Advanced
Micro
Computers
Distributed by
Advanced Micro Devices

MonoBoard Monitor

User's Manual

	REVISION RECORD
REVISION	**DESCRIPTION**
A	Original Release
(3/2/79)	
B	Manual Updated
(8/17/79)	
Publication No. 00680114	

REVISION LETTERS I, O, Q AND X ARE NOT USED

Address comments concerning
this manual to:

ADVANCED MICRO COMPUTERS
Publications Department
3340 Scott Boulevard
Santa Clara, CA 95051

PREFACE

This manual provides operating instructions for the Monitor program provided for the AMC 95/4000 series MonoBoard Computers; it is intended to assist the system designer in developing programs for the MonoBoard Computer. Familiarity with the MonoBoard architecture and the serial I/O interface is assumed.

The information in this publication is believed to be accurate in all respects. However, no responsibility is assumed for errors that might appear in this publication. Advanced Micro Computers disclaims responsibility for any consequences resulting from the use thereof. No part of this manual may be copied or reproduced in any form without prior written permission from AMC.

This product is intended for use only as described in this document. Advanced Micro Computers cannot be responsible for the proper functioning of undescribed features or parameters.

TABLE OF CONTENTS

MONOBOARD MONITOR

1-1. INTRODUCTION

The MonoBoard Monitor provides a basic program development tool for the MonoBoard computer board. The Monitor provides the following via keyboard/CRT command:

- o memory inspection and modification
- o program load, execute, and breakpoint debug
- o program save/load via papertape
- o data save/load via papertape

When the MonoBoard Monitor is initiated at power-on, an identification message followed by a prompt character (<) is displayed. The Monitor then waits for a command via the keyboard. Commands are specified by single alphabetic characters followed by one or more parameters and a carriage return.

1-2. INSTALLATION

The MonoBoard Monitor is supplied on a single 2716 E-PROM. To install the Monitor, insert the E-PROM into the MonoBoard memory socket for address zero.

As shown in figure 1-1, the Monitor resides in the PROM area from address 0000H through 07FFH. Addresses 0800H through 2FFFH are ROM location available to the user. The Monitor working space occupies addresses 3C00H through 3CC3H; all other RAM locations are available to the user.

The standard configuration does not contain memory from location 4000H, through FFFFH. Any references to non-existent memory will cause unpredictable results.

The Monitor uses a terminal connected to the serial I/O port. See the AMC MonoBoard Computer User's Manual for detailed information about CRT/TTY connections and optional jumper connections on the MonoBoard Computer boards.

When power is applied to the MonoBoard, the Monitor program begins execution at address zero and the prompt

```
AMC MONOBOARD MONITOR
>
```

is displayed on the terminal output device.

1-3. MONITOR COMMANDS

Command mode directives are specified with a single alphabetic character following the Monitor prompt (>). These directives can be followed by hexadecimal arguments as described below. These arguments are separated by one or more of the following characters:

```
blank ! " # $ % & ' ( ) * + , - . /
```

Arguments enclosed in ( ) are optional and need not be included.

When a command is not recognized by the Monitor or the command arguments do not conform to the required format, the Monitor responds with a question mark (?). (See table 1-1 for Monitor Command Summary.)

The RUBOUT key on a TTY or the DEL key on a CRT terminal can be used to erase the current character in a line.

