF1's)
        Q                           (exit trigger domain)
        HP                          (enter HP screen)
        <CR>                        (ignore number of snapshots)
        N                           (select no trigger)
        X                           (select no trace qualifier)
        N                           (do not restrict count to interval)
        M                           (use MAIN resource to break)

.U                                  (remove write protect macro)
        MM                          (enter mapper domain)
        M <0>0000,3FFF 0
        Q                           (leave mapper domain)
```

```
.W                                        (write protect macro)
        MM                                (enter mapper domain)
        M <0> 0000,3FFF 0 W               (write protect)
        Q                                 (leave mapper domain)
```

APPENDIX C

ERROR CODES

TRIGGER DOMAIN ERROR CODES

A	Illegal Parameter
B	Bad Digit
C	Illegal Command
D	Missing Equal Sign
E	Number is out of range
F	Bad select operand

APPENDIX D

INSTALLATION

APPENDIX D
INSTALLATION

D.1 CABLE CONNECTORS

D.1.1 CPU Pod Cables to EMS 8000

There are two six foot cables permanently attached to the CPU Pods, Z8001 and Z8002. These permanently attached cables, marked #1 and #2 at the 50-pin connector plugs, connect to the front-panel of the EMS 8000 as follows:

1. Plug the top cable, #1, into the EMS front-panel 50-pin jack labeled CPU POD CABLES, #1. Cable #1 is keyed and should be oriented so that the triangle (Pin 1) on the EMS front panel matches the arrow (Pin 1) on the #1 cable.

NOTE

Cable #1 plugs into the Pod connector J2 (farthest one back on the pod). When the pod is closed, Cable #1 is on the top. This cable is keyed.

2. Plug the bottom cable, #2, into the EMS front-panel 50-pin connector jack labeled CPU POD CABLES, #2 (i.e., to the right of Cable #1).

NOTE

Cable #2 plugs into the Pod connector J1 (farthest front on the pod). When the pod is closed, cable #2 is on the bottom. This cable is keyed.

D.1.2 Emulator Cable to CPU Pod and to Target System

Since the Emulator cable provides the communications link between the Target and the CPU pod, it is sometimes referred to as the Target cable. This cable can be identified by the blue wire along one side of the cable which indicates Pin 1. The size of the DIP connectors for the Z8001 and the Z8002 CPU pods differ. Follow the appropriate set of instructions for connecting this cable to your particular EMS CPU pod and Target.

Emulator Cable for the Z8001 CPU Pod:

1. Plug the 50-pin DIP connector into the Z8001 CPU Pod, matching Pin 1 (indicated by the blue wire along one side of the cable). Only 48 pins are used for the Z8001 communications with the Target.

2. Plug the other end of the Emulator cable into the Target ensuring that Pin 1 is in the proper position. (Pin 1 is marked on the underside of the DIP connector).

Emulator Cable for the Z8002 CPU Pod:

1. Plug the 40-pin DIP connector into the Z8002 CPU Pod, matching Pin 1 (indicated by the blue wire along one side of the cable). All 40 pins are used for the Z8002 communications with the Target.

2. Plug the other end of the Emulator cable into the Target ensuring that Pin 1 is in the proper position. (Pin 1 is marked on the underside of the DIP connector).

D.1.3 Terminal Cable to EMS 8000

An RS-232 cable with 25-pin connectors on each end are used to connect EMS 8000 to the CRT Terminal. Phase I of EMS 8000 supports only the ADM-31 Terminal which should be connected as follows:

1. Connect one end of the RS-232 cable to the 25-pin connector jack (J4) located on the upper right of the rear panel of the EMS 8000 unit (see Figure D.1).

2. Connect the other end of the RS-232 cable to the center jack on the ADM-31 Terminal to be used with EMS 8000.

NOTE

EMS 8000 Phase I requires the ADM-31 Terminal to be set for 9600 baud. This is accomplished by setting the ADM-31 rear-panel switches (lower right) to the number 14.

3. See Section D.2 for instructions on setting EMS 8000 rear panel switches which control the EMS baud rate.

D.1.4 Host Cable to EMS 8000

Any RS-232 cable with 25-pin DIP connectors on both ends can be used to connect the host to the EMS 8000.

1. Connect one end of the RS-232 25-pin plug into the Host RS-232 jack (J3) located on the top left of the EMS 8000 rear panel.

