GenRad
futuredata

GenRad Corp.
futuredata

Universal Logic Analyzer Manual

Preliminary

MICROCOMPUTER
SYSTEMS

GenRad Corp.
futuredata

Universal Logic Analyzer Manual

Preliminary

GenRad Corp.
futuredata

6151 W. Century Blvd.
Suite 1124
Los Angeles, CA 90045

PREFACE

The AMDS Logic Analyzer manual is divided into five sections as follows:

Section 1 - Introduction. Section 1 contains a brief description of the Logic Analyzer's theory and hardware design.

Section 2 - Operation. Section 2 explains the basic procedures involved in running a program with logic analysis.

Section 3 - List of Commands. Section 3 defines the Logic Analyzer commands and associated parameters.

Section 4 - Debugger Modifications. Section 4 provides information on Debugger relocation and interrupt vector modification.

Section 5 - Installation. Section 5 provides step-by-step instructions to integrate the Logic Analyzer and External Probe Assembly hardware with the AMDS.

HOW TO USE THIS MANUAL

Sections. The beginning page of each section is marked with a small black block on the edge of the page. Beginning pages can thus be quickly located by fanning the pages from front to back.

Conventions.

Letters. Commands and statements are printed in upper case and must be typed exactly as shown.

Parameters. Parameters appear in lower case. Optional parameters are enclosed in brackets:

[parameter]

Required parameters are enclosed in braces:

{parameter}

Brackets and braces are shown for clarification purposes only; they are not part of the input.

Dots. A series of horizontal or vertical dots in command syntax indicates that the parameter preceding the dots may be repeated indefinitely.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

Section 1
Introduction

1-1. SYSTEM OVERVIEW

The Logic Analyzer greatly simplifies troubleshooting and debugging tasks by providing a means through which the user can view bus data during program execution. The Logic Analyzer gathers a maximum of 256 traces for up to 24 address lines, up to 16 data lines, the Read/Write line, the I/O line, the DMA line, the Instruction Fetch line, and four external user-defined lines. Information is recorded synchronously with the system or prototype clock to 10 MHz, and stored in an area of memory contained in the Logic Analyzer hardware.

The Trace Qualifier register selects the traces to be stored in Logic Analyzer memory. The type of information to be stored can be restricted, such as a READ or WRITE to a particular location. Three breakpoints are provided to halt data storage (and, optionally, to halt program execution). The breakpoints may be ANDed or ORed. In addition, a count may be set for each breakpoint so that the breakpoint condition(s) will be satisfied only when the condition(s) has occurred n number of times. A delay count may be entered as an element of the Trace Qualifier to continue storage for up to 65,535 qualified bus cycles after the breakpoint conditions have been met.

Recorded information may be displayed in either a parallel state (cycle-by-cycle) format or a serial timing diagram (line-by-line) format. The parallel state display shows 16 traces at a time in hexadecimal form. Each trace represents the values of all of the 48 lines during one bus cycle. The serial timing diagram display shows the logic states of up to 16 selected lines in square wave form. Each line may contain up to 64 states. Special function keys and commands may be used with either form of display to step through all 256 traces.

1-2. PHYSICAL DESCRIPTION

The Logic Analyzer package consists of four printed circuit cards and an external probe assembly. The cards plug into the AMDS card cage and are connected together with a series of cables.

Section 2
Operation

2-1. LOGIC ANALYZER ENTRY AND EXIT

To invoke the Logic Analyzer, from the Debugger display type

A

To return to the Debugger, type the D command in either of the following ways:

D(expression)	where expression consists of a constant, an indirect address, a relative address, a symbolic address, or a combination of such values;
D$	where $ instructs the Logic Analyzer to display the Debugger at the current location of the program counter (PC=); or
D	which returns to the Debugger address previously displayed.

2-2. SETTING UP THE LOGIC ANALYZER

Upon entry to the Logic Analyzer, the CRT screen displays four major parameters as shown in Figure 2-1. These parameters: the AND/OR provision, the Trace Qualifier, the breakpoints, and the timer increment are discussed below.

AND/OR Provision. The AND mode of operation requires that the designated breakpoint conditions be satisfied in sequence before the Logic Analyzer will stop data storage. In other words, the B1 conditions cannot be satisfied until the B0 conditions have been satisfied. The OR provision causes the Logic Analyzer to halt the storage process when any one of the three breakpoint conditions has been met.

