MOTOROLA

M68KVSOM/A1

APRIL 1984

ADDENDUM

TO

VME/10 MICROCOMPUTER SYSTEM

OVERVIEW MANUAL

M68KVSOM/D1

This addendum transmits new and replacement pages for your VME/10 Microcomputer System Overview Manual.

Insert into your M68KVSOM/D1 manual the new and changed pages attached to this addendum. Make certain that the pages you are replacing are removed from your manual. This page of the addendum should be placed after the title page and used as a record page of the changes made to the manual. Pages affected by this addendum are:

i/ii	1-21/1-22	3-9/3-10	8-1/8-2
iii/iv	2-1/2-2	4-1/4-2	9-1/9-2
1-1/1-2	2-3/2-4	4-13/4-14	9-3/9-4
1-3/1-4	2-5/2-6	4-15/4-16	9-9/9-10
1-5/1-6	2-7/2-8	4-17/4-18	
1-17/1-18	3-1/3-2	4-19/4-20	
1-19/1-20	3-5/3-6	5-5/5-6	

MICROSYSTEMS

QUALITY • PEOPLE • PERFORMANCE

16849 PRINTED IN USA (5/84) MPS 3500

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, control and indicator descriptions, initialization procedures, and software-related information for the VME/10 Microcomputer System (hereafter referred to as VME/10). Before unpacking the system, powering it up, and performing any software operations described in this manual, refer to the VME/10 Microcomputer System Installation Guide, M68KVSIG.

The VME/10 provides the single user with 8- and 16-bit hardware/software and instrumentation development support and, eventually, 32-bit support. It incorporates the extended performance of the MC68010 MPU, the MC68451 Memory Management Unit (MMU), VMEbus compatibility, and full VERSAdos multitasking real-time operating system support, including high-level languages such as Pascal. Optionally available is SYSTEM V/68, derived from the UNIX SYSTEM V operating system. SYSTEM V/68 is not fully described in this manual. Refer to the SYSTEM V/68 documentation listed in paragraph 1.11 for further information.

The VME/10 is comprised of a chassis, a keyboard, and a display unit. Refer to Figure 1-1.

1.2 FEATURES

The features of VME/10 are as follows:

- MC68010 16/32-bit Microprocessor Unit (MPU).
- MC68451 Memory Management Unit (MMU).
- Industry-standard VMEbus interface with full bus arbitration logic and software controllable interrupter.
- I/O Channel interface for adding off-board resources such as A/D converters, serial and parallel I/O ports, etc.
- 384K bytes RAM triple-ported between graphics controller, local bus, and VMEbus (note that an additional 256K bytes of RAM is required for SYSTEM V/68 operation).
- Static RAM for storage of user-definable character sets and display attributes.
- Two 28-pin sockets for ROM/PROM/EPROM storage of up to 64K bytes for custom applications.
- Battery backed-up time-of-day clock with 50 bytes of CMOS RAM storage.
- 15-inch video display having the following software-controllable display formats:
    a. 25 lines by 80 characters -- 8 x 10 characters with descenders (10 x 12 font).
    b. 800 x 300 pixel for normal resolution graphics.

FIGURE 1-1. VME/10 Microcomputer System

c. 800 x 600 pixel for high resolution graphics.

d. Pixel graphics with overlaid character displays.

- Monochrome video display standard, with 7-level gray scaling (color optional).

- Detachable full ASCII character set keyboard with cursor control keys, hexadecimal pad, and 16 function keys.

- Mass storage subsystem providing both 5-1/4" floppy disk and 5-1/4" Winchester disk storage units standard.

  Floppy disk

  1M-byte unformatted capacity (655K-byte formatted)

  Winchester disk

  Choice of: (a) 6.38M-byte unformatted capacity (5M-byte formatted)

  (b) 19.1M-byte unformatted capacity (15M-byte formatted)

  (c) 51.9M-byte unformatted capacity (40M-byte formatted)

- Card cage options for feature expansion capability.

  Choice of: (a) Five I/O Channel card cage slots (with 6.38M-byte Winchester option)

  (b) Five VMEbus card cage slots with VMEbus backplane, plus four I/O Channel slots (with 19.1M-byte and 51.9M-byte Winchester options)

- Conformance to ergonomic standards applicable to video display and keyboard.

- TENbug firmware-resident debug/monitor package.

- Firmware-resident power-up/reset and disk-resident module diagnostic self-test.

- VERSAdos real-time multitasking operating system with M68000 Macro Assembler, plus tools and utilities. (SYSTEM V/68 software optionally available.)

- Capability of hosting hardware development tools.

  - HDS-400 for M68000 family 16/32-bit emulation

  - HDS-200 for M6800 family 8-bit emulation

  - Bus state analyzer for logic analysis functions

1.3 SPECIFICATIONS

Table 1-1 lists the specifications for the VME/10.

TABLE 1-1. VME/10 Microcomputer System Specifications

CHARACTERISTIC	SPECIFICATION	
Microprocessor	MC68010	
MPU clock frequency	10 MHz	
Word size		
Data	1-, 8-, 16-bit	
Address	24-bit	
Memory address capability	16M bytes (8 bits/byte)	
Bus standard	VMEbus	
Clock frequency	16 MHz	
Bus cycle time	200 ns (min.)	
Interrupt control	7-level priority	
Bus arbitration	4-level daisy-chained	
Data	16-bit	
Address	24-bit	
Control	Asynchronous	
Temperature		
Operating	10° to 40° C	
Storage	-40° to 60° C	
Relative humidity	20% to 80% (non-condensing)	
Physical dimensions	Chassis & monitor	Keyboard
Length	22.8 in. (57.9 cm)	8.3 in. (21.1 cm)
Width	19.0 in. (48.3 cm)	19.0 in. (48.3 cm)
Height	20.0 in. (50.8 cm)	2.0 in. (5.1 cm)
Weight	50 lbs. (23 kg)	5 lbs. (2.3 kg)
Power requirements (switching power supply)	90-132 Vac, 47-63 Hz, 500 W	
	180-264 Vac, 47-63 Hz, 500 W	

1.4 EQUIPMENT SUPPLIED

Table 1-2 lists the part number and description for the standard system configurations.

TABLE 1-2. Standard System Configuration

PART NUMBER	DESCRIPTION
M68K101-1	VME/10 Microcomputer System, including the MC68010 Microprocessor Unit, MC68451 Memory Management Unit, 384K bytes dynamic RAM, 655K-byte (formatted) 5-1/4" floppy disk unit, 5M-byte (formatted) 5-1/4" Winchester disk unit, 15-inch monochrome video display, and full ASCII keyboard with cursor control keys, hexadecimal keypad, and 16 user functional keys. For 115 Vac, 60 Hz operation. Expansion card cage incorporates five slots for single wide I/Omodule cards, plus ribbon cable and connectors to provide the I/O Channel interface functions to each card slot. System software includes the VERSAdos operating system, plus M68000 Family Structured Macro Assembler, Symbolic Debugger, CRT Editor, and Linkage Editor. A comprehensive diagnostics package is also included. VERSAdos is resident on the Winchester hard disk. System firmware incorporates (a) power-up self-test function,(b) disk bootstrap loader, and (c) TENbug Debug/Monitor package.
M68K102B1	Same as M68K101-1, except as follows: . Expansion card cage provides five slots for double format VMEmodule cards, plus 5-position VMEbus backplane at the rear of the card cage. Also includes four slots for single wide I/Omodule cards, with necessary cabling and connectors to provide the I/O Channel interface to each card slot. . 15M-byte (formatted) 5-1/4" Winchester disk unit replacing 5M-byte unit in M68K101-1).
M68K101-2	Same as M68K101-1, but for 230 Vac, 50 Hz operation.
M68K102B2	Same as M68K102B1, but for 230 Vac, 50 Hz operation.
M68K102C1	Same as M68K102B1, except with 40M-byte (formatted) Winchester disk drive.
M68K102C2	Same as M68K102B2, except with 40M-byte (formatted) Winchester disk drive.

1.5 I/O CHANNEL AND VME EQUIPMENT OPTIONS

Table 1-3 lists the part number and description for the optional equipment.

TABLE 1-3. Optional Equipment

PART NUMBER	DESCRIPTION
Modular expansion options - VMEbus	
MVME200	64K dynamic RAM with byte parity
MVME201	256K dynamic RAM with byte parity
MVME210	Static ROM/RAM module
MVME300	High performance IEEE-488 GPIB Controller with DMA
Modular I/O expansion options - I/O Channel	
MVME400	Dual RS-232C serial port
MVME410	Dual 16-bit parallel port (see NOTE)
MVME420	SASI adapter
MVME435	Buffered 9-track magnetic tape adapter
MVME600	12-bit analog input module
MVME601	16-channel expander for MVME600
MVME605	12-bit analog output module
MVME610	Opto-isolated 120V/240V ac input
MVME615/616	Opto-isolated 120V/240V ac output
MVME620	Opto-isolated 60 Vdc input
MVME625	Opto-isolated 60 Vdc output
MVME935	Remote I/O Channel extender cable connection module
Remote I/O Channel modules	
M68RAD1	Remote intelligent analog conversion module
M68RIO1	Remote I/O solid state relay module

NOTE: This module recommended for parallel printer interface port applications with the VME/10 system.

1.8.3 Symbolic Debugger

The symbolic debugger (SYMbug) program is used to debug other programs whose source code is written in Motorola's Structured Macro Assembler for execution on the M68000 Family MPU's. The language processors, in cooperation with the linkage editor, supply information to SYMbug. This permits the user to describe the debugging requirements to SYMbug in terms close to the language in which the source program was written. SYMbug allows the user to debug in symbolic terminology. SYMbug allows the user to perform the following:

- Examine, insert, and modify program elements such as instructions, numeric values, and coded information (i.e., data in all its representations and formats).
- Control execution, including the insertion of breakpoints into a program and request for breaks or changes in elements of data.
- Trace execution by displaying information at designated points in a program.
- Search programs and data for specific elements and sub-elements.
- Create macro commands allowing user-defined formats and commands.

1.8.4 CRT Text Editor

The CRT-oriented Text Editor provides the capability to create and modify source programs. The editor supports both command and page editing, utilizing the cursor control keys, control characters, and function keys of the VME/10 keyboard chassis to insert, alter, or delete characters and lines within a user text file.

