

VAX 9000 Family Console Command Description

Order Number EK-9000C-CD-001

**digital equipment corporation
maynard, massachusetts**

First Edition, May 1990

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Restricted Rights: Use, duplication, or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

Copyright © Digital Equipment Corporation 1990

All Rights Reserved.
Printed in U.S.A.

The postpaid Reader's Comment Card included in this document requests the user's critical evaluation to assist in preparing future documentation.

FCC NOTICE: The equipment described in this manual generates, uses, and may emit radio frequency energy. The equipment has been type tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such radio frequency interference when operated in a commercial environment. Operation of this equipment in a residential area may cause interference, in which case the user at his own expense may be required to take measures to correct the interference.

The following are trademarks of Digital Equipment Corporation:

BI	KLESI	RSTS	VAX
CI	LA	RSX	FORTRAN
DEC	MASSBUS	RT	VAX MACRO
DECmate	MicroVAX	RV20	VAXBI
DECUS	NI	RV64	VAXcluster
DECwriter	PDP	TA	VAXELN
DHB32	P/OS	TK	VMS
DIBOL	Professional	ULTRIX	VT
DRB32	RA	UNIBUS	Work
EDT	Rainbow	VAX	Processor
KDB50	RD	VAX C	XMI
KDM			

digital™

This document was prepared and published by Educational Services Development and Publishing, Digital Equipment Corporation.

Contents

About This Manual	xi
--------------------------	----

1 Console Command Language

1.1	Command Syntax	1-1
1.1.1	Command Qualifiers	1-3
1.1.1.1	Qualifier Entry	1-3
1.1.1.2	Qualifier Values	1-4
1.1.1.3	CPU and SCU Qualifier	1-4
1.1.1.4	Confirmation Qualifier	1-5
1.1.1.5	/ALL Qualifiers	1-5
1.2	In-Line Help	1-5
1.3	Special Keys	1-7
1.3.1	Predefined Keys	1-7
1.3.2	User-Defined Keys	1-8
1.4	Message Format	1-8
1.5	The Radix of Numbers	1-9
1.6	Expressions	1-10
1.6.1	Types of expressions	1-10
1.6.2	Data Types	1-12

2 Console Command Descriptions

ALLOCATE	2-2
:= (Assign String)	2-3
= (Assign Symbol)	2-5
@ (Execute Procedure)	2-7
BOOT	2-9
CALL	2-11

CLOSE	2-12
CONTINUE	2-14
COPY	2-16
CREATE	2-18
CREATE/DIRECTORY	2-20
CREATE/WINDOW	2-21
DEALLOCATE	2-24
DEASSIGN	2-25
DEBUG	2-27
DEFINE	2-28
DEFINE/KEY	2-30
DELETE	2-32
DELETE/PATTERN	2-34
DELETE/SYMBOL	2-36
DELETE/TRACE	2-38
DELETE/WATCH	2-40
DELETE/WINDOW	2-42
DEPOSIT	2-43
DIRECTORY	2-53
DISMOUNT	2-56
EDIT	2-57
EVALUATE	2-60
EXAMINE	2-62
EXIT	2-73
FIND	2-74
GOTO	2-76
HALT	2-77
HELP	2-78
IF	2-80
INITIALIZE	2-81
INQUIRE	2-85
LABEL	2-87
LOAD	2-88
LOGOUT	2-91
MAIL	2-92
MICROSTEP	2-93

MOUNT	2-95
NEXT	2-96
ON	2-98
OPEN	2-99
PURGE	2-100
READ	2-102
REBOOT	2-104
RECALL	2-105
RENAME	2-107
REPEAT	2-109
RESET	2-110
RESTORE	2-111
RETURN	2-112
RUN	2-113
SAVE	2-115
SCROLL	2-116
SELECT	2-118
SEND	2-120
SENSE	2-121
SENSE CLOCK	2-122
SENSE CPU	2-123
SENSE IO	2-124
SENSE POWER	2-125
SENSE SCU	2-126
SENSE SYSTEM	2-127
SET	2-128
SET ATTN_ACTION	2-129
SET AUTOBOOT	2-130
SET BI_DEVICES	2-131
SET BOOTFLAGS	2-132
SET BOOTSET	2-133
SET CLOCK	2-135
SET COLD_START	2-139
SET COMMAND	2-140
SET CPU	2-141
SET CYCLE	2-142

SET DEFAULT	2-144
SET ERROR_HANDLING	2-145
SET FAULT_ACTION	2-146
SET ISOLATION	2-147
SET KEEP_ALIVE	2-148
SET LABELS	2-149
SET LOGGING	2-150
SET MESSAGE	2-152
SET PATTERN	2-154
SET PERSONAL_NAME	2-157
SET POWER	2-158
SET PROMPT	2-161
SET RADIX	2-162
SET REMOTE	2-163
SET REVISION	2-164
SET SCI	2-165
SET SCM	2-169
SET SCOPE	2-171
SET SCREEN	2-172
SET SERIAL	2-173
SET SNAPSHOT	2-175
SET SOURCE	2-177
SET STEP	2-178
SET TERMINAL	2-179
SET TIME	2-182
SET TRACE	2-183
SET VERIFY	2-187
SET WARM_START	2-188
SET WATCH	2-189
SET XMI_DEVICES	2-192
SHOW	2-193
SHOW ATTN_ACTION	2-194
SHOW AUTOBOOT	2-195
SHOW AVAILABLE	2-196
SHOW BATCH	2-197
SHOW BBU	2-198

SHOW BI_DEVICES	2-199
SHOW BOOTFLAGS	2-200
SHOW BOOTSET	2-201
SHOW CLOCK	2-202
SHOW CONFIGURATION	2-203
SHOW CPU	2-207
SHOW CYCLE	2-209
SHOW DEFAULT	2-210
SHOW DEVICE	2-211
SHOW ENVIRONMENT	2-213
SHOW ERROR_HANDLING	2-214
SHOW FAULT_ACTION	2-215
SHOW FLAGS	2-216
SHOW HISTORY	2-217
SHOW ISOLATION	2-219
SHOW KEEP_ALIVE	2-220
SHOW KEY	2-221
SHOW LOGGING	2-223
SHOW LOGICAL	2-224
SHOW MEMORY	2-225
SHOW MESSAGE	2-226
SHOW MODE	2-227
SHOW NODE	2-228
SHOW PATTERN	2-229
SHOW PERSONAL	2-231
SHOW POWER	2-232
SHOW PROCESS	2-235
SHOW RADIX	2-236
SHOW REMOTE	2-237
SHOW SCI	2-238
SHOW SCM	2-239
SHOW SCOPE	2-243
SHOW SCU	2-244
SHOW SJA	2-245
SHOW SOURCE	2-246
SHOW STEP	2-247

SHOW STRUCTURE	2-248
SHOW SWITCHES	2-250
SHOW SYMBOL	2-251
SHOW SYSTEM	2-252
SHOW TERMINAL	2-253
SHOW THRESHOLD	2-255
SHOW TIME	2-256
SHOW TRACE	2-257
SHOW USERS	2-258
SHOW VERSION	2-259
SHOW WATCH	2-261
SHOW WINDOW	2-262
SHOW XMI_DEVICES	2-263
SHOW ZONE	2-264
START	2-265
STOP	2-267
SUBMIT	2-268
TALK	2-269
TYPE	2-270
UNJAM	2-272
VERIFY	2-273
WAIT	2-275
WRITE	2-276
Z	2-277

3 Lexical Function Description

ASCII	3-2
BITVECTOR	3-3
CLOCK	3-4
CONFIG	3-5
CPU	3-6
EXTRACT	3-8
FIELD	3-9
FILL	3-10
INFORMATION	3-11
INTEGER	3-13

LENGTH	3-14
LOCATE	3-15
LOGICAL	3-16
MCU	3-17
MEMORY	3-18
PARSE	3-19
PART	3-21
POWER	3-22
RADIX	3-23
REVISION	3-24
SCI	3-25
SCU	3-26
SEARCH	3-27
SERIAL	3-28
SID	3-29
SIGNAL	3-30
STRING	3-31
SWITCH	3-32
TIME	3-33
UPC	3-34
VERIFY	3-35

A Command Quick Reference

B Lexical Function Quick Reference

Index

Examples

1-1	Console Command Format	1-1
1-2	COPY Command Format	1-2
1-3	SET Command Format	1-3
1-4	In-Line Help	1-6
1-5	Message Format	1-9
1-6	Expression Usage	1-11

Tables

1-1	CPU Qualifier Arguments	1-4
1-2	Confirmation Responses	1-5
1-3	Predefined Keys	1-7
1-4	User-Defined Keys	1-8
2-1	Window Location Keywords	2-22
2-2	MCM Register Mnemonics	2-45
2-3	PEM Register Mnemonics	2-45
2-4	PEM Address Space Qualifiers	2-46
2-5	RIC Register Mnemonics	2-47
2-6	RIC Address Space Qualifiers	2-47
2-7	SCC Register Mnemonics	2-47
2-8	SJA Register Mnemonics	2-48
2-9	DEPOSIT Special Operators	2-50
2-10	GPR and PSL Register Mnemonics	2-50
2-11	IPR Register Mnemonics	2-51
2-12	EXAMINE Special Operators	2-70
2-13	System Power Buses	2-158
2-14	System I/O Buses	2-159
2-15	System Cabinets	2-159
2-16	Structure Names	2-248
2-17	Operator Control Panel Switches	2-250

About This Manual

This manual describes the commands implemented by the VAX 9000 family service processor unit (SPU), commonly called the console, in console I/O (CIO) mode. Most of the information in this manual is also available on-line (in CIO mode) from the service processor operating system HELP library, through the console HELP command.

Intended Audience

This manual is written for system operators, Digital Customer Services personnel, and other console interface users. This manual assumes the reader is familiar with the VAX 9000 family architecture and the Digital command language (DCL).

Manual Structure

This manual has three chapters, two appendixes, and an index.

- Chapter 1 provides generic information about the console command language.
- Chapter 2 describes the console commands.
- Chapter 3 describes the lexical functions.
- Appendix A is an alphabetical quick reference list of console commands.
- Appendix B is an alphabetical quick reference list of lexical functions.

Manual Conventions

This manual uses the following conventions:

Convention	Meaning
COMMAND argument	In command description format sections, command verbs are uppercase; user-supplied arguments (qualifiers, object parameters, and so on) are lowercase.
COMMAND arg1 COMMAND arg2 COMMAND arg3	Stacked arguments indicate several different argument formats. Only one argument is to be entered.
COMMAND [argument] COMMAND [arg[, ...]]	Brackets ([]) enclose optional arguments. The , ... (comma,ellipsis) indicates a list of similar arguments.
COMMAND [arg1[, ... argn]]	Indicates a limited list, form 1 through n, of similar arguments.
COMMAND {arg1 arg2} COMMAND [arg1 arg2]	The vertical bar () separates argument choices; only one choice is to be entered. Braces ({}) enclose a required argument; brackets ([]) enclose an optional argument.
LEXICAL (argument)	Lexical function arguments, when used, are enclosed in parentheses (()).
LEXICAL [(argument)]	The lexical function argument is optional.
LEXICAL ([argument])	One of several lexical function arguments is required.
(D)	Default.
[mm:nn]	Indicates the field or set, <i>mm</i> through <i>nn</i> . Replaces <mm:nn>.

1

Console Command Language

The console, or SPU, command language merges the standard VAX console command language with basic file-system commands to access the entire SPU operating system from one level. Unlike previous VAX systems, the VAX 9000 family console command language is not implemented as a program running under another operating system.

The console command language resembles the Digital command language (DCL) in its use of lexical functions, command files, symbols, and logical names. While this provides a high degree of familiarity for the majority of VAX 9000 family users, it should be noted that the two languages are not identical.

1.1 Command Syntax

Example 1-1 shows the general format for console commands.

```
>>> VERB [ /qualifiers ] object [ parameters ] Return
```

① ② ③ ④ ⑤

Example 1-1 Console Command Format

- ① The verb is the command name and indicates the command action.
- ② Qualifiers modify the command action and always begin with a slash (/). Qualifiers are usually optional, as indicated by enclosing them in brackets ([]).
- ③ The object parameter receives the command action. The object is a required parameter unless a default object parameter is defined for the command.
- ④ Optional parameters (enclosed in brackets) provide additional command modifiers.
- ⑤ Most commands are not executed until Return is pressed. (Examples of command usage do not show the Return key symbol.)

1-2 Console Command Language

As Examples 1-2 and 1-3 show, the general command format is usually modified to fit specific command requirements.

Example 1-2 shows the COPY command format and a typical COPY command.

```
COPY [/qualifiers] input-file-spec output-file-spec
>>> COPY /NOCONFIRM INPUT.DAT OUTPUT.DAT
      ①      ②      ③      ④
```

Example 1-2 COPY Command Format

- ① COPY is the command verb.
- ② /NOCONFIRM is an optional qualifier specifying that the file is copied as soon as the command is entered; that is, without confirmation (Section 1.1.1.4).
- ③ INPUT.DAT is the first of two required object parameters, and specifies the file to be copied.
- ④ OUTPUT.DAT is the second of two required object parameters, and specifies the duplicate file. There are no optional parameters for this command.

Example 1-3 shows the SET command format and a typical SET command. Note that in SET and SHOW commands, the qualifier follows the object parameter.

```

    SET object [/qualifiers] [parameters]
>>> SET BOOTSET /PRIMARY=1 0,1,2
    ①   ②       ③       ④

```

Example 1-3 SET Command Format

- ① SET is the command verb.
- ② BOOTSET is the object parameter.
- ③ /PRIMARY=1 is the optional qualifier specifying CPU1 as the primary CPU in the boot set.
- ④ The 0, 1, and 2 are required and optional parameters that specify the boot set includes CPU0, CPU1, and CPU2.

1.1.1 Command Qualifiers

Sections 1.1.1.1 through 1.1.1.4 describe several qualifier attributes that apply to many of the console commands.

1.1.1.1 Qualifier Entry

Command qualifiers are always prefixed with a slash (/), as follows:

```
COMMAND /qualifier
```

Most commands allow qualifiers to be entered in any sequence following the command verb and before or after the object and optional parameters. Note that the following command/qualifier sequences are fixed but can be followed by additional parameters:

```

CREATE/DIRECTORY
CREATE/WINDOW
DEFINE/KEY
DELETE/PATTERN
DELETE/SYMBOL
DELETE/TRACE
DELETE/WATCH
DELETE/WINDOW

```

Additionally, some commands (such as LOAD, SHOW STRUCTURE, TEST, and VERIFY) have two sets of qualifiers.

The format section of each command description (Chapter 2) indicates the position of qualifiers and other required and optional command elements.

1.1.1.2 Qualifier Values

Many qualifiers have optional, required, and/or default values. A qualifier value is specified by following the qualifier with a colon (:) or equals (=) character and then the value (Example 1-3). Depending on the command and/or qualifier, the value can be a quoted string, keyword, or numeric value.

1.1.1.3 CPU and SCU Qualifier

Many commands accept a */CPU=cpu-id* qualifier to specify a CPU or CPUs other than the current default. The argument, *cpu-id*, can be an integer value or one of the symbolic names listed in Table 1-1. Generally, if the qualifier is not used (that is, a CPU is not specified), the default CPU is selected. (To specify a CPU as the default, see the SET CPU command description in Chapter 2.)

Table 1-1 CPU Qualifier Arguments

Argument	Description
0, 1, 2, or 3	An integer value. Some commands accept a list; for example: <i>/CPU=(0,1)</i> .
ALL	All CPUs. In some commands <i>/CPU=ALL</i> is similar to <i>/CPU=(0,1,2,3)</i> except that ALL does not detect an error if a CPU does not exist.
AVAILABLE	All CPUs in the available set. CPUs in the available set are permanently allocated to the operating system and cannot be modified by the operator while the operating system is executing. The available set is copied from the boot set when the system is booted.
BOOTPRIMARY	The CPU that will be the primary CPU when the system is next bootstrapped.
BOOTSET	All CPUs in the boot set. The boot set is all CPUs that have passed self-test and are enabled. The boot set CPUs are made available to the operating system to execute in the symmetrical multiprocessing (SMP) environment.
PRIMARY	The first CPU to boot in an SMP environment. The primary CPU controls the booting of the other CPUs in the boot set.

Many of the same commands that accept the CPU qualifier also accept the SCU qualifier. Generally, these commands treat the SCU qualifier as if it were another argument to the CPU qualifier. That is, the command has the same affect on the SCU that it has on the specified CPU.

1.1.1.4 Confirmation Qualifier

Many commands accept the */CONFIRM* qualifier to specify selectively whether a command is to be executed. For example, this qualifier is frequently used in file operations where defaults or wildcard characters are used to specify the same operation for more than one file. Specifying confirmation causes the command to display a prompt requesting an affirmative response for each file to be processed.

The valid responses to the confirmation prompt are listed in Table 1-2.

Table 1-2 Confirmation Responses

Response	Result
Y (yes)	Execute the command.
T (true)	Execute the command.
N (no)	Do not execute the command.
F (false)	Do not execute the command.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

1.1.1.5 /ALL Qualifiers

In most cases, commands that offer an */ALL* qualifier also offer one or more specific object parameters. The two are mutually exclusive; that is, the */ALL* qualifier and a specific object parameter cannot be specified in the same command line.

1.2 In-Line Help

In addition to the *HELP* command, the SPU software provides multilevel in-line help for console commands. As Example 1-4 shows, typing a question mark (?) at some point in the console command line invokes the in-line help facility for that level.

1-6 Console Command Language

>>> ? ❶ Command, one of the following:

ALLOCATE	BOOT	CLOSE	CONTINUE	COPY
CREATE	DEALLOCATE	DEASSIGN	DEBUG	DEFINE
DELETE	DEPOSIT	DIRECTORY	DISMOUNT	EDIT
EVALUATE	EXAMINE	EXIT	FIND	HALT
HELP	IF	INITIALIZE	INQUIRE	LOAD
LOGOUT	MAIL	MICROSTEP	MOUNT	NEXT
OPEN	PURGE	READ	REBOOT	RECALL
RENAME	REPEAT	RESET	RESTORE	RUN
SAVE	SCROLL	SELECT	SEND	SENSE
SET	SHOW	START	STOP	SUBMIT
TALK	TEST	TYPE	UNJAM	VERIFY
WAIT	WRITE	X	Z	

or External command, one of the following:

ERF	DUMP	DIFFERENCES
-----	------	-------------

or Symbol name

or Command file identifier, "@"

>>> SET ? ❷ Parameter, one of the following:

ATTN_ACTION	AUTOBOOT	BI_DEVICES	BOOTFLAGS	BOOTSET
CLOCK	COLD_START	COMMAND	CPU	CYCLE
DEFAULT	ERROR_HANDLING		FAULT_ACTION	ISOLATION
KEEP_ALIVE	LABELS	LOGGING	MESSAGE	PATTERN
PERSONAL_NAME		POWER	PROMPT	RADIX
REMOTE	REVISION	SCOPE	SCM	SCI
SCREEN	SERIAL	SJA	SOURCE	SPU_UPDATE
SNAPSHOT	STEP	TERMINAL	TIME	TRACE
VERIFY	WARM_START	WATCH	XMI_DEVICES	XMI_UPDATE

>>> SET TERMINAL ? ❸ Terminal qualifier, one of the following:

/[NO]BROADCAST	/CPU:	/DEVICE:	/[NO]ECHO
/[NO]EIGHTBIT	/[NO]ESCAPE	/KEYPAD:	/PAGE:
/[NO]PROGRAM /PIO_PORT:	/[NO]TALK_MODE		/WIDTH:

or Terminal name,

or confirm with carriage return

>>> set terminal /keypad:? ❹ Option, one of the following:

APPLICATION	NUMERIC
-------------	---------

Example 1-4 In-Line Help

- ❶ Typing a question mark in response to the console prompt displays a list of console commands.
- ❷ Typing a question mark after a **SET** command displays a list of SET command parameters or options.
- ❸ Typing a question mark after a SET command **parameter** displays a list of command qualifiers.
- ❹ Typing a question mark after a SET command parameter **qualifier** displays a list of qualifier options.

1.3 Special Keys

The console command language interprets LK201 keyboard keys for command recall, command editing, and special functions.

1.3.1 Predefined Keys

Table 1-3 lists the keys defined by the console software.

Table 1-3 Predefined Keys

Keys	Action
Ctrl/A	Switch between insert and overstrike edit mode.
Ctrl/B	Display previous command in recall buffer. Same as Up Arrow.
Ctrl/C	Cancel command processing. Same as Ctrl/Y.
Ctrl/D	Move cursor one character to the left. Same as Left Arrow.
Ctrl/E	Move cursor to end of line.
Ctrl/F	Move cursor one character to the right. Same as Right Arrow.
Ctrl/H	Move cursor to beginning of line.
Ctrl/Q	Resume screen output.
Ctrl/S	Hold screen output.
Ctrl/U	Delete line.
Ctrl/W	Redraw the screen.
Ctrl/Y	Cancel command processing. Same as Ctrl/C.
Ctrl/Z	End of file.
Do	Same as Return key.
Help	Invokes HELP utility.
?	Invokes in-line help. See Section 1.2.
Down Arrow	Display next command in recall buffer.
Left Arrow	Move cursor one character to the left. Same as Ctrl/D.
Right Arrow	Move cursor one character to the right. Same as Ctrl/F.
Up Arrow	Display previous command in recall buffer. Same as Ctrl/B.

1.3.2 User-Defined Keys

In addition to the predefined keys, the user can define function keys, editing keypad keys (except arrows), and numeric keypad keys. To implement user-defined keypad keys, set the keypad mode to application. Table 1-4 lists the keys that the user can define. See the DEFINE/KEY and SET TERMINAL/KEYPAD command descriptions in Chapter 2 for more information.

Table 1-4 User-Defined Keys

Key Name	Key ¹
Editing Keypad	
E1	Find
E2	Insert Here
E3	Remove
E4	Select
E5	Prev Screen
E6	Next Screen
Function Keys	
F6-F14	F6-F14
Do	F15
Help	F16
F17-F20	F17-F20
Numeric Keypad	
PF1-PF4	PF1-PF4
KP0-KP9	0-9
PERIOD	.
COMMA	,
MINUS	-
ENTER	Enter

¹To implement user-defined keypad keys, set the keypad mode to application.

1.4 Message Format

The console command language uses the DCL format for messages. All messages begin with a percent character (%) in column 1. If the message continues to more than one line, the continuation line(s) begin with a minus character (-) in column 1. Example 1-5 shows the message format.

```
%Facility-Severity-Ident, Text
-Facility-Severity-Ident, Text
```

① ② ③ ④

Example 1-5 Message Format

- ① Facility is a 3-character identification code for the SPU operating-system subsystem that produced the message. For example, CLI is the code for the command language interpreter.
- ② Severity is a 1-character code for the message severity, as follows:
 - S = Success
 - I = Informational
 - W = Warning
 - E = Error
 - F = Fatal error
- ③ Ident is a variable-length, message-unique field. It is usually an acronym for the message.
- ④ Text is a brief error description and possible remedy.

1.5 The Radix of Numbers

The console command language uses numbers for VAX data, loop counts, file versions, frequencies, and so on. Normally, the radix is hexadecimal for all VAX data and numbers related to VAX data; and the radix is decimal for all file-related numbers, test numbers, bit numbers, and so on.

The user can specify the radix for all VAX data and numbers related to VAX data. All numbers that are not hexadecimal and do not have a fixed radix are displayed with a radix identifier prefixed to the number. For example:

Decimal numbers are displayed as %D123.
 Octal numbers are displayed as %O173.
 Binary numbers are displayed as %B1111011.

More specifically, the following data types are decimal values:

- Loop count (except /NEXT) and cycle count
- Data size
- Bit number
- Time value
- Voltage, current, and frequency values
- Test number
- File system numbers (generation number, count, keep count)

The following data types are represented in the user-specified radix:

- All VAX data and addresses
- EVALUATE expressions
- CPU ID
- RIC ID
- XMI node and BI node IDs

All numbers input, except file generation numbers, accept the following radix specifiers:

- %D (decimal)
- %B (binary)
- %O (octal)
- %X (hexadecimal)

For example:

```
>>> DEPOSIT R0 %D10
```

deposits 10₁₀ in VAX register R0.

NOTE

Decimal quantities are limited to a value that can be stored in 32 bits. In other words, if the radix is specified as %D, the value cannot exceed 2147483647 (signed) or 4294967295 (unsigned).

1.6 Expressions

The parameters of many commands are defined as an expression. This section describes the kinds of expressions and expression operators, including lexical functions.

1.6.1 Types of expressions

- **Numeric expressions**

Numeric expressions can have any elements that evaluate to a numeric result.

- **String expressions**

String expressions can have any elements that yield a string result.

- **Mixed expressions**

Mixed expressions can have either numeric expressions, string expressions, or both.

- **Lexical functions**

The command language interpreter (CLI) also provides lexical functions that give string or numeric results (for example, the contents of a register) based on a predetermined operation. Lexical functions are used primarily in command files to allow symbol assignment to CPU data. For a description of the lexical functions, see Chapter 3.

- **Arithmetic operators**

The supported arithmetic expression operators are:

+ - / *

- **Relational operators**

The following relational integer operators yield 0 or 1 to reflect false and true expressions:

.EQ. .NE. .GE. .LE. .GT. .LT.

The following relational string operators yield 0 or 1 to reflect false and true expressions:

.EQS. .NES. .GES. .LES. .GTS. .LTS.

- **Boolean operators**

Boolean operators .AND. and .OR. are provided for data manipulation. They are not relational operators; their precedence is higher than relational operators. The unary operator .NOT. is also provided.

Example 1-6 gives several examples of how expressions are used.

```
DEPOSIT R0 R1
EXAMINE/NEXT:10 @
EVALUATE %X100 + (%X45 * 4)
SAVE_R0 = R0
START @PC
REPEAT IF SIGNAL("UPC") .NE. %X1B00 THEN MICROSTEP/NOSPACEBAR 1
IF P1_.EQS._ "" THEN INQUIRE P1 "_What"
```

Example 1-6 Expression Usage

1.6.2 Data Types

The console command language supports three basic data types: integer, bitvector, and string. Some commands automatically convert between data types, for example:

```
>>> WRITE STDOUT XYZ
```

converts `XYZ` to a string data type before it is output.

- Integer data type

An integer data type is limited to 32 bits in length and is a signed value. The range of the value is from `-2147483648` to `2147483647`. Integers can be used in expressions and all operators accept integer data types.

Integers can be converted to strings by binary to ASCII translation, and to bitvectors by truncation or zero fill, depending on the bitvector size.

- Bitvector data type

A bitvector is a 1- to 2048-bit data type. It is used to store large numbers (scan rings, control store words, and so on). Bitvectors can be used as an expression but cannot be used as operands of an expression; that is, the value is accepted and passed, but no operations are performed on the value. Bitvectors can be manipulated with lexical functions.

Bitvectors can be converted to strings by binary to ASCII translation, and to integers by truncation or zero fill, depending on the bitvector size.

- String data type

A string is a 0- to 255-character ASCII string. Strings can be used in expressions and can be mixed with integers.

Strings can be converted to bitvectors or integers by ASCII to binary translation.

2

Console Command Descriptions

The command descriptions in this chapter are nearly identical to the descriptions in the service processor operating system HELP library. In most cases, the two are distinguished only by minor formatting differences to accommodate the different media.

ALLOCATE

Locks the specified unit so that only this process can use the unit. This allows a single process to ensure that no other operations are performed on the specified unit.

