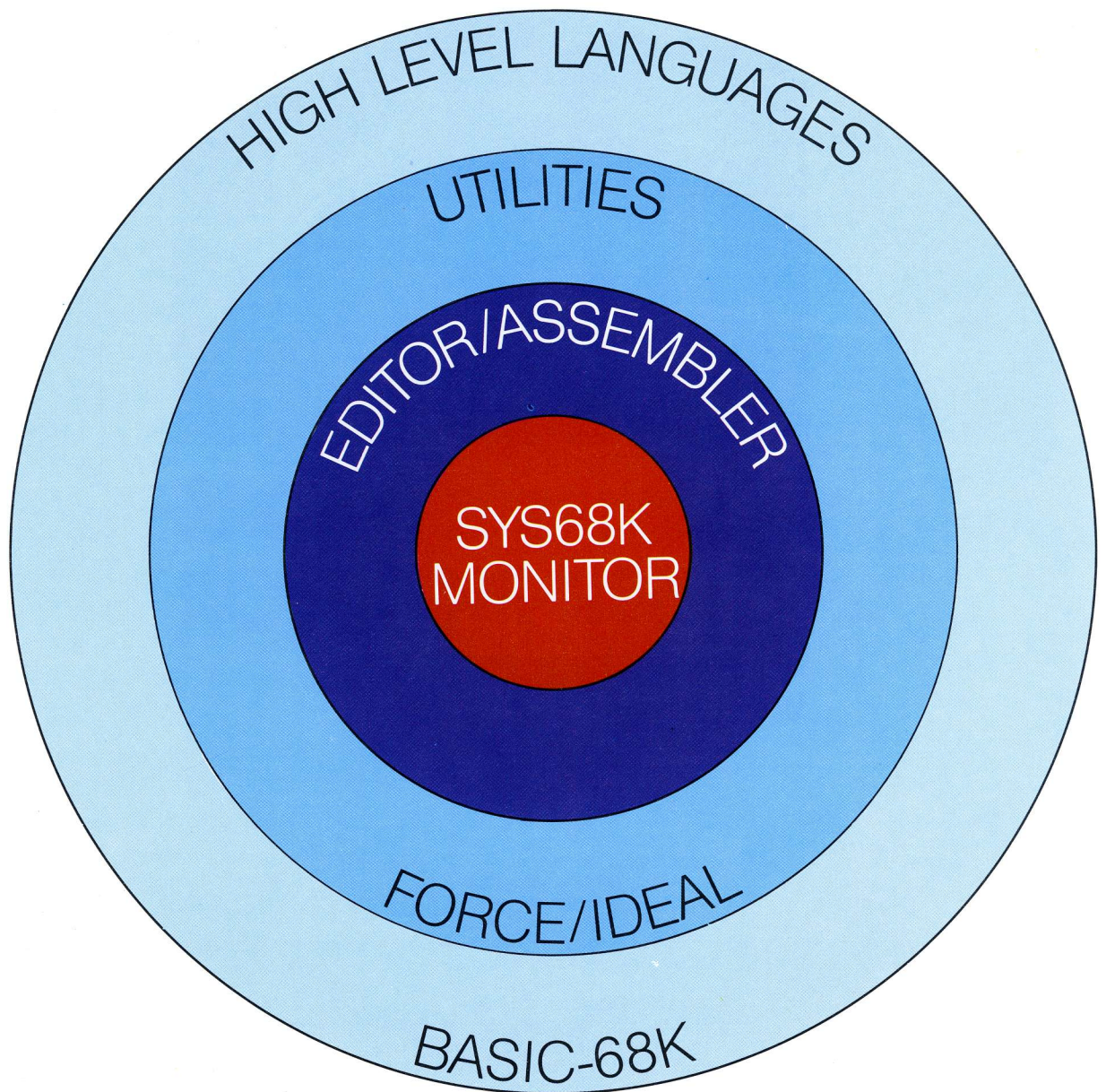




**System 68000 VME**  
**SYS68K MONITOR**  
Debug Package for  
VME-Board Module



## Features of the resident SYS68K MONITOR package:

- EPROM resident system monitor / debugger
- More than 30 commands for debug, up/downline load
- One-line assembler/disassembler for assembly language program development
- Full speed execution of system and user programs operating in the VME bus oriented monoboard microcomputer system
- Terminal capability for up/downline load from a FORCE F7000 system or any host computer
- Powerful software and system debug command set allow access to all VME modules plus the full 16M byte direct address range of the VME system bus
- Includes all required installation and operation documentation