2. Connect the other end of the RS-232 25-pin plug into the MCZ-1 Terminal Port (J-6) located on the MCZ-1 rear panel.

3. Refer to Section D.2 for instructions on setting EMS 8000 rear-panel switches used for EMS and Hot communications.

D.2 SWITCH SETTINGS

1. Arrange the EMS rear-panel switches to the appropriate settings for Phase I as indicated in Tables D-1 and D-2.

Table D-1. Phase I Switch Settings for RS-232 (MCZ-1 Host)

DIP SWITCH	BIT SWITCH	SETTING	COMMENTS
S2	Bits 1 and 2	OFF	See Figure D-1 for
S2	Bits 4 through 8	ON	DIP Switch S-number
S1	Bits 1 through 8	OFF	assignments

2. Set Bits 5 through S3 to represent the hexadecimal value which corresponds to the desired baud rate (see Table D-2).

NOTE

The recommended speed for the MCZ-1 is 38.4K baud. For a longer distance (up to 50 feet) reduce the speed to 19.2K baud.

Table D-2. Phase I Host Baud Rates and Corresponding Hexadecimal Values for Switch Setting

Baud Rate	Hex Value	S3 (Bits 5 through 8) Switch Settings	Comments
300	0	0 0 0 0	
600	1	0 0 0 1	
1200	2	0 0 1 0	
2400	3	0 0 1 1	
4800	4	0 1 0 0	
9600	5	0 1 0 1	
19.2K	6	0 1 1 0	
38.4K	7	0 1 1 1	Recommend Host Baud Rate (38.4K)

Bit 5
Bit 6
Bit 7
Bit 8

3. When using mappable memory, configure the boards as indicated in Table D-3:

Table D-3. Jumper Placements for Mappable Memory

Mappable Memory #1	Place jumper between E1 and E2
Mappable Memory #2	Place jumper between E2 and E3

D.3 CHECKOUT OF INTERNAL CONNECTIONS

1. Ensure that the EMS 8000 boards are plugged into the unit in the proper position indicated by Table D-4, Board Positions for EMS 8000 Phase I.

Table D-4. Board Positions for EMS 8000 Phase I

Name	Plug into Slots from Top
CCU	slot 2
Trigger I	slot 4
Trigger II	slot 5
Probe Interface	slot 6 (optional)
Trace	slot 7
Emulator	slot 8
Mappable Memory #1	slot 9 (optional)
Mappable Memory #2	slot 10 (optional)

2. Ensure that all cable connections, switch settings, and board positions correspond to the instructions given in Sections D.1 and D.2.

3. Verify the following:

   a. Power supply module is present and plugged into backplane and fans.

   b. Rear panel Printed Wire Assembly (PWA) is present and connected to the backplane with 50-pin flat cable. To ensure proper connection, colored side of cable must correspond to pin 1 on both sides.

c. Front panel is present and connected to the backplane with the 16-pin flat cable. To ensure proper connection, colored side of cable must correspond to pin 1 on both sides.

--NOTE--

Before using the test socket of EMS to provide a clock, the test board must be properly installed for the particular CPU pod being used with your EMS.

Installation and/or verifying that the test board is correctly installed for your EMS configuration is very simple:

- With the front panel removed, the test board is installed vertically on the far left side of the unit.

- One end of the test board is marked P1; this connection end is inserted into the test socket if a Z8001 pod is being used.

- The other end of the test board is marked P2; this connection end is inserted into the test socket if a Z8002 pod is being used.

In either case, to insert the test board correctly, all components will be on the left side of the board (as viewed from the front of the EMS unit.)

D.4 AC MODULE CONFIGURATION

If your EMS 8000 has an international ac module, contact your Zilog FAE (Field Applications Engineer).

D.5 POWER UP SEQUENCE

Assuming that the installation procedures in Sections 2.0 through 3.0 have been properly completed, the EMS 8000 should be powered up in the following manner:

1. Turn AC on EMS 8000 first, or on EMS and the Target simultaneously, if they are connected to the same power strip.

2. Turn AC on Target. The target can subsequently be turned off by itself (with the EMS 8000 still on) because of the special target power sense which tristates all signals going to the target system when powered off.

3. Turn AC on Host and Terminal. The AC on Host and Terminal can be turned on and off without worrying about sequence.

4. It is recommended that a Lear Siegler ADM 31 be used with Phase I EMS.

D.6 BOOTING THE SYSTEM UP

1. The following procedure files are included on the EMS Software Disk. (The MCZ-1 and MCZ-2 media is available initially; additional Host's procedure files will be added later.)