Format

ALLOCATE *object*

Parameters

object

Specifies one of the following:

- CPU[id] — Allocate the specified CPU; if a CPU is not specified, the default CPU is allocated. See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.
- MCM — Allocate the clock subsystem.
- PCS — Allocate the power subsystem.
- SCU — Allocate the system control unit, memory, and I/O.

Example

```
>>> ALLOCATE MCM
```

Allocates the clock subsystem to the current process.

:= (Assign String)

Assigns a symbolic name to a character string.

Format

symbol-name :=[=] *string*

Parameters

symbol-name

The symbol name can contain between 1 and 255 characters. It must begin with an alphabetic character, underscore (_), or dollar sign (\$). The remaining characters can be any alphanumeric character from the Digital multinational character set, underscores, or dollar signs.

A single equal sign (:=) in the statement places the symbol in the local symbol table. Double equal signs (:=) place the symbol in the global symbol table. The global symbol table is available at the interactive command level and to command procedure files. Command procedures (see HELP for the @ command) also have unique local symbol tables.

string

The string can be specified as a string literal consisting of any alphanumeric or special characters, or as a symbol or lexical function that evaluates to a string literal.

The := string assignment statement automatically converts lowercase text to uppercase, removes leading and trailing spaces and tabs, and compresses multiple spaces and tabs to a single space.

String literals need not be enclosed in quotation marks ("). However, to preserve case and retain multiple spaces and tabs, place quotation marks around the string. To use quotation marks in a string, enclose the string in quotation marks and use a double set of quotation marks within the string. For example:

```
>>> TEST := "this is a ""test"" string"
>>> SHOW SYMBOL TEST
TEST = "this is a "test" string"
```

To define a symbol as a null string, do not specify a string. For example:

```
>>> NULL :=
```

2-4 Console Commands

`:=` (Assign String)

To request substitution of a symbol or lexical function, place apostrophes (') around the item. For example:

```
>>> A := "Quoted string"
>>> B := 'A'
>>> SHOW SYMBOL B
B = "Quoted string"
```

Examples

1 >>> TIME := SHOW TIME
>>> TIME
10-OCT-1988 16:35:44

The string `SHOW TIME` is assigned to symbol `TIME`. Because the symbol is the first word on the command line, the command interpreter substitutes its string value and executes the command `SHOW TIME`.

2 >>> STAT := @DISK\$HARD:[CONSOLE]STAT.CMD
>>> STAT

This example shows how to define a symbol as a foreign command. The symbol `STAT` is equated to a string that begins with the at sign (@) character (execute command procedure) followed by a file specification. The symbol `STAT` can now be used as a synonym for the command:

```
>>> @DISK$HARD:[CONSOLE]STAT.CMD
```

When `STAT` is subsequently entered on the command line, the command interpreter executes the command procedure file.

3 >>> A := "this is a big space."
>>> SHOW SYMBOL A
A = "this is a big space."
>>> B := 'A'
>>> SHOW SYMBOL B
B = "this is a big space."

This example shows how to request symbol substitution with the string assignment statement. The apostrophes in the definition for symbol `B` cause the symbol to take on the value assigned to symbol `A` instead of the literal string `A`.

= (Assign Symbol)

Assigns a symbolic name to a character string or integer value.

Format

symbol-name **=[=]** *expression*

Parameters

symbol-name

The symbol name can contain between 1 and 255 characters. It must begin with an alphabetic character, underscore (`_`), or dollar sign (`$`). The remaining characters can be any alphanumeric character from the Digital multinational character set, underscores, or dollar signs.

A single equal sign (`=`) in the statement places the symbol in the local symbol table. Double equal signs (`==`) place the symbol in the global symbol table. The global symbol table is available at the interactive command level and to command procedure files. Command procedures (see `HELP` for the `@` command) also have unique local symbol tables.

expression

The expression can be a character string, integer value, symbol, lexical function, or combination of entities.

If the expression evaluates to a string, the symbol is assigned a string value. If the expression evaluates to an integer, the symbol is assigned an integer value. If a symbol is specified in an expression, the value of the symbol is used in evaluating the expression.

To specify a literal character string in an expression, enclose the string in quotation marks (`"`).

2-6 Console Commands

= (Assign Symbol)

Examples

```
1 >>> LIST == "DIRECTORY"
>>> LIST
Directory DISK$HARD:[CONSOLE]

CPU0.CMD;1          CPU1.CMD;1          CPU2.CMD;1          CPU3.CMD;1
DEBUG.CMD;2        LOGIN.CMD;3          Z.CMD;2             Z.CMD;1
Total of 8 files.
```

The assignment statement defines the symbol `LIST` as the character string `DIRECTORY` and places the symbol in the global symbol table. When `LIST` is entered on the command line, the symbol's definition is retrieved from the global symbol table and the command `DIRECTORY` is executed.

```
2 >>> COUNT = 0
>>> LOOP:
>>> COUNT = COUNT + 1
.
.
.
>>> IF COUNT .LT. 5 THEN GOTO LOOP
```

The symbol `COUNT` is defined in a command procedure. The procedure initializes, increments, and tests the value of `COUNT` to control iterations through the code loop. If `COUNT` is less than 5, the `IF` statement causes a branch to label `LOOP`; otherwise, the procedure exits the loop and executes the command following the `IF` statement.

```
3 >>> A = 25
>>> CODE = 4 + INTEGER("6") - 'A'
>>> SHOW SYMBOL CODE
CODE = -15    HEX = FFFFFFF1    Octal = 1777761
```

Two symbols are defined for a mixed arithmetic expression. First, a value of 25 is assigned to the symbol `'A'`. The second statement evaluates an expression containing the integer 4, the lexical function `INTEGER("6")`, and the symbol `'A'`; the result, -15, is assigned to the symbol `CODE`.

NOTE

Assuming the default radix is hexadecimal, symbol `A` must be enclosed in quotes to distinguish it from `A16`.

@ (Execute Procedure)

Executes commands from a command procedure file. Some commands are valid *only* in a command procedure. These are:

CALL GOTO LABEL ON RETURN

Format

@ file-spec [*p1* [*p2* [... *p8*]]]

Parameters

file-spec

The command procedure file to be executed. The default file type is CMD and the default directory is the current directory. Wildcard characters are not allowed in the file-spec.

Command procedures can contain any console command and can be nested to a depth of eight.

[*p1* [*p2* ... [*p8*]]]

Up to eight parameters can be passed to a command procedure. The parameters assign character string values to the symbols named P1 through P8 that are local to the command procedure. Command procedures also have access to the global symbol table.

Separate multiple parameters with spaces. Specify a null parameter with quotation marks (""). Specify a parameter with a character string value containing alphanumeric or special characters using the following restrictions:

- The command interpreter converts lowercase letters to uppercase and delimits parameters with spaces. To preserve lowercase and embedded spaces, enclose the parameter in quotation marks.
- To pass a parameter containing literal quotation marks and spaces, enclose the entire string in quotation marks and use a double set of quotation marks within the string. For example:

```
>>> @TEST1 "Spaced and ""quoted string"""
```

When TEST1.CMD executes, parameter P1 is equated to the string:

```
Spaced and "quoted string"
```

2-8 Console Commands

@ (Execute Procedure)

- If a string contains quotation marks but not spaces, the quotation marks are preserved in the string and the letters within the quotation marks remain in lowercase. For example:

```
>>> @TEST2 abc"def"ghi
```

When TEST2.CMD executes, parameter P1 is equated to the string:

```
ABC"def"GHI
```

- To use a symbol as a parameter, enclose the symbol in apostrophes to force symbol substitution. For example:

```
>>> NAME = "JOHNSON"  
>>> @INFO 'NAME'
```

The symbol NAME is replaced with value "JOHNSON" and the parameter "JOHNSON" is passed as P1 to INFO.CMD.

Examples

```
1 >>> SET VERIFY  
>>> @SYSINIT
```

Executes the system initialization procedure SYSINIT.CMD.

```
2 >>> CREATE EXAMINE.CMD  
DEPOSIT/PHYSICAL/LONGWORD/NEXT=20 200 0  
EXAMINE/PHYSICAL/LONGWORD/NEXT=20 200 Ctrl/Z  
^Z  
>>> @EXAMINE
```

This procedure deposits zeros into 20 consecutive longwords starting at physical address 200 and then examines the longwords.

BOOT

Executes a command procedure that loads, and optionally starts, operating system software.

Format

BOOT *[/qualifiers] [device]*

Qualifiers

/BI=node-id

Specifies the boot device controller's BI node number. The number is loaded into GPR R0 bits [3:0].

/NODE=node-id

The XMI node of the boot device, where node-id is a hexadecimal number.

/PIO_MODE(D)

/NOPIO_MODE

Specifies whether the process enters program I/O (PIO) mode after starting the primary bootstrap.

/R3=register-data

Register-data is additional boot data and flags expressed as a hexadecimal value.

/R5=boot-flags

The numeric value boot-flags sets bootstrap control flags in GPR R5. The flags can be specified with this qualifier or with a DEPOSIT R5 command in the boot procedure. If the /R5 qualifier is to be used, the DEPOSIT R5 command in the boot procedure must be commented out. When this qualifier specifies the flags, they are loaded before the boot procedure is executed. The flags provide additional boot sequence information to the primary bootstrap loader VMB.EXE.

/START (D)

/NOSTART

Specifies whether the processor starts at the completion of the boot procedure. When /NOSTART is specified, the GPRs and VMB.EXE are loaded, but the procedure returns to the console prompt instead of issuing the START command.

2-10 Console Commands

BOOT

/XMI=xmi-id

Specifies the XMI node number (hexadecimal) of the boot device controller or of the BI adapter that maps the boot device's BI controller. The value is loaded into GPR R0 bits [9:4], where [9:8] specify the XJA number and [7:4] specify the node.

Parameters

device

Specifies the device and unit number from which the boot procedure is to be executed. If the parameter is not specified, the default boot procedure DEFBOO.CMD is executed.

The device parameter format is:

dddnnn

where *ddd* is a device name that invokes the command procedure *ddd*BOO.CMD and *nnn* is a unit number that is loaded into GPR R3.

Examples

1 >>> BOOT

Executes the default boot procedure DEFBOO.CMD.

2 >>> B XCI2

Executes XCIBOO.CMD and specifies system boot device unit 2.

3 >>> B/R5=1

Executes DEFBOO.CMD and specifies a conversational boot sequence.

CALL

Transfers control to a labeled subroutine in a command procedure and creates a new procedure level. The next instruction of the calling routine is saved on a stack for return.

Format

CALL *label*

Parameters

label:

The first item on a command line. A label cannot contain embedded blanks. When the CALL command is executed, control passes to the command following the label. The label can precede or follow the CALL statement in the command procedure.

All labels are procedure-level dependent except for labels that define subroutine entry points. Subroutine entry point labels are local to the current command procedure file level and must be unique.

CLOSE

Closes a file that was opened with the OPEN command and deassigns the logical name specified with the OPEN command.

Format

CLOSE *logical-name*

Parameters

logical-name

Specifies the logical name assigned to the open file by the OPEN command.

Examples

```
1  START:
    OPEN/READ FILE TEST.DAT
    LOOP:
      READ/END_OF_FILE=DONE FILE LINE
      .
      .
      .
    GOTO LOOP
    DONE:
      CLOSE FILE
```

The OPEN command opens file TEST.DAT and gives it the logical name FILE. The READ command /END_OF_FILE qualifier transfers control to the line at label DONE when the end of the file is reached. The CLOSE command closes the input file.

```
2 >>> @READFILE
  >>> SHOW LOGICAL/PROCESS
      .
      .
      .
  "INFILE" = "TEST.DAT"
  "OUTFILE" = "NEWTEST.DAT"
  >>> CLOSE INFILE
  >>> CLOSE OUTFILE
```

Command procedure READFILE.CMD is executed. SHOW LOGICAL/PROCESS displays the logical names in the process logical name table, including INFILE and OUTFILE that were assigned by OPEN commands in READFILE.CMD. The CLOSE commands close these files and deassign the logical names.

CONTINUE

Resumes macrocode execution in the default or specified CPU.

Format

CONTINUE [*qualifier*]

Qualifiers

/CPU=cpu-id

The CPU to be continued, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

A CPU list can also be specified, for example: */CPU=(0,1,2,3)*.
CONTINUE/*CPU=ALL* is similar to CONTINUE/*CPU=(0,1,2,3)* except that ALL does not detect an error if a CPU does not exist. If a CPU is not specified, the default CPU is continued.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

/NOLOG

Determines whether to display command results on the terminal.

/PIO_MODE (D)

/NOPIO_MODE

If */NOPIO_MODE* is not specified, the SPU enters program I/O (PIO) mode for the continued processor.

/PIO_PORT={OPA0|OPA1}

Specifies the port to which the terminal is temporarily connected in PIO mode.

Description

When CONTINUE is executed, CPU macrocode execution resumes at the current PC, CPU state changes from HALTED to RUNNING, and, if the default CPU is continued, the console is placed in program I/O (PIO) mode. Characters typed in PIO mode are always sent to the primary console communication register set regardless to which CPU the console is attached. The CPU is not initialized.

Typing Ctrl/P returns the console to console I/O (CIO) mode. If the continued CPU is not the default, the console is not placed in PIO mode.

Examples

1 >>> CONTINUE

Resume program execution in the default CPU and enter PIO mode.

2 >>> CONTINUE/CPU=PRIMARY

Resume program execution in the primary CPU. Enter PIO mode only if the primary CPU is also the default CPU.

3 >>> CONTINUE/CPU=1

Resume program execution in CPU1. Enter PIO mode only if CPU1 is also the default CPU.

COPY

Creates a new file from an existing file, or a group of files from a group of existing files.

Format

COPY *[/qualifiers] input-file-spec output-file-spec*

Qualifiers

/CONFIRM

/NOCONFIRM (D)

Determines whether an affirmative response is required before a file is copied. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	The file is copied.
T (true)	The file is copied.
N (no)	The file is not copied.
F (false)	The file is not copied.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

/CONTIGUOUS

Specifies that the new file is to be contiguous on disk.

/LOG

/NOLOG (D)

/LOG displays the complete input and output file specifications and number of blocks for each file that is copied.

Parameters

input-file-spec

The file or files to be copied. The default device and directory are used unless otherwise specified.

An input file list cannot be specified; however, the asterisk (*) wildcard character can be used in place of any whole or partial field (file name, type, or version) in the file specification.

If the input file specification is omitted, the `From:` prompt is displayed.

output-file-spec

Specifies the name of the file or files to be created. The default device and directory are used unless otherwise specified. If the created file has the same name as an existing file, the created file is given the next higher version number.

The output file specification must include at least one field (file name, type, or version); the corresponding input file field replaces any missing fields.

The asterisk (*) wildcard character can be used in place of any whole field in the output file specification. The field is replaced by the corresponding field from the input file specification.

If the output file specification is omitted, the `To:` prompt is displayed.

Examples

```
❶ >>> COPY
    From: TEST.DAT
    To:   NEWTEST.DAT
```

Copies file `TEST.DAT` to file `NEWTTEST.DAT`, both in the current directory.

```
❷ >>> COPY TEST.TXT TMP
    >>> COPY TEST.TXT .TMP
```

Copy file `TEST.TXT` into files named `TMP.TXT` and `TEST.TMP`. The missing output file fields are duplicated from the corresponding input file fields.

```
❸ >>> COPY A*.* [SAVE]*.*
```

Copies all files beginning with `A` from the default directory to a group of files with the same file names in directory `[SAVE]`.

CREATE

Creates a file from text entered at the terminal. Typing Ctrl/Z closes the file and terminates the command.¹

Format

CREATE *[[/NO]LOG] file-spec*

Qualifier

/LOG

/NOLOG (D)

/LOG displays the complete file specification, including device and directory, after the file is created.

Parameters

file-spec

The file to be created. Defaults are not supplied; if either the file-name field or the file-type field is omitted, the field is null. If the specified file exists, a new version of the file is created. Wildcard characters are not allowed.

¹ Command variants, of the form CREATE/OPTION, are listed in the table of contents and described separately on the following pages.

Example

```
>>> CREATE/LOG A.DAT
First line of file A.DAT
Next line of file
.
.
.
Last line 
^Z
DEVICE:[DIRECTORY]A.DAT;1 created
>>>
```

The **CREATE** command creates file **A.DAT** in the current directory, reads the lines of text from the terminal, and writes them into the file. **Ctrl/Z** terminates the command and closes the file. The **/LOG** qualifier displays the complete file specification.

CREATE/DIRECTORY

Creates a new directory or subdirectory.

Format

CREATE/DIRECTORY *[/[NO]LOG] directory-spec*

Qualifier

/LOG

/NOLOG (D)

/LOG displays the complete directory specification, including device, after the directory is created.

Parameters

directory-spec

The directory or subdirectory to be created. The directory specification must contain a directory name enclosed in brackets ([]). A device name is optional. A subdirectory name begins with a period (.) to separate the directory levels.

When creating a subdirectory, the top and any intermediate level directories must exist; they are not automatically created. For example, to create subdirectory [A.B.C], top level directory [A] and subdirectory [A.B] must exist or be created first. Wildcard characters are not allowed.

Examples

1 >>> CREATE/DIRECTORY [TEST]

Creates the directory [TEST] and places the file TEST.DIR in the top level directory, [000000], of the current device.

2 >>> CREATE/DIRECTORY [TEST.SUB]
>>> SET DEFAULT [TEST.SUB]

The CREATE/DIRECTORY command creates subdirectory [TEST.SUB] and places file SUB.DIR in directory [TEST]. The SET DEFAULT command changes the default directory to the subdirectory.

CREATE/WINDOW

Creates a window of a specified type and position on the screen. If the screen is off or suspended (by SET SCREEN OFF), it is turned on.

Format

CREATE/WINDOW *[/qualifiers] window-name [AT location]*

Qualifiers

/CPU=cpu-id

The CPU associated with trace or microcode windows, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/ECS

Creates a window to trace the EBox control store microcode. An ECS trace window displays the ASCII source code of the microword at the current EBox microPC. The data displayed is read from the EBox microcode listing file. The window is updated every microstep that does not burst clocks; if clocks burst, only the final PC is displayed.

/EXAMINE

Creates an examine window. This window can be used with the EXAMINE/WINDOW command to display memory data in ASCII, or instruction or binary data in the current context (byte, word, quadword, or longword).

/JCS

Creates a window to trace the JBox control store microcode. A JCS trace window displays the ASCII source code of the microword at the current JBox microPC. The data displayed is read from the JBox microcode listing file. The window is updated every microstep that does not burst clocks; if clocks burst, only the final PC is displayed.

/ODOMETER

Creates a window to monitor selected signals in the CPU. An odometer window can monitor the state of up to 256 CPU signals at a time. The SET TRACE/ODOMETER command specifies the signals to be monitored. The value of each signal and the last 32 characters of its name are displayed.

The window is updated every microstep that does not burst clocks; if clocks burst, only the final PC is displayed. When the window is updated following a microstep command, signals that have changed state are highlighted with reverse video.

/UPDATE (D)

/NOUPDATE

Determines whether the specified window is updated on a MICROSTEP or NEXT command. /NOUPDATE is the same as /VIEWONLY.

/VIEWONLY

Specifies a microcode window is display only. The window does not respond to MICROSTEP commands, but it can be scrolled to view the microcode.

Parameters

window-name

A unique window name, having up to 11 characters. The name identifies the window for SCROLL, SELECT, and DELETE/WINDOW commands.

location

The position of the window on the screen. The window can occupy a quarter, third, half, or whole screen. Window location keywords are:

Table 2-1 Window Location Keywords

Keyword	Screen Segment	Keyword	Screen Segment
W1	Whole screen	H1	First (upper) (D)
		H2	Second (lower)
T1	First (upper)	Q1	First (upper)
T2	Second	Q2	Second
T3	Third (lower)	Q3	Third
		Q4	Fourth (lower)

The command window occupies at least five lines. It is automatically extended from the end of the lowest window to the bottom of the screen.

Examples

```
❶ >>> CREATE/WINDOW
   _window_name: TOP
```

Creates a window named TOP and turns the screen on if it was off. The window type defaults to ECS (EBox control store trace window) and the location defaults to H1 (upper half of screen).

```
❷ >>> CREATE/WINDOW/CPU=1 CP1 AT Q1
   >>> CREATE/WINDOW/ODOMETER/CPU=1 OD1 AT Q2
   >>> SET TRACE/ODOMETER=OD1 @TRACE.LIST
   >>> MICROSTEP
   STEP>
```

This example creates ECS window CP1 to trace CPU1 EBox microcode and ODOMETER window OD1 to trace selected CPU1 signals. The SET TRACE/ODOMETER command specifies that signals in file TRACE.LIST are to be traced and displayed in window OD1. The MICROSTEP command places the SPU in space bar step mode (SBSM), causing one clock cycle to be issued each time the space bar is pressed. The SPU exits SBSM when any other key is pressed.

```
❸ >>> CREATE/WINDOW/ODOMETER/CPU=0 CP0 AT Q1
   >>> CREATE/WINDOW/ODOMETER/CPU=1 CP1 AT Q2
   >>> SET TRACE/ODOMETER=CP0 @TRACE.LIST
   >>> SET TRACE/ODOMETER=CP1 @TRACE.LIST
   >>> MICROSTEP/CPU=(0,1)
   STEP>
```

This example creates ODOMETER windows for CPU0 and CPU1 and specifies that each CPU traces the signals in file TRACE.LIST. The MICROSTEP command places the SPU in SBSM for both CPUs so that a clock cycle is issued in both CPUs each time the space bar is pressed.

DEALLOCATE

Unlocks the specified unit. Only the locking process (see ALLOCATE) can successfully issue this command.

Format

DEALLOCATE *object*

Parameters

object

Specifies one of the following:

- CPU[id] — Deallocate the specified CPU; if a CPU is not specified, the default CPU is allocated. See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.
- MCM — Deallocate the clock subsystem.
- PCS — Deallocate the power subsystem.
- SCU — Deallocate the system control unit, memory, and I/O.

Example

```
>>> DEALLOCATE MCM
```

Deallocates the previously allocated clock subsystem.

DEASSIGN

Removes a logical name from a specified logical name table. The **/ALL** qualifier and logical-name parameter are mutually exclusive.

NOTE

Logical names created by the command interpreter (for example, SYS\$INPUT and SYS\$OUTPUT) cannot be deassigned.

Format

DEASSIGN *[/qualifier] logical-name*

Qualifiers

/ALL

Removes all logical names from the specified table. If a logical name table is not specified, the default is the process table. If **/ALL** is specified, do not specify the logical-name parameter.

/PROCESS (D)

Specifies the process (private) logical name table.

/SYSTEM

Specifies the system logical name table.

Parameters

logical-name

The logical name to be deassigned. If the logical name contains any characters other than alphanumerics, dollar signs (\$), or underscores (_), enclose it in quotation marks ("). Wildcard characters are not allowed. If **/ALL** is specified, do not specify the logical-name parameter.

2-26 Console Commands

DEASSIGN

Example

```
>>> DEFINE SOURCE DUA50:[SOURCE]
>>> SHOW LOGICAL SOURCE
"SOURCE" = "DUA50:[SOURCE]" (PROCESS)
.
.
.
>>> DEASSIGN SOURCE
>>> SHOW LOGICAL SOURCE
%CLI-S-NOTRAN, no translation for logical name SOURCE
```

The **DEFINE** command assigns the logical name **SOURCE** to the device and directory specification **DUA50:[SOURCE]**. Subsequent references to logical name **SOURCE** result in references to that directory.

The first **SHOW LOGICAL** command displays the equivalence string and logical name table for the logical name **SOURCE**. The **DEASSIGN** command deassigns the logical name. The second **SHOW LOGICAL** command shows that the name has been deassigned.

DEBUG

Causes the SPU to enter the local debugger.

Format

DEBUG [*/CONFIRM*]

Qualifier

/CONFIRM (D)

/NOCONFIRM

/CONFIRM issues a confirmation prompt before entering the debugger.

DEFINE

Creates a logical name entry in a specified logical name table and assigns an equivalence string to the logical name.²

Format

DEFINE *[/qualifiers] logical-name[:] equivalence-string*

Qualifiers

/LOG (D)

/NOLOG

Determines whether to display a message when a defined logical name supersedes an existing name.

/PROCESS (D)

Places the logical name in the process private logical name table.

/SYSTEM

Places the logical name in the system logical name table.

Parameters

logical-name

The logical name to be defined. It can include any alphanumeric character, dollar signs, or underscores.

Logical names ending with a colon (:) are assumed to be definitions of disk volumes for use in file specifications. The colon is not stored as part of the logical name.

Logical names are placed in the process logical name table by default.

² DEFINE/KEY, a special form of the command, is described separately on the following pages.

equivalence-string

A quoted or unquoted string associated with the logical name.

If the string contains spaces, tabs, or lowercase letters, enclose the string in quotation marks. To use quotation marks in a string, enclose the entire string in quotation marks and use a double set of quotation marks ("") within the string. For example:

```
DEFINE QUOTED_STRING "This is a ""quoted"" string"
```

If the logical name is to be used as a file specification, include the punctuation marks (colons (:), brackets ([]), periods (.)) in the equivalence string that would be required if the string were used directly as a file specification. For example:

```
DEFINE FILE DUA0:[CONSOLE]TEST.DAT
```

Local area network node names must be defined as logical names. The equivalence string is a decimal number with the format:

dd.nnn

where *dd* is the area number and *nnn* is the node number. Limiting the logical name to six characters is consistent with VMS conventions but not necessary.

DEFINE/KEY

Assigns an equivalence string to a terminal key so that pressing the key is equivalent to entering the string on the command line. Up to 32 keys on an LK201 keyboard can be defined.

Format

DEFINE/KEY *[/qualifiers] key-name equivalence-string*

Qualifiers

/ECHO (D)

/NOECHO

Determines whether the equivalence string is displayed when the key is pressed.

/NOECHO and **/NOTERMINAL** cannot be used together.

/LOG

Displays the result of the key definition.

/TERMINAL

/NOTERMINAL (D)

/TERMINAL processes the command line and terminates the current equivalence string when the key is pressed.

/NOTERMINAL allows the equivalence string to be inserted in or appended to the current command line. The command line is processed when Return is pressed, or when a key with the **/TERMINAL** option is pressed. **/NOTERMINAL** and **/NOECHO** cannot be used together.

Parameters

key-name

The name of the key to be defined. The types of keys that can be defined are:

- Numeric keypad keys
- Function keys F6 through F20
- Editing keypad keys (except the arrow keys)

Valid key names are:

Key Name	Key
Editing Keypad	
E1	Find
E2	Insert Here
E3	Remove
E4	Select
E5	Prev Screen
E6	Next Screen
Function Keys	
F6-F14	F6-F14
Do	F15
Help	F16
F17-F20	F17-F20
Numeric Keypad	
PF1-PF4	PF1-PF4
KP0-KP9	0-9
PERIOD	.
COMMA	,
MINUS	-
ENTER	Enter

NOTE

For key definitions to take effect, the keypad must be in application mode (SET TERMINAL/KEYPAD=APPLICATION).

equivalence-string

The string to be processed when the key is pressed. If the string contains spaces, enclose the string in quotation marks.

Example

```
>>> SET TERMINAL/KEYPAD=APPLICATION
>>> DEFINE/KEY/TERMINAL KP0 "MICROSTEP 0"
```

Defines the 0 key on the numeric keypad as the MICROSTEP command with a step count of 0 to prevent the command from entering space bar step mode after each clock cycle.