Trace Qualifier. The Trace Qualifier (Q) specifies the characteristics of the traces to be stored in Logic Analyzer memory. A delay count may be entered to continue storage for a specified number (up to 65,535) of bus cycles after the breakpoint conditions have been met. The allowable Trace Qualifier parameters are listed in Table 2-A.

Breakpoints. The Logic Analyzer provides three breakpoints, labelled B0, B1, and B2. The Logic Analyzer stops the storage process when the specified conditions have been met. The allowable breakpoint parameters are listed in Table 2-B.

Timer Increment. The timer measures the length of time between satisfaction of breakpoints B0 and B1. Three types of measurement units may be selected by the commands listed below:

TN for .1 microsecond (100 nanoseconds)

TU for 1 microsecond

TB for 1 bus cycle

Default Conditions. The default conditions for the above parameters are:

B0: Break immediately on all but DMA cycles

B1, B2: Ignored

Q: Store all traces except DMA

T: Bus cycles

The Trace Qualifier line and each of the breakpoint lines may be reset to their respective default conditions by typing the line name followed by a RETURN. For example:

B0 RETURN

Q RETURN

will reset both the B0 breakpoint line and the Trace Qualifier line to the default conditions listed above.

FIGURE 2-1.

```
AND  COUNT    ADDR  DATA  IO  RW  DMA  M1  EXTR      PC=0000  Eaddr=FFFF
B0   1      =XXXX   XX    X   X   0    X   XXXX
B1   0      =XXXX   XX    X   X   0    X   XXXX      TIMER=0 cycles
B2   0      =XXXX   XX    X   X   0    X   XXXX
Q    0      =XXXX   XX    X   X   0    X   XXXX
```

Table 2-A. Trace Qualifier Parameters

Parameter	Condition	Explanation
COUNT (or C)	=0	No qualification of stored data.
	=1 to 65,535 (decimal)	Number of qualified cycles to be stored after breakpoint conditions are satisfied (DELAY).
ADDR (or A)	=4 Hex Characters	
	=#SYMBOL	Symbolic Debug mode.
	=XXXX	Don't care. Hex characters and "X" can be mixed, e.g.: ABCE X123 X1X1
	={expression}	Store when ADDR equals allowable AMDS {expression}
	>={expression}	Store when ADDR is greater than or equal to {expression}. "X" not allowed.
	<={expression}	Store when ADDR is less than or equal to {expression}. "X" not allowed.
DATA (or D)	=2 Hex Characters	
	=XX	Don't care. Hex characters and "X" can be mixed, e.g.: AB AX 1X
IO	=1	Store on I/O execution.
	=0	Store on other than I/O execution.
	=X	Don't care.
RW	=R	Store on READ cycle (memory or I/O).
	=W	Store on WRITE cycle (memory or I/O).
	=X	Don't care.

Table 2-A. Trace Qualifier Parameters (Concluded)

Parameter	Condition	Explanation
DMA	=1	Store on DMA cycle.
	=0	Store on other than DMA cycle.
	=X	Don't care.
EXTR (or E)	=4 Binary Characters	
	=1	Store if line high.
	=0	Store if line low.
	=X	Don't care.

Table 2-B. Breakpoint Parameters

Parameter	Condition	Explanation
COUNT (or C)	=0	Breakpoint ignored.
	=1 to 65,535 (decimal)	Number of breakpoints past the breakpoint needed to satisfy conditions.
ADDR (or A)	=4 Hex characters.	
	=#SYMBOL	Symbolic Debug mode.
	=XXXX	Don't care. Hex characters and "X" can be mixed, e.g.: ABCE X123 X1X1
	={expression}	Break when ADDR equals any allowable address expression.
	>={expression}	Break when ADDR is greater than or equal to {expression}. "X" is not allowed.
	<={expression}	Break when ADDR is less than or equal to {expression}. "X" is not allowed.
DATA (or D)	=2 Hex characters	
	=XX	Don't care. Hex characters and "X" can be mixed, e.g.: AB AX 1X
IO	=1	Break on I/O execution.
	=0	Break on other than I/O execution.
	=X	Don't care.
RW	=R	Break on READ cycle (memory or I/O).
	=W	Break on WRITE cycle (memory or I/O).
	=X	Don't care.