    a. HOST: This file must be active in order for the Host to respond to the EMS commands. It is not required when in Transparent mode (i.e., when the user is using the Host independently of EMS).

    b. EMS.A, EMS.B, and EMS.C: These files contain the EMS control program to the downloaded into EMS.

    c. Other files: Additional files contain user notes or additional information.

    The files which must be present on the Host's master unit are as follows: HOST, EMS.A, EMS.B and EMS. C.

- With an MCZ-1 or MCZ-2 floppy drive system, the diskette may be inserted in the second floppy drive or the files may be transferred to the system diskette using normal RIO procedures.

- With an MCZ-1 hard disk system, the files should be transferred to the hard disk. The user should use his normal back-up procedures for the EMS software diskette.

2. Having powered up the systems, press the RESET button on EMS (located on the left side of the front panel, next to the MONITOR switch).

   The following message should appear on the terminal screen:

   ```
   EMS Phase 1 (Date)
   Memory Test in Progress...
   ```

   Periods will appear in sequence indicating the test is progressing. At the conclusion of the Memory Test, (10-15 secs.) the following will appear:

   ```
   You are in Transparent mode. EMS.A, EMS.B, and EMS.C should be on host.
   Invoke program host on host computer and the hit the BREAK key.
   ```

3. If the Host is not initialized, it should be booted using its normal procedure. (If using MCZ-1 or MCZ-2 as the Host, push RESET on the host, then carriage return.)

   If the Host has been initialized, nothing is required. The user may wish to verify that the Host operating system is presented ready for commands (entering <CR> to get a RIO prompt can be done on an MCZ system).

4. After receiving the Host prompt (to ready the Host for downloading), enter the command "HOST" followed by a carriage return. Press the terminal BREAK key or the EMS MONITOR button and the EMS operating system will be loaded from the host, downloading EMS.A, EMS.B, and EMS.C into EMS. The download process take about 30 seconds at 35.4K baud, and proportionately longer at slower baud rates.

   At the completion of downloading, the following message will appear:

   EMS 800 (1 or 2) Phase I Monitor Version X.X

5. Set the EMS terminal to **all CAPS** ; you are ready to use EMS.

   In order to verify the proper attachment to the target system the follow simple sequence should be followed:

   Enter R(CR). The register should be displayed. A potential error message is: BAD CLOCK fix it.

   This can happen from the following conditions:

   - the Z8000 does not get the proper clock
   - a bad connection exists between the TARGET system and EMS
   - no power is present to the target system
   - a possible faulty CPU pod on EMS, or
   - a possible faulty Emulator board on EMS

6. If the registers are displayed correctly, the FCW and the PC should be initialized to the values that the user's program would set them to at reset (memory locations <0>0002, 0004, and 0006 for Z8001). This should be done before any initial startup of emulation, so that the start will appear to be a run from reset to the user system. Otherwise, the CPU will treat the initial values fetched from these locations as instructions rather than the values intended for the FCW and PC. (After starting emulation, a reset of the user system should

allow the CPU to fetch the FCW and PC correctly, but the delay from start of emulation to reset could cause execution in erroneous portions of memory. It is much safer to cause emulation to start with the correct FCW and PC already loaded).

7. After initializing the FCW and PC, enter the command "G" (CR) in order to start the emulated processor. The target system should be reset (if required to initialize circuitry in the target system) and checked for correct operation. A possible error message on the EMS at this stage, is "TIMEOUT! This shouldn't happen." This would indicate one of the CPU control lines (BUSRQ, RESET, or WAIT) is stuck in the target or the emulator is not functioning.

8. To stop emulation, press the ESC (escape) key on the EMS terminal.

D.7 TRANSPARENT MODE

It is possible to access all the functions of the host from the EMS terminal by putting the system in Transparent mode either when EMS is booted and goes into the automatic transparent mode or returning to transparent from EMS mode.