DELETE

Deletes one or more files from a mass storage device.³

Format

DELETE *[/qualifiers] file-spec*

Qualifiers

/CONFIRM

/NOCONFIRM (D)

Determines whether an affirmative response is required before each file is deleted. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	Delete the file.
T (true)	Delete the file.
N (no)	Do not delete the file.
F (false)	Do not delete the file.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

/LOG

/NOLOG (D)

Determines whether the file specification and block size of each deleted file are displayed.

³ Command variants, of the form DELETE/OPTION, are listed in the table of contents and described separately on the following pages.

Parameters

file-spec

The file or files to be deleted. The file specification must include the file name, type, and version number. If device or directory are not specified, the default device and directory are assumed. The wildcard can be substituted for a group of files or any field (file name, type, or version) or partial field in the file specification. The DELETE command does not allow a file list.

To delete the latest version of a file, but not earlier versions, either do not specify a version number after the semicolon field separator, specify a version number of 0, or leave one or more spaces after the semicolon.

Examples

1 >>> DELETE COMMON.SUM;2

Deletes the file COMMON.SUM;2 from the default disk and directory.

2 >>> DELETE *.OLD;*

Deletes all versions of files with the file type of .OLD.

3 >>> DELETE A*.*;*

Deletes all files starting with A.

4 >>> DELETE TEST.DAT;

Deletes only the highest numbered version of TEST.DAT.

DELETE/PATTERN

Deletes one or more scan patterns for the specified CPU from the system scan pattern table.

Format

DELETE/PATTERN *[/qualifiers] [pattern-name]*

Qualifiers

/ALL

Deletes all scan patterns for the specified CPU. If */ALL* is specified, do not specify the *pattern-name* parameter.

/CPU=cpu-id

The CPU for which scan patterns are to be deleted, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, scan patterns for the default CPU are deleted.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

/NOLOG (D)

Determines whether the the name of each deleted pattern is displayed.

/RESET

Specifies that all tracepoints, watchpoints, or patterns that were set by any process are to be deleted. This qualifier is primarily for development use and should be avoided.

/SCU

/NOSCU

Determines whether system control unit pattern points are deleted.

Parameters

pattern-name

The name of one or more pattern definitions to be deleted. The asterisk (*) wildcard character is allowed. If /ALL is specified, do not specify the pattern-name parameter. For information on scan patterns, see the SET PATTERN command description.

DELETE/SYMBOL

Deletes one or all symbols from a local or global symbol table.

Format

DELETE/SYMBOL *[qualifiers] [symbol-name]*

Qualifiers

/ALL

Deletes all symbols in the specified symbol table. If a symbol table is not specified, the local symbol table is assumed. If */ALL* is specified, do not specify the symbol-name parameter.

/GLOBAL

Specifies the global symbol table.

/LOCAL (D)

Specifies the local symbol table.

/LOG

/NOLOG (D)

Determines whether the name of each deleted symbol is displayed.

Parameters

symbol-name

The symbol to be deleted. By default, the symbol is assumed to be in the local symbol table. If /ALL is specified, do not specify the symbol-name parameter.

Examples

1 >>> DELETE/SYMBOL/ALL

Deletes all symbol definitions from the local symbol table.

2 >>> DELETE/SYMBOL/LOG FOO
%CLI-I-DELSYM, Local symbol FOO has been deleted

Deletes symbol FOO from the local symbol table and displays an informational message listing the symbol deleted.

3 >> DELETE/SYMBOL/GLOBAL PDEL

Deletes symbol PDEL from the global symbol table.

DELETE/TRACE

Deletes one or more tracepoints from the system tracepoint table for the specified CPU.

Format

DELETE/TRACE *[/qualifiers] [tracepoint-name]*

Qualifiers

/ALL

Deletes all tracepoints for the specified CPU.

Specifying ***/ALL*** causes a special opcode to flush the SCM's tracepoint table. This differs from issuing **DELETE/TRACE *** in that it resynchronizes the tracepoint table. If ***/ALL*** is specified, do not specify the tracepoint-name parameter.

/CPU=cpu-id

The CPU from which tracepoints are to be deleted, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, tracepoints are deleted from the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

/NOLOG (D)

Determines whether the name of each deleted tracepoint is displayed.

/RESET

Specifies that all tracepoints, watchpoints, or patterns that were set by any process are to be deleted. This qualifier is primarily for development use and should be avoided.

/SCU

/NOSCU

Determines whether system control unit tracepoints are deleted.

Parameters

tracepoint-name

The name of one or more tracepoints to be deleted. The asterisk (*) wildcard character is allowed. If */ALL* is specified, do not specify the *tracepoint-name* parameter. For information on tracepoints, see the SET TRACEPOINT command description.

DELETE/WATCH

Deletes one or more watchpoints from the system watchpoint table for the specified CPU.

Format

DELETE/WATCH *[/qualifiers] [watchpoint-name]*

Qualifiers

/ALL

Deletes all watchpoints and tracepoints for the specified CPU.

Specifying */ALL* causes a special opcode to flush the SCM's watchpoint table. This differs from issuing *DELETE/WATCH ** in that it resynchronizes the watchpoint table. If */ALL* is specified, do not specify the watchpoint-name parameter.

/CPU=cpu-id

The CPU from which watchpoints are to be deleted, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, watchpoints are deleted from the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

/NOLOG (D)

Determines whether the name of each deleted watchpoint is displayed.

/RESET

Specifies that all tracepoints, watchpoints, or patterns that were set by any process are to be deleted. This qualifier is primarily for development use and should be avoided.

/SCU

/NOSCU

Determines whether system control unit watchpoints are deleted.

Parameters

watchpoint-name

The name of one or more watchpoints to be deleted. The asterisk (*) wildcard character is allowed. If */ALL* is specified, do not specify the watchpoint-name parameter. For information on watchpoints, see the SET WATCHPOINT command description.

DELETE/WINDOW

Deletes a window from the screen. If the window was visible on the screen, the screen is redrawn to display previously hidden windows.

Format

DELETE/WINDOW *[/ALL] [window-name]*

Qualifier

/ALL

Deletes all windows from the screen and turns off screen mode. If */ALL* is specified, do not specify the *window-name* parameter.

Parameters

window-name

The name of the window to delete. Wildcard characters are not allowed. Note that the command window cannot be deleted. If */ALL* is specified, do not specify the *window-name* parameter.

DEPOSIT

Replaces the contents of a location or series of locations specified by an address expression with data specified by a value expression. A pointer to the location and the context of the data are saved for subsequent DEPOSIT and EXAMINE commands.

The DEPOSIT command can replace the contents of locations in memory, I/O space, general-purpose registers (GPRs), internal processor registers (IPRs), control stores, data structures, register structures, and scan rings.

Format

DEPOSIT *[/qualifiers] address-expression value-expression*

Qualifiers

/ASCII

The value-expression is an ASCII string and must be enclosed in quotation marks (""). Subsequent unqualified references default to the ASCII data type.

/BYTE

Specifies the value-expression is integer data and is to be deposited one byte at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/CODE

Specifies program address space. Valid only with /PEM or /RIC.

/CPU=cpu-id

The CPU in which the data is to be deposited, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, data is deposited in the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/D_FLOAT

The value-expression is a floating-point number in the specified format.

/ECS

The address-expression specifies an EBox control store location.

/EMEMORY

Specifies external memory address space. Valid only with */PEM* or */RIC*.

/EREG

The address-expression specifies an EBox register.

/F_FLOAT

The value-expression is a floating-point number in the specified format.

/GENERAL

The address-expression is a general-purpose register (in the range 0 to 15). Subsequent unqualified references default to general-purpose register address space.

/G_FLOAT

The value-expression is a floating-point number in the specified format.

/IMEMORY

Specifies internal memory address space. Valid only with */PEM* or */RIC*.

/INTERNAL

The address-expression is an internal register (in the range 0 to 255). Subsequent unqualified references default to internal register address space.

/JCS

The address-expression specifies a JBox control store location.

/LENGTH=bits

Valid only when */RING* is specified. Specifies the number of bits (in the current radix) to be deposited. If not specified, the command defaults to the length specified in the configuration database (CDB) file.

/LOG

/NOLOG

Displays the address and value of each location to which data is deposited.

/LONGWORD (D)

Specifies the value-expression is integer data and is to be deposited one longword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/MCM

Specifies a master clock module register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

Table 2-2 MCM Register Mnemonics

BURST	CCR0	CCR1	DIVIDER	FREQUENCY
INTERVAL	POSITION	SERIAL	TRANSFER	

/NEXT[=count]

Deposit data into the address-expression location and the next [count] location[s].

/OCTAWORD

Specifies the value-expression is integer data and is to be deposited one octaword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/PEM

Specifies a power and environmental monitor register, port register, memory address (internal or external), or program space location. A register address-expression can be a numeric literal or one of the following mnemonics:

Table 2-3 PEM Register Mnemonics

ASDREG	BBUREG	BSCREG	CSREG	EMEREG
KEYAREG	KEYBREG	MISREG	OCPDISREG	OCPLEDREG
OCPSWREG	POLLREG	PSREG	PWR1REG	PWR2REG
P1REG	RBDIREG	RBDSREG	RICAVAREG	RTEREG
STCREG	TOFFREG	VSNREG		

The following address space qualifiers can also be specified with the /PEM qualifier:

Table 2-4 PEM Address Space Qualifiers

Qualifier	Address Space Specified
/CODE	Program space
/EMEMORY	External memory
/IMEMORY	Internal memory
/PORT_REGISTER	Port register
/REGISTER	Internal register

/PHYSICAL

The address-expression is a 32-bit physical address. Subsequent unqualified references default to physical address space.

/PORT_REGISTER

Specifies port register address space. Valid only with /PEM or /RIC.

/QUADWORD

Specifies the value-expression is integer data and is to be deposited one quadword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/REGISTER

Specifies internal register address space. Valid only with /PEM or /RIC.

/RIC=ric-id

Specifies a regulator intelligence card register, port register, memory address (internal or external), or program space location. The ric-id can be a numeric literal or one of the following mnemonics:

Table 2-5 RIC Register Mnemonics

ADREG	ASDREG	BIREG	BIXREG	CSREG
DAREG	DLYREG	EMEREG	ESREG	HWSREG
IDREG	LMREG	MGNREG	MUXREG	PFREG
PLREG	PSREG	P1REG	RCREG	RENREG
RSREG	SCSREG	SEREG	VSNREG	XUEREG

The following address space qualifiers can also be specified with the */RIC* qualifier:

Table 2-6 RIC Address Space Qualifiers

Qualifier	Address Space Specified
<i>/CODE</i>	Program space
<i>/EMEMORY</i>	External memory
<i>/IMEMORY</i>	Internal memory
<i>/PORT_REGISTER</i>	Port register
<i>/REGISTER</i>	Internal register

/RING

Specifies a physical scan ring.

/SCC

Specifies a scan controller register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

Table 2-7 SCC Register Mnemonics

ABR	CSR	CCR	DCSR	DMAE
DMAI	DMAM	DMAO	ESR	MSR
RCR	SCR	SDR	SHR	SSR

/SCU

/NOSCU (D)

Specifies a system control unit register.

/SJA

Specifies an SPU-to-JBox adapter register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

Table 2-8 SJA Register Mnemonics

ADDR	CMND	DATHI	DATLO	DMASK
DXCNT	DXCS	DXMEM	DXSPU	FLAG
RETRD	RXCS	RXDB	RXFCT	RXPAR
RXPRM	SJACS	SJCS	TODR	TXCS
TXDB	TXFCT	TXPRM	XJA	

/SPU

Specifies service processor unit memory.

/VERIFY***/NOVERIFY***

Valid only when */RING* is specified. Specifies whether scan ring deposits are verified. To verify scan ring deposits, a test pattern is written to the ring, followed by the value-expression. When the ring is rotated to deposit the data, the test pattern is verified.

/VECTOR=register:element

Specifies vector register number and element number.

/VIRTUAL

The address-value is a 32-bit virtual address. If the address is in P0 or P1 space, a CPU must be specified (*/CPU=cpu-id*) to determine process context. If a CPU is not specified, the default CPU is assumed.

/WORD

Specifies the value-expression is integer data and is to be deposited one word at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

Parameters

address-expression

The address of a location or the starting address of a series of locations to receive data. The address can be a numeric literal, symbolic address, or special operator.

value-expression

The data to be deposited. The value can be a numeric literal, lexical function, or special operator.

Expressions

numeric-literals

An address-expression or value-expression can be a numeric value in any of four radices determined by one of the following radix operators:

- %B*** = Binary
- %D*** = Decimal
- %X*** = Hexadecimal (D)
- %O*** = Octal

For example, in the command:

```
DEPOSIT 100 %D1234
```

the value-expression is a decimal value.

The default radix can be changed with the **SET RADIX** command.

special-operators

The DEPOSIT and EXAMINE commands allow special operators to be used in place of expressions. The following table lists the operators:

Table 2-9 DEPOSIT Special Operators

Operator	Definition
. (period)	Use last referenced location.
* (asterisk)	Use last referenced location.
+ (plus)	Increment last referenced location by current data size (memory references) or by one (all other references).
- (minus)	Decrement last referenced location by current data size (memory references) or by one (all other references).
@ (at)	Use contents of last referenced location as new address (indirect addressing). Or, use data from last DEPOSIT command as new deposit data.

symbolic-address-mnemonics

The DEPOSIT and EXAMINE commands allow mnemonics to be used in place of numeric literal addresses to reference the processor status longword (PSL), general-purpose registers (GPRs), and internal processor registers (IPRs). The qualifiers /GENERAL and /INTERNAL cannot be used with a mnemonic.

The following tables list the GPR and IPR mnemonics. The addresses in the tables are the hexadecimal values that would be required if the /GENERAL or /INTERNAL qualifiers were used. For example: DEPOSIT R0 1234 is equivalent to DEPOSIT/GENERAL 0 1234.

Table 2-10 GPR and PSL Register Mnemonics

Mnemonic	Address	Name
AP	12	Argument pointer
FP	13	Frame pointer
PC	15	Program counter
PSL	-	Processor status longword
R0-R11	00	GPRs R0 to R11
SP	14	Stack pointer

Table 2-11 IPR Register Mnemonics

Mnemonic	Address	Register Name
ASTLVL	13	Asynchronous system trap level
ESP	01	Executive stack pointer
ICCS	18	Interval clock control
ICR	1A	Interval count
IPL	12	Interrupt priority level
ISP	04	Interrupt stack pointer
KSP	00	Kernel stack pointer
MAPEN	38	Memory management enable
NICR	19	Next interval count
P0BR	08	P0 base register
P0LR	09	P0 length register
P1BR	0A	P1 base register
P1LR	0B	P1 length register
PCBB	10	Process control block base
PME	3D	Performance monitor enable
RXCS	20	Console receiver control and status
RXDB	21	Console receiver data buffer
SBR	0C	System base register
SCBB	11	System control block base
SID	3E	System identification
SIRR	14	Software interrupt request
SISR	15	Software interrupt summary
SLR	0D	System length register
SSP	02	Supervisor stack pointer
TBCHK	3F	Translation buffer check
TBIA	39	Translation buffer invalidate all
TBIS	3A	Translation buffer invalidate single
TODR	1B	Time of year
TXCS	22	Console transmit control and status
TXDB	23	Console transmit data buffer
USP	03	User stack pointer

2-52 Console Commands

DEPOSIT

Examples

```
❶ >>> DEPOSIT 100 100
>>> DEPOSIT + 200
>>> EXAMINE .
P 00000104 00000200
```

The first command deposits a value of 100 in physical memory address 100. The second command increments this address by the current data size (4 for longword) and deposits a value of 200 in the resultant location (104).

```
❷ >>> DEPOSIT/VIRTUAL 100 200
>>> DEPOSIT + @
>>> EXAMINE .
00000104 00000200
```

The first command deposits a value of 200 in virtual memory address 100. The second command increments this address by the current data size and deposits the value from the previous DEPOSIT command (200) in the resultant location.

```
❸ >>> DEPOSIT/ASCII 100 "ABC DEF GHI"
>>> EXAMINE/ASCII 100
P 00000100 ABC DEF GHI
>>> EXAMINE/LONG/NEXT:2 100
P 00000100 20434241
P 00000104 20464544
P 00000108 00494847
```

The DEPOSIT command deposits an ASCII character string in physical memory starting at address 100. The EXAMINE commands display the data in ASCII and in longword format.

DIRECTORY

Lists the contents of a directory or provides information about a file or group of files that match a given file specification.

Format

DIRECTORY *[/qualifiers] [file-spec]*

Qualifiers

/DATE

Lists the creation date of each specified file.

/FULL

Lists the following items for each specified file:

- File name, type, and version number
- Number of blocks used and allocated
- Owner's user identification code (UIC)
- Creation and last revision dates
- Expiration and backup dates
- File organization and attributes
- Record format and attributes
- File protection

/OWNER

Lists the owner UIC of each specified file.

/PROTECTION

Lists the file protection of each specified file.

/SIZE

Lists the size in blocks of each specified file.

/TOTAL

Displays only the total number of files and, if */SIZE* is specified, the total number of blocks for all files in the directory.

Parameters

file-spec

The file or files to be listed, according to the following syntax:

Syntax	Action
No file-spec	List all versions of all files in the current directory.
Device only	Use current directory as a default and list all versions of all files in that directory of the specified device.
Device and directory	List all versions of all files in the specified directory of the specified device.
Device, directory, and file name, or file name only	List all files of that name in the specified or current directory of the specified or current device, regardless of file type and version.
Device, directory, and file type, or file type only	List all files of that type in the specified or current directory of the specified or current device, regardless of file name and version.
Wildcards	The asterisk (*) and percent sign (%) wildcard characters can be used in certain fields of the file specification, as follows:

Field	Valid Wildcard
Name	* or %
Type	* or %
Version	* only

Examples

❶ >>> DIRECTORY

Lists all versions of all files in the default device and directory.

❷ >>> DIRECTORY A*

Lists all types and versions of all files beginning with A in the default device and directory.

❸ >>> DIRECTORY DUA0:[CONSOLE]*.CMD

Lists all versions of all files with the CMD file type field in device DUA0 directory [CONSOLE].

DISMOUNT

Closes a previously mounted disk or tape volume and deassigns the logical name, if any, associated with the device. All open files on the volume must be closed before the device can be dismounted.

Format

DISMOUNT *[/[NO]UNLOAD] device-name[:]*

Qualifier

/UNLOAD

/NOUNLOAD

Specifies whether a tape is unloaded on dismount.

Parameters

device-name[:]

The name of the device to be dismounted. The device name can be a physical device name or a logical name assigned to the device.

Example

```
>>> MOUNT DU: TESTVOL DISK
      .
      .
      .
>>> DISMOUNT DISK:
```

The **MOUNT** command mounts disk volume **TESTVOL** on physical device **DUA0** and assigns the logical name **DISK** to the device. The **DISMOUNT** command closes access to the volume, deallocates the device, and deletes the logical name **DISK**.

EDIT

Invokes the EDT text editor.

Format

EDIT *[/qualifiers] input-file-spec*

Qualifiers

/COMMAND[=command-file] (D)

/NOCOMMAND

/COMMAND causes the editor to execute a startup command file. If the qualifier or command-file specification is omitted, the editor executes a system- or user-defined default startup command file. To use some other startup command file, use the command-file specification. For example:

```
>>> EDIT/COMMAND=[CONSOLE.TOOLS]XEDTINI TEST.DAT
```

Wildcard characters are not allowed, and the default startup command file type is *.EDT*.

If */NOCOMMAND* is specified or a startup command file does not exist, the editor begins the editing session in its default state.

/CREATE (D)

/NOCREATE

Determines whether the editor creates a new file if the specified input file does not exist. If */NOCREATE* is specified and the file does not exist, the editor does not create the file and displays an error message.

/JOURNAL[=journal-file]

/NOJOURNAL

Determines whether the editor records the editing session in a journal file. The default journal file specification is the input file name with file type *.JOU*. To specify a different journal file name or type, use the */JOURNAL=journal-file* option. Wildcard characters are not allowed. The journal file allows recovery from an abnormally ended editing session (see */RECOVER*).

/NOJOURNAL prevents the editor from recording the editing session.

/OUTPUT[=output-file]
/NOOUTPUT

Determines whether the editor creates an output file at the end of the editing session. The default output file specification is the same as the input file specification. Use the */OUTPUT=output-file* option to specify a different output file specification. Wildcard characters are not allowed in the file specification.

If you specify */NOOUTPUT* on the command line but then decide to save the editing session, issue the the editor line-mode command *WRITE* to output the text to an external file before ending the session.

NOTE

Specifying */NOOUTPUT* suppresses the creation of an output file but not the creation of a journal file.

/RECOVER

Specifies that the editor read a journal file at the start of the editing session.

If an editing session ends abnormally while journaling is in effect, use the */RECOVER* qualifier to start the next session. When */RECOVER* is specified, the editor processes the commands contained in the journal file to restore the edits made in the previous session. For example:

```
>>> EDIT/RECOVER TEST.DAT
```

The default journal file name is the input file name and the default file type is *.JOU*. If the journal file name is not the same as the input file name or the file type is not *.JOU*, include */RECOVER* and */JOURNAL* on the command line. For example:

```
>>> EDIT/RECOVER/JOURNAL=SAVE.XXX TEST.DAT
```

Parameters

input-file-spec

Specifies the file to be created or edited. Wildcard characters are not allowed. If the file does not exist, and /NOCREATE is not specified, the editor creates the file. The editor does not provide a default file type when creating files. If a file type is not specified, the field is left null. The file must be a disk file on a Files-11 formatted volume.

Examples

```
❶ >>> EDIT/OUTPUT=NEWFILE.TXT OLDFILE.TXT
      1          This is the first line of the file OLDFILE.TXT.
      *
```

The EDIT command opens OLDFILE.TXT for editing and writes the output to file NEWFILE.TXT when the session ends.

```
❷ >>> EDIT/RECOVER OLDFILE.TXT
```

The editor opens file OLDFILE.TXT and processes journal file OLDFILE.JOU to recover from an abnormally ended editing session. Normal interactive editing can resume after the journal file is processed.

EVALUATE

Evaluates a numeric expression and displays the result on the console terminal in the current or specified radix. The current radix is set by the SET RADIX command.

Format

EVALUATE *[/qualifier] expression*

Qualifiers

/DISPLAY=radix-spec

Specifies the radix of the result, where radix-spec can be:

- HEXADECIMAL
- DECIMAL
- OCTAL
- BINARY

The expression is evaluated in the current radix as specified by the SET RADIX command.

/RADIX=radix-spec

Overrides the current radix. The expression is evaluated and the result displayed in the specified radix, where radix-spec can be:

- HEXADECIMAL
- DECIMAL
- OCTAL
- BINARY

Parameters

expression

The numeric expression to be evaluated. The expression can consist of integer values, or symbols or lexical functions that evaluate to integers.

An integer value can be prefixed by a radix operator to override the current radix. The radix operators are:

`%X` = Hexadecimal
`%O` = Octal
`%D` = Decimal
`%B` = Binary

NOTE

If the default radix is hexadecimal, the EVALUATE command interprets the characters A through F (upper or lowercase) as hexadecimal data and not as symbols. To force symbol substitution of these characters, enclose them in apostrophes.

Example

```
>>> SET RADIX HEX
>>> A = 1
>>> B = 2
>>> EVALUATE A + B
00000015
>>> EVALUATE 'A' + 'B'
00000003
```

The first EVALUATE command interprets the characters A and B as hexadecimal data and computes the sum to be 15_{16} . The second EVALUATE command substitutes the values defined for symbols A and B (1 and 2) and computes the sum to be 3.

EXAMINE

Displays the contents of a location or series of locations specified by an address expression. A pointer to the location and the context of the data are saved for subsequent DEPOSIT and EXAMINE commands.

The EXAMINE command can display the contents of locations in memory, I/O space, GPRs, IPRs, control stores, data structures, register structures, and scan rings.

Format

address-expression
EXAMINE *[/qualifiers]* *signal*
structure

Qualifiers

/ASCII[=count]

Displays data in ASCII format. For example:

```
>>> DEPOSIT/ASCII 1000 "This is an ASCII string"  
>>> EXAMINE/ASCII 1000  
P 00001000 This is an ASCII string
```

Binary data is converted to ASCII before it is displayed. If the binary value has no ASCII equivalent, the EXAMINE command displays a period (.). Count is a hexadecimal number.

When /ASCII is specified, or ASCII mode is the default, the EXAMINE command uses hexadecimal as the default radix for numeric literals that are specified on the command line.

/BYTE

Specifies the result is integer data and is to be displayed one byte at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/CODE

Specifies program address space. Valid only with /PEM or /RIC.

/CPU=cpu-id

The CPU in which a location is to be examined, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the location is in the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/D_FLOAT

The result is a floating-point number in the specified format.

/ECS

The address-expression specifies an EBox control store location.

/EMEMORY

Specifies external memory address space. Valid only with */PEM* or */RIC*.

/EREG

The address-expression specifies an EBox register.

/F_FLOAT

The result is a floating-point number in the specified format.

/GENERAL

The address-expression is a general-purpose register (in the range 0 to 15). Subsequent unqualified references default to general-purpose register address space.

/G_FLOAT

The result is a floating-point number in the specified format.

/IMEMORY

Specifies internal memory address space. Valid only with */PEM* or */RIC*.

/INSTRUCTION

The result data type is a VMS macroinstruction. For example:

```
>>> DEPOSIT 1000 5150D0
>>> EXAMINE/INSTRUCTION 1000
    P 00001000 MVL R0,R1
```

2-64 Console Commands

EXAMINE

/INTERNAL

The address-expression is an internal register (in the range 0 to 255). Subsequent unqualified references default to internal register address space.

/JCS

The address-expression specifies a JBox control store location.

/LABEL

Displays a hierarchical list of labels from the current database (CDB) file in the format:

CPU.MCU: Model = mmm Revision = v

as follows:

```
>>> EXAMINE/LABEL
%CPU0.CTL: Model = CTL Revision = A
%CPU0.CTU: Model = CTU Revision = A
%CPU0.DST: Model = DST Revision = A
%CPU0.DTA: Model = DTA Revision = A
%CPU0.DTB: Model = DTB Revision = A
%CPU0.FAD: Model = FAD Revision = A
%CPU0.INT: Model = INT Revision = A
%CPU0.MUL: Model = MUL Revision = A
%CPU0.OPU: Model = OPU Revision = A
%CPU0.UCS: Model = UCS Revision = A
%CPU0.VAD: Model = VAD Revision = A
%CPU0.VAP: Model = VAP Revision = A
%CPU0.VIC: Model = VIC Revision = A
%CPU0.VML: Model = VML Revision = A
%CPU0.VRG: Model = VRG Revision = A
%CPU0.XBR: Model = XBR Revision = A
```

/LENGTH=bits

Valid only when */RING* is specified. Specifies the number of bits (in the current radix) to be displayed. If not specified, the command defaults to the length specified in the current database (CDB) file.

/LOG

/NOLOG

Displays the address and value of each location examined.

/LONGWORD (D)

Specifies the result is integer data and is to be examined one longword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/MCM

Specifies a master clock module register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

MCM Register Mnemonics

BURST	CCR0	CCR1	DIVIDER	FREQUENCY
INTERVAL	POSITION	SERIAL	TRANSFER	

/NEXT[=count]

Examines the address-expression location and the next [count] location[s].

