ZAX

ICD-278 Z80B

Command Reference Guide

ICD 278

Z80B

COMMAND REFERENCE GUIDE

Prepared by

ZAX Corporation
2572 White Road, Irvine, California 92714

Part No. 20-100-03, Rev. B   Printed: December 1983

INTRODUCTION

The information contained in this COMMAND REFERENCE GUIDE relates to emulators directed at Z-80B based devices.

The ICD-278 is capable of initiating 29 different software commands. Each command is structured in a particular format that must be selected and executed accurately. It is therefore important for the design engineer to become familiar with the proper command instruction and results in order to use the emulator commands effectively.

For easy identification and immediate usage, symbolic notation will be used to explain the proper input format. This notation will be in the form of a console keyboard illustration, typical of the type an operator would be using.

Each console keyboard element will symbolize the EXACT input operation which must be performed, including spacing, feature keys, optional characters, numbers, etc. The operator has only to determine which command format to use and then follow the format command configuration displayed for each command.

Each command is defined by its symbolic key letter, the command name, an explanation and an example of typical command usage.

TABLE of CONTENTS

ADDRESS/DATA Parameters

Throughout this guide, the majority of commands will require the user to input address or data information. This section describes the address/data parameters available and explains their usage. Also displayed are the various symbol notations and their explanation.

The address/data common to the ICD commands is displayed to the right of the parameter explanations and characterized by the symbol: □

PHY 16

This parameter represents a hexadecimal PHYSICAL memory address in hex 16-bit. It can be specified by offset registers, addition or subtraction. In a "DON'T CARE" situation, "X" may be specified for the BREAK and EVENT commands (in this case, the parameter cannot be specified by the offset registers, addition or subtraction).

HEX ADRS

This parameter represents a HEXADECIMAL 16-bit memory address.

BIN 16

This parameter represents a BINARY 16-bit physical memory address. It can be used ONLY for a hardware break address in the BREAK and EVENT commands, and is displayed whenever using the BREAK status command.

ASCII 32

This parameter specifies data modification in ASCII codes. All ASCII A thru Z characters(32) may be used.

HEX DATA

This parameter specifies to use WORD or BYTE data (8/16-bit). The valid range is Ø to F and "X" (don't care).

L BYTE

This parameter is used to specify the number of BYTES. The proper format for data entry is;

"L" (number of bytes)

NOTE: The letter "L" must be specified before the number can be entered.

SYMBOL Notation

■ This symbol is used to represent a keyboard character that the operator will utilize. NOTE: Consult your console keyboard operating manual for validation of characters.

▢ This symbol displays to the operator that an address, data group, letter code, etc., is needed at this point in the command format. The options are always explained immediately after this symbol or in the address/data box.

▮ This symbol is used to represent a response from the system via the console display. The symbol will visualize all data exactly as it appears on the console screen.

🅐, 🅑 This symbol indicates additional information may be found in the APPENDIX section.

●, ❶ This symbol locates various notes throughout the Command Reference Guide.

> M ① , ② , ③ , ④ cr

M — IN THIS EXAMPLE, "M" IS THE SYMBOL USED TO INITIATE A MOVE COMMAND. SINCE THE LETTER "M" IS USED TO SPECIFY THE MOVE COMMAND, THERE ARE NO OPTIONS AVAILABLE YET.

① — THIS SPECIFIES A DATA ENTRY THAT IS REQUESTED. THE AVAILABLE ADDRESSES WILL ALWAYS BE DISPLAYED JUST TO THE RIGHT OF THE COMMAND FORMAT EXPLANATION > > > > > PHY 16

② — THIS SPECIFIES A SECOND DATA ENTRY. IN THIS EXAMPLE, THE MEMORY ADDRESS AT WHICH THE DATA TRANSFER IS TERMINATED. THE AVAILABLE ADDRESSES TO THE USER ARE> > PHY 16 L BYTE

③ — THIS SPECIFIES A THIRD DATA ENTRY. IN THIS EXAMPLE, A BEGINNING MEMORY ADDRESS FOR THE DATA TRANSFER. THE AVAILABLE ADDRESS TO THE USER IS > > > > > > > > > > PHY 16

④ — THIS SPECIFIES THE OPTION OF USING EITHER UP OR PU TO DETERMINE HOW DATA IS TO BE TRANSFERRED BETWEEN THE ICD AND THE TARGET SYSTEM.

P U , U P < THE AVAILABLE OPTIONS FOR THE KEYBOARD CHARACTERS WILL ALWAYS BE DISPLAYED TO THE LEFT OF THE EXPLANATION.

EXAMPLE

MOVE THE DATA THAT RESIDES IN THE TARGET SYSTEM MEMORY Ø - ØFFF, TO THE EMULATOR MEMORY. START AT THE ICD ADDRESS, Ø;

```
>M Ø, FFF,Ø,UP
```

EXAMPLE FORMAT

> A ① cr

① SPECIFIES THE MEMORY ADDRESS

x x x x ② CONSOLE DISPLAYS THE ICD ADDRESS

② SPECIFIES A MNEMONIC CODE TO BE USED IN THE Z80 ASSEMBLER

EXAMPLE

IN LINE ASSEMBLER;

```
>A 1ØØ
Ø1ØØ LD HL,AØØØH
Ø1Ø3 PUSH DE
Ø1Ø4 LD DE,BØØØH
Ø1Ø7 EX DE,HL
Ø1Ø8 POP DE
Ø1Ø9 INC HL
Ø1ØA INC DE
Ø1ØB JP Ø1ØØH
Ø1ØE
```

A THE ASSEMBLE COMMAND IS USED FOR WRITING SOFTWARE PATCHES EMPLOYING THE INTERNAL 64K MEMORY OF THE ICD EMULATOR.

```
> B / ① ② , ③ , ④ cr
```

● HARDWARE BREAK
TYPE 1: STANDARD HARDWARE BREAK;

① ; A , B , C SPECIFIES A HARDWARE BREAK NAME.
(IF NONE IS SPECIFIED, ASSUME A THEN B THEN C AFTER WHICH OCCURS THE ERROR MESSAGE; BREAK BUSY)

② SPECIFIES THE BREAK STATUS;

M MEMORY ACCESS
M R MEMORY READ
M W MEMORY WRITE
P PORT ACCESS
P R PORT READ

P W PORT WRITE
O F OP CODE FETCH
I A INTERRUPT ACKNOWLEDGE

③ SPECIFIES A BREAK ADDRESS PHY 16

④ SPECIFIES THE BREAK PASSING COUNT
(DECIMAL 1 TO 65535; DEFAULT=1)

> B / ① ② cr

EXAMPLE ▼

● HARDWARE BREAK
TYPE 2 EVENT TRIGGER BREAK;

① ; A, B, C SPECIFIES A HARDWARE BREAK NAME
(SEE TYPE 1; STANDARD HARDWARE BREAK)

② SPECIFIES AN ARM/INDIVIDUAL BREAK;

A R M THE ARM BREAK OCCURS ONLY AFTER THE EVENT TRIGGER TAKES PLACE

I N D THE INDIVIDUAL BREAK TAKES PLACE WITHOUT REGARD TO THE EVENT TRIGGER

B THE BREAK COMMAND IS USED TO SET THE STANDARD HARDWARE AND SOFTWARE BREAKPOINTS. SUPPLEMENTAL BREAKPOINT COMMANDS INCLUDE SETTING; AN EVENT TRIGGER BREAK, AN ETERNAL BREAK, AN OPERATIONAL BREAK AND DISPLAYING A BREAK CONDITION. SUPPLEMENTAL COMMANDS MAY BE USED IN CONJUNCTION WITH THE STANDARD HARDWARE AND SOFTWARE BREAK COMMANDS.

> B / OPT ① cr

- HARDWARE BREAK
  TYPE 3 EXTERNAL BREAK;

OPT ; X SPECIFIES AN OPTIONAL EXTERNAL BREAK

①; SPECIFIES THE EXTERNAL BREAK TO OCCUR AT THE HI OR LOW EDGE H I , L O ~~W~~

> B / ① ② cr

- HARDWARE BREAK
  TYPE 4 BREAK CONDITION;

①; A, B, C SPECIFIES A HARDWARE BREAK NAME (SEE TYPE 1; STANDARD HARDWARE BREAK)

② MAKES A SPECIFIED BREAK;

O N VALID

O F F INVALID

C L R CLEARED

- NOTE: IF THE SPECIFIC HARDWARE BREAK NAME IS OMITTED, ALL HARDWARE BREAKS ARE AFFECTED.

> B / ① ② , ③ cr

- SOFTWARE BREAK
  TYPE 1 STANDARD SOFTWARE BREAK;

①; 0 - 7 SPECIFIES A SOFTWARE BREAK NAME (MUST SPECIFY)

② SPECIFIES A BREAK ADDRESS PHY 16

③ SPECIFIES A BREAK PASSING COUNT (DECIMAL 1 TO 65535; DEFAULT=1)

CAUTION
Do not PATCH CODE
WHILE USING
SW BKPTS.

> B S = ①

- SOFTWARE BREAK
  TYPE 2 ENABLE/DISABLE BREAK;

①,

E N SPECIFIES TO PLACE SOFTWARE BREAK IN THE ENABLE STATE [2]

D I SPECIFIES TO PLACE SOFTWARE BREAK IN THE DISABLE STATE (NO SOFTWARE BREAK OCCURS)

[2] NOTE: WHEN THE SOFTWARE BREAK IS IN THE ENABLE STATE, EXECUTE MOV A,A (7FH): AND A BREAK WILL OCCUR. BREAKS CAN BE SET USING AN ARBITRARY NUMBER BY GENERATING THEM IN THE USER PROGRAM.