Table 2-B. Breakpoint Parameters (Concluded)

Parameter	Condition	Explanation
DMA	=1	Break on DMA cycle.
	=0	Break on other than DMA cycle.
	=X	Don't care.
EXTR (or E)	=4 Binary characters	
	=1	Break if line high.
	=0	Break if line low.
	=X	Don't care.

Example: Breakpoint and Trace Qualifier Set-Up. Consider the short program below:

```
START   INC   A
        OUT   Ø
        JMP   START
```

When assembled, linked and located at 100H and loaded with the symbol table, the 8080 program is:

Address	Data	
100	3C	INR A
101	D3	OUT 0
102	00	
103	C3	JMP 100H
104	00	
105	01	

The Logic Analyzer might be set up in the following manner:

Desired BREAK Condition	Logic Analyzer Command
Output X'AA'	BØ,D=AA,IO=1
100 times past JMP instr.	BØ,C=100,D=C3,M1=1
Time in microseconds between output of 00 and X'FF'	BØ,D=00,IO=1 B1,D=FF,IO=1 TU
First time program jumps out of normal sequence	0 (or breakpoints) BØ,A<=#START-1,M1=1 B1,A>=106,M1=1
Store only writes to port Ø, break on the 10th write after X'FF' is output	BØ,D=FF,IO=1 Q,C=10,A=XX00,IO=1

2-3. EXECUTING A PROGRAM WITH LOGIC ANALYSIS

To begin Logic Analyzer operation, the user must select the Debugger (JD), load his program (L), and go to the Logic Analyzer (A). The CRT screen will display the default information shown in Figure 2-1. Figure 2-1 presents the set-up information in 8-bit processor form. Each hexadecimal digit in the ADDR and DATA column represents four bits (0 through F), and the information in the remaining columns represents one bit.

After setting the parameters to his particular needs, the user may execute his program (E) either from the Logic Analyzer or he may go back to the Debugger (D) and execute from there.

Execution From Logic Analyzer. A program may be executed from the Logic Analyzer in one of three ways:

E(expression)	where expression is any allowed combination of addresses or symbolic references;
E$	where execution begins at the current program counter (PC=);
E	where execution begins at the cursor of the active side of the Debugger display (Eaddr=).

When program execution is begun in the Logic Analyzer display mode, both the storage process and program execution will halt when the Logic Analyzer breakpoints have been satisfied. The Logic Analyzer display will return to the screen. If the breakpoints have not been satisfied, the Debugger will return to the display.

Execution From Debugger. When program execution is begun in the Debugger display mode, the storage process will halt when the Logic Analyzer breakpoints have been satisfied, and the program will continue until a software or hardware breakpoint is reached, or until the [BREAK] or [RESET] key is depressed manually. The Debugger display will return to the screen. Enter the Logic Analyzer by typing an A.

If the breakpoints were satisfied, the message

BREAKPOINT CONDITIONS SATISFIED

will flash on the display; the "BREAK+0" message represents the break trace line.

If the breakpoints were not satisfied, the Logic Analyzer stops storing data several instructions after program execution has stopped. These instructions are the operating system instructions which interrupt the system, save the user's program status, and issue an OUT to port AF which stops the Logic Analyzer storage process.

Table 2-C. Return Display Conditions

Execution Starting Point	Return Display	
	Debugger	Logic Analyzer
Debugger Display	User must type an A to go to Logic Analyzer, where message will flash if Logic Analyzer breakpoints have been satisfied.	Not Possible
Logic Analyzer Display	Logic Analyzer breakpoints were not satisfied.	Logic Analyzer breakpoints were satisfied. Message will flash.

2-4. INTERPRETATION OF RESULTS

Two display formats are available to users of the Logic Analyzer: a parallel state (hexidecimal) display and a serial (timing diagram) display. The parallel state display is most useful when comparing program execution history stored by the Logic Analyzer with a program listing. The serial display is most useful in determining the timing relations between signals, such as a data line and an external line.

Parallel State Display. The parallel state display shows 16 traces at a time in hexadecimal form. Each trace represents the values of the 48 lines during one bus cycle. As shown in Figure 2-2, the screen is split, the left half displaying the disassembled program, and the right half displaying the bus data.

The up and down arrow keys may be used to step through all 256 traces. To move up or down quickly, the P command may be used. For example:

P+5 moves ahead 5 lines

P-10 moves back 10 lines

The BREAK ± indicator on the reverse videoed line shows the number of qualified bus cycles between the reverse videoed trace and the trace which caused the break conditions to be satisfied (or the last trace stored if the break conditions were not satisfied). The break trace line is highlighted.