SYS68K MONITOR is an EPROM based resident package ready for installation and immediate use with the VME monoboard CPU as well as for VME based microcomputer products. SYS68K MONITOR provides a powerful evaluation and system debugging tool for VME based CPU systems. The EPROM resident package will operate in 16K bytes of ROM space. SYS68K MONITOR uses the first 1024 words of RAM storage for interrupt vectors and temporary storage. The EPROM resident package is delivered in two 8K byte EPROMs. Table 1 lists the commands available to the user. The package permits full speed execution of system and user developed programs operated in a VME based CPU system environment under complete operator control. SYS68K MONITOR may be utilized with the VME based CPU monoboard microcomputer SYS68K/CPU-1 in a stand-alone environment with only a user provided standard RS232C asynchronous ASCII terminal. Alternately, it may be used with a second serial I/O port connected to a host computer for up/downline loading of programs in S-record format. When connected to a host computer in this manner, the VME CPU/SYS68K MONITOR operator terminal combination appears as a normal asynchronous ASCII terminal (a virtual terminal) to the host operating system. The second serial I/O port, the host computer interface, is implemented on the CPU board.

### Program Debugging Capability

In a typical debug session, the user will download his program to a VME CPU board from the host computer used for software development. After loading, SYS68K MONITOR commands

may be used to examine and modify memory, set breakpoints to run particular program segments, and trace program progress. The user may set up and examine a variety of conditions using any of the powerful commands listed in Table 1, such as the register display/set series and the memory block manipulation commands. The Data Conversion command serves as an aid in examining and modifying data by converting hexadecimal to decimal, and decimal to hexadecimal. If corrections or program patches are required, these may be performed and checked in the VME module. The user may save a corrected copy to the host computer files by using the DUMP command for upline load. Creating program patches may be aided by use of the Display Offset command to assist with relocatable and position independent code. The user may also copy all traffic to the serial port debug terminal on a printer attached to an auxiliary port by use of the Printer Attach command. This may be useful for desk debugging following a debug session. The user may communicate with the host computer as a terminal for purposes other than up/downline load by executing the Transparent Mode command or the Send Message to Port 2 command. By using the Port Format command, the serial port may be re-configured for such

attributes as baud rate, stop bits and number of data bits. In the transparent mode, the two serial ports must operate at the same baud rate.

### System Environment

SYS68K MONITOR may be used for debug in total systems environments which include the VME CPU monoboard and other VME bus compatible products. Included as part of the SYS68K MONITOR is a one-line assembler/disassembler function. The assembler/disassembler is a dynamic assembler/editor in which the source program is not saved. Each source line is translated into the proper 68000 machine language code and is stored in memory on a line-by-line basis at the time of entry. In order to display an instruction, the machine code is disassembled and the instruction mnemonic and operands are displayed. Valid 68000 instructions are translated.

### Assembler/Disassembler Capability

The on board assembler does not allow line numbers and labels; however, it is a powerful tool for creating, modifying, and debugging 68000 code. The on board assembler processes each line of a program as an individual unit. Therefore the capabilities of the SYS68K MONITOR assembler are more restricted:

- Label and line numbers are not used. – Labels are commonly used to reference other lines and locations in a program. The one-line assembler has no knowledge of other program lines and, therefore, cannot make the required association between a label and the label definition located on a separate line.
- Source lines are not saved. – In order to read back a program after it has been entered, the machine code is disassembled and then displayed as mnemonic and operands.

### Command Line Edit and Control Functions:

(BREAK)	Abort Command
(DEL)	Delete Character
(CTRL-D)	Redisplay Line (*)
(CTRL-H)	Delete Character (*)
(CTRL-X)	Cancel Command Line (*)
(CR)	Send Line to Memory (*)

\*It has to be noted that these functions are terminal specific. (See User Manual of Terminal.)