1.8.5 Resident Structured Assembler

The M68000 family resident structured macro assembler translates source statements into relocatable machine code, assigns storage locations to instructions and data, performs auxiliary assembler actions designated by the programmer, and optionally produces a cross reference listing. The assembler includes macro and conditional assembly capabilities, plus certain "structured programming" control structures.

1.8.6 Linkage Editor

The Linkage Editor provides the capability of converting one or more separately compiled object units into a loadable object module file. The editor determines segment attributes, calculates address space, searches libraries, resolves external references, relocates object code, and issues error messages. At the end of a linkage process, the editor prints a report that contains a module map, a table of externally defined symbols, and any unresolved or multiply defined symbols.

1.8.7 Diagnostic Package

The diagnostic package verifies the overall functionality of the VME/10 by exposing it to a set of off-line tests. The package provides two levels of diagnostics. The first level is the firmware-resident, power-up/reset test. The second level is comprised of disk-resident diagnostics for more extensive hardware testing. The governing guideline for diagnostics is to provide a comprehensive test package that will isolate a malfunction to a functional block and at least down to the faulty module. For detailed information, refer to the VME/10 Microcomputer System Diagnostics Manual (M68KVSDM).

1.8.8 FORTRAN Compiler (Optional)

FORTRAN is a science/math-oriented high-level programming language. Motorola's FORTRAN compiler for the M68000 family meets and exceeds the ANSI FORTRAN 77 subset specification, providing real-time processing capabilities. The compiler generates relocatable object code which is ROMable and floatable and permits linkage to assembly language routines.

1.8.9 Pascal Compiler (Optional)

Pascal is a high-level, user-oriented language for the M68000 family, based on the language as defined by Niklaus Wirth. Pascal is a highly constructed language which promotes good programming techniques, is self-documenting, and has user-oriented statement forms which simplify program writing. Extensions provided by Motorola include the following: address specification for variables, alphanumeric labels, string types, exit, non-decimal integers, runtime error checking, routine file management, and separate compilation and linking.

1.9 SYSTEM V/68 OPERATING SYSTEM (Optional)

SYSTEM V/68 is the standard UNIX system-derived operating system for the M68000 family of microprocessors. It offers a small, compact kernel which provides process scheduling and input/output facilities to all programs. It also offers a powerful command shell for interactive system controls and an extensive set of utility programs for many tasks such as program development, text processing, and electronic mail.

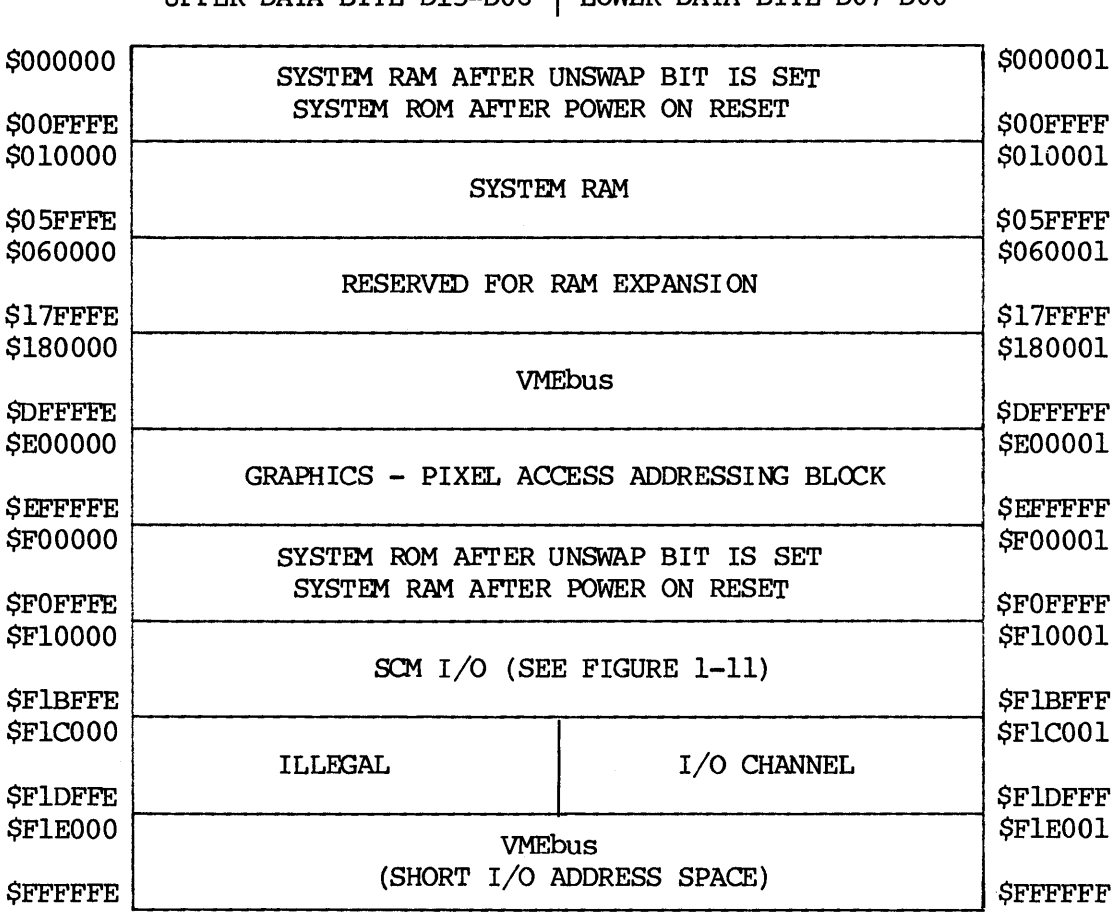
C Language is provided as a part of SYSTEM V/68. The Motorola version of C offers significant enhancements, including new utility programs such as CXREF (cross reference) and CFLOW (flow analysis). A FORTRAN compiler is also supplied.

Future additions to the system will be BASIC and Pascal compilers, Motorola assembler and linker, various device drivers, and networking support.

SYSTEM V/68 is not fully described in this manual. Refer to the SYSTEM V/68 documentation listed in paragraph 1.11 for further information.

1.10 SYSTEM MEMORY MAP

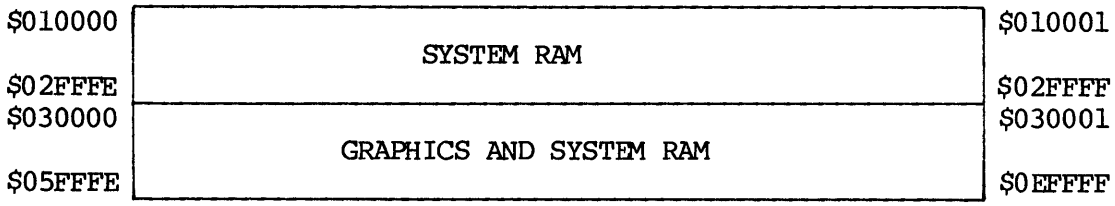
Figures 1-9 through 1-11 depicts the 16-megabyte system memory map.



NOTE: The RAM (384K bytes) on the System Control Module (SCM) can be accessed by off-board devices via the VMEbus. The base address of the dual-ported RAM on the VMEbus is $D00000-D5FFFF.

FIGURE 1-9. VME/10 System Memory Map

HIGH RESOLUTION GRAPHICS



NORMAL RESOLUTION GRAPHICS

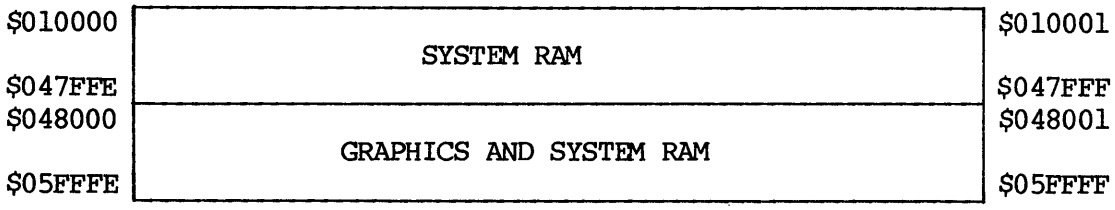


FIGURE 1-10. High and Normal Resolution Graphics

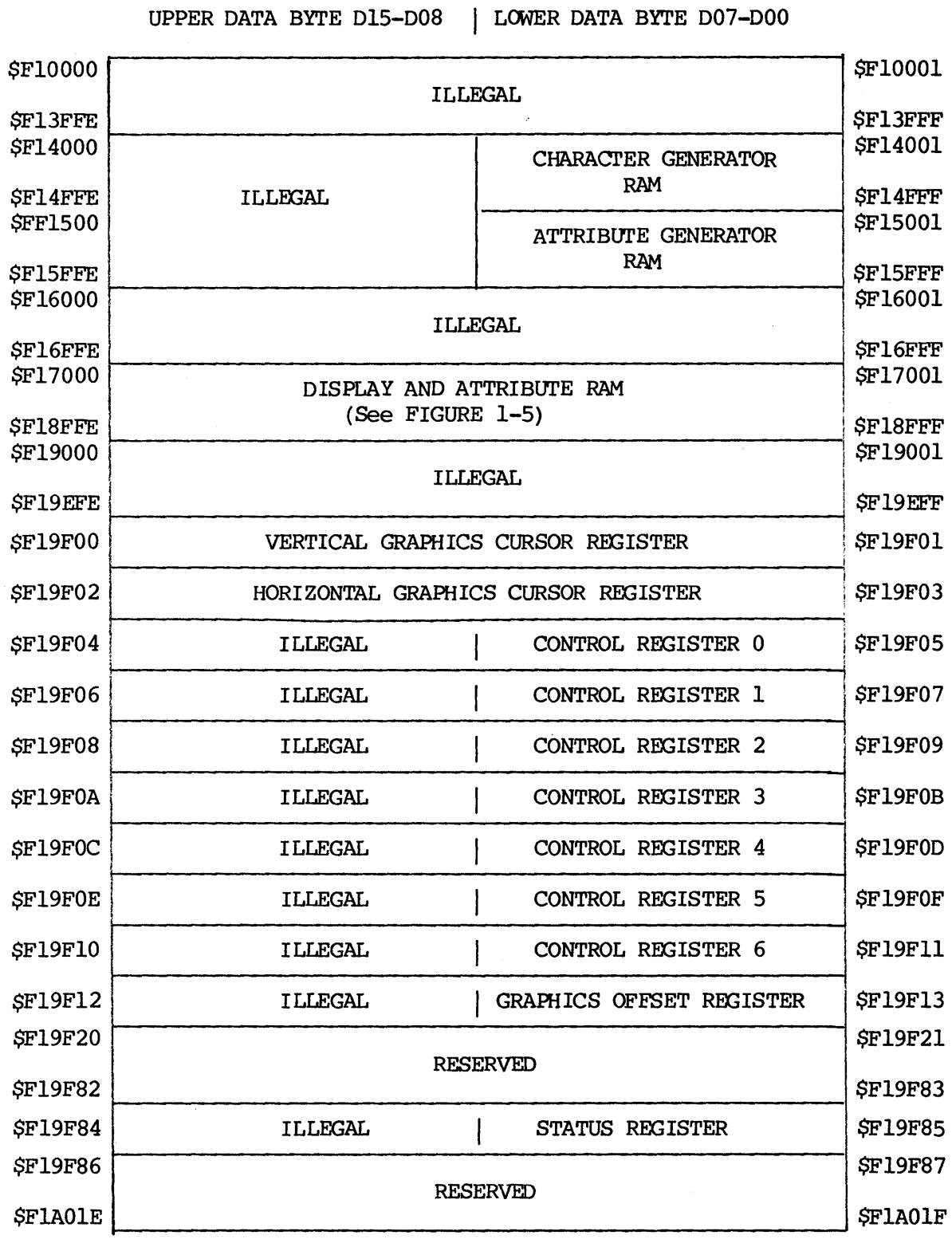


FIGURE 1-11. SCM I/O Memory Map (Sheet 1 of 2)