> B / ① ② cr

● SOFTWARE BREAK
TYPE 3 BREAK CONDITION;

① ; 0 ~ 7 SPECIFIES A SOFTWARE BREAK NAME[1]
(MUST SPECIFY)

② ; MAKES A SPECIFIED SOFTWARE BREAK;

O N VALID

O F F INVALID

C L R CLEARED

❶ NOTE: IF THE SPECIFIC SOFTWARE BREAK NAME IS OMITTED,
ALL SOFTWARE BREAKS ARE AFFECTED.

> B / ① ② cr

● SUPPLEMENTAL BREAK
TYPE 1

① ; T SPECIFIES A $\overline{\text{WAIT}}$ TIMEOUT

E SPECIFIES AN EVENT TRIGGER

W SPECIFIES A WRITE PROTECT

② MAKES ANY OF THE ① COMMANDS;

O N VALID

O F F INVALID

> B / ① ② cr

● SUPPLEMENTAL BREAK
TYPE 2

①; E   SPECIFIES AN EVENT TRIGGER BREAK

②   SPECIFIES A BREAK PASSING COUNT
(DECIMAL 1 TO 65535; DEFAULT=1)

> B ① cr

● SUPPLEMENTAL BREAK
TYPE 3

①; I N I   CLEARS THE EVENT PASSING CONDITION BY RESETTING
THE CONDITION OF THE ARM OCCURRENCE

EXAMPLE: BREAK Command

DISPLAY BREAK STATUS WITHOUT HARDWARE BREAK ON;

```
>B
T (ON)
S (DI)
W (OFF)
```

DISPLAY A.BREAK STATUS WITH HARDWARE BREAK ON;

```
>B OF,1ØØ
>
>B
A (ON)   OF  Ø1ØØ    1     Ø   IND (ØØØØ_ØØØ1_ØØØØ_ØØØØ)
T (ON)
S (DI)
W (OFF)
```

SET BREAK STATUS WITH A 16-BIT PHYSICAL ADDRESS AND ISSUE A BREAK STATUS COMMAND;

```
>B/A M,ØØØX_111X_XXXX_ØØØØ
>B
A (ON)   M  XXXØ    1       Ø   IND (ØØØX_111X_XXXX_ØØØØ)
T (ON)
S (DI)
W (OFF)
```

EXAMPLE: BREAK Command

Set a software break using a hexadecimal address;

```
>B/7 1234
>B
A (ON)   M  XXXØ     1        Ø   IND (ØØØX_111X_XXXX_ØØØØ)
7 (ON)      1234     1        Ø
T (ON)
S (DI)
W (OFF)
```

Set arm break for hardware break A;

```
>B/A ARM
>B
A (ON)   M  XXXØ     1        Ø   ARM (ØØØX_111X_XXXX_ØØØØ)
7 (ON)      1234
T (ON)
S (DI)
W (OFF)
```

Enable a software break;

```
>B S=EN
>B
A (ON)   M  XXXØ     1        Ø   ARM (ØØØX_111X_XXXX_ØØØØ)
7 (ON)      1234
T (ON)
S (EN)
W (OFF)
```

> C 1 ± 2 ± n cr

1 ~ 2 ~ n SPECIFIES THE OPERATION DATA

NOTE: DECIMAL IS VALID FOR -8388608 TO +8388607
HEXADECIMAL VALID FOR 0 TO FFFFFF(H)

EXAMPLE

EXECUTE MULTIPLE HANDLING OF SUBTRACTION AND ADDITION USING HEX AND DECIMAL DATA INTERCHANGABLY;

```
>C 1234+1234
H: ØØØ9A4
D:   2468


>C 1ØØØH-FFFH
H; ØØØØØ1
D:      1
```

C THE CALCULATION COMMAND IS USED TO ADD AND SUBTRACT HEXADECIMAL AND DECIMAL NUMBERS. THE OPERATION OF ADDITION AND SUBTRACTION ON DATA MAY BE PERFORMED TOGETHER ON THE SAME LINE.

> C O ① , ② , ③ , ④ cr

① SPECIFIES THE COMPARISON START MEMORY ADDRESS PHY 16

② SPECIFIES THE END MEMORY ADDRESS PHY 16 L BYTE

③ SPECIFIES THE START MEMORY ADDRESS TO BE COMPARED HEX ADRS

④ SPECIFIES EITHER;

U P TO USE ① FOR THE USER MEMORY, AND ③ FOR THE ICD PROGRAM MEMORY,

P U TO USE ① FOR THE ICD PROGRAM MEMORY AND ③ FOR THE USER MEMORY

● NOTE: UP IS USED TO COMPARE TARGET SYSTEM MEMORY CONTENTS TO EMULATOR MEMORY CONTENTS.

PU IS USED TO COMPARE EMULATOR MEMORY CONTENTS TO TARGET SYSTEM MEMORY CONTENTS.

EXAMPLE

MEMORY COMPARISON OF THE RANGE; Ø TO ØØØ1

```
>CO Ø,FFF,ØØØ1,UP <CR>
 1         2    3          4
 ADDRESS   M    ADDRESS    M
   ØØØØ    C3      ØØØØ    FF
   ØØØ1    Ø6      ØØØ1    8F
```

● NOTE:

1 - TARGET SYS MEMORY ADDRESS
2 - TARGET SYS MEMORY CONTENTS
3 - EMULATOR MEMORY ADDRESS
4 - EMULATOR MEMORY CONTENTS

CO THE COMPARE COMMAND IS USED TO COMPARE THE CONTENTS OF SPECIFIED MEMORIES AND THEN DISPLAY THE UNMATCHING DATA.

EXAMPLE

DUMP MEMORY CONTENTS FROM Ø TO 3Ø;

```
>D Ø,3Ø
          Ø   1   2   3   4   5   6   7   8   9   A   B   C   D   E   F      ASCII CODE
 ØØØ0    21  FF  ØØ  11  34  12  77  12  13  23  B7  C2  Ø6  ØØ  C3  ØØ    !...4.w..#7B..C.
 ØØ1Ø    Ø1  ØØ  ØØ  ØØ  .........                                          ................
 ØØ2Ø    ØØ  ØØ  Ø.........                                                 ................
 ØØ3Ø    ØØ
```

DUMP MEMORY CONTENTS INTO WORD;

```
>D/W  AØØØ,L4Ø
         Ø      2      4      6      8      A      C      E        ASCII CODE
AØØØ  FØ7F   A18F   F8Ø5   9ØØF   FØ1D   AFAF   FØØD   ØØ1C     .p.!.x...p//.p..
AØ1Ø  FØ17   E5AF   FØØ5   24ØF   FØØ5   BDAF   FØED   18ØF     .p/e.p.$.p/=mp..
AØ2Ø  DØØ    84AF   FØØ9   Ø1ØF   D21F   A74F   FØ25   Ø21F     .p/..p...RO'%p..
AØ3Ø  FØØD   A125   FØØ5   ØØØF   F3ØF   A7AF   FØ45   52ØF     .p%!.p...s/'Ep.R
```

> D / ① ② , ③ cr

EXAMPLE ▲

① ; W SPECIFIES TO DUMP DATA INTO WORD UNITS
(IF OMITTED, DATA IS DUMPED IN BYTE UNITS)

② SPECIFIES THE DISPLAY STARTING MEMORY ADDRESS PHY 16

③ SPECIFIES THE DISPLAY ENDING MEMORY ADDRESS PHY 16 L BYTE

D THE DUMP COMMAND IS USED TO DISPLAY THE MEMORY CONTENTS IN EITHER BYTE OR WORD UNITS. THE CONTENTS IS REPRESENTED BY A HEXADECIMAL NUMBER AND ASCII CODE.

> D I ① , ② cr

① SPECIFIES THE STARTING MEMORY ADDRESS (IF OMITTED, DATA IS DISPLAYED FROM THE CURRENT PC) PHY 16

② SPECIFIES THE ENDING MEMORY ADDRESS PHY 16 L BYTE

● NOTE: >DI <CR> , DISPLAYS 12 LINES OF CODE.

● NOTE: IF ① AND ② ARE OMITTED, THE DATA DISPLAYED WILL BE THE CURRENT PROGRAM COUNTER.

EXAMPLE

```
>DI 1ØØ
Ø1ØØ 21ØØBØ  LD HL, BØØØH
Ø1Ø3 11ØØAØ  LD DE, AAAAH
Ø1Ø6 7E      LD A,(HL)
Ø1Ø7 B7      OR A
Ø1Ø8 C2ØØØ3  JP NZ,Ø3ØØH
Ø1ØB 12      LD (DE),A
Ø1ØC 23      INC HL
Ø1ØD 13      INC DE
Ø1ØE C3Ø6Ø1  JP  Ø1Ø6H


>DI  1ØØ,114
Ø1ØØ 21ØØBØ  LD HL,ØBØØØH
Ø1Ø3 11ØØAØ  LD DE,ØAØØØH
Ø1Ø6 7E      LD  A,(HL)
Ø1Ø7 B7      OR  A
Ø1Ø8 C2ØØØ3  JP NZ,Ø3ØØH
Ø1ØB 12      LD (DE),A
Ø1ØC 23      INC HL
Ø1ØD 13      INC DE
Ø1ØE C3Ø6Ø1  JP  Ø1Ø6H
Ø111 ØØ      NOP
Ø112 3AØØ3A  LD  A,(3AØØH)
```

DI THE DISASSEMBLE COMMAND IS USED TO DISASSEMBLE THE MEMORY CONTENTS AND DISPLAY THE RESULT.