/OCTAWORD

Specifies the result is integer data and is to be displayed one octaword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/PEM

Specifies a power and environmental monitor register, port register, memory address (internal or external), or program space location. A register address-expression can be a numeric literal or one of the following mnemonics:

PEM Register Mnemonics

ASDREG	BBUREG	BSCREG	CSREG	EMEREG
KEYAREG	KEYBREG	MISREG	OCPDISREG	OCPLEDREG
OCPSWREG	POLLREG	PSREG	PWR1REG	PWR2REG
P1REG	RBDIREG	RBDSREG	RICAVAREG	RTEREG
STCREG	TOFFREG	VSNREG		

The following address space qualifiers can also be specified with the /PEM qualifier:

PEM Address Space Qualifiers

Qualifier	Address Space Specified
/CODE	Program space
/EMEMORY	External memory
/IMEMORY	Internal memory
/PORT_REGISTER	Port register
/REGISTER	Internal register

/PHYSICAL

The address-expression is a 32-bit physical address. Subsequent unqualified references default to physical address space.

/PORT_REGISTER

Specifies port register address space. Valid only with /PEM or /RIC.

/QUADWORD

Specifies the result is integer data and is to be examined one quadword at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

/REGISTER

Specifies internal register address space. Valid only with /PEM or /RIC.

/RIC=ric-id

Specifies a regulator intelligence card register, port register, memory address (internal or external), or program space location. The ric-id can be a numeric literal or one of the following mnemonics:

RIC Register Mnemonics

ADREG	ASDREG	BIREG	BIXREG	CSREG
DAREG	DLYREG	EMEREG	ESREG	HWSREG
IDREG	LMREG	MGNREG	MUXREG	PFREG
PLREG	PSREG	PIREG	RCREG	RENREG
RSREG	SCSREG	SEREG	VSNREG	XUEREG

The following address space qualifiers can also be specified with the */RIC* qualifier:

RIC Address Space Qualifiers

Qualifier	Address Space Specified
<i>/CODE</i>	Program space
<i>/EMEMORY</i>	External memory
<i>/IMEMORY</i>	Internal memory
<i>/PORT_REGISTER</i>	Port register
<i>/REGISTER</i>	Internal register

/RING

Specifies a physical scan ring.

/SCC

Specifies a scan controller register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

SCC Register Mnemonics

ABR	CSR	CCR	DCSR	DMAE
DMAI	DMAM	DMAO	ESR	MSR
RCR	SCR	SDR	SHR	SSR

/SCU

/NOSCU (D)

Specifies a system control unit register.

/SJA

Specifies an SPU-to-JBox adapter register. The address-expression specifies a register address and can be a numeric literal or one of the following mnemonics:

SJA Register Mnemonics

ADDR	CMND	DATHI	DATLO	DMASK
DXCNT	DXCS	DXMEM	DXSPU	FLAG
RETRD	RXCS	RXDB	RXFCT	RXPAR
RXPRM	SJACS	SJCS	TODR	TXCS
TXDB	TXFCT	TXPRM	XJA	

/SPU

Specifies service processor unit memory.

/SYMBOL=name

Primarily used in command procedures, this qualifier assigns a symbol name to the result and suppresses the display. If the result is less than 32 bits, the data type is integer; otherwise, the data type is bitvector. If the */INSTRUCTION* qualifier is specified, the data type is string.

/VECTOR=register-number:element-number

Specifies vector register number and element number.

/VIRTUAL

The address-value is a 32-bit virtual address. If the address is in P0 or P1 space, a CPU must be specified (*/CPU=cpu-id*) to determine process context. If a CPU is not specified, the default CPU is assumed.

/WINDOW[=window-name]

Displays data in the specified format in the specified window. If window-name is not specified, data is displayed in the default EXAMINE window. The window can be scrolled up or down with the SCROLL command.

/WORD

Specifies the result is integer data and is to be displayed one word at a time. This qualifier is valid only in the context of physical or virtual address space. Subsequent unqualified references default to this data type.

Parameters

address-expression

The address of the location or the starting address of a series of locations to examine. The address can be a numeric literal, symbolic address, or special operator.

signal-name

Specifies a signal name as listed in the current database (CDB) file in the format:

```
%CPUn.label.label.signal<end-bit:start-bit>  
%SPU.label.label.signal<end-bit:start-bit>
```

structure-name

Specifies a control store, register file, data cache, or control structure as listed in the current database (CDB) file in the format:

```
%CPUn.structure-name[location]<end-bit:start-bit>  
%SPU.structure-name[location]<end-bit:start-bit>
```

Expressions

numeric-literals

An address-expression or value-expression can be a numeric value in any of four radixes determined by one of the following radix operators:

- %B** = Binary
- %D** = Decimal
- %X** = Hexadecimal (D)
- %O** = Octal

For example, in the following command the **%D** operator specifies the address as decimal:

```
>>> EXAMINE %D100  
P 00000064 A06267D0
```

The default radix is determined by the **SET RADI**X command.

special-operators

The DEPOSIT and EXAMINE commands allow special operators to be used in place of expressions. The following table lists the operators:

Table 2-12 EXAMINE Special Operators

Operator	Definition
. (period)	Use last referenced location.
* (asterisk)	Use last referenced location.
+ (plus)	Increment last referenced location by current data size (memory references) or by one (all other references).
- (minus)	Decrement last referenced location by current data size memory references or by one (all other references).
@ (at)	Use contents of last referenced location as new address (indirect addressing). Or, use data from last DEPOSIT command as new deposit data.

symbolic-address-mnemonics

The DEPOSIT and EXAMINE commands allow mnemonics to be used in place of numeric literal addresses to reference the processor status longword (PSL), general-purpose registers (GPRs), and internal processor registers (IPRs). The qualifiers /GENERAL and /INTERNAL cannot be used with a mnemonic.

The following tables list the GPR and IPR mnemonics. The addresses in the tables are the hexadecimal values that would be required if the /GENERAL or /INTERNAL qualifiers were used. For example:

EXAMINE R0

is equivalent to

EXAMINE/GENERAL 0

GPR and PSL Register Mnemonics

Mnemonic	Address	Name
AP	12	Argument pointer
FP	13	Frame pointer
PC	15	Program counter
PSL	-	Processor status longword
R0-R11	00	GPRs R0 to R11
SP	14	Stack pointer

IPR Register Mnemonics

Mnemonic	Address	Register Name
ASTLVL	13	Asynchronous system trap level
ESP	01	Executive stack pointer
ICCS	18	Interval clock control
ICR	1A	Interval count
IPL	12	Interrupt priority level
ISP	04	Interrupt stack pointer
KSP	00	Kernel stack pointer
MAPEN	38	Memory management enable
NICR	19	Next interval count
P0BR	08	P0 base register
P0LR	09	P0 length register
P1BR	0A	P1 base register
P1LR	0B	P1 length register
PCBB	10	Process control block base
PME	3D	Performance monitor enable
RXCS	20	Console receiver control and status
RXDB	21	Console receiver data buffer
SBR	0C	System base register
SCBB	11	System control block base
SID	3E	System identification
SIRR	14	Software interrupt request
SISR	15	Software interrupt summary
SLR	0D	System length register
SSP	02	Supervisor stack pointer
TBCHK	3F	Translation buffer check
TBIA	39	Translation buffer invalidate all
TBIS	3A	Translation buffer invalidate single
TODR	1B	Time of year
TXCS	22	Console transmit control and status
TXDB	23	Console transmit data buffer
USP	03	User stack pointer

2-72 Console Commands

EXAMINE

Examples

```
❶ >>> EXAMINE 1000
      P 00001000 D05957C0
```

Displays the contents of address 1000. The default address space is physical and the default size is longword.

```
❷ >>> EXAMINE/INSTRUCTION .
      P 00001000 ADDL2   R7,R9
```

The `/INSTRUCTION` qualifier specifies to display data starting at the current address as an instruction.

```
❸ >>> EXAMINE/LONGWORD/NEXT=2 1000
      P 00001000 D05957C0
      P 00001004 0001FF8F
      P 00001008 00005100
```

The `/LONGWORD` qualifier resets the default data size to longword and `/NEXT=2` specifies to display the next two locations in addition to the starting address.

```
❹ >>> EXAMINE/GENERAL 4
```

Displays the contents of GPR 4. The default data size is longword.

```
❺ >>> EXAMINE SP
```

Displays the contents of the stack pointer (GPR 14).

```
❻ >>> EXAMINE P0BR
```

Displays the contents of the P0 base register (IPR 08).

EXIT

Terminates processing of the current command procedure. If the command procedure was called from within another command procedure, control returns to the calling procedure. Optionally, a return status can be specified.

Format

EXIT */qualifiers [status-code]*

Qualifiers

/ATTN

Returns CPU attention to the caller. Allows a command procedure that traps a CPU attention (with ON ATTN, see the ON command description) to pass the attention to the calling procedure for further processing.

/FAULT

Returns power fault to the caller. Allows a command procedure that traps a power fault (with ON FAULT, see the ON command description) to pass the fault to the calling procedure for further processing.

Parameters

status-code

A numeric value representing success or failure of the exiting procedure.

The status code can be an integer or an expression that equates to an integer value. An odd value indicates failure and an even value indicates success of the procedure. A status code of %X8000000 causes all command procedure file levels to exit.

If no status is specified, the default success status is returned, as follows:

- Bit 0 set = Success
- Bit 0 not set = Warning
- Bit 1 set = Error
- Bit 2 set = Fatal/severe error

FIND

The FIND command does one of the following:

- Specifies the CPU in which GPRs are to be modified.
- Finds the starting address of a 256-Kbyte block of good main memory.
- Finds the starting address of the restart parameter block.

Format

FIND */qualifier [blocks]*

Qualifiers

/CPU=cpu-id

The CPU in which GPRs are to be modified, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, GPRs in the default CPU are modified.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/MEMORY

Returns the starting address of the first page-aligned, 256-Kbyte block of main memory. VMB is loaded at this address.

When a number of blocks is specified with FIND/MEMORY, that number of pages are located, if possible. If that number of pages are located, the SP contains the base address plus 200_{16} .

/RPB

Finds the location of the restart parameter block (RPB) in main memory. If the RPB is found, the SP is set to the base address plus 200_{16} and the PC is set to the address in the RPB.

Parameters***blocks***

The number of main memory blocks to be located.

GOTO

Valid only in a command procedure. GOTO transfers control to a specified label.

Format

GOTO *label*

Parameters

label

The first item on a command line. A label cannot contain embedded blanks. When the GOTO command is executed, control passes to the command following the label. The label can precede or follow the GOTO statement in the command procedure.

All labels are procedure-level dependent except for labels that define subroutine entry points. Subroutine entry point labels are local to the current command procedure file level and must be unique. Labels must be terminated with a colon (:).

HALT

Stops macroprogram execution in a specified processor. The HALT command is usually not used for a multiprocessor because it can leave the multiprocessor in an undefined state.

Format

HALT [/CPU=*cpu-id*]

Qualifier

/CPU=cpu-id

The CPU to be halted, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is halted.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

HELP

The **HELP** command invokes the SPU HELP facility to display information about an SPU command or topic. You can abbreviate any topic name, but ambiguous abbreviations display all matches. In response to the Topic? prompt, you can:

- Type the command, topic, or subtopic on which help is needed.
- Type ? (question mark) to display a list of topics or subtopics.
- Press Return to exit to the previous topic or SPU command level.
- Enter Ctrl/Z to exit to the SPU command level.

IN-LINE HELP

For information on in-line help, see Section 1.2.

Format

HELP [*topic . . .*]

Parameters

topic

Specifies a command, topic, or subtopic on which help is needed. A topic and subtopic, separated by spaces, can be specified. If a topic is not specified, a list of all commands and topics in the HELP library is displayed.

Examples

```
1 >>> HELP
  .
  .
  Topic?
```

Displays information about using the **HELP** command; lists the commands and topics for which help is available; returns the **Topic?** prompt.

```
2 >>> HELP DEFINE
```

Displays information about all the **DEFINE** command subtopics.

```
3 >>> HELP DEFINE/KEY
```

Displays information about all the **DEFINE/KEY** command subtopics.

IF

Valid only in a command procedure. IF allows commands to be executed conditionally.

Format

IF *expression THEN command*

Parameters

expression

Specifies an expression to be evaluated. See Section 1.6 for details on expression syntax.

command

Specifies the command to be executed. The command can be any legal command including GOTO.

The REPEAT command can be used with the IF command to produce a WHILE command as long as the target command is repeatable. See HELP REPEAT for repeatable commands.

INITIALIZE

Initializes various system elements and data structures. When issued without qualifiers, initializes the boot set.

Format

INITIALIZE *[/qualifiers]*

Qualifiers

/BANK_MASK=mask

Valid only with /MEMORY, where mask is a hexadecimal number.

/BRIEF

Valid only with /KERNEL and /SCU. Passes a flag to the SYS\$SYSTEM:INIT.COMD command procedure specifying a brief initialization. The procedure then skips clock subsystem initialization, does not load the various structures, and does not save system state.

/CLOCK

Initializes the clock subsystem.

/CPU=cpu-id

The CPU to be initialized, where cpu-id is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is initialized. This qualifier can also be specified with /KERNEL, /MEMORY, /SCAN, and /SCU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DEBUG

/NODEBUG

Used with /SCM and /SJA.

/FIRMWARE=file-spec

Used with /SCAN or /SCM. Specifies loadable firmware.

/INTERLEAVE=type

Valid only with **/MEMORY**, where *type* is one of the following:

1WAY 2WAY 2WAY_1UNIT 2WAY_2UNIT 4WAY

/IO

Initializes the I/O subsystem. **INITIALIZE/IO** issues a **RESET** to all XMI-to-SCU adapters (XJAs) and scans the XMI bus for I/O adapters. The results are used to configure and load the JBox I/O physical address memory map (IPAMM).

/KERNEL

Initializes the processor after microcode is loaded and the scan subsystem is initialized. **INITIALIZE/KERNEL** should be issued before starting program execution.

/LOG

/NOLOG

Used with **/SCAN**, **/SCM**, and **/SJA**. Determines whether to display command results.

/MEMORY

Initializes the memory subsystem. **INITIALIZE/MEMORY** determines the preferred memory configuration from self-test results and configures and loads the JBox memory physical address memory map (MPAMM) and nonexistent physical address memory map (NPAMM).

NOTE

The */INTERLEAVE* qualifier can be used to override the preferred memory configuration.

/OUTPUT=file-spec

/NOOUTPUT

Valid only with **/MEMORY**. Determines whether the main memory bad block list is output to a file. If *file-spec* is not specified, the default file is [UCODE]DEFECT_LIST.SYS.

/POWER

Invokes the command procedure [SYSEXE]POWER.CMD to initialize the power subsystem.

/RESET

/NORESET

Used with **/SCAN**.

/RESTORE=file-spec
/NORESTORE

Valid only with **/MEMORY**. Determines whether to restore main memory bad block information. If *file-spec* is not specified, the default file is **[UCODE]DEFECT_LIST.SYS**.

/SCAN

Invokes the command procedure **SCAN.CMD** to initialize the scan subsystem.

/SCM

Initializes the scan control module.

/SCU

/NOSCU

Valid only with **/SCAN**. Initializes the SCU.

/SIMULATION

/NOSIMULATION

Valid only with **/SJA**. Specifies whether the SJA driver connects to the specified CPU or a simulation running on a remote host.

/SJA

Initializes the SPU-to-JBox adapter.

/SUNDANCE

/NOSUNDANCE

Valid only with **/SCM**. For use only with the MCU tester. Specifies whether the scan control module is controlling the VAX 9000 scan system or the MCU tester (which requires different firmware).

/TEST(D)

/NOTEST

Valid only with the **/MEMORY** qualifier. This qualifier determines whether test data is used to build the memory bitmap.

/TIMEOUT=seconds

Valid only with **/SCM**, where *seconds* is a decimal number.

/VOLUME

Initializes a disk or tape volume.

2-84 Console Commands
INITIALIZE

/WPT_AREA=kbytes

Valid only with /SCM, where kbytes is a decimal number.

Example

```
>>> INITIALIZE/KERNEL
[Initializing Master Clock Module]
%CLI-I-KNLREV, system kernel revision is A
[Initializing memory subsystem]
[Configured memory size is 10000000 (256 MBytes)]
[Initializing IO subsystem]
[Waiting for self test to complete]
[Setting up XJA adapter 0]
>>>
```

The INITIALIZE/KERNEL command display.

INQUIRE

Valid only in a command procedure. INQUIRE requests input from the terminal during execution of the procedure and assigns a symbol to the input string. The symbol is defined in the local symbol table.

Format

INQUIRE *[/qualifier] symbol-name prompt-string
prompt-symbol*

Qualifiers

/EXPRESSION

/NOEXPRESSION

Specifies whether data input to the prompt-string is to be evaluated as an expression.

/STRING

/NOSTRING

Specifies whether data input to the prompt-string is to be interpreted as a string.

Parameters

symbol-name

An alphanumeric name containing from 1 to 255 characters.

prompt-string | prompt-symbol

The prompt displayed when the INQUIRE command is executed. String values are automatically converted to uppercase, leading and trailing spaces and tabs are removed, and multiple spaces and tabs are compressed to a single space.

To prevent case conversion and retain multiple spaces and tabs, place quotation marks around the string. To use quotation marks in a string, enclose the string in quotation marks and use a double set of quotation marks within the string. If the string includes an at sign (@), enclose the string in quotation marks.

2-86 Console Commands
INQUIRE

If **prompt-string** is not specified, the command interpreter prompts for the string with:

`_PROMPT:`

If **prompt-symbol** is specified, the symbol value is used as the prompt string.

LABEL

Valid only in a command procedure. A label identifies a point in the procedure that can be referenced by GOTO and CALL commands.

Format

label: *[command]*

Parameters

label:

The first item on a command line. A label cannot contain embedded blanks. When the command is executed, control passes to the command following the label. The label can precede or follow the command that references it in the command procedure.

All labels are procedure-level dependent except for labels that define subroutine entry points. Subroutine entry point labels are local to the current command procedure file level and must be unique. Labels must be terminated with a colon (:).

command

A command, preceded by a space, can follow a label on the same line.

LOAD

Loads the contents of a file into memory or into a control store.

Main memory loading is assumed to be in EXE file format. If the EXE header is not present, the file is loaded as a binary image file. If the header is present, it is skipped and the remainder of the file is loaded as a binary image file.

If a control store qualifier is specified, the contents of the file are loaded into the specified control store. Data checking is provided for control store loads and an error is returned if the file format is incorrect.

Format

LOAD *[/structure] [/qualifier] file-spec*

Structures

/ABS

Loads a test case in absolute format. The absolute file format is the same as an EXE file without a header but with an additional block at the end of the file. This block is loaded into memory and is used to set the program starting address and the values of all GPRs and IPRs. The default file type is .CDB.

/CDB

Loads a configuration database. This file describes the system scan rings and contains the routines to load each structure. The default file type is .EXE.

/MAIN_MEMORY (D)

Loads main memory. This is the default if no other structure is specified. The file format is assumed to be standard .EXE. The entire file is loaded as a binary image.

/PEM

Loads the power and environmental monitor firmware. The file format is assumed to be binary data. The /START qualifier must be used to specify the start address. The standard address is 8000₁₆. The default file type is .BIN.

/RIC=ric-id

Loads the specified regulator intelligence card firmware. The file format is assumed to be binary data. The */START* qualifier must be used to specify the start address. The standard address is 8000₁₆. The default file type is .BIN.

/RING[=ring-id]

Loads fixed patterns into an internal scan ring. The input data file specifies ring selects, lengths, and data to be moved into a ring. The MCU broadcast ring (ring 15) can also be loaded. The default file type is .RING.

/STRUCTURE

Loads one or more structures from a binary file. The structure names are contained in the file. The default file type is .LOD.

/TEXT

Loads a structure from a MICRO2-format ULD file. This file is the same format as the input file to ULINK. It performs the function as the .LOD file loaded with */STRUCTURE* but is much slower.

/VECTOR

Loads a text file that specifies the elements of the vector registers. Any register can be loaded and a subset of the elements can be loaded for each vector.

Qualifiers

/CPU=cpu-id

The CPU for nonmemory load, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is loaded.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DATA (D)

/NODATA

Specifies whether the structure data is loaded or only the structure symbol table.

/LOG

/NOLOG

Determines whether load status is displayed.

/REVISION=version

Specifies the version to be compared against the revision information stored in the .LOD file for each structure.

/SCU

/NOSCU

Directs the load to the system control unit.

/START=address

Loads the structure, starting at the specified address.

/SYMBOL

/NOSYMBOL

Determines whether to load control store symbols.

/VERIFY

/NOVERIFY

Verifies a scan ring load.

Parameters

file-spec

The file containing the data to be loaded. Data is loaded into the specified data structure starting at location 0 unless ***/START*** is specified.

LOGOUT

Deletes the current command language interpreter (CLI) process and returns the process to the host system. Only remote processes can log out. Local processes ignore this command.

Format

LOGOUT *[/qualifiers]*

Qualifiers

/BRIEF

Echoes the LOGOUT command.

/FULL (D)

Displays the logout message, which includes user name and date and time of logout.

Examples

1 >>> LOGOUT/BRIEF

The LOGOUT/BRIEF format.

2 >>> LOGOUT
GIBRISH logged out at 15-APR-1988 14:23:45.30

The default LOGOUT (*/FULL*) message format.

MAIL

Mails the specified file to the specified recipient.

Format

```
MAIL [/qualifiers] file-spec recipient[, . . . ]  
    @recipient-list
```

Qualifiers

/FILE=file-spec

The file to send.

/SELF

/NOSELF

Determines whether the sender receives a copy of the mailed file.

/SUBJECT=quoted-string

The subject of the mailed file.

Parameters

file-spec

The file to be mailed.

recipient

Can specify one or more recipients or a list of recipients using the at sign (@).

MICROSTEP

Issues a specified number of clock cycles to a specified CPU.

Format

MICROSTEP *[/qualifiers] step-count*

Qualifiers

/BURST

/NOBURST

Specifies whether to burst the clocks for the number of cycles specified by step-count.

/CPU=cpu-id

Specifies the CPU to be stepped, where cpu-id is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If the cpu-id is BOOTSET, all CPUs and the SCU are stepped together. The SET STEP BOOTSET command permanently selects this option.

If the cpu-id is ALL, all CPU clocks are stepped.

If a CPU is not specified, the default CPU is stepped.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SPACEBAR(D)

/NOSPACEBAR

Determines whether to enter space bar step mode (SBSM) at the end of the specified step-count.

In SBSM, the prompt changes to STEP> and one clock cycle is issued each time the space bar is pressed.

SBSM is exited when any other key is pressed.

/SCU

/NOSCU

Specifies whether to step the SCU clocks. When the SCU clocks are stepped, memory timing is switched from STBY to STEP and one clock is issued (unless /BURST is specified). Memory timing is switched back to STBY and any tracepoints set in the SCU are scanned. The process is repeated for each count in the step-count. On the final count, active tracepoints are displayed followed by the microPC.

Parameters

step-count

A decimal value specifying the number of clock cycles to issue.

If step-count is nonzero, the SPU enters space bar step mode (SBSM, see /SPACEBAR) at the end of the specified count.

If step-count is zero, SBSM is not entered after the clock cycle is generated. This command form is used when a key is defined to step a processor instead of SBSM.

MOUNT

Mounts a specified volume on a specified device.

Format

MOUNT *device volume-name*

Parameters

device

If the device is not specified, the command prompts for it.

The asterisk (*) wildcard character can be used to specify disk device labels to mount the disk present. The label is read from the device and the system logical DISK\$label is defined as the name of the disk device.

The tape device label must be specified; the asterisk (*) wildcard character cannot be used. The system logical name TAPE\$label is defined as the name of the specified tape device. A warning message is issued if the tape label does not match the tape installed in the drive.

volume-name

The volume to be mounted on the specified device. If the volume is not installed on the device, an error message is issued. If the volume name is not specified, the command prompts for it.

NEXT

Executes the specified number of macroinstructions. The processor executes a HALT after the current instruction completes. The PC points to the instruction following the single-stepped instruction.

Format

NEXT *[qualifiers] [step-count]*

Qualifiers

/CPU=cpu-id

Specifies the CPU to be stepped, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU is stepped.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SPACEBAR (D)

/NOSPACEBAR

Determines whether to enter space bar step mode (SBSM) at the end of the specified *step-count*.

In SBSM, the prompt changes to `NEXT>` and one macroinstruction is executed each time the space bar is pressed.

SBSM is exited when any other key is pressed.

/VIRTUAL

/NOVIRTUAL

Specifies whether the instruction decode routine uses virtual or physical memory when decoding and displaying the next instruction.

Parameters

step-count

A decimal value specifying the number of macroinstructions to execute.

If *step-count* is nonzero, the SPU enters space bar step mode (SBSM, see */SPACEBAR*) at the end of the specified count.

If *step-count* is zero, SBSM is not entered after the instruction is executed. This command form is used when a key is defined to step a processor instead of SBSM.

ON

Valid only in a command procedure. Executes a specified command when a specified condition occurs during procedure execution.

NOTE

The ON command is executed only once and then disables itself. The ON target command must reexecute the ON command to reestablish the condition handler.

Format

ON *condition THEN command*

Parameters

condition

One of the following:

ATTN	Scan subsystem unhandled attention
ERROR	CLI process command error
FAULT	Power subsystem unhandled exception
SEVERE	CLI process command severe error
WARNING	CLI process command warning

command

Any valid CLI command.

OPEN

Opens a file for command-level reading and/or writing ASCII record-oriented files. Such files can store temporary data or test case results for later comparison. Files opened with the OPEN command remain open until closed with the CLOSE command.

Format

OPEN *[/qualifier] logical-name file-spec*

Qualifiers

/APPEND

Opens an existing file for write access, starting at the end of the file. The original file content cannot be modified.

/ERROR=label

On error, transfers control to label. Valid only in a command procedure.

/LOG

/NOLOG

Determines whether to display an informational message when the file is opened.

/READ(D)

Opens an existing file for read, starting at the beginning of the file.

/WRITE

Creates a new file.

Parameters

logical-name

Specifies a logical name to be assigned to the open file and placed in the process logical name table.

file-spec

The name of the file or device to be opened for input or output. Wildcard characters are not allowed in the file specification.

PURGE

Deletes all previous versions of files that match a given file specification.

Format

PURGE *[/qualifiers] [file-spec[, . . .]]*

Qualifiers

/CONFIRM

/NOCONFIRM (D)

Determines whether an affirmative response is required before a file is purged. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	The file is purged.
T (true)	The file is purged.
N (no)	The file is not purged.
F (false)	The file is not purged.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

/KEEP=number

Specifies the maximum number of file versions to be retained. If not specified, only the highest numbered version of the file is kept.

/LOG

/NOLOG (D)

Determines whether file-specs are displayed as each file is deleted.

Parameters

file-spec[, ...]

Specifies one or more files to be purged. If omitted, all files in the directory are purged. The version number cannot be specified. Wildcard characters can be substituted for the directory, name, or type fields or partial fields.

Examples

1 >>> PURGE

Deletes all but the highest numbered version of all files in the default directory.

2 >>> PURGE/KEEP=2

Deletes all but the two highest numbered versions of all the files in the default directory.

3 >>> PURGE/KEEP=2 TAMPA::DISK1:[EXAMPLE]*.LIS

Deletes all but the two highest numbered versions of each file with the file type .LIS in the directory EXAMPLE on remote node TAMPA.