	UPPER DATA BYTE D15-D08	LOWER DATA BYTE D07-D00	
$F1A020	ILLEGAL	MC68A45 ADDRESS REGISTER	$F1A021
$F1A022		MC68A45 INTERNAL REGISTER FILE	$F1A023
$F1A024	ILLEGAL		$F1A025
$F1A02E			$F1A02F
$F1A030	ILLEGAL	MC2661 TX/RX DATA REGISTERS	$F1A031
$F1A032		MC2661 STATUS REGISTER	$F1A033
$F1A034		MC2661 MODE 1 AND MODE 2 REG.	$F1A035
$F1A036		MC2661 COMMAND REGISTER	$F1A037
$F1A038	ILLEGAL		$F1A039
$F1A07E			$F1A07F
$F1A080		MC146818 SECONDS REGISTER	$F1A081
$F1A082		MC146818 SECONDS ALARM REG.	$F1A083
$F1A084		MC146818 MINUTES REGISTER	$F1A085
$F1A086		MC146818 MINUTES ALARM REG.	$F1A087
$F1A088		MC146818 HOURS REGISTER	$F1A089
$F1A08A		MC146818 HOURS ALARM REGISTER	$F1A08B
$F1A08C		MC146818 DAY OF THE WEEK REG.	$F1A08D
$F1A08E		MC146818 DAY OF THE MONTH REG.	$F1A08F
$F1A090		MC146818 MONTH REGISTER	$F1A091
$F1A092		MC146818 YEAR REGISTER	$F1A093
$F1A094		MC146818 REGISTER A	$F1A095
$F1A096		MC146818 REGISTER B	$F1A097
$F1A098		MC146818 REGISTER C	$F1A099
$F1A09A		MC146818 REGISTER D	$F1A09B
$F1A09C	ILLEGAL	BATTERY BACKED UP RAM	$F1A09D
$F1A0FE		TIME-OF-DAY CLOCK (MC146818)	$F1A0FF
$F1A100	ILLEGAL		$F1A101
$F1A7FE			$F1A7FF
$F1A800	DMA/MMU		$F1A801
$F1AFFE			$F1AFFF
$F1B000	ILLEGAL		$F1B001
$F1BFFE			$F1BFFF
$F1C000	ILLEGAL	I/O CHANNEL (SEE NOTE)	$F1C001
$F1DFFE			$F1DFFF

NOTE: The standard SYSGEN program dictates that if a printer is included, the MVME410 module should be addressed at $F1C1E1-$F1C1FF. Similarly, if there are serial ports, the MVME400 module should be addressed at $F1C1C1-$F1C1DF. These are recommended addresses reserved for the modules.

FIGURE 1-11. SCM I/O Memory Map (Sheet 2 of 2)

1.11 RELATED DOCUMENTATION

The following manuals are applicable to the VME/10 Microcomputer System.

Hardware

M68000 16/32-Bit Microprocessor Programmer's Reference Manual, M68000UM
VME/10 Microcomputer System Installation Guide, M68KVSIG
VMEbus Specification Manual, MVMEBS
256K/512K Byte Dynamic RAM Memory Module, M68KVM11

VERSAdos

VERSAdos Customer Letter, M68KSYSLET
M68000 Family CRT Text Editor, M68KEDIT
M68000 Family Real-Time Multitasking Software User's Manual, M68KRMS68K
M68000 Family Resident Structured Assembler Reference Manual, M68KMASM
M68000 Family Resident Pascal User's Manual, M68KPASC
M68000 Family Linkage Editor User's Manual, M68KLINK
M68000 Family Resident FORTRAN Compiler User's Manual, M68KFORTRN
VERSAdos Data Management Services and Program Loader User's Manual, RMS68KIO
VME/10 Microcomputer System Diagnostics Manual, M68KVSDM
VME/10 Microcomputer System Reference Manual, M68KVSREF
VME/10 Microcomputer System Command and Graphics Primitives Reference Manual, M68KVSG
TENbug Debugging Package User's Manual, M68KTENBG
SYMbug/DEbug Monitor Reference Manual, M68KSYMBG
VERSAdos System Facilities Reference Manual, M68KVSF
VERSAdos Messages Reference Manual, M68KVMSG
System Generation Facility User's Manual, M68KSYSGEN
VERSAdos Overview, M68KVOVER
Pascal Programming Structures for Motorola Microprocessors, George C. Cherry
Guide to Writing Device Drivers for VERSAdos, M68KDRVGD/D1

SYSTEM V/68

SYSTEM V/68 Customer Letter, M68KVSV68
SYSTEM V/68 Release Description, M68KUNRD
SYSTEM V/68 Transition Aids, M68KUNTA
SYSTEM V/68 Administrator's Manual, M68KUNAM
SYSTEM V/68 Administrator's Guide, M68KUNAG
SYSTEM V/68 Error Message Manual, M68KUNMSG
SYSTEM V/68 User's Guide, M68KUNUG
SYSTEM V/68 Document Processing Guide, M68KUNDP
SYSTEM V/68 Graphics Guide, M68KUNGG
SYSTEM V/68 Programming Guide, M68KUNPG
SYSTEM V/68 Support Tools Guide, M68KSTG
SYSTEM V/68 User's Manual, M68KUNUM
SYSTEM V/68 Operator's Guide, M68KUNOG

Additionally available is a two-volume set of hardware documentation, under the part number 68-M68KVSHD, and various device driver manuals.

CHAPTER 2

HARDWARE/SOFTWARE SYSTEM STARTUP

2.1 INTRODUCTION

This chapter provides system initialization, media backup procedures, and system performance verification for the VME/10. Commands and other input/output (I/O) are presented in this manual in a modified Backus-Naur Form (BNF) syntax. Certain symbols in the syntax are not to be typed; their usage is restricted to the syntactic structure. These symbols and their meanings are as follows:

< > The angular brackets enclose a "syntactic variable", that is replaced in a command line by one of a class of items it represents.

| This symbol indicates that a choice is to be made. One of several items, separated by this symbol, should be selected.

[ ] Square brackets enclose an optional item. The enclosed item may occur zero or one time.

[ ]... Square brackets followed by periods enclose an item that is optional/repetitive. The item may appear zero or more times.

Operator entries are shown underscored for clarity (the underscore is not typed), and are to be followed by pressing the carriage return key (<--'). When a carriage return is the only required entry, it is shown as (CR).

2.2 SYSTEM POWER-UP

After the VME/10 has been correctly installed as directed in the VME/10 Microcomputer System Installation Guide, M68KVSIG, turn on the system by setting the 0/1 rocker switch on the chassis to the 1 position.

2.3 POWER-UP/RESET SELF-TEST

When the VME/10 is powered up, a firmware-resident power-up/reset (PWRT) self-test is performed to verify the functionality of the system resources necessary to boot and initiate the operating system. During the self-test, messages regarding the progress and results of the self-test are displayed. The PWRT takes about 5 seconds to execute, but note that an additional several seconds is required for the Winchester media to spin up to speed before the PWRT is performed. The total length of time required may be up to a minute.

Upon completion of the PWRT after power-up, control of the system is given either to TENbug or to VERSAdos (if resident), depending upon the position of the KYBD LOCK switch (key vertical = locked = VERSAdos, key horizontal = unlocked = TENbug, refer to NOTES in paragraph 2.4.2, Method 1). If SYSTEM V/68 is resident on the hard disk or is to be booted from diskette, bring the VME/10 up in TENbug and then refer to the SYSTEM V/68 documentation for booting, backing up, and operating the system.

The PWRT self-test may also be initiated by the operator with the RESET and ABORT pushbuttons (provided the KYBD LOCK key switch is in the horizontal, unlocked position) when used in the following sequence:

a. Press and hold the RESET pushbutton.
b. Press and release the ABORT pushbutton.
c. Release the RESET pushbutton.

The VME/10 then performs the powerup/reset self-test and gives control to TENbug.

2.4 SYSTEM INITIALIZATION

The VME/10 may be initialized in either of two modes:

a. TENbug (operating system media not required)
b. VERSAdos operating system, if resident on hard disk

NOTE

The VERSAdos operating system and supporting software is resident on the Winchester hard disk when the VME/10 is delivered. After the system has been initialized, backup copies of this software should be made and stored for safekeeping. This procedure is given in paragraph 2.6, Backup Procedure.

Prior to performing these procedures, refer to Chapter 3 for identification and function of the VME/10 switches and keys.

2.4.1 TENbug

TENbug may be entered by either of two methods. Method 1 is used when the system is initially powered up, and method 2 is used to return to TENbug from VERSAdos.