❶ > E ① ② = ③ cr   ❷ > E ① ② cr

TYPE 1 EXAMINE AND CHANGE MEMORY CONTENTS
TYPF 2 EXAMINE MEMORY CONTENTS ONLY

① ; / W SPECIFIES TO CHANGE THE MEMORY CONTENTS ON A WORD BASIS (DEFAULT IS A BYTE BASIS)

② SPECIFIES BEGINNING ADDRESS OF MEMORY TO BE FILLED PHY 16 3

③ SPECIFIES THE MEMORY CONVERSION DATA/CONVERSION STARTING MEMORY ADDRESS HEX DATA ASCII 32 4

● NOTE: MULTI-DISPLAY OF DATA IS POSSIBLE USING THE FORMAT LISTED BELOW;

cr DISPLAYS DATA AT THE NEXT ADDRESS AFTER CHANGE

9 cr DISPLAYS DATA AT THE SAME ADDRESS AFTER CHANGE

Λ cr DISPLAYS DATA AT THE ADDRESS THAT IS DECREMENTED BY 1 AFTER THE CHANGE

/ cr TERMINATES THE EXAMINE COMMAND AFTER THE CHANGE

❸ NOTE: THE SPECIFIED ASCII OR HEX DATA IS WRITTEN TO THE SPECIFIED ADDRESS BUT NOT VERIFIED

EXAMPLE

EXAMINE AND CHANGE MEMORY IN WORD DATA;

```
>E/W 100 <CR>
0100 03FA = CHANGE DATA <CR>
0102 1F21 =
0104 11FD =
0106 FDAB =
0108 200E =
010A FECD =
```

EXAMINE AND CHANGE MEMORY IN BYTE DATA;

```
>E 0100 <CR>
0100 FA = CHANGE DATA <CR>
0101 03 =
0102 21 =
0103 1F =
0104 FD =
0105 11 =
0106 AB =
0107 FD =
```

E THE EXAMINE COMMAND IS USED TO CHANGE THE MEMORY CONTENTS TO EITHER ASCII OR HEXADECIMAL DATA. TWO TYPES OF EXAMINE COMMANDS ARE AVAILABLE TO THE USER; (1)EXAMINE AND CHANGE MEMORY CONTENTS, (2)EXAMINE THE MEMORY CONTENTS ONLY.

❹ NOTE: IF ③ IS OMITTED, THE MEMORY CONTENTS ARE NOT CONVERTED.

> E V ① ② ③ ④ cr

● STANDARD EVENT TRIGGER
TYPE 1;

① ; S T = SPECIFIES AN EVENT TRIGGER STATUS BY NAME;

M	MEMORY ACCESS	P W	PORT WRITE
M R	MEMORY READ	O F	OP FETCH CODE
M W	MEMORY WRITE	I A	INTERRUPT ACKNOWLEDGE
P	PORT ACCESS	E X	COMMAND EXECUTION
P R	PORT READ		

② ; A = SPECIFIES TO DIRECT AN EVENT TRIGGER AT A PARTICULAR BREAK ADDRESS PHY 16 BIN 16 1

③ D = SPECIFIES EVENT DATA (BYTE DATA) D

④ IF BYTE DATA IS USED, SPECIFIES THE DATA BIN 16 HEX DATA 1

$\underbrace{\text{XXXX}}_{?}$ XX

E V ① cr

● TYPE 2;

① SPECIFIES TO;

O N MAKE AN EVENT TRIGGER ENABLED

O F F MAKE AN EVENT TRIGGER DISABLED

C L R CLEAR AN EVENT TRIGGER[2]

❶ NOTE: "X" MAY BE SPECIFIED AS "DON'T CARE" FOR BINARY OR HEXADECIMAL INPUT (1 OR 4 BIT)

● NOTE: IF A STATUS IS SPECIFIED FOR PORT OPERATION, THE HIGH-ORDER 8 BITS OF THE ADDRESS SHOULD BE "X".

❷ NOTE: IF A NEW SETTING IS MADE AFTER AN EVENT CLEAR, THE EVENT GOES TO THE ON STATE.

● NOTE: A NEW EVENT CONDITION HAS PRECEDENCE OVER A PRE-SET EVENT CONDITION, BUT ONLY AT THAT PARTICULAR SETTING.

EXAMPLE ▼

E V THE EVENT COMMAND IS USED TO SET OR RELEASE AN EVENT TRIGGER. THE STANDARD EVENT TRIGGER IS SPECIFIED BY A STATUS NAME, ADDRESS AND EVENT DATA. IT IS ALSO POSSIBLE TO OUTPUT AN EVENT TRIGGER PULSE VIA THE EVENT TRIGGER CONNECTOR DURING A SPECIFIED MACHINE CYCLE(TYPE 2) AN EVENT TRIGGER CAN ALSO BE USED AS A HARDWARE BREAK.

EXAMPLE: EVENT Command

SPECIFY AN EVENT HEX ADDRESS USING 'DON'T CARE';

```
>EV A=AXXØ
>EV
(ON)
STATUS  = ANY
ADDRESS = AXXØ  (1Ø1Ø_XXXX_XXXX_ØØØØ)
DATA    = XX      (XXXX_XXXX)
```

DISPLAY AND EVENT STATUS;

```
>EV ST=OF
>EV
(ON)
STATUS  = OF
ADDRESS = AXXØ  (1Ø1Ø_XXXX_ØØØØ)
DATA    = XX      (XXXX_XXXX)
```

SPECIFY EVENT DATA;

```
>EV D=ØØ1Ø_11XX
>EV
(ON)
STATUS  = OF
ADDRESS = AXXØ   (1Ø1Ø_XXXX_XXXX_ØØØØ)
DATA    = 2X       (ØØ1Ø_11XX)
```

SET EVENT COMMAND USING ADDRESS, DATA AND STATUS;

```
>EV A=ØØFF D=12 ST=MR
>EV
(ON)
STATUS  = MR
ADDRESS = ØØFF   (ØØØ_ØØØØ_1111_1111)
DATA    = 12     (ØØØ1_ØØ1)
```

> F ① , ② , ③ , ④ cr

① ; / W SPECIFIES TO FILL MEMORY ON A WORD BASIS (DEFAULT IS A BYTE BASIS)

② SPECIFIES THE BEGINNING ADDRESS OF THE MEMORY TO BE FILLED PHY 16

③ SPECIFIES THE ENDING ADDRESS OF MEMORY TO BE FILLED (IF OMITTED, DATA TO BE FILLED FROM THE BEGINNING MEMORY ADDRESS IS WRITTEN) PHY 16 L BYTE

④ SPECIFIES DATA TO BE FILLED WITHIN THE BEGINNING AND ENDING ADDRESSES HEX DATA ASCII 32

EXAMPLE

FILL MEMORY WITH "Ø" FROM ADDRESS ØØØØ TO ØØFF;

```
>F ØØØ,ØØFF,Ø
```

FILL MEMORY WITH ASCII CHARACTER FROM Ø TO FF;

```
F ØØØ, ØFF,'ZAX'
```

F THE FILL COMMAND IS USED TO FILL MEMORY WITH SPECIFIED DATA, EITHER HEXADECIMAL OR ASCII CODES.

EXAMPLE

Execute a user program starting at 0H;

```
>G Ø,B

PC      MC         OP                 SP    AF    BC    DE    HL    IX    IY    I    IF(SP)
ØØØB 11ØØBØ    LD   DE,ØBØØØØH    FFFC ØØØØ  ØØØØ  BØØØ  AØØØ  ØØØØ  ØØØØ  ØØ    Ø ØØ4ØØ
<Break  Hardware  A>
```

Continue the execution after break;

```
>G

PC     MC         OP                  SP   AF    BC    DE    HL     IX    IY     I   IF(SP)
15DC 3AØØ3A    LD  A,(3AØØH)     FFB6 3AØØ  141Ø  BØØ1  9FØ1  ØØØØ  ØØØØ   ØØ  Ø Ø3ØØ
<Break Monitor>
```

> G ① , ② , ③ cr

① SPECIFIES THE PROGRAM EXECUTION STARTING MEMORY ADDRESS (IF OMITTED, THE PROGRAM EXECUTION IS CONTINUED FROM THE CURRENT PC) PHY 16

② SPECIFIES THE PROGRAM EXECUTION ENDING MEMORY ADDRESS PHY 16

③ SPECIFIES A SECOND (OPTIONAL) ENDING MEMORY ADDRESS THAT MAY BE USED IN CONJUNCTION WITH THE FIRST ENDING MEMORY ADDRESS PHY 16

● NOTE: WHEN ① AND ② ARE OMITTED, A BREAK IS NOT SPECIFIED.

EXAMPLE ⏶

G THE GO COMMAND IS USED TO EXECUTE THE USER PROGRAM.