D.7.1 Changing from EMS Mode to Transparent Mode

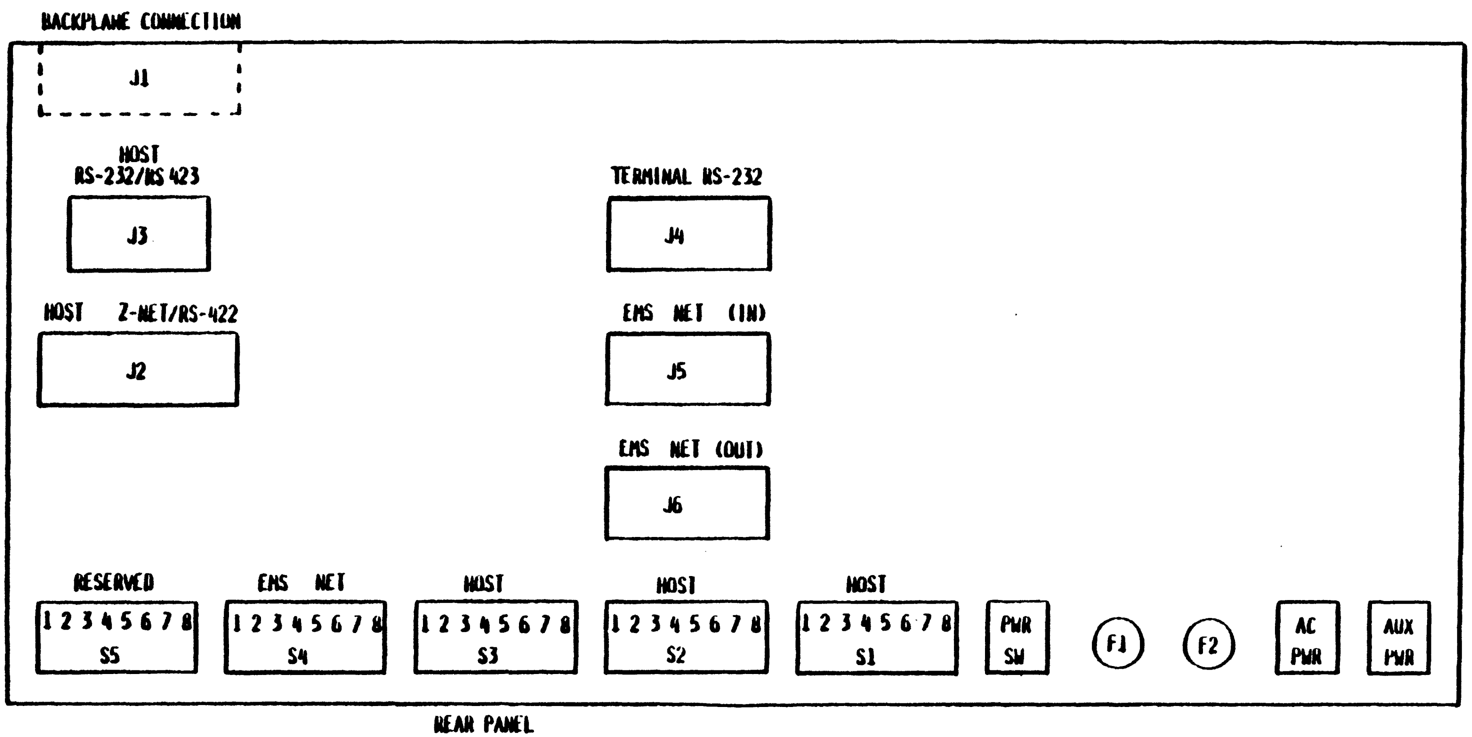
1. If the HOST procedure is active and EMS is active, type in the following key sequence after the prompt (-):
   -TR
   X <CR> (This will not be echoed.)

D.7.2 Changing from Transparent Mode to EMS Mode

1. Type in the following key sequence after the prompt (%):
   %HOST

2. Then press the break key on the EMS terminal. The following prompt should appear:

   (You are now in the EMS 8000 monitor.)



KEY:

J1 INTERNAL BACKPLANE CONNECTION (DOES NOT APPEAR ON REAR PANEL)

J2, J5, J6 } NOT USED IN PHASE I.

S4 RESERVED

S5 NOT USED IN PHASE I

Figure D-1. EMS 8000 Rear Panel

APPENDIX E

A FAMILIARIZATION SESSION

A FAMILIARIZATION SESSION

The following is a very simple succession of commands which will help familiarize the user with EMS Phase I, and does not require a working target. Simply plug the Emulator cable into the ZIF socket in the front panel which supplies a clock. (Follow the instructions given under installation). A more detailed tutorial is available on the EMS Software diskette and can be printed by copying the files EMS READ.* to a printer.

The intent of the following program is to write a simple looping pattern into mappable memory and then to execute it. A breakpoint, a trace qualification strategy, etc...can be set in this program.