Method 1 (enter TENbug at system initial power-up)

a. Set the KYBD LOCK key switch to the unlocked (horizontal) position.

b. Set the power switch to the on position (1). When power is applied, the power-up reset (PWRT) self-test is initiated.

c. If PWRT self-test indicates no errors, and there is no MVME400 (Dual RS-232C Serial Port) present in the VME card cage, the following will be displayed on the monitor:

```
TENbug x.x >
```

d. If PWRT self-test indicates no errors, and there is an MVME400 present, the following is displayed on the monitor:

```
TENbug
```

e. Select the terminal to serve as the default console device, and press any key on its keyboard. The following will be displayed:

```
TENbug x.x >
```

Method 2 (enter TENbug from VERSAdos operating system)

a. When the VERSAdos operating system is running on the VME/10, TENbug is entered by pressing the RESET pushbutton (provided the KYBD LOCK key switch is in the unlocked (horizontal) position and the vectors in location 0-7 have not been destroyed).

b. Go to step c. of Method 1.

2.4.2 VERSAdos Operating System

The VERSAdos operating system may be entered by either of two methods. Method 1 is used when the system is initially powered up. Method 2 is used after the system has entered TENbug.

Method 1 (enter VERSAdos operating system at system initial power-up)

NOTE

When using the power-up boot process, described as follows, VERSAdos is booted from device 0, controller 0 and assumes that VERSADOS.SY has a null catalog. To boot VMES10.VERSADOS.SY, a different file, or boot from a different device/controller, the boot must be initiated from TENbug. Refer to TENbug's BO command described in Chapter 9 and in the TENbug manual.

a. On the chassis operator panel, set the KYBD LOCK key switch to the locked (vertical) position. (Note that the chassis operator panel pushbutton switches RESET and ABORT are inoperative.)

b. Set the power switch to the on position (1). When power is applied, the power-up/reset self-test (PWRT) is initiated. It may require up to one minute for the disk drive to attain running speed and to perform the self-test. The following messages are displayed on the monitor:

```
Power-up test in progress
Waiting for disk to spin up
```

c. After the self-test concludes, assuming no errors have been indicated, the following is displayed:

```
Power-up test complete
```

d. The VME/10 boots the VERSAdos operating system into memory from the Winchester disk media and VERSAdos identifies itself. Unlock the keyboard (KYBD LOCK switch in horizontal position) and press the "uppercase lock" key (⇪) on the keyboard. Make the responses indicated, using current date and time:

```
VERSADOS VERSION: n.nn mm/dd/yy          xxxxxxxxxx
ENTER DEFAULT SYSTEM VOLUME:USER NO.=SYS:0
ENTER DATE (MM/DD/YY)=3/31/83
ENTER TIME (HR:MIN)=7:00
7:00:01 3/31/83 START SESSION 0001 USER 0
```

Two informative chain files, which may be modified by the user, are executed and their messages displayed.

The operating system may have been generated with an automatic logon as volume SYS:, user 0, and session 0001. In this case, current time and date are displayed. They may be changed by the user, with the DATE and TIME commands.

e. To exit the VERSAdos operating system, enter one of the following on the keyboard:

```
=LOG OFF
 14:00:00 3/31/83 END SESSION 0001 USER 0
```

or

```
=OFF
 14:00:00 3/31/83 END SESSION 0001 USER 0
```

or

```
=BYE
```

f. If the power has not been turned off, VERSAdos can be reentered after logoff by pressing the CLEAR/BREAK key and responding to the system prompts.

Method 2 (enter VERSAdos operating system from TENbug)

a. When TENbug is running on the VME/10, ensure that the VERSAdos operating system media is available and that the KYBD LOCK switch is in horizontal position.

b. Enter the BO command from the keyboard to load VERSAdos from the fixed hard disk.

```
TENbug x.x > BO 0,0,VME510.VERSADOS.SY
```

Go to step d. of Method I.

NOTES

(1) After the VERSADOS.SY file has been renamed with a null catalog, as directed in the customer letter, M68KSYSLET, VERSAdos may be booted from TENbug with the simple BO command.

(2) To update an existing system when the VERSAdos release media is floppy diskette, refer to the customer letter, M68KSYSLET.

(3) To update a 5M-byte or 40M-byte system, refer to the customer letter, M68KSYSLET, for a necessary patch.

(4) To boot SYSTEM V/68 from floppy diskettes, refer to the customer letter, M68KVSV68.

2.5 DISK-RESIDENT MODULE DIAGNOSTICS

After the VME/10 has been powered up, extended tests can be performed on the system by executing the disk-resident module diagnostics (DRMD) package. It is recommended that the first-time user perform these extended tests to verify system performance. For detailed information, refer to the VME/10 Microcomputer System Diagnostics Manual, M68KVSDM.

2.6 BACKUP PROCEDURE

The software supplied on the fixed Winchester hard disk should not be exercised until the following procedure has been performed to create a complete backup copy. The backup version should be labeled and stored for safekeeping. Note that when backing up from the fixed disk to floppy diskettes, several diskettes will be required.

The following steps initialize and configure diskettes for VERSAdos and create backup diskettes from the fixed hard disk. For SYSTEM V/68, refer to the documentation listed in paragraph 1.11. User-entered responses are shown underlined and are to be followed by a carriage return.

a. After entering VERSAdos as shown in paragraph 2.4.1, insert a double-sided blank or scratch diskette in the floppy drive. (This procedure assumes that the diskettes have not been initialized; if using initialized diskettes, BACKUP can be used directly, and a Y answer can be given to BACKUP's "CONTINUE (Y/N)?" query in step e., after inserting next initialed diskette.)

b. Call the INIT program and make the entries shown below:

```
=INIT #FD02;V
OK TO INITIALIZE #FD02  (Y/N) ? Y

Configuration change-to exit enter C (configure) or Q (terminate utility).

Data Density of media (S-single,D-double) D > C
DO YOU WANT TO FORMAT DISK (Y/N) ?  Y
START FORMAT
ENTER NEW VOLUME NAME VME1
ENTER USER NUMBER 0
ENTER DESCRIPTION (MAX 20 CHARACTERS) VME/10 BACKUP
DO YOU WANT THE BOOT STRAP (Y/N) ? Y
FILE NAME IS: SYS:0000..IPL.SY
ENTER NEW NAME IF NEEDED (CR)
THE CURRENT LOAD ADDRESS IS $10000
ENTER NEW LOAD ADDRESS (CR)
DO YOU WANT A DUMP AREA (Y/N) ? N
DO YOU WANT TO VALIDATE SECTORS (Y/N)? Y
   VALIDATING SECTORS ....
   0 BAD SECTORS ENCOUNTERED
=
```

NOTE

Because this first diskette must be bootable, the V option must be entered on the INIT command line. When the bootstrap question is answered with a Y, the user will be allowed to enter a LOAD address. The LOAD address for the VME/10 is $10000. A RETURN (CR) verifies this address.)

c. Call the BACKUP program to copy files from the hard disk to the floppy. The filenames are listed as the files are copied:

```
=BACKUP #HD00,#FD02
STARTING FILE-BY-FILE BACKUP PROCESS
COPY ALL FILES, SELECT FILES, OR QUIT (A/S/Q) ? A
DUPLICATE FILE - OK TO COPY (Y/N/Q) ?   VME1:0000..IPL.SY N
VME1:0000..ARESTRRG.HT
 .
 .
 .
VME1:<user no.>.<catalog>.<file name>.<ext>
** OUTPUT DISK FULL **    CONTINUE (Y/N) ? N
=
```

d. Note the full user number, catalog, file name, and extension of the last file copied.

e. Remove the diskette, label it, and set it aside.

f. Insert another double-sided blank or scratch diskette into the floppy drive.

g. Initialize and format the floppy and continue backing up the hard disk as follows. (Note that BACKUP's S option must be used to back up these successive floppies.).

```
=INIT #FD02
OK TO INITIALIZE #FD02  (Y/N) ? Y

Configuration change-to exit enter C (configure)or Q (terminate utility).

Data Density of media (S-single,D-double) D > C
DO YOU WANT TO FORMAT DISK (Y/N) ?  Y
START FORMAT
ENTER NEW VOLUME NAME VME2
ENTER USER NUMBER 0
ENTER DESCRIPTION (MAX 20 CHARACTERS) VME/10 BACKUP
DO YOU WANT THE BOOT STRAP (Y/N) ? N
DO YOU WANT A DUMP AREA (Y/N) ? N
DO YOU WANT TO VALIDATE SECTORS (Y/N)? Y
   VALIDATING SECTORS ....
   0 BAD SECTORS ENCOUNTERED
=BACKUP #HD00,#FD02;S
STARTING FILE-BY-FILE BACKUP PROCESS
COPY ALL FILES, SELECT FILES, OR QUIT (A/S/Q) ? A
ENTER RESTART FILENAME (INCLUDING USER NUMBER)
 <user no.>.<catalog>.<filename>.<ext>          (Enter full file name of
 VME2:<user no.>.<catalog>.<filename>.<ext>      last file copied to
                                                 previous diskette, but
 .                                               do not enter the volume
 .                                               name. The files will be
 .                                               listed as they are
                                                 copied.)
```

i. Each time the ** OUTPUT DISK FULL ** message appears, enter N and repeat the procedure from step d. above until all files are copied from the hard disk (system will return to the VERSAdos "=" prompt without issuing the ** OUTPUT DISK FULL ** message). Use a different volume name for each diskette.

Note that in the preceding examples, the utilities MT and DMT are not required. However, for most routine operations with diskettes that have been initialized previously (have a volume name), the diskettes must be mounted and dismounted. For examples of using MT and DMT for routine input/output operations to floppy diskettes, refer to the VERSAdos System Facilities Manual, M68KVSF.

CHAPTER 3

CONTROLS AND INDICATORS

3.1 INTRODUCTION

This chapter provides control and indicator descriptions for the VME/10 chassis, display unit, and keyboard.

3.2 CHASSIS

The chassis has an operator panel (Figure 3-1) located at the bottom left corner on the front of the chassis.