● NOTE: G cr

ALLOWS SUCCESSIVE EXECUTION OF USER PROGRAM ONCE A BREAK HAS OCCURRED

- TYPE ❶ REAL TIME TRACE STATUS

  > H cr

- TYPE ❷ REAL TIME TRACE COUNTER RESET

  > H C L R cr

- TYPE ❸ MONITOR TRIGGER

  > H ① , ② cr

- TYPE ❹ EVENT TRIGGER

  > H ① , ② cr

TYPE ▶
▼

- TYPE ❺ FORMAT DISPLAY

  > H ① , ② , ③ cr

- TYPE ❻ HISTORY SEARCH

  > H S , ① , ② , ③ cr

---

❸ ① ; B M SPECIFIES THE BEGIN MONITOR TRIGGER. IN THIS MODE, TRACE SECTION IS INITIATED BY THE EMULATION START AND TERMINATED AT A PREDETERMINED BREAK POINT SPECIFIED BY THE RANGE; ②

E M SPECIFIES THE END MONITOR TRIGGER. EMULATION IS INITIATED USING THE GO OR NEXT COMMAND TO START TRACE. THE MONITOR WILL RESPOND WITH A BREAK WHEN THE TRACE IS TERMINATED. THE TRACE RANGE UP TO WHERE THE BREAKPOINT OCCURS IS A MAXIMUM OF 2048.

❸ ② SPECIFIES THE TRACE RANGE(DECIMAL 1 TO 2047)

❹ ① ; B E SPECIFIES THE BEGIN EVENT TRIGGER. IN THIS MODE, TRACE BEGINS WHEN AN EVENT POINT IS PASSED AFTER EMULATION HAS BEGUN.

C E SPECIFIES THE CENTER EVENT TRIGGER. IN THIS MODE, THE TRACE SECTION IS RECOGNIZED BOTH BEFORE AND AFTER THE EVENT POINT.

M E SPECIFIES THE MULTIPLE EVENT MODE. IN THIS MODE, A TRACE IS PERFORMED EACH TIME AN EVENT POINT IS PASSED DURING LOOP PROCESSING.

E E SPECIFIES THE END EVENT TRIGGER. IN THIS MODE, EMULATION IS INITIATED UŞING THE GO OR NEXT COMMAND TO START THE TRACE. WHEN THE EVENT POINT IS PASSED, THE TRACE TERMINATES.

❹ ② SPECIFIES THE TRACE RANGE(DECIMAL 1 TO 2047). WHEN ② IS OMITTED IN THE SPECIFICATION OF BE,BM & CE, THE DEFAULT IS 2047 AND 1023 WHEN OMITTED FOR ME.

❺ ①; M SPECIFIES THE DISPLAY IN MACHINE CYCLE FORMAT

D SPECIFIES THE DISPLAY IN MACHINE CYCLE AND DISASSEMBLE (EXCEPT FOR OPERATION CODE FETCH)

❺ ❻ ② SPECIFIES THE DISPLAY OR SEARCH INITIATION STORAGE POINTER (DECIMAL 1 TO 2047)

❺ ❻ ③ SPECIFIES THE DISPLAY OR SEARCH TERMINATION STORAGE POINTER (DECIMAL 1 TO 2047)

❻ S SPECIFIES TO SEARCH FOR THE REALTIME TRACE CONTENTS

❻ ①; / Ⓐ / Ⓑ / Ⓒ SPECIFIES A STATUS TO BE SEARCHED FOR

Ⓐ SPECIFIES THE SEARCH ADDRESS PHY 16

Ⓑ SPECIFIES THE SEARCH DATA HEX DATA

Ⓒ SPECIFIES SEARCH STATUS BY NAME;

M R	MEMORY READ CYCLE	I A	INTERRUPT ACKNOWLEDGE CYCLE		
M W	MEMORY WRITE CYCLE	O F	OPERATION CODE FETCH		
P R	PORT READ CYCLE	P W	PORT WRITE CYCLE	H A	HALT ACKNOWLEDGE CYCLE

H THE HISTORY COMMAND IS USED TO SET A TRIGGER MODE FOR REALTIME TRACING, AND TO DISPLAY OR SEARCH FOR THE USER PROGRAM OPERATION TO BE TRACED. THERE ARE SIX DIFFERENT HISTORY COMMANDS AVAILABLE TO THE USER; REAL TIME TRACE STATUS, REAL TIME TRACE COUNTER RESET, MONITOR TRIGGER, EVENT TRIGGER, FORMAT DISPLAY AND HISTORY SEARCH.

Ⓐ See APPENDIX A

▼ EXAMPLE

EXAMPLE: HISTORY Command

Display the History status;

```
>H
Clock Counter = ØØØØØØAA/
Storage Mode  = EM
Storage Size  = 14/    14
```

Display the Real Time Trace in machine cycle;

```
>HM,2Ø
POINT   ADDR    DT   ST
ØØ2Ø    ØØØ3    11   M1
ØØ19    ØØØ4    ØØ   MR
ØØ18    ØØØ5    Ø4   MR
ØØ17    ØØØ6    E5   M1
ØØ16    ØØ85    Ø3   MW
ØØ15    ØØ84    ØØ   MW
ØØ14    ØØØ7    D5   M1
ØØ13    ØØ83    Ø4   MW
ØØ12    ØØ82    ØØ   MW
ØØ11    ØØØ8    21   M1
ØØ1Ø    ØØØ9    ØØ   MR
ØØØ9    ØØØA    AØ   MR
ØØØ8    ØØØB    11   M1
ØØØ7    ØØØC    ØØ   MR
ØØØ6    ØØØD    BØ   MR
ØØØ5    ØØØE    7E   M1
ØØØ4    AØØØ    3A   MR
ØØØ3    ØØØF    77   M1
ØØØ2    AØØØ    3A   MW
```

EXAMPLE: HISTORY Command

Display the realtime trace in machine cycle and disassemble code;

```
>H D
POINT   ADDR     DT       ST        OP
ØØ23    ØØØØ     21ØØØ3            LD   HL,Ø3ØØH
ØØ2Ø    ØØØ3     11ØØØ4            LD   DE,Ø4ØØH
ØØ17    ØØØ6     E5                PUSH HL
ØØ16    ØØ85     Ø3       MW
ØØ15    ØØ84     ØØ       MW
ØØ14    ØØØ7     D5                LD   HL,ØAØØØH
ØØ13    ØØ83     Ø4       MW
ØØ12    ØØ82     ØØ       MW
ØØ11    ØØØ8     21ØØAØ            LD   HL,ØAØØØH
ØØØ8    ØØØB     11ØØBØ            LD   DE,ØBØØØH
ØØØ5    ØØØE     7E                LD   A, (HL)
ØØØ4    AØØØ     3A       MR
ØØØ3    ØØØF     77                LD   (HL).A
ØØØ2    AØØØ     3A       MW
```

Search for the contents of real time trace for a memory read;

```
>H S,///MR
Point   T    Addr DT   ST
ØØ13         Ø1Ø1 ØØ   MR
ØØ12         Ø1Ø2 AØ   MR
ØØ1Ø         Ø1Ø4 ØØ   MR
ØØØ9         Ø1Ø5 BØ   MR
ØØØ7         AØØØ FF   MR
```

● NOTE: The real time counter counts the CPU state number (CPU clock) in 32 bits.

● NOTE: The pointer at which a program is initiated with a Go or NEXT command and the point at which the program was previously started, are displayed as the real time trace starting storage pointers.
If these pointers exceed 2047, a FULL message is displayed.

> I ① cr

① SPECIFIES EITHER THE SYSTEM, PARTIAL, OR ALL INCIRCUIT MODE;

0 THE SYSTEM MODE ENABLES DEBUGGING (SOFTWARE ONLY) USING THE ICD PROGRAM MEMORY. IN THIS MODE THE TARGET SYSTEM I/O AND INTERRUPT SIGNALS ARE IGNORED

1 THE PARTIAL INCIRCUIT MODE ENABLES DEBUGGING USING THE ICD PROGRAM AND TARGET SYSTEM MEMORIES. THE SPECIFIED MAPPING BECOMES VALID AS DO THE SYSTEM I/O INTERRUPT SIGNALS

2 THE ALL INCIRCUIT MODE ENABLES DEBUGGING USING ONLY THE TARGET SYSTEM MEMORY. MEMORIES SPECIFIED AS READ/WRITE AND READ ONLY (R/W,RO) ARE OPERATED BY THE USER MEMORY (US)

● NOTE: WHEN NO PARAMETER IS SPECIFIED, THE CURRENT STATE IS DISPLAYED

EXAMPLE

SPECIFY THE PARTIAL AND SYSTEM INCIRCUIT MODES;

```
>I 1
>I
IN-CIRCUIT MODE 1

>I Ø
>I
IN-CIRCUIT MODE Ø
```

I THE INCIRCUIT COMMAND IS USED TO SET THE ICD MAPPING MODE.

● See APPENDIX B

> I D cr

EXAMPLE

```
>ID
ICD-278 FOR Z8Ø      V1.Ø
```

I D

THE IDENTIFICATION COMMAND IS USED TO DISPLAY AN ICD DEVICE NAME AND A VERSION OF THE INSTALLED FIRMWARE.

```
> L / ① ② , ③ , ④ cr
```

① SPECIFIES EITHER THE;

- T TERMINAL PORT (IGNORE SOFTWARE HANDSHAKE)
- P TERMINAL PORT (PERFORM SOFTWARE HANDSHAKE)
- A AUXILIARY PORT (IGNORE SOFTWARE HANDSHAKE)
- H HOST PORT (PERFORM SOFTWARE HANDSHAKE)

② SPECIFIES THE NAME OF THE FILE TO BE LOADED[1]

③ SPECIFIES THE BIAS OF THE OBJECT TO BE LOADED (IF OMITTED, THE OBJECT STARTING ADDRESS IS SPECIFIED)

④ SPECIFIES THE LOAD MESSAGE ASCII 32 HEX DATA

[1] NOTE: THIS PARAMETER IS VALID ONLY WHEN THE UTILITY SOFTWARE PROGRAM ZICE IS AVAILABLE.