READ

Reads a line from a specified file and assigns a symbol to the line.

Format

READ *[/END_OF_FILE=label] logical-name symbol-name*

Qualifiers

/END_OF_FILE=label

Control is transferred to label if the end-of-file condition is detected. Valid only in command procedures.

/ERROR=label

Control is transferred to label if an error condition is detected. Valid only in command procedures.

Parameters

logical-name

The logical name assigned by the OPEN command when the file was opened. STDIN, STDOUT, SYS\$INPUT, and SYS\$OUTPUT are the predefined logical names for the terminal or controlling file.

symbol-name

A 1- through 255-character alphanumeric name equated to the contents of the record being read. The first character must be an alphabetic letter, underscore (_), or dollar sign (\$). The command interpreter places the symbol name in the local symbol table for the current command level. If the symbol was previously defined, the READ command redefines it to the new value being read.

Example

```
>>> OPEN INFILE NAMES.DAT
>>> LOOP:
>>> READ/END_OF_FILE=ENDIT INFILE NAME
      .
      .
      .
>>> GOTO LOOP
>>> ENDIT:
>>> CLOSE INFILE
```

The **OPEN** command opens the file **NAMES.DAT** for input and assigns it the logical name **INFILE**. The **READ** command reads records from file **INFILE** into symbol **NAME**, and it specifies label **ENDIT** to receive control when the last **INFILE** record has been read. The procedure loops until all records in the file have been processed.

REBOOT

Reboots the SPU. The command causes the SPU to enter kernel mode and execute a HALT instruction. The SPM attempts to reboot if the Startup switch is set to Boot.

Format

REBOOT {/CONFIRM|/NOCONFIRM}

Qualifier

/CONFIRM (D)

/NOCONFIRM

Must be specified to avoid accidental execution. Determines whether an affirmative response is required before the reboot is executed. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	Reboot.
T (true)	Reboot.
N (no)	Do not reboot.
F (false)	Do not reboot.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

Example

```
>>> REBOOT/NOCONFIRM
```

The SPU is rebooted without confirmation.

RECALL

Recalls previously entered commands from the command buffer. The command buffer stores up to 20 commands; the most recently entered command is number 1. If the qualifier and parameters are omitted, the most recently entered command (that is, number 1) is recalled. The RECALL command is not given a number and cannot be recalled.

Format

RECALL [*/ALL*] [*command* | *index*]

Qualifier

/ALL

Displays all commands in the command buffer with their numbers. If */ALL* is specified, do not specify the command or index parameter.

Parameters

command

The leading substring of the command to be recalled. The leading substring is the minimum number of characters required to identify uniquely the wanted command from other commands in the buffer.

index

The recall buffer number of the command to be recalled.

2-106 Console Commands

RECALL

Example

```
>>> RECALL/ALL
 1 EVE RECALL.SDML
 2 help read
 3 mail
 4 help read
 5 help purge
 6 mail
 7 MAIL
 8 set ho 3d
 9 mail
10 WIPEM
11 mail
12 help logout
13 SPELL LOAD.SDML
14 WIPEM
15 MAIL
16 mail
17 help lexicals
18 WIPEM
19 mail
20 help lexicals f$leng
```

Typical contents of a full command buffer.

RENAME

Changes the directory, file name, file type, or version of an existing file specification.

NOTE

Files cannot be renamed from one device to another.

Format

RENAME *[/qualifiers] input-file-spec[, . . .] output-file-spec*

Qualifiers

/LOG

/NOLOG (D)

Determines whether the file-spec is displayed for each renamed file.

/CONFIRM

/NOCONFIRM (D)

Determines whether an affirmative response is required before a file is renamed. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	The file is renamed.
T (true)	The file is renamed.
N (no)	The file is not renamed.
F (false)	The file is not renamed.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

Parameters

input-file-spec, . . .]

One or more file specifications that are to be changed. Wildcard characters are allowed in the directory, file name, file type, or version number fields of the file specification. All files with specifications that match the wildcard fields are renamed.

output-file-spec

The new file specification. The device, directory, file name, and file type of the input-file-spec provide defaults for output-file-spec fields that are not specified or that contain a wildcard character. Wildcard characters in corresponding fields of input-file-specs and output-file-specs result in multiple rename operations.

Version numbers are applied to output-file-specs according to the first of the following rules that applies:

1. The output-file-spec specified version number.
2. The input-file-spec version number if either the input-file-spec or output-file-spec has an asterisk (*) in the version number field.
3. Version number 1 if the output-file-spec file name and file type fields do not currently exist.
4. The highest existing version number +1 if the output-file-spec file name and file type fields currently exist.

Example

```
>>> RENAME/LOG PS*.* [DIAG]ONTHA.*  
%CLI-I-RENAMED, DUA0:[TEST]PSN.QT;1 renamed to DUA0:[DIAG]ONTHA.QT;1
```

Any file in the default directory having PS as the first two characters of the file name is renamed to file name ONTHA in directory [DIAG] and has the same file type as the input file.

REPEAT

Repeats the execution of a command until Ctrl/C is typed, an error occurs, or an IF command expression becomes FALSE.

The REPEAT IF command is a special case that performs the same function as a WHILE command. Loops can be constructed as follows:

```
>>> LOOP_COUNT = 0
>>> REPEAT IF LOOP_COUNT.LEQ.100 LOOP_COUNT=LOOP_COUNT+1
```

The REPEAT command increments LOOP_COUNT until it reaches 101 and then returns control to the command interpreter.

If the target command of a REPEAT IF command is not repeatable, the command executes as if REPEAT was not entered.

Format

REPEAT *[/COUNT=number] command*

Qualifier

/COUNT=number

The command is repeated the specified number of times. If not specified, the command is repeated forever.

Parameters

command

The command to be repeated. If the command is not repeatable, execution terminates after the command is executed once.

RESET

Scans the RESET vector from the CDB into the specified CPU. Only scan latches are affected and all are modified. Then the command file [SYSEXE]RESET.CMD is executed. This file can contain any valid SPU commands.

Format

RESET */qualifiers*

Qualifiers

/CPU=cpu-id

The CPU to be reset, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is reset. All error checking is turned on.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Determines whether the SCU is reset. All error checking is turned on and all ports are turned off.

RESTORE

Restores previously saved scan state or SPU context information (see the SAVE command description). A qualifier must be specified; if not, the command prompts for additional input. The qualifiers allow the saved component states to be restored at the same time with one RESTORE command or individually with separate commands. Scan state is restored on units that are not running. If the system clocks are running on any of the specified units, an error message reports the running unit(s).

Format

RESTORE *qualifiers*

Qualifiers

/CPU=cpu-id

The CPU on which the scan state is to be restored, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU state is restored.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Specifies whether to restore the SCU scan state.

/SPU

/NOSPU

Specifies whether to restore the SPU context state.

RETURN

Valid only in a command procedure. Returns control to the procedure at the top of the CALL stack. The stack is decremented.

Format

RETURN

RUN

Loads the specified program and creates a job to execute it. The current terminal is attached to the new job until it completes unless the **/DETACH** qualifier is specified. If **/DETACH** is not specified, the CLI is suspended until the program completes.

Format

RUN *[/qualifier] file-spec [command-line]*

Qualifiers

/DEBUG

/NODEBUG

The **/DEBUG** qualifier causes the job to enter the debug wait state before starting execution. If this qualifier is specified without the remote debugger present, the job hangs. The job must be continued with the remote debugger.

/DETACHED

/NODETACHED

The **/DETACHED** qualifier starts a server job to execute the specified program. The CLI continues after starting the job.

/JOB_PRIORITY=level

Specifies the job priority level, from 1 (high) to 31 (low).

/KERNEL_STACK=size

Specifies the size of the kernel stack as a decimal number. Unlike the user stack, the kernel stack is not automatically increased when necessary.

/LOAD

/NOLOAD

The **/LOAD** qualifier causes the specified program to be unloaded and reloaded before execution. This ensures that the latest version of the image file is executed.

/MAXIMUM_MESSAGES=number

Specifies the maximum number (decimal) of messages that can be pending on the job's port.

/MODE={KERNEL | USER}

Specifies the initial execution mode.

/PARAMETERS=quoted-string

Passes parameters to the program to be run.

/POWER_RECOVERY

/NOPOWER_RECOVERY

The */POWER_RECOVERY* qualifier specifies that the job requires powerfail restart.

/PROCESS_PRIORITY=number

Specifies the process priority (decimal) for the first process in the job.

/USER_STACK=size

Specifies the initial size (decimal) of the user stack. If necessary, the user stack is automatically extended.

Parameters

file-spec

Specifies the program to be loaded and executed.

[command-line]

The command line is passed to the specified program, where it is interpreted (or ignored).

SAVE

Save scan state or SPU context state. The state is saved in a global area of SPU memory where it can be referenced by SPU software or restored (see the RESTORE command description). Scan state is typically saved as part of the system initialization procedure INIT.CMD to establish a system reset vector. The SPU uses the vector internally during the recovery phase of CPU/SCU error handling.

A qualifier must be specified; if not, the command prompts for additional input. The qualifiers allow the component states to be saved at the same time with one SAVE command or individually with separate commands. Scan state is saved on units which are not running. If the system clocks are running on any of the specified units, an error message reports the running unit(s).

Format

SAVE *qualifiers*

Qualifiers

/CPU=cpu-id

The CPU on which the scan state is to be saved, where cpu-id is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU scan state is saved.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Specifies whether to save the SCU scan state.

/SPU

/NOSPU

Specifies whether to save the SPU context state.

SCROLL

Valid only in screen mode. Scrolls a specified window left, right, up, or down a specified amount. In addition, microcode windows can be scrolled to the current, next, or previous microPC, or to a specified microcode address.

Format

SCROLL *[/qualifiers] [window-name]*

Qualifiers

/CURRENT

Valid only in microcode and EXAMINE windows. Scrolls the window to the microcode listing location that corresponds to the current microPC of the control store associated with the window.

/DOWN[=*lines*] (*D*)

Scrolls the window down (forward). The default is six lines.

/LEFT[=*columns*]

Valid only in microcode windows. Scrolls the window to the left. The default is eight columns.

/NEXT[=*lines*]

Valid only in microcode windows. Scrolls the window forward the specified number (decimal) of lines. If lines are not specified, scrolls to the microcode listing location corresponding to the next microPC.

/PREVIOUS[=lines]

Valid only in microcode windows. Accurate only if the clocks are single-stepped. Scrolls the window backward the specified number (decimal) of lines. If lines are not specified, scrolls the window to the microcode listing location corresponding to the previous microPC.

/RIGHT[=columns]

Valid only in microcode windows. Scrolls the window to the right. The default is eight columns.

/TO=address

Valid only in microcode and EXAMINE windows. Scrolls the window to the microcode listing location corresponding to the specified hexadecimal address.

/UP[=lines]

Scrolls the window up (backward). The default is six lines.

Parameters

window-name

The name of a window created by a CREATE/WINDOW command.

If a window is not specified and if the currently selected window type supports the specified scrolling operation, the currently selected window is scrolled.

SELECT

Selects and optionally relocates and resizes a window. The command window is resized as required. If screen mode is suspended (by SET SCREEN OFF), it is automatically enabled.

NOTE

The command window cannot be selected.

Format

SELECT *[/NEXT] window-name [AT location]*

Qualifier

/NEXT

Selects the next least-recently created window.

Parameters

window-name

The window to be selected. The specified window becomes the default window for the SCROLL command, and the window name is highlighted in the associated information line.

location

The screen segment the window is to occupy, specified by one of the following keywords:

Keyword	Screen Segment	Keyword	Screen Segment
W1	Whole screen	H1	First (upper) (D)
		H2	Second (lower)
T1	First (upper)	Q1	First (upper)
T2	Second	Q2	Second
T3	Third (lower)	Q3	Third
		Q4	Fourth (lower)

The window is resized to conform to the new location. If the window type does not support resizing, the command fails.

NOTE

The bottom five lines of the screen are reserved for the command window.

SEND

Broadcasts a message to all terminals or a specified terminal.

Format

SEND *[/qualifier] [terminal-name] string*

Qualifiers

/ALL

Broadcasts the message string to all terminals. If */ALL* is specified, do not specify the terminal-name parameter.

/BELL

Sends one bell character with the message.

/OPCOM

Sends the message through VMS to the operator communications manager.

/URGENT

Appends an urgent header to the message.

Parameters

terminal-name

The terminal to receive the message. If */ALL* is specified, do not specify the terminal-name parameter.

string

The message text. It is limited to one line and is displayed on the line following the SEND header. The header displays the message source, urgency, and time.

SENSE

Determines the configuration and current status of various VAX 9000 system components. During system initialization, the command options determine the appropriate data files and configuration, and whether to attempt a cold boot, warm boot, or hot boot.⁴

Format

SENSE *option*

⁴ Command variants, of the form SENSE OPTION, are listed in the table of contents and described separately on the following pages.

SENSE CLOCK

Reads the current state of the clock subsystem and the MCM revision and serial numbers.

Format

SENSE CLOCK

SENSE CPU

Executes scan hard-core tests:

- 0 (SCI loopback)
- 1 (CD loopback)
- 4 (sense rings)

to determine the revision and configuration of the specified CPU. The command destroys the state of the CPU scan subsystem and does not execute if the CPU clocks are running.

Format

SENSE CPU *cpu-id*

Parameters

cpu-id

The CPU to be sensed, specified by numeric literal 0, 1, 2, or 3.

SENSE IO

Displays information about the I/O subsystem.

Format

SENSE IO

SENSE POWER

Reads the power subsystem status to determine the cabinet configuration, BBU state, and power status. This information and the front panel switches determine the power-on action.

Format

SENSE POWER

SENSE SCU

Executes scan hard-core tests:

- 0 (SCI loopback)
- 1 (CD loopback)
- 4 (sense rings)

to determine the revision and configuration of the SCU. This command destroys the state of the SCU scan subsystem and does not execute if the SCU clocks are running.

Format

SENSE SCU

SENSE SYSTEM

Executes a **SENSE POWER** command followed by a **SENSE CLOCK** command. This sets up the state of the clock and power subsystems (which must be known before executing the **SENSE CPU** and **SENSE SCU** commands). Next, each CPU and then the SCU are sensed.

Format

SENSE SYSTEM */qualifiers*

Qualifiers

/COMPARE

Compares the system part and serial number information to the saved history records. If used with **/LOG**, any differences will be displayed.

/LOG

/NOLOG

Determines whether to display command results.

SET

Defines or modifies characteristics of the processors or the service processor (SPU) subsystem.⁵

Format

SET *option*

⁵ Command variants, of the form SET OPTION, are listed in the table of contents and described separately on the following pages.

SET ATTN_ACTION

Defines a command to be executed when the scan subsystem generates an unhandled scan attention.

Scan attentions are normally serviced by the error handling subsystem. If an attention is not recognized, or the error handler is disabled, the attention may be delivered to all CLI processes that have established an ATTN_ACTION.

Format

SET ATTN_ACTION *command*

Parameters

command

The command to be executed following a scan attention.

Example

```
>>> SET ATTN_ACTION @FIX_IT
```

Executes the procedure FIX_IT.CMD following an unhandled scan attention.

SET AUTOBOOT

Controls the automatic bootstrap flags.

Format

SET AUTOBOOT {*ON/OFF*}

Parameters

ON/OFF

Determines whether automatic bootstrap is enabled (ON) or disabled (OFF).

SET BI_DEVICES

Loads the BI device names and IDs from the specified file.

Format

SET BI_DEVICES *[parameters]*

Parameters

file-spec

Specifies a file that translates BI node IDs into device names. If *file-spec* is not specified, the default file is [UCODE]BI_DEVICES.DAT.

device[, ...]

Specifies one or more devices. If not specified, the default file [UCODE]BI_DEVICES.DAT is used.

SET BOOTFLAGS

Sets the default value of the bootstrap flags in R5.

Format

SET BOOTFLAGS *value*

Parameters

value

A hexadecimal value to be loaded into R5.

SET BOOTSET

Specifies the set of CPUs that operates in an SMP environment when the system is next bootstrapped.

Format

SET BOOTSET *[/qualifiers] cpu-id [, . . .]*

Qualifiers

/ENABLE=(cpu-id [, . . .])

Explicitly includes one or more CPUs in the boot set. To specify more than one CPU, separate the CPU identifiers with commas and enclose the list in parentheses.

/DISABLE=(cpu-id [, . . .])

Explicitly excludes one or more CPUs in the boot set. To specify more than one CPU, separate the CPU identifiers with commas and enclose the list in parentheses.

/PRIMARY=cpu-id

Specifies the CPU to become the primary CPU when the system is next bootstrapped. The specified CPU must be a member of the boot set.

Parameters

cpu-id

Specifies the CPUs that form the boot set. Included CPUs are enabled, excluded CPUs are disabled.

CPUs are specified by numeric literals 0, 1, 2, and 3. If the primary CPU is not specified (see */PRIMARY*) or the previous primary is not available, the lowest numbered CPU in the list becomes the primary CPU.

2-134 Console Commands
SET BOOTSET

Examples

1 >>> SET BOOTSET 0,1,2,3

In a quad CPU system, includes all available CPUs in the boot set. CPU0 is the boot primary by default.

2 >>> SET BOOTSET/PRIMARY=1 0,1

Includes CPU0 and CPU1 in the boot set and specifies CPU1 as the boot primary.

SET CLOCK

Modifies the state of the master clock subsystem.

Format

SET CLOCK *[/qualifiers] [ON(D)|OFF]*

Qualifiers

/ATTENTION[=*attention-name*]

/NOATTENTION

Determines whether a clock subsystem fault is reported to the console as an attention and entered in the error log. The attention-name argument can be:

- PHASE_ERROR

Indicates that a clock synchronization error exists between clock distribution chips on the MCUs.

- UNLOCKED_ERROR

Indicates that the master clock module (MCM) frequency synthesizer has lost internal synchronization.

Both attentions are enabled by default.

/CPU=*cpu-id*

Specifies the CPU in which the clock subsystem is to be modified, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU's clock subsystem is modified.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/EMULATION

/NOEMULATION

When emulation is specified, MCM control is disabled and SCM driver loopback functions are enabled for system testing.

/FREQUENCY=value

The frequency, in MHz, of the master clock.

The frequency value must be in the range of 332 to 580₁₀. If the value is not a multiple of four, the frequency is automatically rounded up to the next higher multiple of four.

/INTERVAL=cycles

The interval, in machine cycles, during which system clocks are turned off. For example, if cycles is 0, clocks are generated every machine cycle; if cycles is 1, clocks are generated every other machine cycle.

The interval value must be in the range of 0 to 255₁₀.

/LOG

/NOLOG

Determines whether an informational message is displayed when the command is executed.

/POSITION=value

The position of reference clock relative to master clock. The position value must be in the range of 0 to 1023₁₀.

CAUTION

This value is set during initialization and should not need to be modified. If this value is 0, the MCM may not be able to synchronize the main and reference clocks.

/SAMPLE_RATE=hertz

The sampling rate, in Hz, of the master-clock-to-reference-clock phase alignment relationship. The rate argument can be one of the following:

- OFF (disable the sample loop)
- 488
- 976
- 1950

The normal sampling rate is 488 Hz.

/SCU

/NOSCU

Specifies the system control unit clock group. Valid only when setting the clock state on or off.

/SYNCH=synch-option

The control mode for master-clock-to-reference-clock phase alignment. The synch-option argument can be one of the following:

- AUTOMATIC
The MCM automatically controls the phase alignment.
- MANUAL
The console, through the MCM position register, controls the phase alignment.
- NONE
Phase alignment is disabled.

Parameters

ON/OFF

Turns clocks on or off in the specified CPU(s). If not specified, the clocks are turned on.

2-138 Console Commands

SET CLOCK

Examples

1 >>> SET CLOCK ON

Starts the clock in the default CPU.

2 >>> SET CLOCK/CPU=ALL ON

Starts the clock in all available CPUs.

3 >>> SET CLOCK/CPU=(0,1)/SCU ON

Starts the clocks in CPU0 and CPU1 and in the SCU.

4 >>> SET CLOCK/CPU=(0,1)/SCU OFF

Stops the clocks in CPU0 and CPU1 and in the SCU.

5 >>> SET CLOCK/FREQUENCY=400

Sets the clock frequency to 400 MHz.

6 >>> SET CLOCK/FREQ=401/LOG
%CLI-I-NOTMULT4, specified frequency has been rounded to 404 MHz

Attempts to set a clock frequency that is not a multiple of 4 MHz. The CLI displays an informational message and automatically rounds the frequency up to the next multiple of 4 MHz. Note that /NOLOG would inhibit the informational message.

SET COLD_START

Sets or clears the cold-start flag for the specified CPU. This command is used primarily for debug.

Format

SET COLD_START [/CPU=*cpu-id*] {ON|OFF}

Qualifier

/CPU=cpu-id

Specifies the CPU in which the cold-start flag is to be set or cleared, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the cold-start flag is set or cleared in the default CPU

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

Parameters

ON/OFF

Determines whether the cold-start flag is set.

SET COMMAND

Adds commands to the process command table.

Format

SET COMMAND *[/qualifiers] file-spec*

Qualifiers

/CLEAR

/NOCLEAR

Clears the command tables.

/LOG

/NOLOG (D)

Determines whether to display memory and keyword usage for each new command.

Parameters

file-spec

Specifies the name of a command definition file. The default file type is .CLD.

Example

```
>>> SET COMMAND TEST
```

Adds the commands in TEST.CLD to the process command table.

SET CPU

Specifies the default CPU for commands that reference a CPU.

Format

SET CPU *[/[NO]LOG] cpu-id*

Qualifier

/LOG

/NOLOG

Determines whether to display command results.

Parameters

cpu-id

The default CPU. The *cpu-id* can be an integer value or symbolic name, as follows:

Integer	0, 1, 2, or 3
BOOTPRIMARY	Next primary processor
PRIMARY	Current primary processor

NOTE

The SET CPU command does not allow a list of processors.

Examples

1 >>> SET CPU 0

Specifies CPU0 as the default CPU.

2 >>> SET CPU PRIMARY

Specifies the PRIMARY CPU as the default CPU.

SET CYCLE

Modifies the clock cycle counter.

Format

SET CYCLE *[/qualifiers] count*

Qualifiers

/CPU=cpu-id

The CPU in which the clock cycle counter is to be modified, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU's clock cycle counter is modified.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/INTERVAL=value

The number of time units per cycle.

/SCU

/NOSCU

Determines whether the system control unit clock cycle counter is modified.

Parameters

count

The count to be loaded into the clock cycle counter.

The clock cycle counter is the time base for the SET TRACE and SET PATTERN commands and is incremented each time the clock is stepped. A null count clears the counter.

Examples

1 >>> SET CYCLE

Clears the clock cycle counter in the default CPU.

2 >>> SET CYCLE/CPU=1/INTERVAL=1000 1

In CPU1, sets the clock cycle counter value to 1 and sets the interval to 1000.

SET DEFAULT

Changes the default device and/or directory name. The new default is applied to all subsequent file specifications that do not include a device or directory name. A fatal message is displayed if the target directory does not exist.

Format

```
SET DEFAULT directory-spec[:]
```

Parameters

directory-spec[:]

Specifies the device and/or directory name to be used as the default device or directory in file specifications. Terminate a physical device name with a colon (:). Enclose a directory name in brackets ([]). The relational directory specifiers [-] or [--] can be specified to set default one level above the current level. For example:

```
SET DEFAULT [-.FILES]
```

The device and directory can also be specified as a logical name. For example:

```
SET DEFAULT SYS$LOGIN
```

Examples

```
❶ >>> SET DEFAULT [UCODE]
```

Changes the default directory to [UCODE]. The default device does not change.

```
❷ >>> SET DEFAULT DISK2:
```

Changes the default device to DISK2. The default directory does not change.

```
❸ >>> SET DEFAULT DISK0:[TEST]
```

Changes the default device to DISK0 and the default directory to [TEST].

SET ERROR_HANDLING

Sets the state of the CPU/SCU error handling system.

Format

SET ERROR_HANDLING *[/qualifiers]* {ON|OFF}

Qualifiers

/MATCH

/NOMATCH

Determines whether macroPC, microPC, and physical address match handling for the CPUs is on or off. If */MATCH* is specified and a match condition is detected, the error handling system signals a match exception to the CLI. If */NOMATCH* is specified, detected match conditions are treated as errors.

/RECOVERY

/NORECOVERY

Determines whether error recovery for the CPUs and SCU is enabled. If */RECOVERY* is specified, the error handling system attempts to recover from any errors that occur. If */NORECOVERY* is specified, error recovery is not attempted and errors are signaled to the CLI.

/REPORTING

/NOREPORTING

Determines whether error log entries are generated for all recovered and nonrecovered CPU and SCU errors.

Parameters

ON|OFF

Determines whether the CPU/SCU error handling system is on or off. If error handling is ON, errors are reported as specified by the qualifiers. If error handling is OFF, errors are reported to the CLI.

SET FAULT_ACTION

Specifies a command to be executed when an unhandled exception is detected in the power subsystem.

Format

SET FAULT_ACTION *command*

Parameters

command

Specifies the command to be executed following a power exception.

Power subsystem exceptions are normally serviced by the error handling subsystem. If the error handler fails to recognize an exception or is disabled, all CLI processes that established a POWER FAULT action are notified.

SET ISOLATION

Loads isolation data for the MCU tester.

Format

SET ISOLATION *[/qualifiers] file-spec*

Qualifiers

/CPU=cpu-id

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

Displays command results.

/SYBIL

Specifies isolation for SYBIL patterns.

Parameters

file-spec

The file containing the isolation data.

SET KEEP_ALIVE

Controls the state of the SPU keep-alive monitor.

Format

SET KEEP_ALIVE [/CPU=*cpu-id*] {ON|OFF|MANUAL}

Qualifier

/CPU=*cpu-id*

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

Parameters

ON/OFF/MANUAL

Determines whether the keep-alive monitors in the CPUs are enabled and how the SCM polls the monitors.

If a keep-alive failure is detected when MANUAL is enabled, the file SYS\$SYSTEM:KAF.CMD is executed.

If a keep-alive failure is detected when ON is enabled, the CLI handles the failure internally by logging the error and rebooting the CPU.

Examples

1 >>> SET KEEP_ALIVE ON

Turns the keep-alive monitor on.

2 >>> SET KEEP_ALIVE OFF

Turns the keep-alive monitor off.

SET LABELS

Appends bit-range labels to each line of data displayed by the SPU.

Format

SET LABELS {*ON/OFF*}

Parameters

ON/OFF

Determines whether bit-range labels are displayed.

Example

```
>>> SET LABELS ON
>>> EXAMINE ECS[0]
%CPU0.ECS[0]<145:0> = F9007702 03621800 08601F00 0206942E ! Bit 127:0
                                     2C3F3 ! Bit 145:128

>>> SET LABELS OFF
>>> EXAMINE ECS[0]
%CPU0.ECS[0]<145:0> = F9007702 03621800 08601F00 0206942E
                                     2C3F3
```

The labels indicate that the first line of data shows the value of bits 127 through 0, and the second line bits 145 through 128.

SET LOGGING

Opens a specified file and copies all terminal output to the file. Up to five logging files can be open at the same time. The command does not affect normal terminal output. All terminal output does not go to the logging file. For example:

```
>>> WRITE SYS$OUTPUT "FOO"
```

will not go to the logging file.

Format

```
SET LOGGING [/qualifiers] {ON|OFF}
```

Qualifiers

/ALL

Specifies that the command affects all log files.