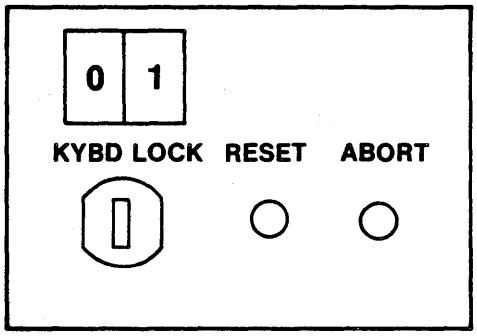


FIGURE 3-1. Operator Panel

The controls perform the following functions:

a. 0 | 1 - The power on/off rocker-arm switch is used to turn on power to the VME/10. The '0' represents the off position; the '1' represents the on position.

b. KYBD LOCK - The KYBD LOCK key switch controls a bit in a register which is monitored by TENbug. When the key switch is in the locked (vertical) position, the VME/10 enters the VERSAdos operating system, if resident. When the key switch is in the unlocked (horizontal) position, the VME/10 enters TENbug. Also, when the key switch is in the locked position, the keyboard keys and the front panel pushbutton switches RESET and ABORT are inoperative. This feature provides protection from inadvertent interrupts during system usage.

c. RESET - When this momentary-action pushbutton switch is pressed, it resets the VME/10 logic circuits. If the VME/10 is in the VERSAdos operating system, TENbug is entered by pressing RESET (provided the KYBD LOCK key switch is in the unlocked position).

d. ABORT - When this momentary-action pushbutton switch is pressed, the VME/10 enters TENbug, but the VME/10 logic circuits are not reset. After an abort, the user can enter 'G' to continue execution of the current program prior to the abort.

There are two indicators located at the front of the chassis. When either the Winchester or floppy disk drive is accessed, the respective indicator becomes illuminated.

3.3 DISPLAY UNIT

The display unit has a rear panel which contains a rotary adjustment control (=D) used for varying the screen intensity.

3.4 KEYBOARD CONSOLE

The keyboard console is partitioned into four basic functional groups:

a. Typewriter keyboard
b. Cursor control keypad
c. Hex/edit keypad
d. User function keys (F1-F16)

NOTE

The exact action taken when a key is depressed depends upon the port configuration and the program communicating to the keyboard display unit. Refer to the Data Management Services Manual (RMS68KIO) for detailed information for VERSAdos and to the documentation listed in paragraph 1.11 for SYSTEM V/68.

Refer to Figure 3-2 for keyboard assembly.

3.4.2 Typewriter Keyboard

The typewriter keyboard contains the following numerics, alphabetic, symbol and special character selection:

a. Numerics (0-9)
b. Alpha characters (A-Z)
c. Symbol characters
d. Special characters (delete, carriage return, and forward tab)

Table 3-2 lists the characters and codes generated when a key is depressed when the typewriter keyboard is in a specific or multi-mode of operation.

3.4.2.1 Numerics (0-9). Numerics are obtained by depressing the respective numeric key when in the normal mode. Refer to Table 3-2.

3.4.2.2 Alphabetic Characters (a-z). Lowercase alphabetic characters are obtained by depressing the respective alphabetic key when in the normal mode. To obtain uppercase characters, depress the CAPS LOCK key (remains down) or depress and hold the shift key, then depress the respective alpha key. Refer to Table 3-2.

3.4.2.3 Symbol Characters. Symbols are obtained by depressing the respective symbol key when in the normal or shift mode. Refer to Table 3-2.

3.4.2.4 Special Characters. The special character keys are defined as follows:

a. The DEL (delete) key erases the character before the cursor.

b. The <--| (carriage return) key moves the cursor to the beginning of the next line and signals the end of a line typed to the computer.

c. The -->| (forward tab) key moves the cursor to the next tab position in certain programs (e.g., the VERSAdos editor).

TABLE 3-2. Standard Typewriter Keyboard Character Code

KEY	DESCRIPTION	MODE			
		NORMAL	CAPS LOCK	SHIFT	CTRL
A	Alphabetic A	a	A	A	$01
B	Alphabetic B	b	B	B	$02
C	Alphabetic C	c	C	C	$03
D	Alphabetic D	d	D	D	$04
E	Alphabetic E	e	E	E	$05
F	Alphabetic F	f	F	F	$06
G	Alphabetic G	g	G	G	$07
H	Alphabetic H	h	H	H	$08
I	Alphabetic I	i	I	I	$09
J	Alphabetic J	j	J	J	$0A
K	Alphabetic K	k	K	K	$0B
L	Alphabetic L	l	L	L	$0C
M	Alphabetic M	m	M	M	$0D
N	Alphabetic N	n	N	N	$0E
O	Alphabetic O	o	O	O	$0F
P	Alphabetic P	p	P	P	$10
Q	Alphabetic Q	q	Q	Q	$11
R	Alphabetic R	r	R	R	$12
S	Alphabetic S	s	S	S	$13
T	Alphabetic T	t	T	T	$14
U	Alphabetic U	u	U	U	$15
V	Alphabetic V	v	V	V	$16
W	Alphabetic W	w	W	W	$17
X	Alphabetic X	x	X	X	$18
Y	Alphabetic Y	y	Y	Y	$19
Z	Alphabetic Z	z	Z	Z	$1A

3.4.3 Cursor Control Keypad

The cursor control keypad provides the following:

a. Cursor control
b. Functions: CLEAR/BREAK, RESET
c. Special character: ESC

3.4.3.1 Cursor Control. The cursor control keys are used in special programs (e.g., the editor). Table 3-3 indicates the cursor control keys and their respective character codes and functions.

TABLE 3-3. Cursor Control Keys

KEY	FUNCTION	CHARACTER CODE	FUNCTION PERFORMED WHEN DEPRESSED
<--	Cursor left	$08	Moves cursor left one column.
-->	Cursor right	$0C	Moves cursor right one column.
↑	Cursor up	$0B	Moves cursor up one line in same column.
↓	Cursor down	$0A	Moves cursor down one line in same column.
|<--	Backward tab	$DB	Moves cursor left to previous tab position.
-->|	Forward tab	$09	Moves cursor right to next tab position.
SEL			Not implemented.
CLR TAB SET			Not implemented.
	Home	$C0	Moves cursor to left-most column in top line.

3.4.3.2 Functions (CLEAR/BREAK, RESET). The function keys are described as follows:

a. The shifted value of the CLEAR/BREAK key (CLEAR) causes all unprotected positions in the display to be filled with spaces. The cursor moves to the home position.

b. The non-shifted value of the CLEAR/BREAK key (BREAK) generates a 'special break condition' signal which is recognized by the VERSAdos operating system and allows the user to log on.

c. The shifted value of the RESET/ESC key (RESET) initializes the screen to the power-on condition.

3.4.3.3 Special Character (ESC). The non-shifted value of the RESET/ESC key (ESC) generates an ASCII escape character ($1B).

3.4.4 Hex/Edit Keypad

The hex/edit auxiliary keypad performs two modes -- hexadecimal keyboard entry and editing functions -- which are controlled by the PAD/FUNC key (refer to paragraph 3.4.1). The ENTER key (carriage return) generates a character code $0D which moves the cursor to the beginning of the next line (left margin). The ENTER key is not affected by the PAD/FUNC key.

3.4.4.1 Hexadecimal Mode. When the PAD/FUNC key is depressed, the keypad can be used as a hexadecimal keypad utilizing characters 0-F and also a comma (,) and a period (.).

3.4.4.2 Edit Mode. When the PAD/FUNC key is in the normal position (not depressed), the keypad can be used for editing purposes when in special programs (e.g., the VERSAdos editor). Table 3-4 defines the editing notations and the character codes generated.

CHAPTER 4

SOFTWARE DESCRIPTION

4.1 INTRODUCTION

The VME/10 Microcomputer System package includes the VERSAdos operating system and associated development system software furnished on the fixed Winchester media. VERSAdos consists of a powerful set of file-handling utilities, security capability, real-time multitasking kernel, a system generation facility, an M68000-family assembler and linkage editor, a CRT-oriented text editor, diagnostics, and both symbolic and non-symbolic debuggers.

Optionally available are Pascal and FORTRAN compilers and various cross assemblers, cross linkers, and a cross Pascal compiler; the latter make it possible to assemble or compile and link programs for 8-bit "target" systems using the VME/10 as the "host" design station.

Also available is the SYSTEM V/68 operating system for the VME/10. SYSTEM V/68 is not described in this manual. For complete information, refer to the list of SYSTEM V/68 documentation in paragraph 1.11.

The VME/10 firmware contains a resident monitor/debugger, TENbug, useful not only as the "bootstrap" of the operating system or other program, but as a simple, easy-to-use debug tool.

This chapter and those that follow provide not only general descriptions of the features and functions of various components of the VME/10 software, but step-by-step examples which can be performed by the new user for familiarization with the system.

NOTE

> Before using the software furnished on the Winchester disk, a backup copy should be made and stored for safekeeping, as directed in Chapter 2.

System firmware error messages are listed in the TENbug Debugging Package User's Manual, M68KTENBG. Operating system error messages are described in the VERSAdos Error Messages Manual, M68KVMSG.

4.2 VERSAdos

4.2.1 Functional Overview

VERSAdos is a modular, multilayered operating system that provides a convenient and user-friendly interface between the user and system hardware. It provides a solution to general-purpose program generation requirements associated with the development of microprocessor-based systems, as well as the execution requirements of dedicated, real-time, multitasking application systems. The modular nature of the operating system permits configuration of the VME/10 for a variety of host/target applications. This flexibility reduces the costs and problems normally encountered during system integration by permitting extensive debugging to be performed on a compatible hardware/software configuration prior to the integration process.

VERSAdos operations are task oriented. A task is a program, complete with its associated data area, that performs a functional unit of work. Application programs are performed as tasks and are executed according to their priorities, scheduling requirements, and availability of required resources.

VERSAdos is responsible for accepting, checking, interpreting, and expediting user application requests. During execution of a task, the operating system may request assistance from various operating system support routines not directly accessible to the application program. These support routines assist in operator control, memory management, task segmentation, and input/output control for various hardware subsystems. This permits execution of more than one task at a time, thereby allowing several application programs to be operating independently on the system. This also relieves the application program from the necessary chore of direct interaction with the system hardware. Instead, application programs communicate their input/output requests to the system via the operating system using an established protocol.

The operating system is divided into four major layers, with each layer further subdivided into other layers. The four major layers are: the Real-Time Multitasking Executive (RMS68K) layer, the I/O layer, the File Management layer, and the Session Management layer. This structure is shown in Figure 4-1.