EXAMPLE

LOAD THE TEST HEX FILE ON DISKETTE;

```
> L TEST
100

>L/T ,100
200
```

L THE LOAD COMMAND IS USED TO LOAD EITHER A HEX FILE ON A DISKETTE OR THE OBJECT PROGRAM (INTEL FORMAT) FROM EITHER SPECIFIED PORT IN MEMORY.

Ⓓ See APPENDIX D

> M ① , ② , ③ , OPT cr

① SPECIFIES THE MEMORY ADDRESS AT WHICH THE DATA TRANSFER IS INITIATED PHY 16

② SPECIFIES THE MEMORY ADDRESS AT WHICH THE DATA TRANSFER IS TERMINATED PHY 16 L BYTE

③ SPECIFIES THE BEGINNING MEMORY ADDRESS TO WHICH THE DATA TRANSFER IS MADE PHY 16

OPT,[1] U P SPECIFIES TO MOVE DATA FROM THE THE TARGET MEMORY TO THE ICD PROGRAM MEMORY

P U SPECIFIES TO MOVE DATA FROM THE ICD PROGRAM MEMORY TO THE TARGET SYSTEM MEMORY

[1] NOTE: WHEN OPT IS OMITTED, DATA IS MOVED ACCORDING TO THE SPECIFICATION OF THE MEMORY MAP.

EXAMPLE

MOVE MEMORY OF ØH THROUGH FFFH:

```
>M Ø,FFF,Ø,UP
```

M THE MOVE COMMAND IS USED TO MOVE MEMORY BETWEEN THE ICD AND THE TARGET SYSTEM.

> M A ① , ② = ③ cr

① SPECIFIES THE BEGINNING ADDRESS TO DESIGNATE MAPPING PHY 16

② SPECIFIES THE MAP ENDING ADDRESS PHY 16 L BYTE[1]

③ SPECIFIES THE PROGRAM MEMORY, EITHER;

- R O READ ONLY
- R W READ/WRITE
- N O THE AREA IN WHICH NO MEMORY EXISTS[2]
- U S THE TARGET SYSTEM MEMORY

[1] NOTE: WHEN ② IS OMITTED, 1K BYTES ARE SPECIFIED

[2] NOTE: A BREAK OCCURS IN THIS AREA IF ACCESSED BY A PROGRAM

● NOTE: IF NO PARAMETER IS SPECIFIED, THE CURRENT MAPPING CONDITION IS DISPLAYED.

EXAMPLE

SPECIFY Ø TO FFF AS THE USER MEMORY;

```
>MA Ø,FFF = US
```

DISPLAY THE STATUS OF THE MEMORY MAP;

```
>MA
IN-CIRCUIT MODE 1 (US=RW)
ØØØØ-ØFFF = US
1ØØØ-FFFF = RW
```

DISPLAY THE STATUS OF THE MEMORY MAP WHEN THE INCIRCUIT MODE IS NOT 1;

```
>MA
IN-CIRCUIT MODE Ø (US=RW)
ØØØØ-ØFFF = US
1ØØØ-FFFF = RW
```

MA

THE MAP COMMAND IS USED TO SET THE ICD MEMORY MAP. THE TARGET SYSTEM OR ICD PROGRAM MEMORY IS USED ON A 1K-BYTE BASIS AND MUST BE SPECIFIED.

● SEE INCIRCUIT COMMAND

EXAMPLE

PERFORM A SINGLE STEP TRACE, FIVE STEPS FROM THE CURRENT PC;

```
>N 5

PC     MC       OP                SP    AF    BC    DE    HL    IX    IY    I    IF(SP)
Ø1Ø 3AØØAØ    LD   A,(ØAØØØH)   ØØ2E  Ø5ØE  A5ØE  Ø3Ø1  4312  ØØØØ  ØØØØ  ØØ   Ø ØØ1A
Ø1Ø3 77       LD   (HL),A       ....  ....  ....  ....  ....  ....  ....  ..   . ....
Ø1Ø4 FEØ5     CP   5            ....  Ø542  ....  ....  ....  ....  ....  ..   . ....
Ø1Ø6 C2ØØØ1   JP   NZ,Ø1ØØH     ....  ....  ....  ....  ....  ....  ....  ..   . ....
Ø1Ø9 21ØØAØ   LD   HL,ØAØØØH    ....  ....  ....  ....  AØØØ  ....  ....  ..   . ....
```

PERFORM A SINGLE STEP TRACE FOR FIVE STEPS CHANGING PC;

```
>R PC,2ØØ
>N 5

PC     MC       OP                SP    AF    BC    DE    HL    IX    IY    I    IF(SP)
Ø2ØØ 3AØØ3A   LD  A,(3AØØH)     ØØ2E  3A42  A5ØE  Ø3Ø1  AØØØ  ØØØ0  ØØØØ  ØØ   Ø ØØØØ
Ø2Ø3 Ø0       NOP               ....  ....  ....  ....  ....  ....  ....  ..   . ....
Ø2Ø4 3AØØ3A   LD  A,(3AØØH)     ....  ....  ....  ....  ....  ....  ....  ..   . ....
Ø2Ø7 ØØ       NOP               ....  ....  ....  ....  ....  ....  ....  ..   . ....
Ø2Ø8 3AØØ3A   LD  A,(3AØØH)     ....  ....  ....  ....  ....  ....  ....  ..   . ....
```

● NOTE: THE STARTING ADDRESS CAN BE CHANGED BY CHANGING PC WITH A REGISTER COMMAND.

> N ① cr

① SPECIFIES THE NUMBER OF SINGLE STEPS (Ø TO 15535; IF OMITTED, 1 IS ASSUMED)

EXAMPLE ⬆

N THE NEXT COMMAND IS USED TO DISPLAY ALL TRACE DATA BY N STEPS FROM THE CURRENT POINTER.

> O & ① = ② cr

① SPECIFIES ONE OF FOUR OFFSET REGISTERS, EITHER;

0 , 1 , 2 , 3

② SPECIFIES THE OFFSET REGISTER SET VALUE HEX ADRS

● NOTE: IF NO PARAMETER IS SPECIFIED, THE CURRENT SET VALUE IS DISPLAYED.

EXAMPLE

```
>O
&Ø = ØØØØ
&1 = ØØØØ
&2 = ØØØØ
&3 = ØØØØ

>O &3=AØØØ
>O
&Ø = ØØØØ
&1 = ØØØØ
&2 = ØØØØ
&3 = AØØØ
```

O THE OFFSET COMMAND IS USED TO SET A VALUE IN AN OFFSET REGISTER FOR RELATIVE ADDRESSING. ALL OF THE OFFSET REGISTERS CAN BE USED FOR THE ICD MEMORY ADDRESSING PARAMETERS.

❶ > P ① = ② cr   ❷ > P ② cr

- ● TYPE 1 EXAMINE AND CHANGE PORT CONTENTS
- ● TYPE 2 EXAMINE ONLY

① SPECIFIES THE BEGINNING ADDRESS OF THE PORT TO BE CHANGED (THE ASCII OR HEX DATA IS WRITTEN TO THE SPECIFIED ADDRESS) PHY 16 L BYTE

② SPECIFIES DATA TO CHANGE THE PORT ASCII 32 HEX DATA

● NOTE: MULTI-DISPLAY OF DATA IS POSSIBLE USING THE VARIOUS FORMATS BELOW;

CR DISPLAYS DATA AT THE NEXT PORT AFTER CHANGE

, cr DISPLAYS DATA AT THE SAME PORT AFTER CHANGE

Λ cr DISPLAYS DATA AT PRECEDING PORT AFTER CHANGE

/ cr TERMINATES THE PORT COMMAND AFTER CHANGE

EXAMPLE

EXAMINE AND WRITE OUT A PORT;

```
>P FF=12 12 34 56 78
```

READ AND CHANGE PORT IN BYTES USING MULTI-DISPLAY FORMAT;

```
>P 55
55 55:12
56 56:23
57 57:'B'
58 58:'D',
58 58:ØØ^
59 59:12
58 58:/
```

READ AND CHANGE PORT IN WORDS USING MULTI-DISPLAY FORMAT;

```
>P 23
23 23:ØØ
24 24:12
25 25:33
26 26:/
```

P THE PORT COMMAND IS USED TO CHANGE THE PORT CONTENTS WITH ASCII DATA OR HEXADECIMAL NUMBERS. TWO TYPES OF PORT COMMANDS EXIST; ONE EXAMINES AND CHANGES THE PORT CONTENTS AND THE OTHER EXAMINES THE PORT CONTENTS ONLY.

> P I ① = ② cr

① SPECIFIES EITHER;

B U S R Q BUS REQUEST

N M I NON-MASKABLE

I N T R INTERRUPT

PI INT = DI
OR
PI INT = EN

② SPECIFIES EITHER;

E N TO PLACE THE INTERRUPT OR BUS STATE REQUEST SIGNAL IN THE ENABLE STATE

D I TO PLACE THE INTERRUPT OR BUS STATE REQUEST SIGNAL IN THE DISABLE STATE

● NOTE: IF NO PARAMETER IS SPECIFIED, THE CURRENT BUSRQ, NMI AND INTR CONDITIONS ARE DISPLAYED AS MASKED.