/DISABLE

Temporarily disables output logging to the specified file(s). The file remains open.

/ENABLE

Reenables output logging to the specified file(s).

/FILE[=file-spec]

The log file. If file-spec is not specified, the default is CONSOLE.LOG in the current directory.

If /FILE is not specified, the output is sent to the printer port. SET LOGGING ON enables the printer port and SET LOGGING OFF disables the printer port.

Parameters

ON/OFF

Turns terminal output logging on and off. To turn off logging to a file, the file must be specified exactly as it was when logging was enabled.

Examples

1 >>> SET LOGGING ON

Copies terminal output to remote printer port.

2 >>> SET LOGGING/FILE=[UCODE]UCODE.LOG

Copies terminal output to [UCODE]UCODE.LOG.

SET MESSAGE

Determines the format for error, warning, and informational messages.

NOTE

At least one message field must be displayed. Specifying /NOFACILITY/NOSEVERITY/NOIDENT/NOTEXT displays all fields.

Format

SET MESSAGE */qualifiers* [file-spec]

Message Format

All messages from the SPU command language start with either a percent sign (%) or a minus (-) in column 1. The percent sign denotes the start of a message, the minus indicates message continuation. The message format is:

```
%Facility-Severity-Ident, Text  
-Facility-Severity-Ident, Text
```

where:

Field	Description
Facility	Three-character identification code for the facility that produced the message. For example, CLI is the code for the command language interpreter.
Severity	Single-character code for the message severity, as follows: S = Success I = Informational W = Warning E = Error F = Fatal error
Ident	A variable-length, message-unique field. It is usually an acronym for the message.
Text	A brief error description and possible remedy.

Qualifiers

/FACILITY

/NOFACILITY

Determines whether the facility name prefix is displayed.

/IDENT

/NOIDENT

Determines whether the identification prefix is displayed.

/SEVERITY

/NOSEVERITY

Determines whether the severity level is displayed.

/TEXT

/NOTEXT

Determines whether the message text is displayed.

Parameters

file-spec

The current message file.

SET PATTERN

Specifies a set of signals that is driven by patterns in a file.

Format

```
SET PATTERN [/qualifiers] pattern-name  
[/qualifiers] signal-list  
[/qualifiers] @file-name  
/SYBIL pattern-name
```

Qualifiers

/ABSOLUTE

Sets absolute times. The time field is controlled by the count variable (see SET CYCLE) and can specify an absolute or a relative count. Absolute counts specify the time at which the pattern should be applied. Absolute times less than the current count are discarded.

/ALL

Select all patterns. If /ALL is specified, do not specify the pattern-name parameter.

/CPU=cpu-id

Specifies the CPU to which patterns are applied, where cpu-id is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, patterns are applied to the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DISABLE

Disable the pattern. Patterns can be disabled to prevent signals from being driven. The pattern definition is maintained and is resumed when the pattern is enabled.

/ENABLE

Enable the pattern. When a pattern is enabled, patterns that appear before the current time are discarded.

/FILE=file-name

The pattern file file-name.

/LOG

/NOLOG

Determines whether changes are displayed.

/NAME=pattern-name

Sets a pattern point name.

/ODOMETER[=window-name]

Displays changes in an odometer window.

/RELATIVE

Sets relative time. The time field is controlled by the count variable (see SET CYCLE) and can specify an absolute or a relative count. Relative counts are added to the current count to determine at which time the pattern should be applied.

/SCU

/NOSCU

Determines whether pattern points apply to the system control unit.

/SYBIL

Remote patterns can be executed using the /SYBIL qualifier. The SET/SYBIL command creates a network link to the node defined by logical name SYBIL\$NODE.

/VERIFY

/NOVERIFY

Determines whether to display a message when patterns are applied.

Parameters

signal-list

The signals to be driven. To specify more than 1 (up to 256) signal, separate the signal names with commas, as follows:

```
>>> SET PATTERN [/qualifiers] signal-spec[,signal-spec...]
```

The asterisk (*) wildcard character can be used to specify several like-named signals. For example:

```
>>> SET PATTERN [/qualifiers] E3.*
```

file-name

The file containing the pattern.

pattern-name

The default pattern-name format is PATTERN dd , where dd is a decimal number to make the name unique. The pattern can also be explicitly named with the /NAME qualifier. When the pattern is established, it must be referenced by its assigned name.

For SYBIL patterns, the pattern-name is part of the name of the file that contains the patterns.

Description

The pattern file format is shown below:

```
! Comments
! Time   Signal States
10      11100111111
11      11000111111
```

A pattern deletes itself when no more activity is scheduled.

SET PERSONAL_NAME

Sets the user's personal name for use with the MAIL command.

Format

SET PERSONAL_NAME *quoted-string*

Parameters

quoted-string

A user-specified string.

SET POWER

Modifies the state of the power subsystem.

The power subsystem is partitioned into voltage buses in processor cabinets. The cabinets can be selected as a set to modify all voltage groups within a cabinet, or each individual voltage can be modified.

Only the local CLI process can execute the SET POWER command.

Format

SET POWER *[/qualifiers]* *[ON|OFF]*

Qualifiers

/BUS=bus-name

Specifies the power or I/O buses affected by the command, as follows:

Table 2-13 System Power Buses

Bus Name	Volts	Model 200 Cabinet	Model 400 Cabinet	Tester
A	+5.0	SCU	SCU	MCM
B	+5.0	BBU	BBU	None
C	-3.4	SCU/CPA	SCU	None
D	-5.2	SCU/CPA	SCU	None
E	-3.4	CPB	None	None
F	-5.2	CPB	None	None
J	-3.4	None	CPA	UNIT 0
K	-5.2	None	CPA	UNIT 0
M	-3.4	None	CPB	UNIT 1
N	-5.2	None	CPB	UNIT 1

Table 2-14 System I/O Buses

Bus Name	Bus Name
XMIA_7214A	XMIB_7214A
XMIA_7214B	XMIB_7214B
XMIA_7215A	XMIB_7215A
XMIA_7215B	XMIB_7215B

/CABINET=cabinet-id

The cabinets to which the command applies. The cabinet-ids represent the buses that supply the cabinet (see */BUS*).

Table 2-15 System Cabinets

Cabinet ID	Model 200 Buses	Model 400 Buses	Tester Buses
SCU	A, B, C, D	A, B, C, D	None
CPA	C, D	J, K	None
CPB	E, F	M, N	None
UNIT_0	None	None	J, K
UNIT_1	None	None	M, N
5VOLT	None	None	A, B

/COUNTERS

Clears the counters in the power subsystem driver.

/IO

Modify the state of the I/O power supplies.

/MARGIN=margin-keyword

/NOMARGIN

Determines the voltage conditions for checking system operation. Margin keywords are as follows:

HIGH
LOW
NOMINAL
NONE

Parameters

ON(D) / OFF

Turns power on or off. If not specified, the default is ON.

SET PROMPT

Defines a new string for the CLI command prompt.

Format

SET PROMPT *[prompt-string]*

Parameters

prompt-string

Specifies the string to replace the default CLI prompt (>>>). The string can consist of one or more ASCII characters. Any ASCII character can be used in the string.

To include spaces or lowercase letters in the string, enclose the string in quotation marks. Otherwise, letters are automatically converted to uppercase, and leading and trailing spaces are removed.

If no string is specified, the default prompt is restored.

NOTE

The default prompt is required by some external utilities.

Example

```
>>> SET PROMPT "What now?> "  
What now?> SHOW TIME  
18-MAR-1989 10:29:15
```

The **SET PROMPT** command replaces >>> with What now?>.

SET RADIX

Specifies a default radix for parsing numeric literals.

Format

SET RADIX *radix-name*

Parameters

radix-name

The new default radix is specified by one of the following keywords:

- HEXADECIMAL
- DECIMAL
- OCTAL
- BINARY

The default radix is command-specific. In general, numbers that are values for processor variables are hexadecimal and expression numbers are decimal.

The default radix can be overridden on the command line by preceding the numeric value with one of the following radix operators:

- %D (decimal)
- %O (octal)
- %B (binary)
- %X (hexadecimal)

If the radix is other than hexadecimal, the specifier precedes displayed numbers.

SET REMOTE

Controls access by the remote port. Only the local CLI process can execute this command.

Format

SET REMOTE *[/[NO]PASSWORD] [ON|OFF]*

Qualifier

/PASSWORD=string

/NOPASSWORD

Determines whether a password is needed for remote port access.

The console software maintains one password for remote port access. The password can be set only by the local CLI process and must be entered before the remote port is allowed access to the CLI process. The password is not displayed.

Parameters

ON|OFF

Enables or disables access by the remote port.

SET REVISION

Sets the revision level of the operating system or planar.

Format

SET REVISION *[qualifiers] revision-string*

Qualifiers

/CPU=cpu-id

Valid only with **/PLANAR**. Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/OS

Specifies that the operating system's revision is to be set.

/PLANAR

Specifies that the planar's revision is to be set.

/SCU

Valid only with **/PLANAR**. Specifies the SCU.

Parameters

revision-string

The operating system or planar revision level. For the planar, a single letter in the range A to Z.

SET SCI

Sets the state of the scan control interconnect of a specific port.

The primary use of the SET SCI command is for debugging solid faults in the scan distribution subsystem. Note that any intervening scan commands, including SHOW CLOCK, potentially alter the state of the SCI. If the SET SCI or SHOW SCI commands are in use, no other scan activity should be allowed.

Format

SET SCI *[/qualifiers]*

Qualifiers

/BROADCAST={0 | 1}

/NOBROADCAST

Sets the SCI BDCST line to 1 to select all MCUs on a port for a specified operation. Sets the SCI BDCST line to 0 to use the address on the SCI SELECT lines.

/BYPASS={0 | 1}

/NOBYPASS

Determines whether the port is in bypass mode. Sets the SCI BYPASS to 1 to bypass the extra latches in the data path or to 0 to use the latches.

If the port is not in bypass mode (*/BYPASS=0*), the command issues two extra B-phase clocks to ensure that data is propagated to the output of the two-stage delay in the SCM module. The two clocks are issued before any specified clock options. For example:

```
>>> SET SCI/DATA=1/CLOCK=STEP_A
```

will cause the following sequence:

1. Set data out to 1.
2. Issue two B-phase clocks.
3. Issue one A-phase clock.

Note that */CLOCK=FORCE_B* will defeat this propagation and data out is unpredictable.

2-166 Console Commands
SET SCI

Data in is latched by the two B-phase clocks and is reported as the latched value, not the actual value of the line. Except when performing manual port loopback tests, this command is expected, and highly recommended, to be used with the port in bypass mode.

/CDS={0|1}
/NOCDs

Sets the SCI CDS line to 0 to hold the address on the SCD SCI SELECT lines; sets the SCI CDS line to 1 to load the address.

/CLOCK=state

Sets the SCI clock to one of the following states for the current SCI command. The clock function is always executed after any B-phase clocks required to propagate data to the I/O pins.

State	Description
[NO]FORCE_A	[Clear or] set the SCI A CLK line.
[NO]FORCE_B	[Clear or] set the SCI B CLK line.
RUN	Free run both SCI CLK lines.
STEP	Step one cycle of both SCI CLK lines.
STEP_A	Step one cycle of the SCI A CLK line.
STEP_B	Step one cycle of the SCI B CLK line.
STOP	Stop both SCI CLK lines.

/DATA={0|1}

Specifies the value on the SCI DATA IN line (this is data from the SCM module). A two-stage timing delay in the SCI data path can be disabled with the SCI BYPASS. To ensure that the data set with the /DATA qualifier is propagated to the SCI pin, the SCM module issues two B-phase clocks if SCI BYPASS is not asserted (see /BYPASS).

/DEFAULT

Overrides the value set with any other qualifier and sets the following SCI lines to the state indicated:

SCI Line	State
SCI BDCST H, L	0, 1
SCI BYPASS H, L	0, 1
SCI CDS H, L	0, 1
SCI FCT 1 H, L	0, 1
SCI FCT 0 H, L	0, 1
SCI SEL 3 H, L	0, 1
SCI SEL 2 H, L	0, 1
SCI SEL 1 H, L	0, 1
SCI SEL 0 H, L	0, 1
SCI A CLK H, L	0, 1
SCI B CLK H, L	0, 1

/FUNCTION=state

Sets the SCI FCT[1:0] lines to the specified state, as follows:

State	Description
NOP	Set FCT lines to 0 (NOP).
SCAN	Set FCT lines to 1 (SCAN).
STRAM_LOAD	Set FCT lines to 2 (STRAM_LOAD).
SCAN_LOAD	Set FCT lines to 3 (SCAN_LOAD).

2-168 Console Commands
SET SCI

/PORT=port-id

Specifies the port affected by the command where port-id is the port name or number, as follows:

Port Name	Port Number
CPU0	0
CPU1	1
CPU2	4
CPU3	5
SCU	2
MCM	6

/SELECT={0|1}

Sets the SCI SELECT [3:0] lines to the specified value.

SET SCM

Controls operation of the scan control module.

Format

SET SCM *[/qualifiers]*

Qualifiers

/ATTENTIONS=mask

/NOATTENTIONS

Allows the SCM to determine whether CPU and SCU attentions are enabled. Mask bit 4 is for the SCU and bits [3:0] are for CPU3 through CPU0. The INIT.CMD procedure sets up and controls the mask. This qualifier does not control MCU attention enable bits and does not affect MCM attentions.

/BI_VERIFY

/NOBI_VERIFY

Determines whether the SCM verifies BI transfers by comparing the source and destination buffers after the transfer is complete. Verification is normally disabled.

/BYPASS

/NOBYPASS (D)

Determines whether the SCI normal mode latches are enabled. Normally, the latches are not bypassed. Bypass mode is only for diagnostic use, and scan rates faster than 400 ns are not reliable in bypass mode.

/CACHE

/NOCACHE

Determines whether the SCM internal scan ring cache is enabled. When */CACHE* is specified, the SCM caches scan data to minimize scan I/O operations and to improve performance of error handling software and initialization procedures. The cache is automatically swept when the system clocks are started or when the cache is disabled.

2-170 Console Commands
SET SCM

/RATE=scan-rate

One of the following:

100NS (D)	200NS	300NS	400NS
800NS	1600NS	3200NS	EXTERNAL

The SCAN.CMD procedure selects the 100 ns default scan rate, which should not be changed.

/SCAN_VERIFY

/NOSCAN_VERIFY

Determines whether the SCM verifies scan operations by performing a pattern test on the target ring before applying scan-write data or after reading scan-read data. Normally this feature is enabled.

/VECTOR_PROCESSOR=mask

/NOVECTOR_PROCESSOR

The mask indicates to the SCM which optional vector processor units are present.

/VERIFY

/NOVERIFY

Controls the /SCAN_VERIFY and /BI_VERIFY qualifiers as a pair.

SET SCOPE

Sets the default scope for labels.

Format

SET SCOPE */[NO]LOG label-spec*

Qualifier

/LOG

/NOLOG

Determines whether to display command results.

Parameters

label-spec

Specifies a label name in the current model (current scope) or an absolute label reference. A label specification can have one of the following forms:

Type	Format	Specifies a Label:
Absolute	%NET[.label[.label]]	At the top level
Absolute	%SCU[.label[.label]]	At the top level
Absolute	%CPU0[.label[.label]]	At the top level
Absolute	%CPU1[.label[.label]]	At the top level
Absolute	%CPU2[.label[.label]]	At the top level
Absolute	%CPU3[.label[.label]]	At the top level
Relative	label[.label[.label]]	In this model
Indirect	-.label[.label]]	In the model that contained this model's label

Relative and indirect references can appear anywhere in a label-spec (the label-spec after %nnn is a relative specification). Absolute specifications can appear only as the first label. Specifying CPU n rather than NET overrides the default set with SET CPU. Specifying SCU rather than NET is the only way to examine SCU signals as there is no SET DEFAULT command for the SCU.

SET SCREEN

Controls the state of screen mode. Use the `SHOW WINDOW` command to display the state of the screen.

Format

SET SCREEN {*ON*|*OFF*}

Parameters

ON (D)

If the screen is off, creates the command window and sets it to the entire screen. If the screen was suspended, it is refreshed.

When screen mode is on, the window manager controls the screen. Input is restricted to the command window and the output from most commands is displayed in the window. The command window is created automatically when the screen is first turned on.

The `CREATE/WINDOW` command and various `SET TRACE` commands turn screen mode on.

OFF

Suspends the screen and resets the scrolling region to the entire screen. The windows are not deleted. Selecting a window or creating a new window turns the screen on again.

When the screen is off, the screen scrolls normally, but window definitions remain. The windows are updated and redisplayed when the screen is again turned on.

SET SERIAL

Allows manual serial number entry for components that do not have SPU-readable serial numbers. The SPU operating system saves this information in data file SYS\$SYSTEM:MANUAL.DAT, from which it is retrieved by the SENSE command.

Format

SET SERIAL */qualifiers 10-digit-serial-number*

Qualifiers

/AIE

Specifies the serial number is for the SPU AIE (T1034) adapter.

/AIO

Specifies the serial number is for the SPU AIO (T1031) adapter.

/CPU=cpu-id

Valid only with */PLANAR*. Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DAC

Specifies the serial number is for one of the daughter array cards.

/IDENTIFIER=dac-number

Valid only with */DAC*, *dac-number* is a decimal number in the range 0 through 15.

/PLANAR

Specifies the serial number is for the specified CPU or SCU planar module.

2-174 Console Commands
SET SERIAL

/SCM

Specifies the serial number is for the SPU SCM (T2050) adapter.

/SCU

Valid only with /PLANAR. Specifies the SCU.

Parameters

10-digit-serial-number

Specifies the serial number according to DEC STD 012 and as seen on module serial number tags.

SET SNAPSHOT

Enables or disables the snapshot function when a hardware keep-alive is detected. The command can also be used to trigger a snapshot for testing purposes or to enable snapshots to be taken from command procedures.

Format

SET SNAPSHOT [*/qualifiers*] {ON|OFF|TRIGGER}

Qualifiers

/CPU=cpu-id

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is used. See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/EXCLUDE=section

One of the following:

CONTEXT	HISTORY	IO	LATCH
MCM	MICRO	PAMM	PCS
REVISION	SPU	STACK	

/FILENAME=file-spec

Specifies the name and type for the snapshot data file. The default *file-spec* is SNAPSHOT.DAT.

/SELECT=section

One of the following:

CONTEXT	HISTORY	IO	LATCH
MCM	MICRO	PAMM	PCS
REVISION	SPU	STACK	

/SCU

/NOSCU

Specifies the system control unit.

Parameters

ON

Enables the snapshot function.

OFF

Disables the snapshot function.

TRIGGER

Enables the snapshot function in command procedures and for testing.

SET SOURCE

Establishes a directory path for the CREATE/WINDOW/ECS command. The directory path specifies the location of the MCR file for the currently loaded microcode. It overrides the initial MCR file directory created at assembly and contained in the microcode LOD file.

Format

SET SOURCE *directory*

Parameters

directory

Specifies the location of the MCR file for the currently loaded microcode.

SET STEP

Sets the step characteristics that control the NEXT (macrostep) command.

Format

SET STEP *parameter*

Parameters

INSTRUCTION(D)

NOINSTRUCTION

Determines whether results are to be displayed in assembler notation.

SET TERMINAL

Specifies the characteristics of the specified terminal.

Format

SET TERMINAL *[/qualifiers] [terminal-name]*

Qualifiers

/BROADCAST (D)

/NOBROADCAST

Determines whether the terminal can receive broadcast messages such as SEND and exception messages.

/CPU=cpu-id

The CPU to which the terminal is logically connected in program I/O (PIO) mode, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the terminal is connected to the default CPU. This qualifier is incompatible with any terminal characteristic qualifier.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DEVICE={LA100|VT200}

Specifies the terminal device type to the console software.

/ECHO (D)

/NOECHO

Determines whether the terminal echoes (displays) keyboard input.

/EIGHTBIT
/NOEIGHTBIT

Determines whether the terminal uses 8-bit or 7-bit ASCII character codes. If ***NOEIGHTBIT*** (7-bit code) is specified, the most significant bit is forced to zero.

/ESCAPE
/NOESCAPE

Determines whether ANSI standard escape sequences transmitted from the terminal are handled as a single multiple-character terminator.

If ***/ESCAPE*** is specified, the terminal driver checks escape sequences for syntax before passing them to the program.

/KEYPAD={APPLICATION|NUMERIC}

Determines the terminal's keypad mode.

The ***/KEYPAD=APPLICATION*** qualifier sets the terminal keypad to application mode to enable the ***DEFINE/KEY*** facility.

/PAGE=lines

Specifies the number of lines in a terminal page.

/PIO_PORT={OPA0|OPA1}

Determines to which port the console terminal is connected. ***OPA0*** is the normal console port and ***OPA1*** is the remote port.

/PROGRAM
/NOPROGRAM

Causes the process to enter program I/O (PIO) mode. This qualifier is incompatible with any terminal characteristic qualifier.

/TALK_MODE
/NOTALK_MODE

Determines whether the terminal accepts TALK requests.

/WIDTH=characters

Specifies the number of characters per terminal line.

Parameters

terminal-name

Specifies the terminal affected by the command. If *terminal-name* is not specified, the controlling terminal is affected.

Example

```
>>> SET TERMINAL/DEVICE=LA100
```

Establishes the console terminal as an LA100 and sets the default characteristics for that terminal type.

SET TIME

Resets the system time.

Format

SET TIME *time-string*

Parameters

time-string

A string in the format: "DD-MMM-YYYY HH:MM:SS.DD"

The string must be quoted if a new date is entered.

Examples

1 >>> SET TIME 03:21

Sets system time to 03:21. The date is unchanged.

2 >>> SET TIME "18-Mar-1989 16:30"

Sets system date to 18-MAR-1989 and system time to 16:30.

SET TRACE

Establishes, enables, or disables tracepoints for the specified list of scan signals.

Use DELETE/TRACE to remove tracepoints and SHOW TRACE to display tracepoint status.

Format

```
SET TRACE [qualifiers] signal-list  
          @signal-file
```

Qualifiers

/ABSOLUTE

Specifies that times in comparison files (see */COMPARE*) are absolute. the clock cycle count (specified with */FROM*) when signal

The */ABSOLUTE* and */RELATIVE* qualifiers are mutually exclusive.

/ALL

With */ENABLE* or */DISABLE*, enables or disables all tracepoints. If */ALL* is specified, do not specify */NAME*.

/COMPARE

Compares the signals in a signal comparison file to the current machine state. The signal comparison file is specified with */FILE*.

/CPU=cpu-id

The CPU on which signals are to be traced, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If multiple CPUs are specified, a duplicate tracepoint is established on each processor. If a CPU is not specified, signals are traced on the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DISABLE

Temporarily inhibits trace reporting. The tracepoint definition is maintained, but trace activity is suspended until the tracepoint is enabled.

/NAME specifies the tracepoint to be disabled.

/ENABLE

Enables reporting of a tracepoint.

/NAME specifies the tracepoint to be enabled.

/FILE=file-spec

Specifies the file to which trace results are sent, or a signal comparison file (see */COMPARE*).

/FROM=cycle

Tracing begins when the */FROM* cycle count is reached, or immediately if */FROM* is not used. Tracing ends when the */TO* count is reached or when the trace is deleted with *DELETE/TRACE*.

/LOG

/NOLOG

Displays the result of the trace action.

/NAME=tracepoint-name

The name assigned to a tracepoint when it is established, or the name of a tracepoint to be enabled (*/ENABLE*) or disabled (*/DISABLE*).

If */NAME* is not specified when a tracepoint is established, the *SET TRACE* command automatically assigns a name. If the name is already in use, a warning message is issued.

Wildcard characters can be used when enabling or disabling similarly named tracepoints.

If */ALL* is specified, do not specify */NAME*.

/ODOMETER[=odometer-window-name]

Creates an odometer window to display trace activity, or assigns a previously created window (*CREATE/WINDOW/ODOMETER*) to the tracepoint.

If */ODOMETER* is specified without a window name, the *SET TRACE/ODOMETER* command automatically assigns a name.

/RELATIVE

Specifies that times in comparison files (see */COMPARE*) are relative (can be forced by preceding the time with a plus (+).)

/ABSOLUTE and */RELATIVE* are mutually exclusive.

/SCU

/NOSCU

Determines whether tracepoints apply to the system control unit.

/TO=cycle

Tracing begins when the */FROM* cycle count is reached, or immediately if */FROM* is not used. Tracing ends when the */TO* count is reached or when the trace is deleted with *DELETE/TRACE*.

/VECTOR

Creates a file and outputs the traced signal values and times to the file. The file can be used with *SET TRACE/COMPARE* and *SET PATTERN* commands.

Parameters

signal-file

The file that lists the signals to be traced. The at sign (@) is required to denote a signal-file. For example:

```
>>> SET TRACE [/qualifiers] @SIGNALS
```

signal-list

The signals to be traced. To specify more than 1 (up to 256) signal, separate the signal names with commas, as follows:

```
>>> SET TRACE [/qualifiers] signal-spec[,signal-spec...]
```

The asterisk (*) wildcard character can be used to specify several like named signals. For example:

```
>>> SET TRACE [/qualifiers] E3.*
```

Description

A single tracepoint can identify up to 256 signals, either directly on the command line or through a file.

When a tracepoint is established, traced signals are monitored after each step of the clock in the selected processor(s). Signals that change state between clock steps are displayed on the terminal.

Tracepoints are evaluated only when the clock is single-stepped as a result of a MICROSTEP command; MICROSTEP/BURST does not display signal transitions. The trace action is executed before the completion of the MICROSTEP command.

The /FROM and /TO qualifiers specify the starting and ending clock cycle counts between which tracing occurs. See also the SET CYCLE command description.

Signals can be simultaneously traced (and/or watched) by all users (tracepoint tables are shared resources).

SET VERIFY

Determines whether command and data lines from command procedures are displayed at the terminal.

Format

SET VERIFY {*ON/OFF*}

Parameters

ON(D)/OFF

Turns verification on and off.

SET VERIFY ON causes commands in command procedures to be displayed as they are executed. SET VERIFY OFF cancels this action.

SET WARM_START

Sets or clears the warm-start flag for the specified CPU. This command is used primarily for debug.

Format

SET WARM_START [/CPU=*cpu-id*] {ON|OFF}

Qualifier

/CPU=*cpu-id*

Specifies the CPU in which the warm-start flag is to be set or cleared, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the warm-start flag is set or cleared in the default CPU

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

Parameters

ON/OFF

Determines whether the warm-start flag is set.

SET WATCH

Establishes, enables, or disables watchpoints for the specified list of scan signals.

Use DELETE/WATCH to remove watchpoints and SHOW WATCH to display watchpoint status.

Format

```
SET WATCH [/qualifiers] signal-list DO command  
          @signal-file
```

Qualifiers

/ALL

With /ENABLE or /DISABLE, enables or disables all watchpoints. If /ALL is specified, do not specify /NAME.

/CPU=cpu-id

The CPU on which signals are to be watched, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If multiple CPUs are specified, a duplicate watchpoint is established on each processor. If a CPU is not specified, signals are watched on the default CPU.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/DISABLE

Temporarily inhibits watch reporting. The watchpoint definition is maintained, but watch activity is suspended until the watchpoint is enabled.

/NAME specifies the watchpoint to be disabled.

/ENABLE

Enables reporting of a watchpoint.

/NAME specifies the watchpoint to be enabled.