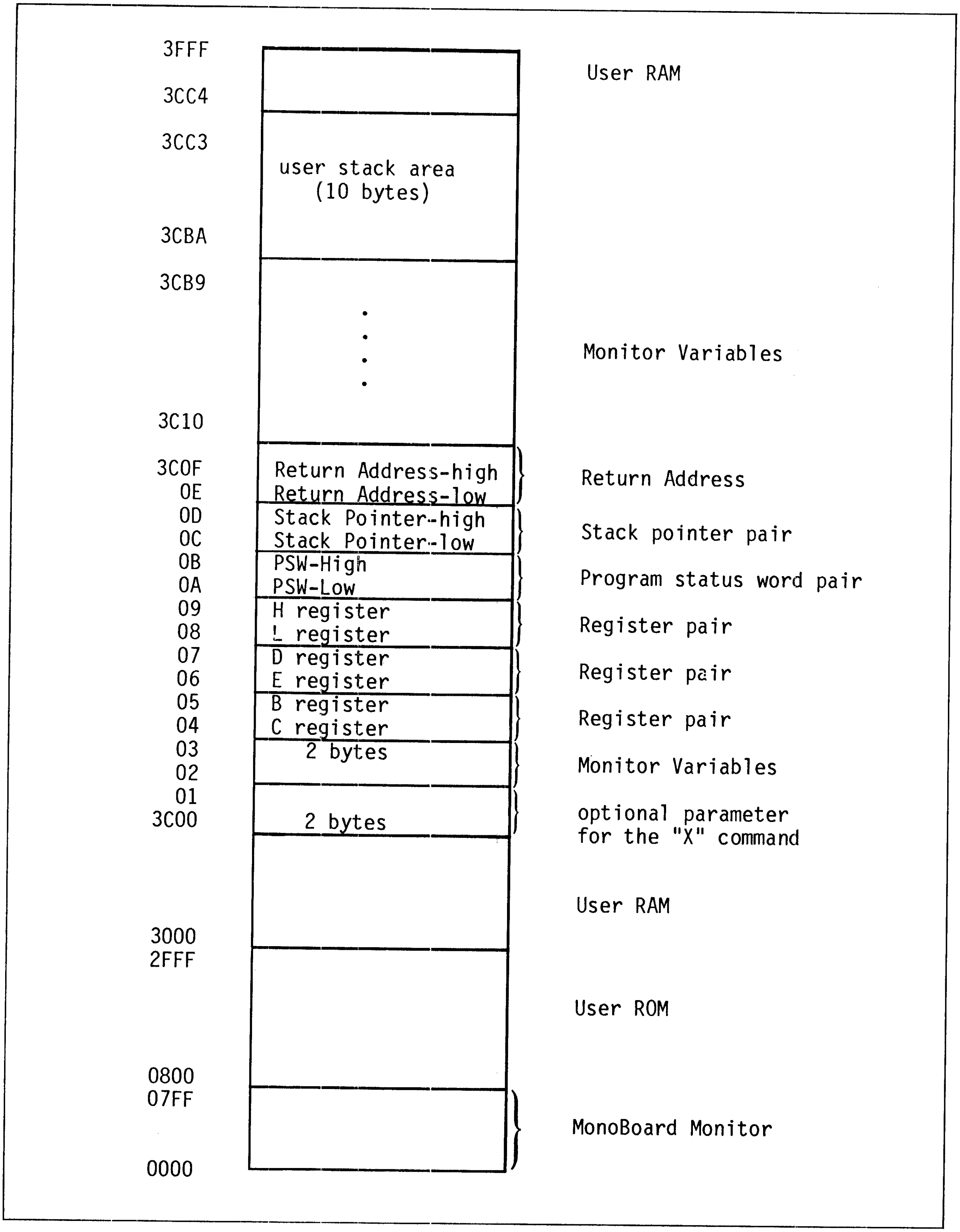

Figure 1-1. Monitor Memory Map.

TABLE 1-1. MONITOR COMMAND SUMMARY.

Command	Function
C X,Y	COMPARE memory strings at X and Y
D X(,Y)	DISPLAY memory from X through Y
F X,Y,Z	FILL memory from X through Y with Z
G (X) (B=BBCC) (D=DDEE) (H=HHLL) (P=AAFF) (S=SSSS) (;Y)	"GO", substituting register values as specified, with breakpoint at Y (if specified)
M X,Y,Z	MOVE memory string from X through Y to location Z and following
R (X)	READ hex tape with optional X bias load address
S X,Y1,Y2,...YN	SUBSTITUTE into memory starting at X, the Y values as specified. Eacy Y value specifies one byte.
T (same parameters as G above)	TRACE and dump registers at breakpoint
W X,Y	WRITE hex tape from X through Y
X X(,Y)	EXECUTE program at location X and store the value Y at top of Monitor RAM

Parameters enclosed in parentheses are optional. Comma means any acceptable separator.

Note: Semicolon shown in the G and T commands must be entered with the parameter.

The CTRL and the Q keys are used together (Control-Q) at any time to cancel a keyboard entry or terminate any operation.

In the following command format descriptions and examples, commas and blanks are used as argument separators; however, any of the previously listed separators may be used as desired.

Compare Memory C,XXXX,YYYY

This command compares the two memory ranges starting at locations XXXX and YYYY. It returns the first locations for which a difference is detected, followed by the contents of each. The compare terminates if either range reaches location FFFF without a mismatch.

Examples:

```
>C 3000, 3CC4
3200 23 3EC4 27
```

Display Memory D,XXXX,YYYY

This command displays the contents of memory from locations XXXX through YYYY. When YYYY is not specified, the contents of memory at location XXXX are displayed. If XXXX is greater than YYYY, the display will wrap-around to location 0 following the display of location FFFF.