Serial (Timing Diagram) Display. The timing diagram is constructed from the data stored in the Logic Analyzer, and uses four special graphics characters to produce the timing diagrams:

‾ high

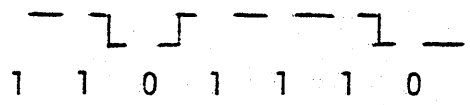
_ low

⌐ rising knee

¬ falling knee

The 64 bits of information, representing the status of one line, are examined, and the appropriate symbol (high or low) is produced for each bit until a change of state is detected.

For example, the bit string 11011100 appears in the timing diagram as:



Note that the signals change (high or low) between the bits.

The up, down, left, and right arrow keys may be used to step through all 256 traces. To quickly move left or right across the screen, the P command may be used, e.g.:

P+5 moves 5 traces to the right

P-10 moves 10 traces to the left

To view one or more lines in non-sequential order, the FS command followed by the desired line number(s) is used. For example:

FS,A02,D1,RW

moves the A02, D1, and RW lines to the top of the display.

The BREAK $\pm$ indicator on the reverse videoed line shows the number of qualified bus cycles between the reverse videoed trace and the trace which caused the break conditions to be satisfied (or the last trace stored if the break conditions were not satisfied).

The timing diagram is continuous only if the Logic Analyzer is storing all machine cycles; thus, all trace qualifier (Q) parameters which effect storage must be set to "Don't Care (X)".

FIGURE 2-2. SERIAL DISPLAY

```
ND  COUNT    ADDR  DATA  IO  RW  DMA  MI  EXTR      PC=0000  Eaddr=0000
0   1       =XXXX   XX    X   X   0    X  XXXX
1   0       =XXXX   XX    X   X   0    X  XXXX      TIMER=0 cycles
2   0       =XXXX   XX    X   X   0    X  XXXX
    0       =XXXX   XX    X   X   0    X  XXXX

PCODE    MNEMONIC  OPERAND              ADDR  DATA  IO  RW  DMA  MI  EXTR
33E74     JP       X'743E'              0038  C3    0   R   0    1   1111
                                        0039  3E    0   R   0    0   1111
                                        003A  74    0   R   0    0   1111
5         PUSH     AF                   743E  F5    0   R   0    1   1111
                                        6FF9  00    0   W   0    0   1111
                                        6FF8  42    0   W   0    0   1111
A0F88     LD       A,(X'880F')          743F  3A    0   R   0    1   1111
                                        7440  0F    0   R   0    0   1111  BREAK-8
                                        7441  88    0   R   0    0   1111
                                        880F  00    0   R   0    0   1111
606       OR       X'06'                7442  F6    0   R   0    1   1111
                                        7443  06    0   R   0    0   1111
3F6       OUT      (X'F6'),A            7444  D3    0   R   0    1   1111
                                        7445  F6    0   R   0    0   1111
                                        06F6  06    1   W   0    0   1111
                                        7446  D3    0   R   0    1   1111
```

FIGURE 2-3. PARALLEL DISPLAY

```
ND  COUNT    ADDR  DATA  IO  RW  DMA  MI  EXTR      PC=4841  Eaddr=4840
0   1       =XXXX   XX    X   X   0    X  XXXX
1   0       =XXXX   XX    X   X   0    X  XXXX      TIMER=0 cycles
2   0       =XXXX   XX    X   X   0    X  XXXX
    0       =XXXX   XX    X   X   0    X  XXXX

                                  BREAK+0

A15                                                          A15
A14                                                          A14
A13                                                          A13
A12                                                          A12
A11                                                          A11
A10                                                          A10
A09                                                          A09
A08                                                          A08
A07                                                          A07
A06                                                          A06
A05                                                          A05
A04                                                          A04
A03                                                          A03
A02                                                          A02
A01                                                          A01
A00                                                          A00
```

WAVEFORM DISPLAY

IN

PREPARATION

Section 3
List of Commands

COMMAND

A

EXPLANATION

The A command enters the Logic Analyzer from the Debugger.

COMMAND

D

EXPLANATION

The D command returns to the Debugger from the Logic Analyzer.

COMMAND

$\left\{\begin{matrix}A\\O\end{matrix}\right\}$

EXPLANATION

A selects the AND mode of operation; the breakpoints must be satisfied in sequence before the Logic Analyzer will halt data storage. O selects the OR mode which causes the Logic Analyzer to halt data storage when any one of the specified conditions has been met.