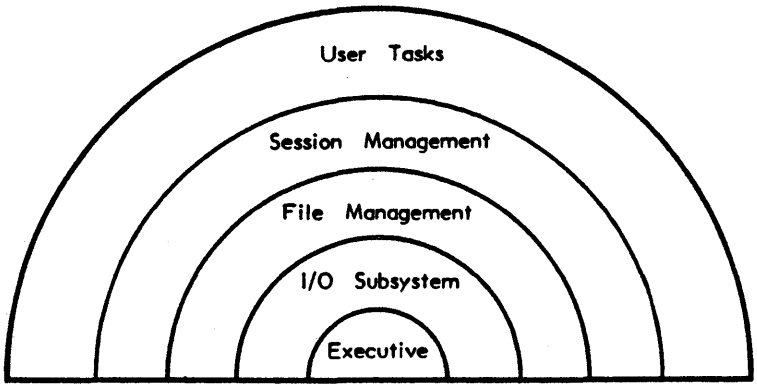


FIGURE 4-1. VERSAdos Structure

COPY

The COPY utility copies a file onto the same volume under a new file name, or onto another volume under the same or a new name. Options allow a file to be appended to the end of an existing file, packing of data in an indexed sequential file, character-by-character comparison of existing files with display of byte differences within records, and character-by-character comparison of a copied file and the original with display of byte differences within records. Output can be sent to a printer if part of the system, or to the display terminal for a quick look at the contents of a file.

DEL

The Delete utility removes a file name from a disk directory and frees all space allocated to that file. Options allow a list of files or a "family" of files with like parameters (e.g., same catalog or same extension) to be deleted with one command, and/or to direct a list of files deleted (normally displayed on the CRT) to an output file or to a printer.

DIR

Each VERSAdos disk contains a file directory which was established in sector 0 of the disk when the disk was initialized. Information describing the disk space allocation, location, and attributes of each file contained on the disk is stored in this directory. Part or all of the information entered for each file can be obtained by using the DIR utility. Options provide greater detail about type, size, and location of each file. See the REPAIR utility in the VERSAdos System Facilities Reference Manual, M68KVSF, for complete descriptions of the VERSAdos file structure.

DMT

This utility, used in conjunction with the MT utility, enables the VME/10 to handle disks of unlike formats. DMT performs the complementary function of the MT utility. It forces VERSAdos to release control of a mounted floppy disk and to reject input/output requests to a new disk until the MT command has been reissued. Before using DMT the floppy must be off-line -- i.e., the floppy drive door be open.

DISPATCH

The use of this utility is privileged; i.e., only logon user 0 may use it. It is used in conjunction with BATCH job processing, to change dynamically the number of batch jobs that are able to execute.

DUMP

DUMP is a utility that allows examination and/or modification of one or more sectors of disk data. The basic command provides a display of the contents of a disk, a file, or a portion of a file, in hexadecimal; alternatively, the dump may be directed to a printer or into another file. Specifying the interactive option allows certain sectors of the disk or file to be read into a change buffer in memory; bytes may be individually examined, changed, and read back to the disk to replace the original version.

DUMPANAL

DUMPANAL is an interactive utility used to analyze the contents of a system crash dump, if the data has been saved in a file by means of the firmware-resident monitor's DB command. DUMPANAL lists various system tables and memory locations as they appear in the dump file.

EMFGEN

This utility allows the user to add error messages and/or alter existing messages in the error message file, ERRORMSG.SY, which is used by VERSAdos' error message handler to issue most system messages.

FREE

Knowledge of unallocated space on a disk is often needed for file creation or editing, or before copying a file. The FREE utility determines and displays the total number of available sectors and the size of the largest available block of contiguous sectors in decimal and hexadecimal representation for a specified volume.

INIT

All blank diskettes for use on the VME/10 must be formatted and initialized with the INIT utility before their first use. Formatting establishes a sector/track pattern on the diskette which is compatible with the VME/10 and VERSAdos. Initializing creates a Volume Identification Block (VID) on the diskette which can be recognized by VERSAdos. The VID includes a user-supplied volume I.D., description, and ownership. A disk file directory is also created by INIT. If directed to do so, INIT will check the disk for bad sectors; if any are found, INIT will write their locations into the Sector Lockout Table (SLT) so data cannot be written to them.

Used diskettes can also be initialized with INIT to clear the file directory. (Disks containing wanted files should not be initialized, as their directory entries will be altered so as to be unrecognizable by VERSAdos, and new data will overwrite their contents.) The formatting function need not be performed when initializing a used VERSAdos disk. (Note: Formatting destroys all data on a disk.)

An option allows specification of the address of the bootstrap file. For the VME/10, the VERSAdos bootstrap file is named SYS:0..IPL.SY, and it must be loaded at location $10000.

Although INIT can be used on hard disks, the fixed hard disk furnished with the VME/10 was formatted and initialized at the factory and contains all VERSAdos files. It should not be re-initialized unless these files are to be replaced.

LIB

The Library utility makes useful software routines available for use by more than one program or more than once in a program. These routines, or program modules, are created in assembly or high-level language; put into a file using the editor; assembled or compiled; and combined into a "library" file or files with the LIB utility. These user-created library files, along with those supplied with the system and with optional high-level languages, can then be linked and made accessible to application programs. LIB offers several interactive commands to aid in manipulation of the modules while creating library files.

LIST

Using the LIST utility, all or part of an ASCII disk file can be displayed, written to a separate file, or (if a printer is part of the system) printed. Selectable options allow specification of beginning and/or ending lines; numbering of lines; prompt for wider or narrower line length and longer or shorter page length specification; prompt for heading; and interactive mode. In interactive mode, if the heading prompt option or non-standard length and width prompt option were specified, these parameters can be supplied. Lines to be listed can also be specified while in interactive mode.

MBLM

Object files which were assembled using the M68000 Family Cross Macro Assembler are in S-record format. These files cannot be linked into load modules, but can be transported to the VME/10 and then converted to loadable and executable files by means of the MBLM utility.

MIGR

ASCII programs filed on MDOS-format diskettes can be converted to VERSAdos format with the MIGR utility. MDOS is the resident operating system for Motorola's EXORciser computer. Because EXORciser's standard drives are typically EXORdisk 8" floppies, and VME/10's floppy drive is 5-1/4", an EXORdisk must be available to the VME/10 in order to use MIGR.

MT

MT allows VERSAdos to access disks of differing media formats. It must be used before performing I/O operations to a floppy diskette on the VME/10 (except for the first diskette accessed after powerup). In turn, the DMT utility must be used after the diskette has been taken off line, to release the device. If the diskette is of VERSAdos format (contains a VERSAdos V.I.D.), entering the MT command and the device designation is all that is required. If the diskette is of foreign format, however, it may be accessed after mounting when configuration data has been supplied by the user during MT's interactive dialog.

NOVALID

If system security level 2 or 3 is in effect, and a user password file exists, NOVALID is used to delete specified user number records from the file.

PATCH

Changes can be made to executable load module files with the PATCH utility. Interactive subcommands allow the display and change of portions of a file after it has been read into memory. This makes it possible to make changes to a program without having to change the source and reassemble it. PATCH includes a one-line disassembler and a one-line assembler.

PRTDMP

The Print Dump utility, PRTDMP, allows dumping part or all of memory to a file after an abort of a load module. The file or a portion of it can then be displayed or routed to a printer for examination. To use this utility, the load module must have been linked with the linker's D option. Interactive commands vary the type of output.

RENAME

This utility is used to change a file's name, catalog, and/or extension. The system administrator (logon user 0) may also change a file's user number. User 0 or the volume owner may change a file's protection key.

REPAIR

REPAIR is an interactive utility used to repair the various logical structures of disks and files if they have become damaged. These structures include:

VID	Volume I.D. block
SAT	Sector allocation table
CFGA	Configuration area (media format)
SDB	Secondary directory block (catalog list)
SDE	Secondary directory entry (catalog entry)
PDB	Primary directory block (file name list)
PDE	Primary directory entry (file name entry)
FAB	File allocation block (list of data blocks)
DB	Data block (list of sequential records)
HDR	Header
SLT	Sector lockout table
DTA	Diagnostic test areas

REPAIR can be used to recover a deleted file, if the file's DB and FAB have not been reallocated.

SCRATCH

This utility quickly erases the VID of a used diskette so that it can be reused. Only the disk's owner or logon user 0 can SCRATCH a disk. The disk also may be reformatted with SCRATCH. After using this utility, the disk must be reinitialized by INIT.

SESSIONS

The SESSIONS utility is used to determine the current online sessions and the batch jobs in queue for execution. Information is displayed by device number (terminal) and sessions number for online sessions and by user number and session number for batch jobs.

SNAPSHOT

The SNAPSHOT utility copies the display on the CRT screen to a file or, if present, a printer.

SPL/SPOOL

VERSAdos offers a spooling capability whereby a particular volume can be designated as storage media for a queue of files awaiting time-consuming background tasks such as batch and chain processing and printing. This frees the system for foreground operations. SPL must be installed in session 0001. SPOOL may then be accessed whenever needed in subsequent sessions. SPOOL includes a list of subcommands for initiating, monitoring, and cancelling spooling functions.

SYSGEN

The SYSGEN facility makes it possible to customize the operating system, deleting unwanted parameters and adding others to accommodate additional peripheral equipment. Furnished with VERSAdos are SYSGEN command files and chain files for several specific system types which facilitate this process. The files for a particular system are identified by their catalog name; e.g., the command file for the VME/10 is named VMES10.SYSCMD.CD. This file reflects the exact configuration of the VERSAdos software furnished for VME/10 uses. By examining this file, the user can determine whether any of several system attributes should be redefined. If changes are made to SYSCMD, SYSGEN must then be used on the file to incorporate the changes into the system, before the changes can take effect.

SYSANAL

SYSANAL is an interactive operating system debugging utility. It provides a means of examining system tables in VERSAdos, and at any part of memory while VERSAdos is running. Output is to the display screen or to a printer if one is available.

TRANSFER

The ASCII file TRANSFER utility allows up- or downloading of printable ASCII files such as source code or S-records between the VME/10 and another system. The systems may be connected directly between serial ports, or by phone lines/modems. Both systems must be configured for the same baud rate and character makeup. TRANSFER uses two associated programs, ULOAD and DLOAD. The receiving system must have a Pascal compiler to compile ULOAD and DLOAD.

UPLOADS

UPLOADS is used to migrate S-records from some external source to a VERSAdos system. The S-records must be received through an MVME400 dual port serial module or the VME/10 I/O Channel which is connected to the source system via a direct RS-232C hardware configuration.

4.2.5.2 Examples. Following are some typical examples of some frequently used utilities. They may be used for familiarity with the system.