Examples:

```
>D 0818,0824
6618 34 8A 68 C9 00 21 55 DC 4B 7D
 32 57 DC

>D 100
0100 48
```

Fill Memory F,XXXX,YYYY,ZZ

This command fills the contents of memory locations XXXX through YYYY with the value ZZ. Both XXXX and YYYY parameters must be supplied. If XXXX is greater than YYYY, the fill will wrap-around to location 0 after filling location FFFF.

Examples:

```
>F 3000, 3100, FF
```

(The value FF is written into locations 3000 through 3100.)

Go to User Program, and Substitute Values for Program Registers G (XXXX) (B=BBCC) (D=DDEE) (H=HHLL) (P=AAFF) (S=SSSS) (;YYYY)

This command starts or resumes execution of user code. If an address XXXX is specified, execution starts there. Otherwise execution resumes at the address following the last breakpoint. A breakpoint address can be selected using the G command by entering a semicolon followed by an address YYYY which specifies the breakpoint address. The breakpoint address cannot be set to a ROM location. Also, breakpoint is good for only one break. Registers are assigned values as pairs. The substitution must be specified with a pair name as follows:

B	B,C registers;
D	D,E registers;
H	H,L registers;
P	A register and flags;
S	Stack Pointer.

The pair name must be immediately followed by the = replacement operator.

When no substitution is specified for a pair, these registers retain their last contents. User code execution is suspended by an escape during I/O. The Monitor will also breakpoint a program when an RST 3 or RST 5 instruction is encountered during execution.

Examples:

>G

(resumes execution of user code from point last suspended, with old register values)

>G 1000

(starts execution of user code at 1000 with old register values)

>G S=3CD0, D=20

(resumes execution of user code from point last suspended, but with stack at 3CD0 and reg D=0 and reg E=20)

>G 2400 H=4200

(starts execution at 2400 with register H=42 and register L=00)

Move Contents of Memory M,XXXX,YYYY,ZZZZ

This command copies the contents of memory at locations XXXX through YYYY into the memory, starting at location ZZZZ. The source range must be specified with XXXX less than or equal to YYYY.

Examples:

>M 100,120,655C
>M,423C,423F,4000

Read Hexadecimal Paper Tape R (XXXX)

This command reads a hexadecimal formatted paper tape (prepared using the W command) into memory. The tape is read from the TTY tape reader. It is loaded into memory at locations specified in the records on the tape. When a bias, XXXX, is specified, the destination location is computed from the (2's complement) sum of the bias and the record addresses. Each tape ends with an end-of-file (null) record.

Examples:

>R

(reads in tape to the memory locations specified by the tape records)

>R 100

(reads in paper tape into memory at locations 100 greater than those specified by the tape records)

>R FF00

(reads in paper tape into memory at locations 100 less than those specified by the tape records)

Substitute Into Memory S,XXXX,Y1,Y2, Y3,....,YN

This command substitutes specified byte values Y1 through YN into memory starting at location XXXX.

Example:

S 3F00 1,2,4,8,10,20,40,80

(replaces the contents of memory at locations 3F00 through 3F07 with the byte value powers of 2)

Go To User Program and Substitute in Trace Mode T (XXXX) (B=BBCC) (D=DDEE) (H=HHLL) (P=AAFF) (S=SSSS) (;YYYY)

This command functions identically to the G command, except that when an RST 5 instruction is executed, the Monitor displays the register contents and continues executing user code at the next instruction. Note that the semicolon is required when the breakpoint option (;YYYY) is selected.

Example:

>T 3CD0,B=0

(starts execution at 3CD0 with all registers set at their prior values, except for registers B and C which are both 0)

Write Hexadecimal Paper Tape
W,XXXX,YYYY

This command outputs to the TTY the contents of memory from locations XXXX through YYYY. The output includes tape leader preceding the first hex record, an end-of-file as the final record, and tape leader following the end-of-file.

The formatted tape consists of records of ASCII coded hexadecimal bytes. When the tape is created, each byte is written from memory as two ASCII characters. The first character is the display code for the high order hex digit, and the second character is the display code for the low order hex digit. For example, the byte 45_{16} would be punched out 34_{16}, 35_{16}.

Each tape record consists of:

1) a colon (:),
2) two ASCII coded hex digits specifying the number (up to 24) of bytes punched from memory,
3) four ASCII coded hex digits specifying the first memory location written to the record,
4) two ASCII coded zeroes,
5) the ASCII coded hex for data punched out,
6) the (2's) complement of the checksum of 2),3),4),5).