EXAMPLE

DISPLAY PIN STATUS;

```
>PI
IN-CIRCUIT MODE 1
NMI     (EN) = Ø
BUSRQ   (EN)
INTR    (EN) = Ø
```

DISABLE BUSREQUEST AND DISPLAY THE STATUS;

```
>PI BUSRQ = DI
>PI
IN-CIRCUIT MODE 1
NMI     (EN) = Ø
BUSRQ   (DI)
INTR    (EN) = Ø
```

P I

THE PIN COMMAND IS USED TO MASK OR UNMASK THE TARGET SYSTEM INTERRUPTION SIGNAL WHEN THE INCIRCUIT MODE IS 1. WHEN THE INCIRCUIT MODE IS Ø, ALL THE INTERRUPTIONS OF THE TARGET SYSTEM ARE IGNQRED AND ARE VALID WHEN THE INCIRCUIT MODE IS 2.

● SEE INCIRCUIT COMMAND

```
> P R 1 cr
```

1 SPECIFIES EITHER TO;

ON OUTPUT THE CHARACTERS SENT FROM THE ICD TO BOTH TERMINAL AND HOST/AUX PORTS

OFF OUTPUT DATA TO THE CONSOLE ONLY

EXAMPLE

TURN PRINT ON AND OFF;

```
>PR ON
>
>
>PR OFF
```

PR THE PRINT COMMAND IS USED TO OUTPUT CHARACTERS SENT FROM THE ICD TO THE TERMINAL AND HOST/AUX PORTS OR THE CONSOLE.

> Q cr

EXAMPLE

ENABLE THE QUIT COMMAND;

```
>Q
```

Q THE QUIT COMMAND IS USED TO RETURN CONTROL BACK TO THE HOST COMPUTER SYSTEM. THIS COMMAND IS VALID ONLY WHEN THE UTILITY SOFTWARE PROGRAM, ZICE IS AVAILABLE. WHEN OPERATIONAL, THE "Ø4" CODE IS TRANSMITTED TO THE HOST COMPUTER SYSTEM.

EXAMPLE

DISPLAY REGISTER WITH TITLES;

```
>R

PC   SP   SZHPNC A  BC   DE   HL   IX   IY   A'F'  B'C'  D'E'  H'L'  I  IF (SP)(HL)
ØØØØ ØØØØ ØØØØØØ ØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ  ØØØØ  ØØØØ  ØØØØ  ØØ  Ø ØØ21 21
```

CHANGE HL TO A VALUE OF AØØØ;

```
>R HL=AØØØ
>R

PC   SP   SZHPNC A  BC   DE   HL   IX   IY   A'F'  B'C'  D'E'  H'L'  I  IF (SP)(HL)
ØØØØ ØØØØ ØØØØØØ ØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ ØØØØ  ØØØØ  ØØØØ  ØØØØ  ØØ  Ø ØØ21 3A
```

❶ > R cr ❷ > R RESET cr ❸ > R ① = ② cr

EXAMPLE

● TYPE 1 REGISTER STATUS (CURRENT REGISTER CONTENTS)
● TYPE 2 REGISTER RESET (INITIALIZES REGISTER TO Ø)
● TYPE 3 REGISTER CHANGE (SPECIFIED CHANGE BELOW)

① SPECIFIES ANY OF THE REGISTERS;

A, F, B, C, D, E, H, L, I

A', F', B', C', D', E', H', L', I'

BC, DH, HL, IX, IY, SP

BC', DE', HL'

CY, S, Z, HC, P, N, IF(IFF1)

② SPECIFIES THE REGISTER CHANGE DATA HEX DATA

● NOTE: WHEN NO PARAMETER IS SPECIFIED, THE CURRENT CONTENTS OF ALL THE REGISTERS ARE DISPLAYED.

R THE REGISTER COMMAND IS USED TO CHANGE THE Z8Ø CPU REGISTER AND PROGRAM COUNTER. THERE ARE THREE TYPES OF REGISTER COMMANDS AVAILABLE; DISPLAY, RESET AND CHANGE.

> S ① , ② , ③ , ④ cr

① ; / D SPECIFIES TO SEARCH FOR UNMATCHED DATA
(DEFAULT; SEARCH IS MADE FOR MATCHED DATA)

② SPECIFIES THE MEMORY ADDRESS TO INITIATE A SEARCH PHY 16

③ SPECIFIES THE MEMORY ADDRESS TO TERMINATE A SEARCH PHY 16 L BYTE

④ SPECIFIES THE SEARCH DATA ASCII 32 HEX DATA

EXAMPLE

```
>S Ø,L3Ø,12
ØØ12


>S/W 1ØØ,11Ø,1234


>S/D Ø,1Ø,ØØ
ØØØ1
ØØØ2
ØØØ3
 --
 --
 --
 --
ØØØ9
ØØØA
ØØØB
 --
 --
 --
 --
ØØ1Ø
```

S THE SEARCH COMMAND IS USED TO SEARCH FOR THE MEMORY CONTENTS AND DISPLAY THE SPECIFIED MATCHED OR UNMATCHED ADDRESS.

> S A / ① , ② , ③ , ④ , ⑤ , ⑥ cr

① SPECIFIES WHICH PORT TO CONSIDER, EITHER THE;

- T TERMINAL PORT (IGNORE SOFTWARE HANDSHAKE)
- P TERMINAL PORT (PERFORM SOFTWARE HANDSHAKE)
- A AUXILIARY PORT (IGNORE SOFTWARE HANDSHAKE)
- H HOST PORT (PERFORM SOFTWARE HANDSHAKE)

② SPECIFIES THE NAME OF THE FILE TO CREATE [1]

③ SPECIFIES MEMORY ADDRESS TO INITIATE OBJECT CREATION PHY 16

④ SPECIFIES MEMORY ADDRESS TO TERMINATE OBJECT CREATION PHY 16

⑤ SPECIFIES THE START ADDRESS OF THE USER PROGRAM PHY 16 HEX ADRS

⑥ SPECIFIES THE SAVE MESSAGE ASCII 32 HEX DATA

[1] NOTE: THIS PARAMETER IS VALID ONLY WHEN THE UTILITY SOFTWARE PROGRAM, ZICE IS AVAILABLE.

EXAMPLE

GENERATE THE TEST.HEX FILE ON DISKETTE;

```
>SA TEST,Ø,37FF,Ø
```

SA

THE SAVE COMMAND IS USED TO CREATE A USER PROGRAM ON A DISKETTE AS THE HEX FILE OR DUMP THE USER PROGRAM TO A SPECIFIED PORT USING THE INTEL FORMAT.

Ⓓ **See APPENDIX D**

EXAMPLE

Check the TERMINAL port input state;

```
>DI 9ØØØ,9ØØC
9ØØØ  1EØ1    LD   E,1
9ØØ2  ØØ      NOP
9ØØ3  B7      OR   A
9ØØ4  CAØ29Ø  JP   Z,9ØØ2H
9ØØ7  77      LD   (HL),A
9ØØ8  Ø4      INC B
9ØØ9  FEØD    CP   ØDH
9ØØB  C8      RET Z
9ØØC  77      LD   (HL),A
>SU/7 ON
>B/7 9ØØ2
>G 9ØØØ,9ØØ8

PC     MC      OP      SP    AF    BC    DE    HL    IX    IY    I   IF(SP)
9ØØ8  Ø4      INC B  ØØ2E  FFØ8  1CØE  Ø3Ø1  AØØØ  ØØØØ  ØØØØ  ØØ  Ø ØØØØ
<Break Hardware A>
```

● NOTE: Only a break set with a BREAK command can be used as a supervisor call

● NOTE: If a supervisor call and a separate specified break exist at the same point, the supervisor call will not be executed.

> S U ① ② cr

① SPECIFIES EITHER;

/ C SET HARDWARE BREAK C AS A SUPERVISOR CALL

/ 7 SET SOFTWARE BREAK 7 AS A SUPERVISOR CALL

/ U SET USER SOFTWARE BREAK AS A SUPERVISOR CALL

② SPECIFIES EITHER;

O N SET BREAK IF USED AS A SUPERVISOR CALL

O F F SET BREAK IF NOT USED AS A SUPERVISOR CALL

● NOTE: IF NO PARAMETER IS SPECIFIED, THE CURRENT SET STATE IS DISPLAYED.

EXAMPLE

S U THE SUPERVISOR COMMAND IS USED TO SET ANY GIVEN BREAK POINT AS A SUPERVISOR CALL. WHEN EXECUTED, DATA IS DIRECTLY TRANSFERRED (WITHOUT STOPPING AT THE SUPERVISOR BREAKPOINT) BETWEEN THE PROGRAM AND THE ICD SERIAL INTERFACE.

● NOTE: THE FUNCTION CODE IS STORED IN THE E REGISTER AND INPUT/OUTPUT DATA IS EXECUTED IN THE A REGISTER.