COMMAND

$T\left\{\begin{matrix}N\\U\\B\end{matrix}\right\}$

EXPLANATION

The T command sets the timer to measure the length of time between satisfaction of B0 and B1 in either .1 μs, 1 μs, or 1 bus cycle increments.

COMMAND

```
B{n} [,{C|COUNT} = {dec}] [,{A|ADDR} = [>|<] {expression|hex4}] [{D|DATA} = {expression|hex2}]
     [,RS = {R|W|X}] [,IO = {0|1|X}] [,M1 = {0|1|X}] [,DMA = {0|1|X}] [{E|EXTR} = {binary 4}]
```

EXPLANATION

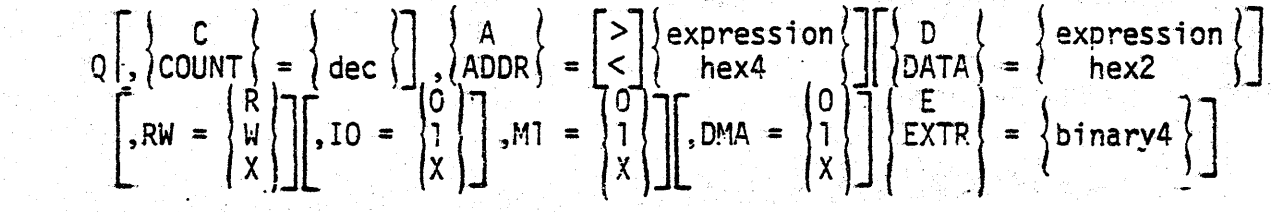
n	specifies which breakpoint (0, 1 or 2) is to be defined.
COUNT	allows the user to select the number of times (up to 65,535) the specified condition must occur before the breakpoints can be satisfied.
ADDR	specifies the breakpoint address. This parameter may be a set number of 4 or 6 hexadecimal digits (depending on processor type), or an expression. ADDR may be set equal to (=), greater than or equal to (>=), or less than or equal to (<=). Don't care (X) symbols may be specified only when using the equal (=).
DATA	specifies the value of the data lines. This number is entered in 2 or 4 hexadecimal digits (depending on processor type).
RW	specifies a Read (R) or Write (W) bus cycle, or both (X).
IO	specifies an I/O port (1) or memory address cycle (0), or both (X).
M1	specifies an op-code fetch cycle (1), or non-op-code data (0), or both (X).
DMA	specifies a DMA cycle (1) or non-DMA cycle (0) or both (X). This is actually the HALT line.
EXTR	specifies the value of the 4 external lines in binary.

Note: X specifies that any value will satisfy the breakpoint conditions for that parameter.

EXAMPLES:

Correct:	B0,ADDR>=1025	
	B0,A=10X4,DATA=33	
Incorrect:	B1,ADDR<=10X1	(X not allowed with <=)
	B2,ADDR>1040	(> not allowed)

COMMAND



EXPLANATION

X	represents a don't care condition for the four bits in that nibble. Individual bits cannot be set in hexadecimal notation. However, in binary form, they can be set to X in place of 1 or 0.
COUNT	specifies the number of qualified bus cycles (up to 65,535) that will be stored after the Logic Analyzer breakpoints have been met.
ADDR	specifies the address(es) by either a set number, a 4- or 6-digit hexadecimal number (depending on processor type), or an expression. This parameter may be set equal to (=), greater than or equal to (>=), or less than or equal to (<=). Don't care (X) symbols may be used with the equal (=).
DATA	specifies the value of the data lines in 2- or 4-digit hexadecimal form.
RW	specifies a Read (0) or Write (1) bus cycle, or either (X).
IO	specifies an I/O port (1) or memory access cycle (0), or either (X).
MI	specifies an op-code fetch (1) or non-op code data, or either (X).
DMA	specifies a DMA cycle (1) or non-DMA cycle (0), or either (X).
EXTR	specifies the value of the 4 external lines in binary.

EXAMPLES

Correct:	Q,DATA=37,M1	
	Q,COUNT=4,ADDR=0100	
Incorrect:	Q,ADDR>=10XX	(X not allowed with >=)
	Q,DATA<=40	(<= not allowed on DATA parameter)

COMMAND

FP

FS[,{line name}] ...