2-190 Console Commands
SET WATCH

/FROM=cycle

Watching begins when the /FROM cycle count is reached, or immediately if /FROM is not used. Watching ends when the /TO count is reached or when the watch is deleted with DELETE/WATCH.

/LOG

/NOLOG

Displays the result of the watch action.

/NAME=watchpoint-name

The name assigned to a watchpoint when it is established, or the name of a watchpoint to be enabled (/ENABLE) or disabled (/DISABLE).

If /NAME is not specified when a watchpoint is established, the SET WATCH command automatically assigns a name. If the name is already in use, a warning message is issued.

Wildcard characters can be used when enabling or disabling similarly named watchpoints.

If /ALL is specified, do not specify /NAME.

/SCU

/NOSCU

Determines whether watchpoints apply to the system control unit.

/TO=cycle

Watching begins when the /FROM cycle count is reached, or immediately if /FROM is not used. Watching ends when the /TO count is reached or when the watch is deleted with DELETE/WATCH.

Parameters

signal-file

The file that lists the signals to be watched. The at sign (@) is required to denote a signal-file. For example:

```
>>> SET WATCH [/qualifiers] @SIGNALS DO [command]
```

signal-list

The signals to be watched. To specify more than 1 (up to 256) signal, separate the signal names with commas, as follows:

```
>>> SET WATCH [/qualifiers] signal-spec[,signal-spec...] DO [command]
```

Several like-named signals can be specified with the asterisk (*) wildcard character. For example:

```
>>> SET WATCH [/qualifiers] E3.* DO [command]
```

command

A command to be executed following a state transition of any watched signal. The command can be any single line SPU command, including no command, or a command procedure.

Description

A single watchpoint can identify up to 256 signals, either directly on the command line or through a signal-file.

When a watchpoint is established, watched signals are monitored after each step of the clock in the selected processor(s). Signals that change state between clock steps cause the command entered on the SET WATCH command line to be executed.

Watchpoints are evaluated only when the clock is single-stepped as a result of a MICROSTEP command; MICROSTEP/BURST does not apply to watchpoints. The watch action is executed immediately after the MICROSTEP command completes.

/FROM and /TO specify the starting and ending clock cycle counts between which watch activity is to take place. See also the SET CYCLE command description.

Signals can be simultaneously watched (and/or traced) by all users (watchpoint tables are shared resources).

SET XMI_DEVICES

Loads the XMI device names and IDs from the specified file.

Format

SET XMI_DEVICES *[file-spec]*

Parameters

file-spec

Specifies a file that translates XMI node IDs into device names. If *file-spec* is not specified, the default file is [UCODE]XMI_DEVICES.DAT.

SHOW

The **SHOW** option command displays characteristics of the processors or the service processor (SPU) subsystem.⁶

Format

SHOW *options*

⁶ Command variants, of the form **SHOW OPTION**, are listed in the table of contents and described separately on the following pages.

SHOW ATTN_ACTION

Displays the command to be executed when the scan attention interrupt is delivered to the CLI.

The scan attention interrupt is delivered to the CLI only if error handling is disabled. For debug purposes, this function provides the means to execute a command procedure and save system failure data; this function is not used by the SPU error handling subsystem.

Format

SHOW ATTN_ACTION

Example

```
>>> show attn_action  
No ATTN_ACTION currently specified  
>>>
```

The SHOW ATTN_ACTION command display.

SHOW AUTOBOOT

Shows the state of the automatic bootstrap flags.

Format

SHOW AUTOBOOT

Example

```
>>> show autoboot
  CPU 0 Auto Boot: ENABLED
  CPU 1 Auto Boot: ENABLED
  CPU 2 Auto Boot: ENABLED
  CPU 3 Auto Boot: ENABLED
>>>
```

The SHOW AUTOBOOT command display.

SHOW AVAILABLE

Displays the set of processors passed to the operating system for inclusion in the symmetrical multiprocessor (SMP).

Format

SHOW AVAILABLE

Example

```
>>> show available  
CPU0  
>>>
```

The SHOW AVAILABLE command display.

SHOW BATCH

Displays the state of the batch streams.

Format

SHOW BATCH

Example

```
>>> SHOW BATCH  
  
      EWBA Batch Process Jobs  
      20-AUG-1989 13:50:12.43  
  
Username      Port ID      Status  
CONSOLE      001100CD    Executing  
/LOG=DUA50:[CONSOLE]TEST.LOG  
DUA50:[CONSOLE]TEST.CMD
```

The SHOW BATCH display format.

SHOW BBU

Shows the state of the battery backup unit.

Format

SHOW BBU

Example

```
>>> show bbu
  BBU 1: Available, Charged
  BBU 2: Available, Charged
>>>
```

The SHOW BBU command display.

SHOW BI_DEVICES

Displays the node names and device types for all devices loaded with the SET BI_DEVICES command.

Format

SHOW BI_DEVICES

SHOW BOOTFLAGS

Shows the setting of the permanent bootstrap flags.

Format

SHOW BOOTFLAGS

Example

```
>>> show bootflags  
  Boot flags: 00000000  
>>>
```

The SHOW BOOTFLAGS command display.

SHOW BOOTSET

Displays the current definition of the boot set.

The boot set is the set of processors that are available to execute the operating system. The boot set is displayed as a list of CPUs.

Format

SHOW BOOTSET

Example

```
>>> SHOW BOOTSET
      CPU0, CPU1, CPU3
>>>
```

The bootset includes CPU0, CPU1, and CPU3.

SHOW CLOCK

Displays the current state, frequency, and period of the clocks to each module.

Format

SHOW CLOCK *[/FULL]*

Qualifier

/FULL

Displays additional information about the internal state of the clock module, as follows:

- Frequency
- Period
- Interval
- Phase-locked loop (PLL) status
- Phase control and current phase register contents

Example

```
>>> show .clock
CPU0      CPU1      CPU2      CPU3      SCU
----      ----      ----      ----      ---
Stop      Stop      Stop      Stop      Run

Frequency = 500MHz
Cycle time = 16.000ns
>>>
```

The SHOW CLOCK command display.

SHOW CONFIGURATION

Displays the configuration of various system components. The command can display the version and type of all components as well as the serial number for most components.

The configuration of all system components is saved in a history file when system power is turned on. History file data can be displayed with the /DATE qualifier.

Format

SHOW CONFIGURATION */qualifiers*

Qualifiers

/CLOCK

Displays the configuration of the clock subsystem.

/CPU=cpu-id

Displays the configuration of the specified CPU, where cpu-id is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU's configuration is displayed.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

The configuration is displayed in the following format:

MCU	TYPE	NAME	REVISION	SERIAL
00+		ABC	0000	
01+		DEF	0000	
02+		XYZ	0000	

NOTE

A plus (+) following the MCU ID indicates self-test passed; a minus (-) indicates self-test failed.

2-204 Console Commands
SHOW CONFIGURATION

/DATE=date

Displays the configuration for the specified date, where date is a quoted string or an unquoted field.

/IO

Displays the I/O configuration of all four XMI buses and any BI buses connected to the XMI bus, in the following format:

XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
00	07+	0C04	XWATCH	FFFF	NOT AVAILABLE		
00	0D+	2001	DWMBA/A	0000	NOT AVAILABLE		
00	0E+	2001	DWMBA/A	0000	NOT AVAILABLE		
01	07+	0C04	XWATCH	FFFF	NOT AVAILABLE		
01	0D+	2001	DWMBA/A	0000	NOT AVAILABLE		
01	0E+	2001	DWMBA/A	0000	NOT AVAILABLE		
XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
00	01+	2107	DWMBA/B	0000	NOT AVAILABLE		
00	06+	410E	TBK50	0000	NOT AVAILABLE		
XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
01	01+	2107	DWMBA/B	0000	NOT AVAILABLE		
01	06+	410E	TBK50	0000	NOT AVAILABLE		
XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
02	01+	2107	DWMBA/B	0000	NOT AVAILABLE		
02	06+	410E	TBK50	0000	NOT AVAILABLE		
XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
03	01+	2107	DWMBA/B	0000	NOT AVAILABLE		
03	06+	410E	TBK50	0000	NOT AVAILABLE		

NOTE

A plus (+) following the node or XBI ID indicates self-test passed; a minus (-) indicates self-test failed.

/KERNEL

Displays kernel configuration.

/MCU=mcu-id

Displays information about the specified MCU.

/MEMORY

Displays memory configuration.

/OUTPUT=file-spec

Outputs the command results to the specified file. The default is the current terminal name.

/POWER

Displays power subsystem configuration in the following format:

NODE	TYPE	NAME	EEPROM	EPROM	SERIAL NUMBER
			VERSION	VERSION	
0+	PEM	31 (49)	31 (49)
11+	RIC	31 (49)	31 (49)
12+	RIC	31 (49)	31 (49)
13+	RIC	31 (49)	31 (49)
21+	RIC	31 (49)	31 (49)
22+	RIC	31 (49)	31 (49)
23+	RIC	31 (49)	31 (49)

NOTE

A plus (+) following the RIC ID indicates self-test passed; a minus (-) indicates self-test failed.

/RINGS

Displays testing and CDB ring lengths.

/SCU

Displays the configuration of the SCU in the following format:

MCU	TYPE	NAME	REVISION	SERIAL
00+		ABC	0000	
01+		DEF	0000	
02+		XYZ	0000	

NOTE

A plus (+) following the MCU ID indicates self-test passed; a minus (-) indicates self-test failed.

/SPU

Displays the configuration of the installed SPU modules, in the following format:

NODE	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
02+	0120	SPM	0000 (0)	T2051-00.A		20004000
06+	0121	SCM	0037 (55)	T2050-00.A		2000C000
07+	410F	DEBNA	0248 (584)	T1034-00.C		2000E000
0A+	410D	KFBTA	004B (75)	T1031-00.A		20014000

NOTE

A plus (+) following the node ID indicates self-test passed; a minus (-) indicates self-test failed.

2-206 Console Commands
SHOW CONFIGURATION

/SYSTEM

Displays the system cabinet configuration in the following format:

System Configuration Summary

System type: UNKNOWN

CPA Cabinet: Absent

CPB Cabinet: Absent

IOA Cabinet: Absent

IOB Cabinet: Absent

SCU Cabinet: Absent

CPU 0: Absent

CPU 1: Absent

CPU 2: Absent

CPU 3: Absent

SCU: Absent

XJA 0: Absent

XJA 1: Absent

XJA 2: Absent

XJA 3: Absent

MMU 0: Absent

MMU 1: Absent

/XMI=xmi-id

Displays the configuration of the installed XMI modules, in the following format:

XJA	XMI	TYPE	NAME	VERSION	IDENTIFIER	S/N	BASE
-----	-----	------	------	---------	------------	-----	------

NOTE

A plus (+) following the node ID indicates self-test passed; a minus (-) indicates self-test failed.

SHOW CPU

Displays the current default CPU or the state of the CPU(s).

Format

SHOW CPU *[/qualifiers] [cpu-id]*

Qualifiers

/ALL

Displays the state of all CPUs.

/FULL

Displays the state of the default CPU.

Parameters

cpu-id

Specifies the CPU. If not specified, the default CPU's state is displayed.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

2-208 Console Commands

SHOW CPU

Examples

1 >>> SHOW CPU
Default CPU is 0

Displays the default CPU.

2 >>> SHOW CPU/FULL
CPU 0 is present and has a CDB file loaded
Unit revision: A00
Unit CDB file: DUA50:[UCODE]CPUA00.CDB
VBOX: Available
Unit state flags:
Power On
Not Broken
Not Initialized
Clocks Not On
Not Running
Hard Core Not Executed
Pattern Test Not Executed

Displays the default CPU's status.

SHOW CYCLE

Displays the current cycle counter for the specified CPU. The interval is multiplied by the cycle count to determine the time. This is used with TRACE and PATTERN points.

Format

SHOW CYCLE *[/qualifiers]*

Qualifiers

/CPU=cpu-id

The CPU containing the cycle counter to be displayed, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU's cycle counter is displayed.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Determines whether to display the system control unit cycle counter.

Example

```
>>> SHOW CYCLE
Cycle = 5, Interval = 1000, Time = 5000 for CPU 0
```

Shows cycle for the default CPU. The cycle count is always a decimal value.

```
>>> SHOW CYCLE/SCU
Cycle = 5, Interval = 1000, Time = 5000 for CPU 0
Cycle = 0, Interval = 1, Time = 0 for SCU
```

Shows cycle for the default CPU and SCU.

SHOW DEFAULT

Displays the current default directory specification.

Format

SHOW DEFAULT

Example

```
>>> SHOW DEFAULT  
DISK$HARD:[SYS4.CONSOLE]
```

The default directory is the **CONSOLE** subdirectory of the **SYS4** directory on device **DISK\$HARD**.

SHOW DEVICE

Displays the status and interrupt level of devices known to the service processor operating system.

Format

SHOW DEVICE *[device-name]*

Description

Physical device information is not available to the command language interpreter (CLI). A device is considered off-line if the device driver is not present or if the device was not configured in the system. A device is on-line if the driver is present and the device can be accessed.

Network terminal devices (RTA) are considered on-line if the specified unit is in use. The base device (RTA0) status is the same as for other devices.

If the drive is present and mounted, disk volumes are shown as mounted with the volume name free block count. Tape volumes cannot be displayed and volume names are not available.

Parameters

device-name

Displays the status and interrupt level of the specified device. The device name is assumed to end with * (asterisk wildcard). Therefore:

```
>>> SHOW DEVICE D
```

displays information on all devices with names that start with d.

2-212 Console Commands

SHOW DEVICE

Display Format

>>> SHOW DEVICE

Device Name	Driver Status	Driver IPL	Volume Name	Free Blocks	Total Blocks
DUA50	Mounted	20	DISK\$HARD	105840	136408
DUA51	Offline	20			
DUA52	Offline	20			
DUA53	Offline	20			
VMA50	Online	20			

Device Name	Driver Status	Driver IPL
MUA7	Online	21

Device Name	Driver Status	Driver IPL
PRINTER	Online	20
CONSOLE	Online	21
REMOTE	Online	22
RTA0	Online	0
RTA1	Online	0
RTA2	Offline	0
RTA3	Offline	0
RTA4	Offline	0

Device Name	Driver Status	Driver IPL
XBA	Online	20
PCS	Online	21
SJA	Online	22
SCM	Online	20

The devices are displayed in groups, as follows:

1. Disk devices
2. Tape devices
3. Terminal devices
4. Generic devices

SHOW ENVIRONMENT

Displays the environmental conditions of the system.

Format

SHOW ENVIRONMENT *[/qualifiers]*

Qualifiers

/CONTINUOUS

Causes the display to be refreshed every 2 seconds or at intervals specified with the */INTERVAL* qualifier.

/INTERVAL=seconds

Specifies the continuous display refresh interval in seconds.

/OUTPUT=file-spec

Specifies the file to which the information is to be written.

Display Format

```
>>>SHOW ENVIRONMENT
          WCU0      WCU1      AIR          SCU      CPA      CPB
AIR
INLET TEMP      22.87C  00.00C  FAN 1 TEMP    00.00C  28.72C  00.00C
  STATUS        NOM    OPEN   STATUS        NOM    OPEN   OPEN
OUTLET TEMP     23.18C  00.00C  FAN 2 TEMP    28.10C  00.00C  00.00C
  STATUS        NOM    OPEN   STATUS        NOM    OPEN   OPEN
COOLANT
INLET TEMP      23.75C  00.00C  FAN 3 TEMP    00.00C
  STATUS        WARM   NOM    STATUS        NOM
OUTLET TEMP     22.74C  00.00C  AMBIENT       00.00C
  STATUS        WARM   NOM    STATUS        NOM
STATUS
WCU STATE      FLT    OK    AIR FLOW 1    OK    OK    OK
PUMP A         RUN    OK    AIR FLOW 2    OK    OK    OK
PUMP B         OK    RUN   AIR FLOW 3    OK    OK    OK
BLOWER A       OK    OK    AIR FLOW 4    OK    OK    OK
BLOWER B       OK    OK    AIR FLOW 5    OK
PRESSURE       OK    OK    AIR FLOW 6    OK
  LEVEL        OK    OK
  FLOW         OK    OK
PAN DETECT     OK    OK
```

Temperature sensors and status information for each cabinet are displayed.

SHOW ERROR_HANDLING

Shows the current error handling settings.

Format

SHOW ERROR_HANDLING

Examples

```
❶ >>> show error_handling

System Error Handling Statistics

Options -
  State:           DISABLED
  Recovery:        ENABLED
  Reporting:       ENABLED
  Match Detect:    ENABLED
  Error interval:  0
  Cache Threshold: 0
  VBOX Threshold:  0
  CPU Threshold:   0

>>>
```

The SHOW ERROR_HANDLING command display.

SHOW FAULT_ACTION

Displays the command executed when the power fault interrupt is delivered to the CLI. This function is for debug purposes and is not used by the SPU error handling subsystem.

Format

SHOW FAULT_ACTION

Example

```
>>> show fault_action  
No FAULT_ACTION currently specified  
>>>
```

The SHOW FAULT_ACTION command display.

SHOW FLAGS

Displays the state of the cold-start and warm-start flags for each CPU. This command is used for debug.

Format

SHOW FLAGS *[/CPU=cpu-id]*

Qualifier

/CPU=cpu-id

Displays flags of the specified CPU, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU's flags are displayed.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

Example

```
>>> SHOW FLAGS
CPU 0 Boot Flags
  WARM_START: OFF
  COLD_START: OFF
```

Both flags are off.

SHOW HISTORY

Displays the contents of the PC history file in the specified EBox.

Format

SHOW HISTORY *[/qualifiers]*

Qualifiers

/BINARY

/NOBINARY

Determines whether the history buffer is output to a binary file. The default file is SYS\$LOGIN:HISTORY.DAT.

/CPU=cpu-id

Displays the specified CPU's history buffer, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, displays the default CPU's history buffer.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/INSTRUCTION

/NOINSTRUCTION

Specifies whether the contents of memory at the saved address is to be decoded and displayed in assembler mnemonics.

/MAXIMUM=locations

Specifies the maximum number of locations to display. The default is the last 16 locations.

/OUTPUT=file-spec

Saves the history buffer in the specified file rather than displaying it.

/PHYSICAL

Specifies that history buffer addresses are physical addresses.

2-218 Console Commands
SHOW HISTORY

/SCU
/NOSCU

Determines whether to display the system control unit history buffer.

/VIRTUAL

Specifies that history buffer addresses are virtual addresses. The SPU assumes that the current process is the process recorded in the history file.

/WINDOW[=*window-name*]

Specifies the window in which the history buffer is to be displayed.

Example

```
>>> show history
PC history for CPU 0 (starting with oldest PC)
  8022DF9B
  8022DF9D
  8022DFAD
  8022E17E
  8022E182
  8022E188
  8022E18F
  8022E1B3
  8022E1B8
  8022E1BA
  8022E1BF
  8022E1C5
  8022E1CA
  8022E1CC
  8022E1CE
  8022E1D3
>>>
```

The SHOW HISTORY command display.

SHOW ISOLATION

This command is for use with the two-hole tester and the SYBIL process. It displays the isolation data that has been loaded for each MCU and CPU. It can also display the isolation data for a specific bit in the broadcast ring.

Format

SHOW ISOLATION */qualifiers [broadcast-ring-bit-number]*

Qualifiers

/CPU=cpu-id

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is used.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/MCU=mcu-id

Displays the isolation data that has been loaded for the specified MCU.

/SYBIL

Displays the SYBIL isolation data that has been loaded.

Parameters

broadcast-ring-bit-number

Displays the isolation data for the specified broadcast ring bit. For example:

```
>>> SHO ISOLATION 5
Flex: FP4.67, Bit: 5, MCU: 0, Unit: 0 (SYB4_IO_PIN_H<92>)
Flex: FP4.203, Bit: 710, MCU: 1, Unit: 0 (SYB1_IO_PIN_H<80>)
Flex: FP4.67, Bit: 5, MCU: 1, Unit: 0 (SYB4_IO_PIN_H<92>)
Flex: FP4.203, Bit: 710, MCU: 0, Unit: 0 (SYB1_IO_PIN_H<80>)
>>>
```

SHOW KEEP_ALIVE

Displays the state of the keep-alive monitor.

Format

SHOW KEEP_ALIVE

Example

```
>>> show keep_alive
```

```
Keep Alive Monitor State
```

```
Options -
```

```
  CPU 0: MANUAL
```

```
  CPU 1: MANUAL
```

```
  CPU 2: MANUAL
```

```
  CPU 3: MANUAL
```

```
>>>
```

Displays state of keep-alive monitor.

SHOW KEY

Displays the DEFINE/KEY definition of the specified key.

Format

SHOW KEY *[/ALL] [key-name]*

Qualifiers

/ALL

Displays the definition of all keys defined with the DEFINE/KEY command. If */ALL* is specified, do not specify the key-name parameter.

/FULL

Displays the flags associated with the key definition. For example:

```
>>> SHOW KEY MINUS
    MINUS = SC/TO:SIGNAL("CS[" + 'UPC()' + "]<11:0>")
>>> SHOW KEY/FULL MINUS
    MINUS = SC/TO:SIGNAL("CS[" + 'UPC()' + "]<11:0>") (NOECHO,TERMINAL)
>>>
```

Parameters

key-name

Displays the DEFINE/KEY definition of the specified key. If */ALL* is specified, do not specify the key-name parameter.

2-222 Console Commands
SHOW KEY

Valid key names are:

Key Name	Key
Editing Keypad	
E1	Find
E2	Insert Here
E3	Remove
E4	Select
E5	Prev Screen
E6	Next Screen
Function Keys	
F6-F14	F6-F14
Do	F15
Help	F16
F17-F20	F17-F20
Numeric Keypad	
PF1-PF4	PF1-PF4
KP0-KP9	0-9
PERIOD	.
COMMA	,
MINUS	-
ENTER	Enter

SHOW LOGGING

Displays the name and status of the current terminal log files if any exist.

Format

SHOW LOGGING

Example

```
>>> show logging
  Current log file(s) open:
    DUA50:[CROWLEY]ESDP_1.LOG
>>>
```

The SHOW LOGGING command display.

SHOW LOGICAL

Displays the equivalence string assigned to a logical name with the DEFINE command.

Format

SHOW LOGICAL *[qualifiers] [logical-name]*

Qualifiers

/ALL

Displays definitions for all logical names. If omitted, the command prompts for the logical-name parameter. If /ALL is specified, do not specify the logical-name parameter.

/PROCESS

Searches only the process logical name table for the specified logical-name. If logical-name is not specified, displays all the entries in the process logical name table.

/SYSTEM

Searches only the system logical name table for the specified logical-name. If logical-name is not specified, displays all the entries in the system logical name table.

Parameters

logical-name

The logical name for which the definition is to be displayed. If omitted, the command prompts for the logical-name parameter. If /ALL is specified, do not specify the logical-name parameter.

SHOW MEMORY

Displays the status of main memory.

Format

SHOW MEMORY

Example

```
>>> show memory
      System Memory Resources at 7-MAR-1990 09:32:13.90
Physical Memory Usage (pages): Total   In Use   Free     Largest
Main Memory (16.00Mb)         32768   19930   12838   12784
System Memory Usage (pages):  Total   In Use   Free     Largest
System Memory                 18632   13830   4802    0
Dynamic Memory Usage (blocks): Total   In Use   Free
Kernel Pool (128 bytes per)  3000    902     2098
Page Table Usage (Slots):    Total   In Use   Free
Page Table Slots              200     94     106
Port Object Usage (ports):   Total   In Use   Free
Port statistics                256     57     199
>>>
```

Displays memory resources.

SHOW MESSAGE

Displays information about the current message file and message format settings.

Format

SHOW MESSAGE [*message-id*]

Parameters

message-id

Displays the specified message, where *message-id* is a hexadecimal number.

Examples

```
❶ >>> show message
Message file: DUA50:[SYSEXEC]SYSMSG.EXE (0)
Message flags:
                /NOFACILITY
                /NOSEVERITY
                /NOIDENT
                /NOTEXT
>>>
```

The SHOW MESSAGE default command display.

```
❷ >>> show message 1
%KERNEL-S-SUCCESS, normal successful completion
>>>
```

The SHOW MESSAGE display of a specified message.

SHOW MODE

Displays the current mode and defaults for the EXAMINE and DEPOSIT commands.

Format

SHOW MODE

Example

```
>>> show mode
  Address Space: MEMORY
  Data Context: LONGWORD
  Default Address: 00000000
  Default Radix: HEXADECIMAL
>>>
```

The SHOW MODE command display.

SHOW NODE

Displays the current network node and address.

Format

SHOW NODE

Example

```
>>> SHOW NODE
  Initial node name: MRBONZ
  Initial node address: AA-00-04-00-19-1D (7.281)

  Current node name: MRBONZ
  Current node address: AA-00-04-00-19-1D (7.281)
>>>
```

The SHOW NODE command display.

SHOW PATTERN

Displays information about the specified pattern file. The name of the file and affected signals are displayed with the cycle count of the next event in the file.

Format

SHOW PATTERN *[/qualifiers] [pattern-name]*

Qualifiers

/CPU=cpu-id

The CPU to which the pattern applies, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU is selected.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/NODE=node-id

Valid only with /SYBIL. Specifies the remote node name or node number (in the format *z.nnn*) where the pattern file resides.

/REMOTE

Displays a list of available remote (that is, SYBIL) patterns rather than listing pattern points.

/SCU

/NOSCU

Determines whether to display system control unit pattern points.

/SYBIL

Specifies SYBIL remote patterns.

2-230 Console Commands
SHOW PATTERN

Parameters

pattern-name

The name of the pattern point; if not specified, all patterns are displayed.

SHOW PERSONAL

Shows the user's personal name.

Format

SHOW PERSONAL

Examples

```
❶ >>> show personal
    No personal name set
>>>
```

The SHOW PERSONAL command display.

SHOW POWER

Displays various regulator (group) voltage measurements.

Format

SHOW POWER *[/qualifier]*

Qualifiers

/BUS=bus-name

Specifies the buses affected by the command, as follows:

Bus Name	Volts	Model 200 Cabinet	Model 400 Cabinet	Tester
A	+5.0	SCU	SCU	MCM
B	+5.0	BBU	BBU	None
C	-3.4	SCU/CPA	SCU	None
D	-5.2	SCU/CPA	SCU	None
E	-3.4	CPB	None	None
F	-5.2	CPB	None	None
J	-3.4	None	CPA	UNIT 0
K	-5.2	None	CPA	UNIT 0
M	-3.4	None	CPB	UNIT 1
N	-5.2	None	CPB	UNIT 1

/CABINET=cabinet-id

The cabinets to which the command applies. The cabinet-ids represent the buses that supply the cabinet (see /BUS).

Cabinet ID	Model 200 Buses	Model 400 Buses	Tester Buses
SCU	A, B, C, D	A, B, C, D	None
CPA	C, D	J, K	None
CPB	E, F	M, N	None
UNIT_0	None	None	J, K
UNIT_1	None	None	M, N
5VOLT	None	None	A, B

If a cabinet is not specified, the command applies to all cabinets.

/CONTINUOUS

Causes the display to be refreshed every 2 seconds or at intervals specified with the /INTERVAL qualifier.

/COUNTERS

Displays the event counters in the power subsystem driver.

/INTERVAL=seconds

Specifies the continuous display refresh interval in seconds.

/IO

Displays the state of the I/O power supplies.

/OUTPUT=file-spec

Logs command output in the specified file.