Ⓒ See APPENDIX C

EXAMPLE

Set the trace mode;

```
>T A,1ØØ,4Ø5
```

Display trace status;

```
>T
(ON)     ALL  Ø1ØØ-Ø3ØØ
```

Execute trace example;

```
>G 1ØØ

PC   MC      OP                 SP    AF    BC    DE    HL    IX    IY    I   IF(SP)
Ø1ØØ 21ØØAØ  LD   HL,ØAØØØH     ØØØ2  ØØ23  1481  15Ø1  AØØØ  ØØØØ  ØØØØ  ØØ  Ø 8185
Ø1Ø3 11ØØBØ  LD   DE,ØBØØØH     ....  ....  ....  BØØØ  ....  ....  ....  ..  . ....
Ø1Ø6 D5      PUSH DE            ØØØØ  ....  ....  ....  ....  ....  ....  ..  . 8ØØØ
Ø1Ø7 11ØØØ3  LD   DE,Ø3ØØH      ....  ....  ....  Ø3ØØ  ....  ....  ....  ..  . ....
Ø1ØA 7E      LD   A,(HL)        ....  FF23  ....  ....  ....  ....  ....  ..  . ....
Ø1ØB 12      LD   (DE),A        ....  ....  ....  ....  ....  ....  ....  ..  . ....
Ø1ØC 13      INC  DE            ....  ....  ....  Ø3Ø1  ....  ....  ....  ..  . ....
Ø1ØD C1      POP  BC            ØØØ2  ....  BØØØ  ....  ....  ....  ....  ..  . 8185
Ø1ØE C3ØØØ4  JP   Ø4ØØH         ....  ....  ....  ....  ....  ....  ....  ..  . ....
Ø4ØØ Ø18Ø16  LD   BC,168ØH      ....  ....  168Ø  ....  ....  ....  ....  ..  . ....
Ø4Ø3 8Ø      ADD  A,B           ....  1511  ....  ....  ....  ....  ....  ..  . ....
Ø4Ø4 Ø3      INC  BC            ....  ....  1681  ....  ....  ....  ....  ..  . ....
Ø4Ø5 FØ
```

Release the trace mode;

```
>T CLR
```

● NOTE: Trace displays ">" with the <ESC> key to accept a command.

● NOTE: Trace is executed by the "GO" command, which starts the display.

❶ > T ① ② , ③ , ④ cr   ❷ > T ⑤ cr

SPACE SPACE

❶ ① ; / S SPECIFIES THE SINGLE STEP MODE (IN THIS MODE, THE ALL OR JUMP TRACE IS EXECUTED EVERY TIME THE SPACE BAR OR RETURN KEY IS PRESSED)

② SPECIFIES EITHER;

A ALL MODE; ALL COMMANDS ARE TRACED AND DISPLAYED

J JUMP MODE; ONLY BRANCH COMMAND IS TRACED AND DISPLAYED

③ SPECIFIES THE BEGINNING ADDRESS OF THE MEMORY AREA TO BE TRACED AND DISPLAYED(DEFAULT IS Ø) PHY 16

④ SPECIFIES THE ENDING ADDRESS OF THE MEMORY AREA TO BE TRACED AND DISPLAYED (DEFAULT IS FFFFF) PHY 16

❷ ⑤ MAKES THE CURRENT TRACE MODE;

O N VALID

O F F INVALID

C L R CLEARED

● NOTE: IF NO PARAMETER IS SPECIFIED, THE CURRENT TRACE MODE SET STATE IS DISPLAYED.

EXAMPLE

T THE TRACE COMMAND IS USED TO SET THE ICD TRACE MODE.

EXAMPLE

USER COMMAND WITH DESIGNATED TERMINATOR;

```
>U !

>A  DIR B:
B: PIP              COM : STAT          COM : DUMP          COM : LOAD          COM
B: ED               COM : ASM           COM : SYSGEN        COM : MOVCPM        COM
```

> U ① cr

① SPECIFIES THE CODE USED AS A TERMINATOR

● NOTE: THE FOLLOWING <u>CANNOT</u> BE SPECIFIED AS THE TERMINATOR CODE;

ESC, NAK, SP, BS, cr

EXAMPLE ▲

U THE USER COMMAND ALLOWS THE CONSOLE TERMINAL TO BE USED TEMPORARILY AS THE TERMINAL FOR THE HOST COMPUTER SYSTEM. WHEN THIS COMMAND IS EXECUTED, THE ICD SENDS ALL CODES (EXCLUDING TERMINATOR CODES RECEIVED FROM THE TERMINAL PORT) TO THE HOST PORT WITHOUT ASSUMING THEM TO BE ICD COMMANDS.

> V / ① , ② , ③ , ④ cr

① SPECIFIES EITHER THE;

- T TERMINAL PORT (IGNORE SOFTWARE HANDSHAKE)
- P TERMINAL PORT (PERFORM SOFTWARE HANDSHAKE)
- H AUXILIARY PORT (IGNORE SOFTWARE HANDSHAKE)
- A HOST PORT (PERFORM SOFTWARE HANDSHAKE)

② SPECIFIES THE OBJECT NAME TO COMPARE[1]

③ SPECIFIES THE BIAS OF THE FILE OR OBJECT TO COMPARE[2]

④ SPECIFIES THE VERIFY MESSAGE ASCII 32 HEX DATA

[1] NOTE: THIS PARAMETER IS VALID ONLY WHEN THE UTILITY SOFTWARE PROGRAM ZICE IS AVAILABLE.

[2] NOTE: IF OMITTED, THE OBJECT STARTING ADDRESS IS ASSUMED

EXAMPLE

COMPARE THE TEST.HEX FILE ON DISKETTE WITH THE MEMORY CONTENTS;

```
>V TEST
ADRS  M  O
ØØØØ


>V/T ,1ØØ
 ADRS  M  O
Ø2ØØ
```

NOTE: "ADRS", "M" AND "O" INDICATE THE MEMORY ADDRESS, CONTENTS OF THE MEMORY ADDRESS AND OBJECT, RESPECTIVELY.

V THE VERIFY COMMAND IS USED TO COMPARE A HOST COMPUTER FILE OR OBJECT IN THE INTEL FORMAT, WITH THE CONTENTS OF THE ICD MEMORY.

D See APPENDIX D

ERROR MESSAGE	COMMAND OCCURRENCE	DISPLAYED WHEN:
UNABLE BREAK ADDRESS	BREAK; GO	WHEN A SOFTWARE BREAK IS SPECIFIED IN THE NONRAM AREA
MULTI BREAK ADDRESS	BREAK, GO	WHEN A SOFTWARE BREAK IS DUPLICATED AT THE SAME ADDRESS
WARNING UNABLE SOFT BREAK	BREAK	IF A SOFTWARE BREAK IS SET AT THE ADDRESS PRESENTLY NOT MAPPED IN RAM
*** FILE NOT FOUND	ALL	WHEN NO FILE EXISTS
*** DISK READ ERROR	ALL	WHEN DISK READ ERROR OCCURS
*** CHECK SUM ERROR	ALL	WHEN SUM CHECK ERROR OCCURS
*** DISK WRITE ERROR	ALL	WHEN DISK WRITE ERROR OCCURS
*** NO DIRECTORY SPACE	ALL	WHEN NO BLANK AREA IS AVAILABLE

ERROR MESSAGE	COMMAND OCCURRENCE	DISPLAYED WHEN:
C?>	ALL	WHEN A COMMAND CODE ERROR OCCURS;
P?>	ALL	WHEN A PARAMETER CODE ERROR OCCURS;
/?>	ALL	WHEN A MODIFIER CODE ERROR OCCURS;
		--- WITH A <BELL> (CONTROL +G), WHEN THE INPUT COMMAND STATEMENT CONTAINS AN ERROR
MEMORY WRITE ERROR AT XXXX	ASSEMBLE, EXAMINE, LOAD MOVE, FILL	DUE TO MEMORY MODIFICATION ERROR (XXXX - IS THE MEMORY ADDRESS AT WHICH THE ERROR OCCURS)
MEMORY TIMEOUT ERROR AT XXXX I/O TIMEOUT ERROR AT XX	ASSEMBLE, DUMP, EXAMINE FILL, LOAD, MOVE, COMPARE, VERIFY, SAVE PORT, SEARCH	IF MEMORY I/O IN THE TARGET SYSTEM DOES NOT RESPOND TO ICD ACCESS, TIMEOUTS A WAITSTATE
XXXX INPUT ERROR	EXAMINE, ASSEMBLE, PORT	AS AN INPUT ERROR. RE-ENTER DATA AFTER ERROR MESSAGE.
BREAK BUSY	BREAK, GO	WHEN THE BREAK SPECIFICATION EXCEEDS THE LIMIT

ERROR MESSAGES

APPENDIX A

HISTORY Command Supplement

REAL-TIME TRACE SPECIFICATIONS

Word width	: Maximum 32 bits
Word size	: 2K words
Acquisition cycle	: CPU machine cycle (150 ns min.)
Effective trace section	: 2046 machine cycles (1.09 ms min.;6 MHz)
Fixed trace data	: A0-A15 MREQ IORQ RD WR M1 S0, S1
Triggering	: The start and end of tracing coordinated with the start and end of emulation can be specifed, as can the event triggered delays.
Display	: Machine cycle display or inverted assembly display.

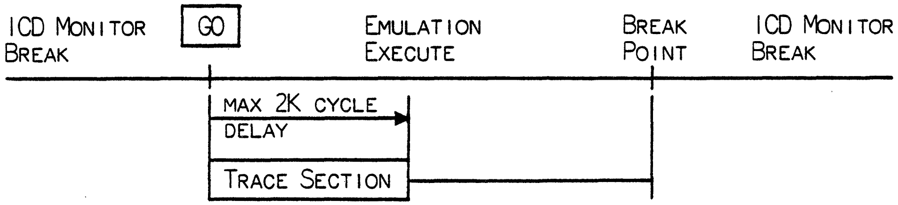
REAL-TIME COUNTER SPECIFICATIONS

Count bit width	: 32 bits
Count clock	: CPU clock (number of states)
Effective count time	: 716 sec.: 6 MHz

TRIGGER MODES

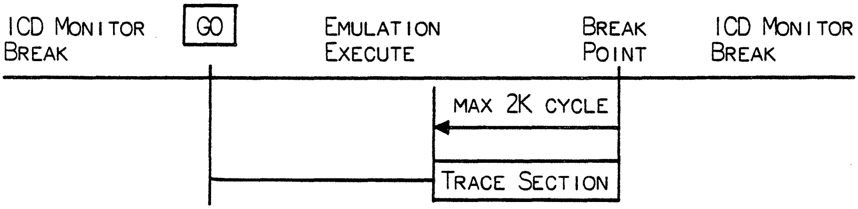
An event trigger with an event point specification or a monitor trigger with a breakpoint specification, is used to designate the trace start and end points. Six trigger modes are available with the ICD HISTORY command.