EXPLANATION

F is the format command. P selects the parallel (cycle-by-cycle) mode which displays 16 traces at a time. Each trace represents the values of the 48 lines during 1 bus cycle.

S selects the serial (line-by-line) display mode in which the logic states of a selected line may be viewed 64 at a time.

FS[,{line name}] ... allows the user to view any of the following lines in a specified order:

A15	most significant bit of ADDR	D07	most significant bit of DATA
A14		D06	
A13		D05	
A12		D04	
A11		D03	
A10		D02	
A09		D01	
A08		D00	least significant bit of DATA
A07		IO	
A06		RW	
A05		DMA	
A04		M1	
A03		E0	
A02		E1	
A01		E2	
A00	least significant bit of ADDR	E3	

COMMAND

EXPLANATION

The up arrow moves forward through displayed data.

COMMAND

EXPLANATION

The down arrow moves backward through displayed data.

COMMAND

EXPLANATION

The right arrow moves displayed data to the right. This command applies only to the serial display.

COMMAND

EXPLANATION

The left arrow moves displayed data to the left. This command applies only to the serial display.

COMMAND

P{+}{count}

EXPLANATION

The P command is used to quickly move up and down in parallel mode or right and left in serial mode. The count parameter must be a decimal number between 0 and 65,535.

Section 4
Debugger Modification

4-1. DEBUGGER RELOCATION

As supplied, the Debugger program is configured for loading into the upper portion of memory in a 48K system. The Debugger can, however, be relocated to accommodate the programmer's individual needs, with the following restrictions.

No modules may be loaded at the following locations:

1) A000 to BFFF (in a 48K system this area contains no memory)
2) E800 to FFFF (in a 48K system this area contains no memory)
3) D500 to D5FF (contains the boot loader work area)
4) E000 to E7FF (contains the boot PROM)

The modules listed below must be loaded at the locations specified:

1) Module WORKA (the command file processor work area) must be loaded to D600
2) Module DEBG256 (which cannot be mapped to user memory) must be loaded on a 256-byte boundary
3) Module CRT must be loaded on a 2K-byte boundary.

When the new locations have been decided upon, the modifications should be entered into command file DEBUG.L which, when executed, will relink the Debugger. The Linker Memory Map may be printed and used to compare the modules' new locations with the above list of restrictions.

An example Linker Memory Map is shown on the following page. Notice that the Debugger program has been moved to location 200 in the lower portion of memory, and DEBG256 has been relocated to 4700.

```
-------- LINKER V01 ----------

ADDR   RSEG      FILE         LENGTH

0200   DEBUGR    DEBUGZ80. R  17C4
19C4   BRKRST    BRKRST. R    000C
19D0   ANALYZ    ANALYZ. R    05EB
1FBB   SETPARM   LASCAN. R    037E
2339   LADCON2   LADCON3. R   0091
23CA   LAPAR     LAPARD. R    0323
26ED   LASERD    LASERD. R    0174
2861   RDLAD     RDLADT. R    00A6
2907   DIO       NDIO. R      0A07
330E   LALOAD    LALOAD. R    030E
361C   KEYIN     KEYIN. R     0295
38B1   LINEIN    LINEIN. R    00EE
399F   KIO       KIO. R       01D2
3B71   DISAM80   Z80. R       034A
3EBB   TABLE1    Z80TAB. R    07C8
4700   DEBG256   DEBUGZ80. R  00E5
D600   WORKA     KEYIN. R     0000
D600   WORKA     LINEIN. R    0000 D
D600   WORKA     NDIO. R      0000 D
D800   CRT       DEBUGZ80. R  07B0
D800   CRT       KIO. R       07D0 D
```

4-2. MODIFYING DEBUGGER INTERRUPT VECTOR USAGE

The Debugger requires three interrupt vectors as listed below:

1) the software breakpoint feature
2) Logic Analyzer operation, the [BREAK] key, and the hardware breakpoint feature
3) the [RESET] key

These functions use the vector locations shown in Table 4-A.