The last record of the tape is an end-of-file. The end-of-file may be any record with an ASCII coded zero in the record length field. Examples:

```
>W 3001,3005
:054001002386A10504A7
:00
```

Execute External Utility
X,XXXX,(YYYY)

This command initiates special programs external to the Monitor. This allows user-designated code to execute as an extension of the Monitor, and preserves the state of the user code outside the Monitor memory area. This enables the programmer to run his own diagnostic, debug, or other special routines as he would a standard Monitor function.

The Monitor interface to the external routine provides a return address on the stack, and then transfer control to location XXXX. If the optional (YYYY) argument is specified, it is decoded as a hex value and stored at locations 3C00 and 3C01.

Example:

>X 800

1-4. EXECUTION-MODE INTERFACES

The Monitor initiates user-mode execution of code in response to either the G or T directive (as described above). Monitor extended execution may also be initiated with an X directive (as described above).

To re-enter the Monitor, user code must transfer control to one of the Monitor entry points. This usually is done by executing one of the RST instructions shown in table 1-2.

TABLE 1-2. RST INSTRUCTIONS.

Entry Point (Hex Address)	Instruction	OP Codes (Hex)	Condition	Response
0	RST 0	C7	System reset	Monitor re-initialized.
8	RST 1	CF	TTY input	Character is input to register A.
10	RST 2	D7	TTY output	Character is output from register A.
18	*RST 3	DF	Temporary task breakpoint; only executed once	Execution of user code is suspended, register contents and code location are displayed, Monitor enter command mode.
20	RST 4	E7	Reserved	Returns with no action.
28	*RST 5	EF	Permanent task breakpoint or tracepoint	Same as for RST 3.
30	RST 6	F7	Service routine address Specification	Monitor recognizes the word following the RST 6 instruction as the starting address of an interrupt service routine. The service routine address is stored for use by an RST 7 interrupt and the program continues executing at the byte following the address word.
38	**RST 7	FF	Interrupt	User registers are saved in the register save area shown in figure 1-1 and the service routine whose address was supplied by the last RST 6 instruction is executed. When the service routine completes, control is returned to the Monitor where the registers are restored. Control is then returned to the user's code.

*These instructions are used to force traces and breakpoints in user programs. The Monitor replaces the instruction at the specified location with the breakpoint RST instruction. When program execution is resumed, the original instruction is executed first.

**Saved registers are not stacked; thus only a single level of interrupt may be handled by RST 7. The location of the interrupt service routine must be specified prior to using RST 7; use RST 6 to accomplish this. User registers are saved in locations 3C04H-3C0FH in the working space.

1-5. TTY I/O INTERFACE

During execution, user code can freely use the Monitor character I/O routines.

When RST 1 is executed, the Monitor TTY input routine is initiated. The Monitor routine waits for the first character input from the TTY and returns the 7-bit ASCII code in register A. Control-Q (11H) is intercepted by the routine and causes automatic suspension of any active user-code and returns to Monitor command mode.

When RST 2 is executed, the Monitor TTY output routine is initiated. The Monitor routine transmits the contents of register A to the TTY. If an escape character is input at this time, it will be detected by this routine, causing automatic suspension of the user task and return of control to the Monitor.

1-6. SAMPLE MONITOR OPERATION

Figure 1-2 is an example of how Monitor commands might be used. Characters following the > prompt are user entered Monitor commands. The results of the commands are shown on the line or lines following the prompt line.

```
>D3D08
3D08  F0

>D 3D08 3D12
3D08  F0 8E F0 8F F0 0F F0 0F F0 0F F0

>S,3D08,1,2,3,4,FF

>M3D08,3D0B,3D0E

>D3D08,3D12
3D08  01 02 03 04 FF 0F 01 02 03 04 F0

>C 3D08 3D0E
3D0C FF   3D12 F0

>F3D08,3D10,0

>D3D08.3D12
3D08  00 00 00 00 00 00 00 00 00 04 F0

>W 3D08,3D42
:183D080000000000000000000004F02FF00FF08FF0DFF0AFF00FF48F12
:183D2000F00FE00FE00FE00FF00FF00FB00FF00F700F700FE00FF00F17
:0B3D3800F00FF00FD00FF00FF00DF0B7
:00

>R200

>D.3F08.3F12
3F08  00 00 00 00 00 00 00 00 00 04 F0
```

Figure 1-2. Monitor Example.

COMMENT SHEET

TITLE: MONOBOARD MONITOR
PUBLICATION NO: 00680114B

COMMENTS: (Describe errors, suggested additions or deletions, and include page numbers, etc.)

From: Name: ______________________ Position: ______________________

Company: ______________________

Address: ______________________

ADVANCED MICRO COMPUTERS
3340 Scott Boulevard
Santa Clara, California 95051
Distributed by
Advanced Micro Devices