● BEGIN MONITOR



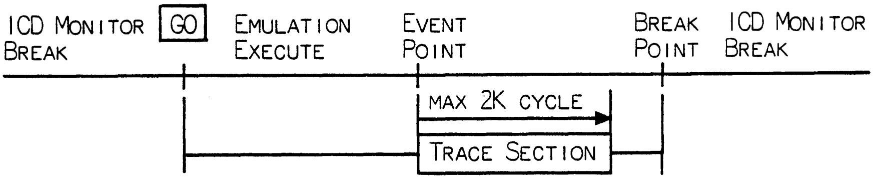
BEGIN MONITOR
TRACING BEGINS AT THE START OF EMULATION (INITIATED BY A GO COMMAND) AND THE END OF TRACING IS AUTOMATICALLY ASSUMED BY THE TRACING RANGE. THE TRACING RANGE IS SPECIFIED BY A DELAY SETTING OF UP TO 2K CYCLES AFTER EMULATION HAS BEGUN.

● END MONITOR



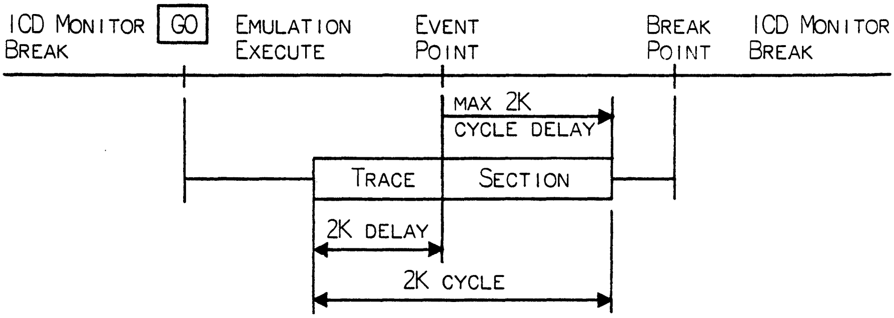
END MONITOR
TRACING BEGINS AT THE START OF EMULATION (INITIATED BY A GO COMMAND) AND ENDS WHEN CONTROL IS BACK WITH THE MONITOR ON BREAK.
NOTE: THE TRACE SECTION IS 2K CYCLES MAXIMUM BEFORE THE BREAKPOINT.

● BEGIN EVENT



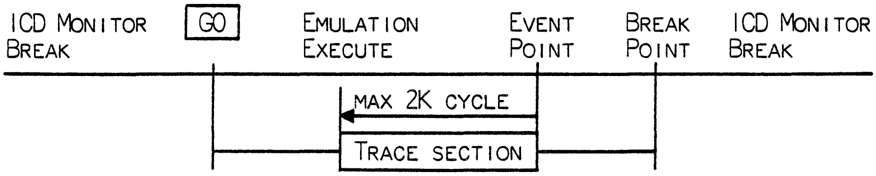
BEGIN EVENT
TRACING BEGINS ON PASSING THE EVENT POINT AFTER THE START OF EMULATION (INITIATED BY A GO COMMAND) AND ENDS AT THE END OF THE TRACING RANGE. THE TRACING RANGE IS SPECIFIED BY A DELAY SETTING OF UP TO 2K CYCLES IMMEDIATELY AFTER PASSING THE FIRST EVENT POINT.

● CENTER EVENT



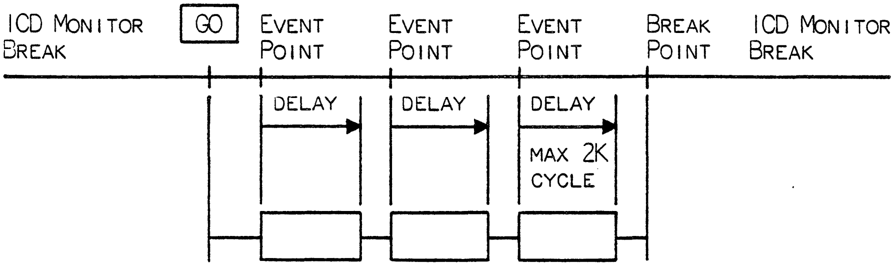
CENTER EVENT
TRACING BEGINS AT THE START OF EMULATION (INITIATED BY A GO COMMAND). THE TRACE SECTION IS SPECIFIED ON EITHER SIDE OF THE EVENT POINT (UP TO 2K MAXIMUM). THE END OF THE TRACE IS DETERMINED BY A DELAY OF UP TO 2K CYCLES AND THE BEGINNING OF THE TRACE IS SPECIFIED BY A MAXIMUM OF UP TO 2K CYCLES MINUS THE DELAY SPECIFICATION.

● END EVENT



END EVENT
Tracing begins at the start of emulation (initiated by a GO command) and ends after passing the first event point. The trace section is a maximum of 2K cycles up to the event point.

● MULTIPLE EVENTS



MULTIPLE EVENTS
Tracing is performed each time an event is passed where multiple events are handled as in loop processing. The trace setting is specified by a delay setting of up to 2K cycles immediately after passing the event point.

APPENDIX B

IN-CIRCUIT Map/Pin Setting

INCIRCUIT MODE	Command set	Mapping set Command				Pin set Command	
		RO	RW	US	NO	EN	DI
I 0	System Mode	RO	RW	(RW)	NO	(DI)	DI
I 1	Partial Incircuit Mode	RO	RW	US	NO	EN	DI
I 2	Full Incircuit Mode	(US)	(US)	US	NO	EN	(EN)

IN-CIRCUIT COMMAND MAP/PIN SETTING

(RW) : OPERATES AS RW REGARDLESS OF MAP COMMAND SETTING.

(US) : OPERATES AS US REGARDLESS OF MAP COMMAND SETTING.

(EN) : OPERATES IN EN STATE REGARDLESS OF THE PIN COMMAND SETTING.

(DI) : OPERATES IN DI STATE REGARDLESS OF THE PIN COMMAND SETTING.

FUNCTION CODE KEY

(1) Function code ØØ or 1Ø: on character input from the port.

Input parameters;
Register Register E : ØØH Specifies the TERMINAL port.
1ØH Specifies the HOST/AUX port.

End state;
Register Register A : Entry characters

Characters input from the port specified with an input parameter are stored in register A. If there is no character input from the specified supervisor call, control does not return to the program until character input occurs.

(2) Function code Ø1 or 11 : Port input state signal fetch.

Input parameter;
Register Register E : Ø1H Specifies the TERMINAL port.
11H Specifies the HOST/AUX port.

End state;
Register Register A : ØØH There is no input data.
FFH There is input data

This function informs the system if input data from the port specified with an input parameter.

(3) Function code Ø2 or 12 : One character output to port.

Input parameter;
Register Register E : Ø2H Specifies the TERMINAL port.
12H Specifies the HOST/AUX port.

End state;
Register Register A : Output characters

This function outputs characters stored in register A to the port specified with an input parameter. If output of the preceding data is not completed at the supervisor call, control does not return to the target program until output is completed.

(4) Function code Ø3 or 13 : Port output state signal fetch.

Input parameter;
Register Register E : Ø2H Specifies the TERMINAL port.
13H Specifies the HOST/AUX port.

End state;
Register Register A : ØØH Output has not been completed.
FFH Output has been completed.

This function informs the system if data from the port specified by a parameter has been completed.

APPENDIX C

SUPERVISOR Function Code Setting

FUNCTION	FUNCTION CODE	DATA OUT	DATA IN
	E - reg	A - reg	
TERMINAL Port data in	0 0	-	RECEIVE DATA
HOST/AUX Port data in	1 0	-	RECEIVE DATA
TERMINAL Port input status read	0 1	-	Input status
HOST/AUX Port input status read	1 1	-	Input status
TERMINAL Port data out	0 2	Output data	-
HOST/AUX Port data out	1 2	Output data	-
TERMINAL Port output status read	0 3	-	Output status
HOST/AUX Port output status read	1 3	-	Output status

◀See FUNCTION CODE KEY

APPENDIX D

OBJECT I/O Procedure

APPENDIX D DISPLAYS THE OBJECT I/O PROCEDURE FOR THE COMMANDS;

LOAD SAVE VERIFY

MODE	REMOTE	LOCAL
/T	X	X
/P	0	0
/A	X	X
/H	0	0
DEFAULT	0 (HOST)	X (TERMINAL)

0 : DENOTES SOFTWARE HANDSHAKE IS PEFORMED. FOR THE CORRECT COMMUNICATION PROTOCOL, PLEASE REFER TO;

SOFTWARE SPECIFICATIONS OF HOST COMPUTER SYSTEM

NOTE: FOR THE LOAD MESSAGE TO BE USED, /T AND /A MUST BE SPECIFIED AT THE OBJECT INPUT PORT.
NOTE: THE OBJECT NAME MAY BE SPECIFIED AS THE USER DEFINITION LOAD MESSAGE IF /P AND /H ARE DESIGNATED DURING SOFTWARE HANDSHAKE.

ZAX

Zax Corporation
2572 White Road, Irvin, California 92714
(714) 474-1170 ● 800-421-0982 ● TLX 183829