Table 4-A. Debugger Use of Interrupt Vectors

Function	Processor Type			
	Z80	8085	8080	6800/02
[RESET] key	X'00'	X'00'	X'00'	X'FFFE'
Logic Analyzer Interrupts	NMI (X'66')	RST5.5 (X'24')	RST6 (X'30')	X'FFFC'
[BREAK] key	NMI (X'66')	RST5.5 (X'24')	RST6 (X'30')	X'FFFC'
Hardware breakpoint	NMI (X'66')	RST5.5 (X'24')	RST6 (X'30')	X'FFFC'
Software breakpoint	RST6 (X'30')	RST6 (X'30')	RST7 (X'36')	X'FFFA'

According to user needs, the software breakpoint vector may be changed to any one of the software restarts (0-7) by relinking the Debugger with the DEBUG.L command file. Note, however, that the following effects will be produced:

1) If RST0 is used, [RESET] key operation will be lost.
2) If RST6 is used (8080 only), the Logic Analyzer, [BREAK] key, and hardware breakpoint feature will not be operable.

With these restrictions in mind, the vector interrupts may be changed by entering the following statement:

```
J^1:DEBUG.L,n,          where n is the restart location (0-7)
```

If the NMI (Z80 or 6800/02), RST5.5 (8085), or RST6 (8080) interrupt is used by the programmer, operation of the Logic Analyzer, hardware breakpoint, and [BREAK] key may be retained by following the steps below:

1) After loading the Debugger, enter the address of the user interrupt routine in location DEBG256. Use low-high format for 8080, 8085, or Z80 systems and high-low format for 6800/02.
2) In location X'66' (Z80, 8085) or location X'30' (8080) store a jump instruction to DEBG256+2. In location FFFC (6800/02) store the address of DEBG256+2 in high-low format.

When any of these interrupts occur, the Debugger checks the cause of the interrupt (Logic Analyzer breakpoint, hardware breakpoint, or [BREAK] key). If the interrupt is due to none of these causes, the Debugger will jump to the user's interrupt routine.

Section 5
Installation

5-1. LOGIC ANALYZER INSTALLATION

The Logic Analyzer consists of three Analyzer cards and one Analyzer Interface card, which are labelled as follows:

Analyzer A is labelled 10160A under J1.

Analyzer B is labelled 10161A near J1.

Analyzer C is labelled 10162A on the left edge of the card.

The Analyzer Interface card is labelled 10159A on the upper right corner.

1. Place the three Analyzer cards in consecutive order (A, B, and C). Attach the 4-inch 3-connector 40-pin cable to J2 on each card.
2. Place the Analyzer Interface card in front of Analyzer Card A and attach J4 on the Analyzer Interface card to J1 on Analyzer Card A with the shorter 2-connector 50-pin cable.
3. Use the longer 2-connector 50-pin cable to connect J1 on the Analyzer Interface card to J1 on Analyzer Card B.

5-2. EXTERNAL PROBE ASSEMBLY CONNECTION

Connect the 20-conductor cable on the External Probe Assembly to J6 on the Analyzer Interface card.

5-3. CPU AND ICE INTERFACE CARD CONNECTION

Position the CPU card and the In-Circuit Emulator Interface (ICE) card in front of the Logic Analyzer card set. Connect the CPU and ICE cards by the longer 3-connector, 40-pin cable as follows:

a) For Z80 CPU and ICE Interface card sets, connect the cable to J2 on the Analyzer Interface card.

b) For 8085 CPU and ICE Interface card sets, connect the cable to J3 on the Analyzer Interface card.

c) For 6802 CPU and ICE Interface card sets, connect the cable to J5 on the Analyzer Interface card.

d) For 8080 CPU and ICE interface card sets, all connections are via the 100 pin mother board.

The 3-connector, 40-pin cables replace the 2-connector, 40-pin cables used to connect the CPU and ICE Interface cards.

Table 5-A. Jumper Connections

	6800 No Ice	6800 Ice	8080 No Ice	8080 Ice	Z80 No Ice	Z80 Ice
INT	X		X		X	
ICE		X		X		X
68M	X	X				
SBM			X	X	X	X
* R/S						

	8085 No Ice	8085 Ice	6802 No Ice	6802 Ice	8080-1 No Ice	8080-1 Ice
INT	X		X		X	
ICE		X		X		X
68M			X	X		
SBM	X	X			X	X
* R/S						

* R/S Indicates a RUN/STOP mode for interrupt. Reserved for later expansion capability.

5-4. JUMPER CONNECTIONS

The Analyzer Interface jumper connections are listed in Table 5-A. above.

GenRad

futuredata

11205 SOUTH LA CIENEGA BOULEVARD LOS ANGELES CALIFORNIA 90045 TWX 910-328-7202 TEL (213) 641-7700