Boot VERSAdos as described in Chapter 2. Insert a blank diskette in the floppy drive, and make the following entries:

```
=INIT #FD02
OK TO INITIALIZE #FD02 (Y/N) ? Y

Configuration change-to exit enter C (configure) or Q (terminate utility).

Data Density of media (S-single,D-double) D > C
DO YOU WANT TO FORMAT DISK (Y/N) ? Y
START FORMAT
ENTER NEW VOLUME NAME VOL1
ENTER USER NUMBER 0
ENTER DESCRIPTION (MAX 20 CHARACTERS) PRACTICE ONE
DO YOU WANT THE BOOTSTRAP (Y/N) ? N   (Prints only for user 0)
DO YOU WANT A DUMP AREA (Y/N) ? N     (Prints only for user 0)
DO YOU WANT TO VALIDATE SECTORS (Y/N) ? Y
   VALIDATING SECTORS ....
   0 BAD SECTORS ENCOUNTERED
= COPY 0..PATCH.LO,VOL1:0..PATCH.LO
=
```

Remove the diskette and dismount it by entering:

```
=DMT #FD02
DISMOUNT Version xxxxxx x
```

Insert another blank diskette, repeat the INIT #FD02 sequence above, giving this diskette a volume name of VOL2 and a description of PRACTICE TWO.

At the VERSAdos prompt, enter:

```
=COPY 0.*.*.XX,VOL2:0.*.*.XX
COPY ALL OR SELECT FILES (A/S) ? A
 .
 .
 .
FILES COPIED = n

=COPY 0.*.*.NW,VOL2:0.*.*.NW
COPY ALL OR SELECT FILES (A/S) ? A
 .
 .
 .
FILES COPIED = n
=
```

These commands will copy various news and instructional files to the diskette. The asterisk (*), or "wild card", selects all files on the default volume (the fixed disk) with a blank catalog name and extensions of XX and NW. The files are listed as they are copied.

```
=DIR #FD02
DEVICE FD02 IS VOLUME VOL2
USER NUM= 0000        DESC= VOLUME TWO
=DIR VOL2:0.*.*.*;S
DIR VERSION xxxxxx x  mm/dd/yy  hh:mm:ss
 .
 .
 .
```

All files on the diskette are listed alphabetically on the CRT screen.

Change default volume to the floppy with the USE session control command:

```
=USE VOL2:
SYSTEM VOLUME =SYS:
USE DEFAULT VOLUME = VOL2:0..
USER NUMBER = 0
USER TASK =
SESSION = 0001
TERMINAL = CNSL
OPTION(S) SET =
=DIR ;E
DIR VERSION xxxxxx x  mm/dd/yy  hh:mm:ss
```

Each file on VOL2 is listed, with all directory information as to file type, size, location, and protection.

```
=COPY TRANSFER.XX,#
```

The contents of the ASCII File Transfer instructional file are displayed on the screen. To halt display, hold down the CTRL key and press the W key. To resume display, type any key.

```
=LIST TRANSFER.XX;L=1
```

The file contents are listed on the screen in LIST's format, with page heading and line numbers.

```
=DEL TRANSFER.XX
DELETED VOL2:0000..TRANSFER.XX
=DIR
  .
  .
  .
```

(Note that TRANSFER is no longer listed in the volume directory.)

Use the FREE utility to ascertain how much space is left on the diskette (dddd = decimal, $hhh = hexadecimal):

```
=FREE
VOLUME VOL2:
    dddd/$hhh        TOTAL SECTORS AVAILABLE
    dddd/$hhh        LARGEST CONTIGUOUS SECTORS
          xx%        OF SECTORS ARE AVAILABLE
=USE SYS:
   .
   .
   .
```

Remove the diskette and dismount it:

```
=DMT VOL2:
DISMOUNT Version xxxxxx x
```

Insert the diskette VOL1 and mount it:

```
=MT #FD02
MOUNT Version xxxxxx x
VOL1 has been mounted
Total Vdos sectors       2552
=DUMP VOL1:0..PATCH.LO,#;I
DUMP VERSION xxxxxx x
>D $7,$8
```

The I option instructs DUMP to enter the interactive mode, and the # in the output field calls for output to the console screen (default mode; #PR will result in output to the printer). In interactive mode, the D subcommand asks for a dump of sectors $7 and $8. They are displayed on the screen in hexadecimal, with printable ASCII characters at the right-hand side.

```
>QUIT
=OFF
09:52:15  9/15/83  END SESSION 0001 USER 0
```

4.3 SOFTWARE DEVELOPMENT

A user may custom-configure an operating system to suit a particular application by using the VERSAdos System Generation Facility (SYSGEN) to modify any of several system attributes, including:

- Type and number of devices
- Number of logical units per task
- Amount of memory space for:
  - Global segment table
  - Trace table
  - Device connection queue
- Number of file assignments in the system at one time

A file must be created to contain a series of commands from the SYSGEN command set. In addition, utility programs not containing interactive dialog may be invoked from within this "command file", allowing a utility or selected portions of a utility to be run as if it had been called directly.

Furnished on the VERSAdos media are command files which represent the configurations of the furnished VERSAdos versions, along with chain files which can be used to perform the SYSGEN. These command files can be listed to identify their parameters, and if a different configuration is required, the command files can be modified and a new operating system generated with SYSGEN.

The following paragraphs contain concepts to be considered when designing an operating system, and a brief listing of the SYSGEN command set. For full information on the SYSGEN process -- the SYSGEN command set, user-changeable parameters, and the SYSGEN command syntax -- refer to the System Generation Facility User's Manual, M68KSYSGEN, which includes a typical example of a SYSGEN command file for the VME/10.

Refer also to the M68000 Family Real-Time Multitasking Software User's Manual, M68KRMS68K, for a more detailed discussion of design concepts.

PRINT

Output records to the printer. A vertical range of records may be specified; default is the entire file. The records may optionally be printed out double- or triple-spaced by specifying the option D or T. For example, PRINT prints the entire edit file, single-spaced, on printer #PR. PRINT 100-199 #PR2 D prints records 100 through 199, double-spaced, on printer #PR2.

Q[UIT]

Save the edit file in a VERSAdos file, terminate the edit session, and return control to VERSAdos. For example, QUIT closes the edit file and writes it to the disk under the output file name. If no changes were made during editing, no output file is created. If, after editing, the changes made are not wanted, exiting with QUIT A prevents the output file from being created. Note, however, that the A option on QUIT is only valid if the output file was to have been sequential (either because the input file was sequential or because the S option was specified on the E command line).

R[ANGE]

Establish default values for the vertical and horizontal ranges of the CHANGE, FIND, PRINT, and SAVE commands. The original defaults are entire record and/or entire file. RANGE is used to change these defaults. For example, R 1-100 changes the vertical range for the FIND, CHANGE, PRINT, and SAVE commands. R :10-30 leaves the vertical range unchanged, but sets the default horizontal range to columns 10 through 30. Entering R or RANGE alone returns defaults to entire record/entire file.

SAVE

Save part or all of the edit file in a VERSAdos file, and continue editing. For example, SAVE *-100 VOL1:0.CAT1.FILE2.SA creates a new file named CAT1.FILE2.SA on VOL1, and copies 100 lines into it from the edit file, beginning at the current line. SAVE FILEDUP.SA copies the entire edit file into a new file named FILEDUP on the current volume.

TAB

Specify tab stops. The default tab stops are set at every 10th column. The column numbers of desired tab locations to be added to existing settings can be specified, or an option letter (A, C, F, or P) can be specified to set tabs at locations convenient for Assembler, COBOL, FORTRAN, or Pascal source programming. When an option letter is specified, previous tab settings are cleared. Specifying TAB with no tab stops or option letter sets tabs at the default settings, or at settings specified on the E command line, if any. For example, TAB 25,35 adds stops at columns 25 through 35, but does not change any settings already in effect. TAB A sets tab stops at columns 1, 11, 18, and 37, convenient for Assembly language source code.

U[P]	Move the record pointer upwards. For example, U 25 moves the cursor to the 25th line preceding the current line.
X[TRACT]	Copy the records from the XTRACT buffer placed there by ADUP, DUP, AMOV, or MOVE, and insert them above the current record in the file. The records still remain in the buffer. XTRACT may also be used simply to empty the buffer. For example, XTRACT or X copies the contents of the buffer into the edit file, above the current line. XTRACT A or X A does not copy the contents of the buffer into the edit file, but deletes them from the buffer.

LINE MODE ONLY (;L OPTION OR WITHIN CHAIN FILES OR WHEN CONNECTED TO A MODEM)

COLM	Display the ruler of column spacings.
DTAB	Delete tab stops.
I[NSERT]	Enter insert level, for adding records to the edit file.
L[IST]	List records in the edit file.
STAB	Specify tab stops.
V[ERIFY]	Display records that are altered or record pointer changes.

5.1.3 Examples

The following simple examples are intended to illustrate various functions of the editor.

With VERSAdos running, insert the diskette VOL1 created in the example in Chapter 4 (or use another formatted, initialized "scratch" diskette) and enter:

```
= MOUNT #FD02
  .
  .
  .
=
```

Change defaults to the diskette (if volume name is not VOL1, substitute the actual volume name for VOL1):

```
= USE VOL1:0
  .
  .
  .
=
```

CHAPTER 8

PASCAL COMPILER

8.1 INTRODUCTION

Source programs written in VERSAdos' Pascal for the VME/10 are compiled with the M68000 Family Pascal Compiler (optional purchase) and then linked with applicable library routines by the M68000 Family Linkage Editor to create an executable load module. They may be linked, also, with other Pascal subprograms and assembly language subroutines.

The Pascal compiler consists of three separate programs which are run sequentially. The first and third programs are required; the second program, which is optional, is an optimizer which reduces code size and increases its efficiency, thereby increasing the speed at which the finished Pascal program executes. The three programs are named PASCAL, POPTIM, and PASCAL2, and are also referred to respectively as Phase 1, Phase 1.5 (or optimizer), and Phase 2. Each program processes its input file in a single pass and generates the input file for the next program.

8.2 SOURCE PROGRAMS

8.2.1 Pascal Source Programs

Various options can be specified from within the Pascal program that affect the compiler's source, listing, and object output, control runtime checks, change stack and heap size, and call for fast floating point arithmetic.

These options are specified in the source file as a Pascal comment, with an additional symbol which informs the compiler that the comment is an "option comment". Two forms may be used:

{$<option>} or (*$<option>*)

Options are specified as alphabetic characters, followed immediately by a plus, minus, or equal sign. More than one option can be specified within the same comment, separated by commas but not spaces. The option comments generally may be specified anywhere a comment is normally allowed.