NOTE

All command lines of this familiarization section end in carriage return, except for ESC (escape key).

-MM	(enter mappable memory domain)
if a Z8001 is present type:	
m-MS 0 M	(map SEG, 0 as mappable)
m-DS	(display segment mapping)
m-M <0> 4000,6000 S	
m-D <0>	
else if a Z8002 is present type:	
m-M 4000,6000 S	(map address 4000 of target to 6000 of MAPPABLE MEMORY)
m-D	(A display of the mapping scheme will appear. Note that the U indicates unmapped memory.)
then for both Z8001 and Z8002 type:	
m-Q	
-F 5000 50 E800	(Fill 50 locations with Jump Relative to Next Location)
-D 5010	
E8F7	(Jump Relative 8 words back)
Q	
R PC	

PC <00> 0000 - 5000	(Set PC to 5000)
FCW 8000 - 4000	(Set FCW to 4000)
-G	(Emulation is now started and the program should be looping between 5000 and 5010.)
<esc> (escape key)	(Step emulation and check that the program was in fact looping in the right range by verifying PC value.)
-T	(Enter Trigger Domain)
t-MT 1 AD=6000 ST=IF1	(Set breakpoint at address=6000, with status=opcode fetch (1F1).)
t-MS	(Display the Main Trigger Status)
t-Q	(Exit Domain)
-G	(The feedback screen, continuously updated by the Z80 CCU during emulation, is now flashed at you. Notice that the MAIN Sequence never happens (Main Sequence recognized; N) and this is not surprising since the address 6000 is never seen in our loop program. Note that the other triggers are seen because they are all programmed as don't care.)
<esc>	(Escape key)
-T	(Enter Trigger Domain)
t-MT 1 AD=5000	
t-Q	
-G	(The feedback screen now shows: Main sequence recognized: Y and this is normal since address 5000 is now seen.)

<esc>

We are now going to cause a break at address 5000:

-HP	(Go and program the trace)

	OLD	NEW
Number of snapshots (hex:nnn)	1 (size 400)	-<CR>
Trace Trigger (M,A,I,C,N)	N	-<CR>
Trace Qualifier (M,A,I,X)	X	-<CR>
Trace in Interval Only (Y,N)	N	-<CR>
Emulation Breakpoint (M,A,I,C,S,N)	N	-M<CR>

The above set up should cause a break to happen on each occurrence of location 5000.

-G	The machine should break right away showing the values of the registers. Note that the PC indicates 5002 which is the address of the instruction following the one with AD=5000.)
-HP	Leave everything the same, but set up a trace qualification of all cycles and undo the break on MAIN TRIGGER by typing the following responses:

	OLD	NEW
Number of snapshots (hex:nnn)	1 (size 400)	-<CR>
Trace Trigger (M,A,I,C,N)	N	-<CR>
Trace Qualifier (M,A,I,X)	X	-<CR>
Trace in Internal Only (Y,N)	N	-<CR>
Emulation Breakpoint (M,A,I,C,S,N)	N	-N<CR>

-G	(The machine is now filling its trace memory)
<esc>	(Escape key)

-HD (History Display)

You can now look at the instructions in the order in which they have been executed. Note the looping from 5000 to 5010. The status display says IF1 since the only thing that happens in that loop is opcode fetches. The four characters after that C S W R indicate CPU cycles. System mode, Word accesses, and Read cycles. The opposite would be DMA cycles, Normal mode Byte accesses and Write cycles. The external probe fields, E1 and E2 don't show anything meaningful unless they have been hooked up.

APPENDIX F

EMS 8000 PHASE I SPECIFICATIONS

EMS 8000 PHASE I SPECIFICATIONS

Clock Rate:	500 kHz to 6MHz
CRT	RS-232 connector with RS-423 buffering. Terminal Baud Rate: 9600 baud
Host Computer	RS-232 connector with RS-423 buffering. or RS-422 interface Host Baud Rate: 300 to 19.2 k (selected by rear-panel switch)
EMS Network Daisy Chain	In and Out - 25-pin D connectors containing Send data and Group Break signals.

Electrical:

Voltage*/Maximum Load Current:	90-140/10A 180-260/05A
Frequency	47 - 63 Hz
Phase	1
Plug	Nema Ref. L5-15P
Receptacle	Nema Ref. L5-15R
Power Cord Length	10.0 feet
Fuse	10A/5A

* Voltage conversion must be performed by authorizeds Zilog personnel. Contact your local Zilog Sales Office to request this service.