- Limited error indication. – The one-line assembler will show a question mark (?) under the portion of the source statement where an error probably occurred, or will display the word "ERROR" or another short message.
- Only one directive (DC.W) is accepted.
- No macro handling capability is included.
- No conditional assembly is used. The symbolic language used to code source programs for processing by the assembler is called 68000 assembly language. This language is a collection of mnemonics representing:
  - Operations
    - 68000 machine instruction operation codes
    - Directive (pseudo-op)
- Operators
  - Special symbols
 A source program is a sequence of source statements arranged in a logical way to perform a pre-determined task. Each source statement occupies a line and must be either an executable instruction or a DC.W assembler directive. Each source statement follows a consistent source line format.

**Table 1 SYS68K MONITOR COMMAND Summary**

COMMAND	DESCRIPTION
BF <address1> <address2> <data><CR>	Block Fill memory – from add1 through add2 with data
BM <address1> <address2> <address3><CR>	Block Move – move from add1 through add2 to add3
BR[ <address>[: <count>] ... ]<CR>	Set/display Breakpoint
BS <address1> <address2> <data><CR>	Block Search – search add1 through add2 for data
BT <address1> <address2><CR>	Block Test of memory
DC <expression><CR>	Data Conversion
DF<CR>	Display Formatted registers
DU[n] <address1> <address2>[ <string>]<CR>	Dump memory to object file
GO[ <address>]<CR>	Execute program
GD[ <address>]<CR>	Go direct
GT <address><CR>	Exec prog: temp breakpoint
HE<CR>	Help; display monitor commands
LO[n][ ;<options>]<CR>	Load Object file
MD <address>[ <count>]<CR>	Memory Display
MM <address>[ <data>][: <options>]<CR>	Memory Modify
MS <address> <data1> <data2> ... <CR>	Memory Set – starting at addr with data1, data2, ..
NOBR[ <address> ... ]<CR>	Remove Breakpoint
NOPA<CR>	Printer Detach
OF<CR>	Offset
PA<CR>	Printer Attach
PF[n]<CR>	Set/display Port Format
RM<CR>	Register Modify
TM[ <exit character>]<CR>	Transparent Mode
TR[ <count>]<CR>	Trace
TT <address><CR>	Trace: temp. breakpoint
VE[n][ =string]<CR>	Verify memory/object file
.AO - .A7[ <expression>]<CR>	Display/set address register
.DO - .D7[ <expression>]<CR>	Display/set data register
.RO - .R6[ <expression>]<CR>	Display/set offset register
.PC[ <expression>]<CR>	Display/set program counter
.SR[ <expression>]<CR>	Display/set status register
.SS[ <expression>]<CR>	Display/set supervisor stack
.US[ <expression>]<CR>	Display/set user stack
MD <address>[ <count>];DI<CR>	Disassemble memory location
MM <address>;DI<CR>	Disassemble/ Assemble memory location

## Ordering Information:

The SYS68K/Monitor is supplied with CPU-1/CPU-2/CPU-3.

SYS68K/CPU-1 B/FW Part No. 800013	Monitor User's Guide for CPU-1
SYS68K/CPU-2/SUM Part No. 800025	Monitor User's Guide for CPU-2
SYS68K/CPU-3/SUM Part No. 800038	Monitor User's Guide for CPU-3

© Copyright 1983  
Design FORCE COMPUTERS

Note:  
FORCE COMPUTERS reserves the right to make changes to the product herein to improve reliability function or design. FORCE COMPUTERS does not assume any liability arising out of the application or use of product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

**FORCE COMPUTERS INC.**  
2041 Mission College Blvd.  
Santa Clara, California 95054  
Phone (408) 988-8686  
TLX 172465  
FAX (408) 980 93 31

**FORCE COMPUTERS GmbH**  
Daimlerstraße 9  
D-8012 Ottobrunn/München  
Telefon (0 89) 6 09 20 33  
Telex 5 24 190 forc-d  
Telefax (0 89) 6 09 77 93