Many of the options may alternatively be specified on the Pascal command lines (paragraph 8.3). The option comment characters and their meanings are:

OPTION	DEFAULT	FUNCTION
A=n	A=4	Specify the number of bytes used for integer arithmetic.
C-	C+	Generate an input file for the optimizer or Phase 2. Eliminating this file reduces the time necessary to generate the listing and any errors.

OPTION	DEFAULT	FUNCTION
D+	D-	This combines the K and R options to (1) generate code to perform runtime checks which verify that array indices and subrange type variables are in range, and (2) include executable unit numbers in the executable object code.
E	none	Page eject for Phase 1 listings.
F=<filename>		Include the file specified by <filename> in the source. Immediately after the line which contains this comment option, Phase 1 will start obtaining its source input from the file indicated by <filename> (which must conform to the rules for specifying a file name for the operating system). When the end of the "include file" is encountered, Phase 1 will return to getting its source from the original source file at the point it left off.
G+	G-	Keep object files output by the compiler or optimizer which contain errors (normally deleted).
H=n	H=4096	Specify the size of the program heap in bytes.
I-	I+	Pass any external files specified on the command line to the program at start-up.
K+	K-	Include executable unit numbers in the executable object code. The executable unit numbers relate to statements and are found on the source listing.
L-	L+	Generate a source listing in the Phase 1 listing file, on the printer, or on the CRT.
O+	O-	Enter source statements as comments in the Phase 2 input.
P+	P-	Include executable unit numbers in the executable object code, but only at function/procedure entry and exit points.
Q+	Q-	Use fast floating point.
R+	R-	Generate code to perform runtime checks which verify that array indices and subrange type variables are in range.
S=n	variable	The value specified by n will be the default stack/heap size in bytes used by the program. If specified, n must be at least 768.
W+	W-	Generate a warning during Phase 1 processing if non-standard Pascal features are used. Standard Pascal comprises only the language features proposed by Jensen and Wirth.

CHAPTER 9

DEBUG CAPABILITY

9.1 INTRODUCTION

A load module often requires debugging to overcome deficiencies which come to light when the program runs in an actual application. Supplied with VERSAdos are two debug monitor programs -- DEbug and SYMbug. In addition to these, a firmware-resident debug monitor program, TENbug, is supplied in the ROM of the VME/10 System.

9.2 TENbug

TENbug is the resident firmware monitor and debugging package for the VME/10. The 32K-byte firmware (stored in ROM or EPROM devices) provides a self-contained programming and operating environment. TENbug may be entered directly at system power-up or from VERSAdos. These two methods are given in Chapter 2.

TENbug interacts with the user through predefined commands that are entered via the terminal. The commands fall into five general categories:

a. Commands which allow the display or modification of memory.

b. Commands which allow the display or modification of the various internal registers of the MC68010.

c. Commands which allow execution of a program under various levels of control.

d. Commands which control access to the various input/output resources on the board.

e. Commands which allow selection of video graphics resolution.

An additional function called the TRAP #15 I/O handler allows the user program to utilize various routines within TENbug.

For complete information on TENbug, refer to the TENbug Debugging Package User's Manual, M68KTENBG.

9.2.1 Command Set

TENbug's debugging functions are performed in response to the entering of simple "primitive" commands, with or without associated parameters and options. Several of the commands are set and reset pairs -- the reset function is specified by preceding the command with NO. The entry of a command line is always followed by pressing the carriage return key (<--|). TENbug checks each entry for validity, returning an error message if incorrect, or processing the command and displaying an interpretation of the parameter values if correctly entered.

Table 9-1 lists the primitive commands supported.

TABLE 9-1. TENbug Commands by Type

COMMAND	DESCRIPTION
[NO]BARS	Draw graphics test pattern.
BF	Block fill.
BH	Boot and halt (not used when SYSTEM V/68 is resident).
BI	Block initialize.
BM	Block move.
BO	Boot operating system.
[NO]BR	Set and remove breakpoints.
BS	Block search;
BT	Block test.
[NO]CH	Alter character display map.
CRT	Alter CRT control registers.
CS	Checksum.
DC	Data conversion/evaluation.
DF	Display formatted registers.
DU	Dump memory (S-records).
GD	Go direct.
G[O]	Install breakpoints and go.
[NO]GR	Alter graphics display map.
GT	Go until address.
HE	Display commands/registers.
IOC	Issue RWIN1 command.
IOP	Issue physical read/write.
IOT	Teach RWIN1 a configuration.
LO	Load (S-records).
MD	Memory display.
M[M]	Memory modify.
OF	Offset register display.
[NO]PA	Printer attach/detach.
PF	Port format.
TM	Transparent mode.
T[R]	Trace.
TT	Trace until address.
VE	Verify (S-records).
VM	Toggle video map.
.A0 - .A7 [<expression>]	Display/set address register.
.D0 - .D7 [<expression>]	Display/set data register.
.DFC [<expression>]	Display/set destination function code.
.PC [<expression>]	Display/set program counter.
.R0 - .R6 [<expression>]	Display/set relative offset register.
.SFC [<expression>]	Display/set source function code.
.SR [<expression>]	Display/set status register.
.SSP [<expression>]	Display/set supervisor stack pointer.
.USP [<expression>]	Display/set user stack pointer.
.VBR [<expression>]	Display/set vector base register.
BREAK	Abort command or process.
DEL	Delete character.
CTRL-D	Redisplay line.
CTRL-H	Delete character.
CTRL-W	Suspend output; any character continues.
CTRL-X	Cancel command line.
<--!	Process current/previous command line.

9.2.2 TENbug Examples

The following example assumes that the system has been initialized and the furnished software has been backed up, as directed in Chapter 2 (VERSAdos) or the SYSTEM V/68 documentation (SYSTEM V/68).

a. Ensure that the KYBD LOCK key switch on the VME/10 chassis is in the unlocked (horizontal) position.

b. Press the on/off switch to the "1" position and wait for the hard disk to spin up.

c. Press and release the RESET pushbutton on the VME/10 chassis. TENbug will take control and display its prompt (if system includes an MVME400, press any key on the keyboard after pressing RESET, to get the full prompt):

```
TENbug x.y >
```

d. Display and alter the contents of MC68010 registers by typing in the commands shown underscored (underscore is not to be typed), following each entry with a carriage return. (The initial register values displayed will differ from these.)

```
TENbug x.y > DF                          (Display formatted registers)
PC=00F02C9E SR=2704=.S7..Z.. USP=FFFFFFFF SSP=000007C4 VBR=00000000 SFC=2 DFC=2
D0-7 00300030 00000804 00000000 00000000 4D505520 00000020 00000000 00000000
A0-7 00F1A031 00F0133C 00F008AA 00000458 0000049A 00000536 00000536 000007C4
  PC=F02C9E

TENbug x.y > .R1 3000                    (Set register R1 offset to 3000)

TENbug x.y > OF                          (Display offset registers)
R0-7 00000000 00003000 00000000 00000000 00000000  0000000 00000000 00000000

TENbug x.y > .PC 40000                   (Change value in program counter)

TENbug x.y > .SSP C00                    (Set supervisor stack pointer)

TENbug x.y > DF                          (Display formatted registers)
PC=00040000 SR=2704=.S7..Z.. USP=FFFFFFFF SSP=00000C00 VBR=00000000 SFC=2 DFC=2
D0-7 00300030 00000804 00000000 00000000 4D505520 00000020 00000000 00000000
A0-7 00F1A031 00F0133C 00F008AA 00000458 0000049A 00000536 00000536 00000C00
  PC=03D000+R1

TENbug x.y >
```

e. Return to VERSAdos with the boot command:

```
TENbug x.y > BO
```

or to SYSTEM V/68 with the boot command:

```
TENbug x.y > bo
```

9.3 DEbug

DEbug is a VERSAdos-resident monitor program, used to debug other programs whose source code is written in assembly language for execution on the MC68010. The language processor and the linkage editor suppy information to the DEbug monitor.

DEbug allows the user to examine, insert, and modify program elements such as instructions, numeric values, and coded data.

Execution can be controlled by DEbug, via the insertion of breakpoints into a program.

DEbug uses an extensive set of primitive commands for manipulation and examination of foreground tasks. A set of task-level commands may be used on foreground or background tasks and are applicable to both the single and multitasking modes of operation.

9.3.1 Command Line

The DEbug program is invoked as follows:

= DEBUG [<program name>]

Specifying the name of the load module to be debugged enters single task mode. The first four letters of the program name are then included in the DEbug prompt. Typing DEbug without a program name enters multitask mode. The maximum number of tasks to be debugged must then be specified, and either the LOAD or ATTA command must be used before any of DEbug's primitive commands can be used.

9.4.3 SYMbug Commands

SYMbug primitive commands are listed in the following table.

TABLE 9-3. SYMbug Primitive Commands

COMMAND SYNTAX	DESCRIPTION
AS [<address> [<value>[;<mask>]]]	Address stop
BF <address1> <address2> <data>[;<length>]	Block fill
BM <address1> <address2> <address3>	Block move
[NO]BR [<address>[;<count>]]...	Set/reset breakpoint
BS <address1> <address2> <data>	Block search
CR [<count>]	Command repeat
DC <expression>	Define constant or Data convert
DE [<default option>]	Defaults
DF	Display formatted registers
FR <file name>	File read
FS <file name>	File save
G[O] [<address>]	Go (execute)
HE[LP] [<command>]	Display commands
[NO]MA [<name>]...	Set/reset macro define
MAE	Macro edit
MD <address> [<count>][;<option>]	Memory display
MM <address>[;<option>]	Memory modify
MS <address> <data>	Memory set
OF	Display Offset register
Q[UIT]	Quit (go to VERSAdos)
[NO]SD [<local> [<value>]]	Set/reset symbol define
T[R] [<count>]	Trace
ATTA <task name>[,<terminal>|#*]	Attach task
DETA [<task name>]	Detach task
EVEN [<task name>],<exception #>	Event definition
LOAD <file> [<command line>]	Load (task)
MASK [<task name>],<exception #>	Mask exception
STAR [<task name>|ALL]	Start task(s)
STAT [<task name>,<status>]	Status definition
STOP [<task name>|ALL]	Stop task(s)
TASK <task name>[,<note level>]	Task notify
TERM <task name>	Terminate task
WAIT	Wait task
BREAK	Abort command
CTRL-S	Redisplay line
CTRL-H	Delete character
CTRL-W	Suspend output (See NOTE)
CTRL-X	Cancel command line
<--'	Send line for execution

NOTE: After CTRL-W has been used, the entry of any character will cause the output display to continue.