Dimensions:

Width	20.0 in.	(50.8 cm)
Depth	25.0 in.	(63.5 cm)
Height	7.0 in.	(17.8 cm)

Unit Weight: 65.0 lbs. (29.5 kg)

Shipping Weight: 80.0 lbs. (36.4 kg)

Transportation/Storage Environment

Temperature	-50 to 125 degrees C
Humidity	5% to 95% (No condensation)

Operating Environment: 0 to 40 degrees C

Humidity 10% to 90% (No condensation)

Front Panel: RESET and MONITOR button. CMU POD CABLES #1 and #2 sockets, Self-Test Pass and Fail light indicators. CPU Pod Test socket. Ext Probe 1 and Ext Probe 2 and Phys Addr Low and Phys Addr High indicators.

Rear Panel: AC power connector, fuse and spare fuse holder, auxiliary AC power (out), on-off power switch, rear-panel access door.

Zilog Sales Offices

West

Sales & Technical Center
Zilog, Incorporated
1315 Dell Avenue
Campbell, CA 95008
Phone: (408) 446-4666
TWX: 910-338-7621

Sales & Technical Center
Zilog, Incorporated
18023 Sky Park Circle
Suite J
Irvine, CA 92714
Phone: (714) 549-2891
TWX: 910-595-2803

Sales & Technical Center
Zilog, Incorporated
15643 Sherman Way
Suite 430
Van Nuys, CA 91406
Phone: (213) 989-7485
TWX: 910-495-1765

Sales & Technical Center
Zilog, Incorporated
2918 N. 67th Place #2
Scottsdale, AZ 85251
Phone: (602) 990-1977

Sales & Technical Center
Zilog, Incorporated
1750 112th Ave. N.E.
Suite D161
Bellevue, WA 98004
Phone: (206) 454-5597

Midwest

Sales & Technical Center
Zilog, Incorporated
890 East Higgins Road
Suite 147
Schaumburg, IL 60195
Phone: (312) 885-8080
TWX: 910-291-1064

Sales & Technical Center
Zilog, Incorporated
28349 Chagrin Blvd.
Suite 109
Woodmere, OH 44122
Phone: (216) 831-7040
FAX: 216-831-2957

South

Sales & Technical Center
Zilog, Incorporated
2711 Valley View, Suite 103
Dallas, TX 75234
Phone: (214) 243-6550
TWX: 910-860-5850

Zilog, Incorporated
7113 Burnet Rd.
Suite 207
Austin, TX 78757
Phone: (512) 453-3216

Technical Center
Zilog, Incorporated
1442 U.S. Hwy 19 South
Suite 135
Clearwater, FL 33516
Phone: (813) 535-5571

East

Sales & Technical Center
Zilog, Incorporated
Corporate Place
99 South Bedford St.
Burlington, MA 01803
Phone: (617) 273-4222
TWX: 710-332-1726

Sales & Technical Center
Zilog, Incorporated
110 Gibraltar Road
Horsham, PA 19044
Phone: (215) 441-8282
TWX: 510-665-7077

Sales & Technical Center
Zilog, Incorporated
240 Cedar Knolls Rd.
Cedar Knolls, NJ 07927
Phone: (201) 540-1671

Zilog Sales
3300 Buckeye Rd.
Suite 401
Atlanta, GA 30341
Phone: (404) 451-8425

United Kingdom

Zilog (U.K.) Limited
Babbage House, King Stre
Maidenhead SL6 1DU
Berkshire, United Kingdom
Phone: (628) 36131
Telex: 848609

France

Zilog, Incorporated
Tour Europe
Cedex 7
92080 Paris La Defense
France
Phone: (1) 788-14-33
TWX: 611445F

West Germany

Zilog GmbH
Zugspitzstrasse 2a
D-8011 Vaterstetten
Munich, West Germany
Phone: 08106 4035
Telex: 529110 Zilog d.

Japan

Zilog, Japan KK
TBS Kaikan Bldg.
3-3 Akasaka 5-Chome
Minato-Ku, Tokyo 107
Phone: (03) 587-0528
Telex: ESSOEAST J22846

Zilog, Inc. 10460 Bubb Road, Cupertino, California 95014 Telephone (408)446-4666 TWX 910-338-7621

Printed in USA

03-0123-01