2-234 Console Commands
SHOW POWER

Example

>>> SHOW POWER /IO

	XMI0				XMI1			
	7214 A	7215 A	7214 B	7215 B	7214 A	7215 A	7214 B	7215 B
STATUS								
REG OK	OK	OK	OK	OK	OK	OK	OK	OK
BIAS 0	OK	OK	OK	OK	OK	OK	OK	OK
BIAS 1	OK	OK	OK	OK	OK	OK	OK	OK
BIAS 2	OK	OK	OK	OK	OK	OK	OK	OK
BIAS 3	OK	OK	OK	OK	OK	OK	OK	OK
	BI0				BI1			
	REG A	REG B	REG C	REG D	REG A	REG B	REG C	REG D
STATUS								
REG OK	OK	OK	OK	OK	OK	OK	OK	OK
	BI2				BI3			
	REG A	REG B	REG C	REG D	REG A	REG B	REG C	REG D
STATUS								
REG OK	OK	OK	OK	OK	OK	OK	OK	OK

SHOW POWER /IO display format.

SHOW PROCESS

Displays information about the current CLI process, including allocated devices and process statistics.

Format

SHOW PROCESS

Display Format

>>> SHOW PROCESS

20-AUG-1989 13:54:05.57 RTA1: User: CONSOLE
Job ID: 0016 Port ID: 000400B3 Image: CLI
Priority: 17 Default directory: DUA50:[CONSOLE]
CPU time: 0 00:00:02.25 Pages: 273 P0 pages: 1073 P1 pages: 25

Devices allocated: RTA1:

Process	Priority	State	Run time	Memory Pages
1	8	Run	0 00:00:02.16	13
2	1	Wait	0 00:00:00.06	4
3	1	Wait	0 00:00:00.00	2
4	1	Wait	0 00:00:00.00	2
5	1	Wait	0 00:00:00.03	2
6	8	Wait	0 00:00:00.00	2

SHOW RADIX

Displays the current radix setting.

Format

SHOW RADIX

Example

```
>>> SHOW RADIX
    Default radix is hex
>>>
```

The current radix is hexadecimal.

SHOW REMOTE

Displays the state of the remote port.

Format

SHOW REMOTE

Example

```
>>> SHOW REMOTE
  Front panel access: DISABLED
  Software controlled access: ENABLED
  Remote status: 1 ACTIVE LINK
>>>
```

The SHOW REMOTE command display.

SHOW SCI

Displays the current state of the SCI. The value of the SCI DATA IN line is displayed with the saved state of the SCI output lines (the output lines cannot be directly read).

NOTE

If the SET SCI/SHOW SCI commands are in use, no other scan activity should occur. Any intervening scan commands, including SHOW CLOCK, could change the SCI state.

Format

SHOW SCI *[/qualifiers]*

Qualifiers

/DATA_ONLY

/NODATA_ONLY

Determines whether only the data is displayed.

/PORT=port-id

Displays SCI state of the specified port, where port-id is the physical port number₁₆ or port name, as follows:

CPU0 CPU1 CPU2 CPU3 MCM SCU

Example

```
>>> show sci
Current SCI Saved State Information for port 0 (CPU0)
  SELECT<3:0>           0
  FUNCTION<1:0>         0      (NOP)
  CDS                   0
  BYPASS                0
  BROADCAST             0
  DATA OUT (from SCM)  0
  CLOCK A               0      (NOP)
  CLOCK B               0      (NOP)
  DATA IN (to SCM)     0      (Data from latch in SCD logic)
>>>
```

The SHOW SCI command display.

SHOW SCM

Displays information about the scan control module.

Format

SHOW SCM *[/qualifiers]*

Qualifiers

/BAD_PAGE_MAP

Displays bad page block, as follows:

```
>>> SHOW SCM /BAD_PAGE_MAP
```

```
Scan Control Module BAD_PAGE_BLOCK
```

```
00000000 00000000 00000000 00000000 : Page 000:07F
00000000 00000000 00000000 00000000 : Page 080:0FF
00000000 00000000 00000000 00000000 : Page 100:17F
00000000 00000000 00000000 00000000 : Page 180:1FF
00000000 00000000 00000000 00000000 : Page 200:27F
00000000 00000000 00000000 00000000 : Page 280:2FF
00000000 00000000 00000000 00000000 : Page 300:37F
00000000 00000000 00000000 00000000 : Page 380:3FF
```

```
>>>
```

2-240 Console Commands
SHOW SCM

/ENTRY_BLOCK

Displays last entry block, as follows:

>>> SHOW SCM /ENTRY_BLOCK

Scan Control Module LAST_ENTRY_BLOCK

```
Valid flag 00000001
R0 - R3    00000000 1F830A10 1F817600 00000005
R4 - R7    1F8004E8 00000000 1F80103C 00017600
R8 - R11   00A3C600 00000001 00000000 0001C602
AP FP SP PC 1F8033F8 1F8033E4 1F802BF0 1F8009AE
PSL        00000200 bi_stop
00(SP)     55000004
04(SP)     0C16EF9E
08(SP)     D0540000
0C(SP)     ACD05204
10(SP)     53D45108
14(SP)     FD8F5191
18(SP)     53D60B13
1C(SP)     5062417E
SCM CSR    FFFFFFFBF
Reserved   115055C0 F88F780C 407E5051 54C05062
Checksum   905E2A3B valid
>>>
```


/ERROR_BLOCK

Displays error block, as follows:

```
>>> SHOW SCM /ERROR_BLOCK
Scan Control Module LAST_ERROR_BLOCK

Error Code  00000008 FATAL_MACHINE_CHECK
SCB Vector  00000004
DType       04442121
BI CSR      05010806
BER         00000000 no_errors
Port PC     00000006
Port PS     B8040740
Port PE     00000001
Port PD     1F8072D4
00 (SP)     0000000C
04 (SP)     00000080
08 (SP)     9D86B914
0C (SP)     0A00000B
10 (SP)     1F8072D4
14 (SP)     00010000
18 (SP)     00000000
1C (SP)     00000000
SCC CSR     000040C2 PCE
SCC CLK CTL 40000001
SCC DMA CSR 00000000
SCC DMA OUT 1F821B10
SCC DMA IN  1F822380
SCC DMA MSK 1F800000
SCC DMA EXP 1F811810
DYRC CSR    00003000
DYRC ADDR   00005CC0
BCI3 CSR    FC7F0000
BCI3 EV Sts 00000006 no_errors
BCI3 DM Cnf 1FDFC000
BCI3 BI Adr C096DA20
BCI3 II Adr DF8223D0
Reserved    04040118 00000004
Checksum    7CE8A9C2 valid

>>>
```

2-242 Console Commands

SHOW SCM

/MEMORY

Displays SCM memory statistics, as follows:

```
>>> SHOW SCM /MEMORY
```

Scan Control Module Memory Resources

ROM Space (pages)	Total	In Use	Free	
Firmware	256	158	98	

RAM Space (pages)	Total	In Use	Free	Largest
Static Storage	181	181	0	
CDB Storage	795	774	21	21
SCC Buffer Area	35	25	10	10
Watchpoint Area	12	0	12	12

```
>>>
```

Example

```
>>> SHOW SCM
```

Scan Control Module Statistics

```
Revision levels -  
  SCM Firmware: V68      SCM RBD: V61
```

```
Options -  
  Scan Verification: DISABLED  
  BI Verification:   DISABLED  
  Signal Caching:   ENABLED  
  Attention Polling: DISABLED  
  Scan Mode:        NORMAL  
  Scan Clock Rate:  100NS  
  Vector proc mask: 01  
  Attention mask:   00
```

```
>>>
```

Displays scan control module statistics.

SHOW SCOPE

Displays the current default scope for use with the EXAMINE, DEPOSIT, SET TRACE, SET WATCH, and SET PATTERN commands. The scope display format (same as EXAMINE/LABEL) is as follows:

```
Label: Model=Model-name, Revision=Model-revision
```

Format

```
SHOW SCOPE
```

Example

```
>>> show scope
      Scope = %CPU0, Model = CPU, Revision = A
>>>
```

The SHOW SCOPE command display.

SHOW SCU

Displays the current state of the SCU.

Format

SHOW SCU

Display Format

```
>>> SHOW SCU
  SCU is present and has a CDB file loaded
  Unit revision: A00
  Unit CDB file: DUA50:[UCODE]SCUA00.CDB
  DA1: Broken
  Unit state flags:
    Power On
    Not Broken
    Not Initialized
    Clocks Not On
    Not Running
    Hard Core Not Executed
    Pattern Test Not Executed
```

SHOW SJA

Displays the current state of the SJA.

Format

SHOW SJA

Example

```
>>> SHOW SJA

SPU/Jbox Adapter Statistics

Revision levels -
  SJA Array: A          SPM Module: A

Options -
Memory type:          EMULATED      Register type:      EMULATED
Looback:              DISABLED      DMA mode:           DISABLED
Primary CPU ID:      00              Simulated O/S:     DISABLED, INACTIVE
Memory mode:         STEP            Cur Mem state:     STBY
Debug mode:          DISABLED        CPUCNF access:     DISABLED
MMU present mask:    00              XJA present mask:  00
Check MMU hndshk:    ENABLED          Check XJA hndshk:  ENABLED
Trace mode:          DISABLED        PF Int enable:     DISABLED

Attentions -
Received ATTNs:      0
                    CPU0 CPU1 CPU2 CPU3 SCU
Attention mask:      0000 0000 0000 0000 0000
Hot check mask:     0000 0000 0000 0000 0000
Clock check mask:   0000 0000 0000 0000 0000
>>>
```

The SHOW SJA command display.

SHOW SOURCE

Displays the source directory for microcode trace windows.

Format

SHOW SOURCE

Example

```
>>> SHOW SOURCE  
DISK$HARD: [SYS4.CONSOLE]
```

The source directory for the microcode trace windows is the CONSOLE subdirectory of the SYS4 directory on device DISK\$HARD.

SHOW STEP

Displays the current STEP attributes. The INSTRUCTION mode state and the BOOTSET mode state are displayed.

Format

SHOW STEP

Example

```
>>> SHOW STEP
      Instruction
>>>
```

The SHOW STEP command display.

SHOW STRUCTURE

Displays the structure current state, version, symbol table state, and file-spec from which the structure was loaded.

Format

SHOW STRUCTURE [*structure-name*] [*/qualifiers*]

Parameters

structure-name

Specifies one structure. If not specified, all structures are displayed. Valid structure names are as follows:

Table 2-16 Structure Names

BP	CACHE0	CACHE1	ECC0	ECC1
ECS	EREG	FRAM	IPAMM	JCS
MPAMM	NPAMM	PCHB	TAG0	TAG1
TAGRM	TBRAMS	VALIDRAMS	VIC	VICA
VICB	VREG			

Qualifiers

/CPU=cpu-id

Specifies the CPU that contains the structure(s), where *cpu-id* is one of the following:

0 1 2 3
ALL AVAILABLE BOOTPRIMARY BOOTSET PRIMARY

If a CPU is not specified, the default CPU structure information is displayed.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU
/NOSCU

Determines whether to display system control unit structure information.

Display Format

>>> show structure

```
Status of structures on CPU 0
```

Name	State	Signal name	Version	Symbols
----	-----	-----	-----	-----
VREG	Initialized	%CPU.VREG[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]VREG.LOD			
BP	Initialized	%CPU.BP[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]BP.LOD			
EREG	Initialized	%CPU.EREG[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]CONSTANT0.LOD			
CACHE0	Initialized	%CPU.CACHE0[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]CACHE0.LOD			
CACHE1	Initialized	%CPU.CACHE1[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]CACHE1.LOD			
FRAM	Initialized	%CPU.FRAM[]	A48e*DE	NOT Loaded
Loaded from	file DUA50:[UCODE]FRAM.LOD			
VALIDRAMS	Initialized	%CPU.VALIDRAMS[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]VALIDRAMS.LOD			
PCHB	Undefined	%CPU.PCHB[]	NONE	NOT Loaded
ECC0	Initialized	%CPU.ECC0[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]ECC0.LOD			
ECC1	Initialized	%CPU.ECC1[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]ECC1.LOD			
VICA	Initialized	%CPU.VICA[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]VICA.LOD			
VICB	Initialized	%CPU.VICB[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]VICB.LOD			
ECS	Initialized	%CPU.ECS[]	E272	Loaded
Loaded from	file DUA50:[UCODE]AQUARIUS.LOD			
VIC	Initialized	%CPU.VIC[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]VIC.LOD			
TAG0	Initialized	%CPU.TAG0[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]TAG0.LOD			
TAG1	Initialized	%CPU.TAG1[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]TAG1.LOD			
TBRAMS	Initialized	%CPU.TBRAMS[]	NONE	NOT Loaded
Loaded from	file DUA50:[UCODE]TBRAMS.LOD			

>>>

Displays information about all the structures for the default CPU (CPU0).

SHOW SWITCHES

Displays the current position of the Startup and Service Processor Access switches on the operator control panel. The possible values are:

Table 2-17 Operator Control Panel Switches

Startup Switch Positions	Service Processor Access Switch Positions
BOOT	REMOTE
RESTART_BOOT	REMOTE_DISABLE
RESTART_HALT	LOCAL
HALT	LOCAL_DISABLE

Format

SHOW SWITCHES

Example

```
>>> SHOW SWITCHES
  Boot Switch: HALT
  Access Switch: REMOTE
```

The Boot switch is set to Halt and the Access switch is set to Remote.

SHOW SYMBOL

Displays symbol values defined by the symbol assignment command. If /ALL or symbol-name are not specified, the command issues the input prompt:

_SYMBOL:

Format

SHOW SYMBOL *[/qualifiers] [symbol-name]*

Qualifiers

/ALL (D)

Displays all symbols in the specified table. If /ALL is specified, do not specify the symbol-name parameter.

/GLOBAL

Displays only global symbols.

/LOCAL

Displays only local symbols.

/structure_qualifier

Displays the symbols loaded for the specified structure. Valid structure qualifiers are:

/BP	/CACHE0	/CACHE1	/ECC0	/ECC1
/ECS	/EREG	/FRAM	/IPAMM	/JCS
/MPAMM	/NPAMM	/PCHB	/TAG0	/TAG1
/TAGRM	/TBRAMS	/VALIDRAMS	/VIC	/VICA
/VICB	/VREG			

Parameters

symbol-name

The symbol to be displayed. If omitted, all symbol definitions are displayed. If symbol-name contains a wildcard character, all matching symbol definitions are listed. If /ALL is specified, do not specify the symbol-name parameter.

SHOW SYSTEM

Displays the processes currently defined in the SPU software system. The memory usage of each process is displayed with the CPU usage. The display format is similar to the VMS SHOW SYSTEM command.

Format

SHOW SYSTEM *[/[NO]PROCESS]*

Qualifiers

/PROCESS

/NOPROCESS

Specifies whether to display subprocess information for each job in the system.

Example

```
>>> show system
EWBAA V10.8(332) on node SPUS18 13-MAR-1990 19:08:50.90
  Pid  Process name  State  Pri  CPU  RW Mem PO Mem P1 Mem
00020000 XBDRIVER    Wait   1  0 00:00:00.17  45  65  16
00030000 CONSOLE_31    Wait   2  0 00:00:09.15  19  43  50
00040000 EDEBUGREM     Wait   3  0 00:00:00.01   2  23   4
00050000 BUDRIVER_31    Wait   4  0 00:00:45.64 159 143  58
00060000 MBDRIVER_31    Wait   4  0 00:00:00.04  52  36  24
00070000 VMDRIVER      Wait  16  0 00:00:00.15   9 107  42
00080000 FALSERVER     Wait  16  0 00:00:00.01   1  12   2
000B0000 SCMDRIVER     Wait   5  0 00:00:08.48 566 127  54
000C0000 SJADRIVER     Wait  16  0 00:00:00.81 846 246  74
000D0000 PCSDRIVER     Wait   5  0 00:00:00.81 408  98  48
000E0000 RTDRIVER      Wait   4  0 00:00:00.00   4  43   2
000F0000 REMOTE      Wait   6  0 00:00:00.02   9  43  13
00100000 PRINTER      Wait   6  0 00:00:00.01   9  42  13
00110000 CONTROL      Wait  12  0 00:00:00.03  42 283  21
00120000 CLI          Run   17  0 00:01:06.67 703 1035 97
00130000 EFM          Wait  12  0 00:00:00.41 432  79  10
>>>
```

The SHOW SYSTEM command display.

SHOW TERMINAL

Displays the current terminal characteristics.

Format

SHOW TERMINAL [*terminal-name:*]

Parameters

terminal-name:

Displays the specified terminal characteristics, where *terminal-name* must be terminated with a colon (:). For example:

```
>>> SHOW TERMINAL CONSOLE
%CMD-W-SYNTAXERR, illegal command - check command description
\CONSOLE\
>>> SHOW TERMINAL CONSOLE:
Terminal: CONSOLE      Device_type: VT100      Connected_to: OPAl
      Input:   9600      LFfill: 0      Width:   80      Parity: None
      Output:  9600      CRfill: 0      Length:  24
Terminal Characteristics
Echo
Escape
No Eightbit
No Passall
Broadcast
Talk_Mode
>>>
```

If not specified, the current terminal characteristics are displayed.

2-254 Console Commands

SHOW TERMINAL

Example

```
>>> SHOW TERMINAL
Terminal: CONSOLE:   Device_type: VT100   Connected_to: OPA0
  Input: 9600   LFFill: 0   Width: 80   Parity: None
  Output: 9600   CRFill: 0   Length: 24
```

Terminal Characteristics

```
Echo
Escape
No Eightbit
No Passall
Broadcast
Talk_Mode
```

The current terminal characteristics.

SHOW THRESHOLD

Displays the current error handling system thresholds.

Format

SHOW THRESHOLD

Example

```
>>> show threshold
Error thresholds and current counts

Cache Cache
Set 0 Set 1 Vbox CPU
Threshold: 0 0 0 0
CPU 0: 16384 0 784 0
CPU 1: 0 0 0 0
CPU 2: 0 0 0 0
CPU 3: 0 3584 0 0
>>>
```

The **SHOW THRESHOLD** command display.

SHOW TIME

Displays the current date and time in the following format:

```
DD-MMM-YYYY HH:MM:SS
```

Format

```
SHOW TIME
```

Example

```
>>> SHOW TIME  
11-JAN-1990 10:17:49
```

The current date and time display.

SHOW TRACE

Displays information about the specified tracepoint.

Format

SHOW TRACE *[/qualifiers] [tracepoint-name]*

Qualifiers

/CPU=cpu-id

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is used.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Determines whether to display system control unit tracepoints.

Parameters

tracepoint-name

The tracepoint to be displayed.

Example

```
>>> SHOW TRACE TRACE1
TRACE1 /FROM=0 /TO=0 /CPU=0 (enabled) USER=0
      %CPU0.DECODE_PC<31:0>
      %CPU0OX_UPC<11:0>
      %CPU0.LOAD_UPC<11:0>
```

Tracepoint TRACE1 information display.

SHOW USERS

Displays the user name, port ID, and terminal name of processes currently logged in to the SPU.

Format

SHOW USERS

Example

```
>>> SHOW USERS
      SPU/ELN Interactive Users
      25-AUG-1988 00:00:21.01

Username      Port ID      Terminal
CONSOLE       000200EA    CONSOLE:
MYNAME        000400D7    RTA1:
```

User information display.

SHOW VERSION

Displays the version of the specified object. Hardware object displays include the hardware part number and serial number, if available. Software object displays include the facility or module name.

Format

SHOW VERSION */object*

Objects

/ALL

Valid only with */CLI* and */RTL*. Displays all data for the subsystem.

/CLI (D)

Displays command language interpreter version.

/CLOCK

Displays the top level clock subsystem revision.

/FACILITY=name

Valid only with */CLI* and */RTL*. Displays more information about the module specified by facility-name.

/RTL

Displays the SPU runtime (RTL) library version.

/SCM

Displays the scan control module version.

2-260 Console Commands
SHOW VERSION

Examples

```
❶ >>> SHOW VERSION

      EWBA X10.7(324)
DEB;47.0 FILE;22.0 MAC;69.0 MAIN;75.0 SET;62.0 TEST;27.0 WIN;22.0
Built by EVANS on node SALTON at 17-FEB-1990 13:00:17.51
>>>
```

Displays CLI version information. Note that CLI is the default if no qualifiers are specified.

```
❷ >>> SHOW VERSION /RTL

      SPURTL X10.5(322)
CMD;30.0 CSA;7.0 ERH;27.0 KNL;47.0 LIB;92.0 PCS;26.0 RMS;8.0
SCM;35.0 SJA;37.0 TRM;13.0
Built by EVANS on node SALTON at 17-FEB-1990 11:52:30.90
>>>
```

Displays runtime library version information.

SHOW WATCH

Displays information about the specified watchpoint.

Format

SHOW WATCH [*/qualifiers*] *watchpoint-name*

Qualifiers

/CPU=cpu-id

Specifies the CPU, where *cpu-id* is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is used.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/SCU

/NOSCU

Determines whether to display system control unit watchpoints.

Parameters

watchpoint-name

The watchpoint to be displayed; if omitted, all watchpoints are displayed.

Example

```
>>> SHOW WATCH WATCH1
WATCH1 /FROM=0 /TO=0 /CPU=0 (enabled) USER=0
Command to execute is EXAMINE %CPU0.*PC*
%CPU0.DECODE_PC<31:0>
%CPU0OX_UPC<11:0>
%CPU0.LOAD_UPC<11:0>
```

Watchpoint WATCH1 information display.

SHOW WINDOW

Displays information about the specified window, including its name, type, location, visibility, and plane. This information appears in the command window.

Format

SHOW WINDOW [*window-name*]

Display Format

The window information is displayed in the following format:

Type	Name	Start	End	Visible	Partial	Update	Plane
Examine	TEST	2	9	N	N	Y	1
Odometer	NEW	11	19	Y	Y	Y	2
*Microcode	UCODE	2	9	Y	Y	Y	0

The window name was assigned when the window was created. *Start* and *End* are the first and last line numbers of the window. *Visible* and *Partial* indicate (yes or no) whether a complete or partial window is visible. *Plane* is the internal window ID.

Parameters

window-name

Specifies the window about which information is to be displayed.

SHOW XMI_DEVICES

Displays the node names and device types for all devices loaded with the SET XMI_DEVICES command.

Format

SHOW XMI_DEVICES

SHOW ZONE

Displays information about the memory allocation zone manager.

Format

SHOW ZONE

Example

```
>>> SHOW ZONE
```

```
Zone ID: 808D8088 Zone type: Shared User
```

```
  Extend by 312 page(s)
```

```
  Maximum of 5000 page(s) in zone
```

```
  Currently 312 page(s) in zone
```

```
  No fill on allocation
```

```
  No fill on deallocation
```

```
  Area extension is ENABLED
```

```
  Area allocated from system pool
```

```
  Area list
```

```
    Block address: 808E0C00
```

```
    Block size:    00027000
```

```
  Free block list
```

```
    Block address: 808E0D78
```

```
    Block size:    00026E80
```

```
>>>
```

The SHOW ZONE command display.

START

Begins execution in the specified CPU at the specified address.

Format

START *[/qualifiers] address-expression*

Qualifiers

/CPU=cpu-id

The CPU to be started, where *cpu-id* is one of the following:

0	1	2	3	
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY

If a CPU is not specified, the default CPU is started.

See Section 1.1.1.3 for more information on specifying a CPU. See the SET CPU command description for more information on specifying the default CPU.

/LOG

/NOLOG

Determines whether to display command results.

/PIO_MODE(D)

/NOPIO_MODE

Determines whether to enter program I/O (PIO) mode. If PIO mode is entered, typing Ctrl/P returns the SPU to console I/O (CIO) mode and displays the following message:

```
[Entering Console IO mode. Please type 'CONTINUE' to return.]
```

/PIO_PORT={OPA0|OPA1}

Determines to which port the console terminal is connected. OPA0 is the normal console port and OPA1 is the remote port.

Parameters

address-expression

The address of a location. The address can be a numeric literal, symbolic address, special operator, or lexical function. See Section 1.6 for more information on expressions.

STOP

Stops a process on the SPU or scan control module (SCM).

Format

STOP */qualifier*

Qualifiers

/JOB=port-id

The port ID (hexadecimal) of the batch job to be stopped (deleted). The SHOW BATCH command lists the port ID.

/PROCESS=process-id

The process to be stopped, where process-id is a hexadecimal number.

/SCM

Issues a BI STOP command to the SCM.

SUBMIT

Enters the specified command procedure file in a batch job queue.

Format

SUBMIT *[/[NO]LOG] file-spec*

Qualifier

/LOG

/NOLOG

Determines whether to display command results.

Parameters

file-spec

The command procedure file to be entered in the batch queue. The default file type is .CMD.

TALK

Enables parallel control/communications between the local terminal and a remote process. When talk mode is enabled, the following message is displayed:

```
%CLI-I-LINK, terminal link established from xxx
```

In talk mode, the local and remote terminals act as a single logical terminal to the remote process, with the characteristics of the remote terminal. Input from either terminal is echoed on both terminals and is treated as part of the remote process's command stream.

Pressing Ctrl/P on the terminal that initiated the link disables talk mode. When talk mode is disabled, any commands in progress continue.

Format

TALK *terminal-name*

Parameters

terminal-name

The name of the remote terminal, for example:

```
>>> TALK RTA1:  
%CLI-I-LINK, terminal link established from RTA1
```

TYPE

Displays the contents of the specified file(s). The stop (Ctrl/S) and start (Ctrl/Q) scroll keys suspend and resume screen scrolling.

Format

TYPE *[/qualifiers] file-spec[, . . .]*

Qualifiers

/CONFIRM

/NOCONFIRM (D)

Determines whether an affirmative response is required before each file is typed. Valid responses to the confirmation prompt are:

Response	Result
Y (yes)	The file is typed.
T (true)	The file is typed.
N (no)	The file is not typed.
F (false)	The file is not typed.
Q (quit)	Abort command processing.

Responses can be upper or lowercase. The first character of the response is checked for Y, T, or Q. Any response other than Y, T, N, F, or Q is interpreted as N or F.

/PAGED

/NOPAGED

Displays the file one screen page at a time.

Parameters

file-spec[, . . .]

The file to be displayed. The default file type is .CMD.

To specify two or more files, separate the file specifications with either commas or plus signs. The files are displayed in the order listed.

The asterisk (*) wildcard character can be used in place of any whole or partial field (file name, type, or version) and the percent (%) wildcard character can be used in place of any single character in the file specification. The command displays all files that satisfy the file description.

Examples

1 >>> TYPE COMMON.DAT

Displays the contents of file COMMON.DAT.

2 >>> TYPE TEST

Displays the contents of file TEST.CMD.

UNJAM

Performs an I/O subsystem reset. The SCU treats this as a power-up initialization and the XMI interface executes a RESET sequence. This, in turn, causes a BI RESET, and so on.

Format

UNJAM [*/XJA=xja-id*]

Qualifier

/XJA=xja-id

Specifies the hexadecimal number of the XJA to be reset.

VERIFY

Compares the specified structure data with the specified file-spec data. Any mismatches cause the mismatched structure and file data to be displayed.

Format

VERIFY */structure [/qualifiers] file-spec*

Structures

/MAIN_MEMORY

Compares main memory.

/PEM

Compares PEM firmware.

/RIC=ric-id

Compares RIC firmware. The ric-id is a hexadecimal number.

/STRUCTURE

Compares structure microcode.

Qualifiers

/CPU=cpu-id

Specifies the CPU that contains the structure, where cpu-id is one of the following:

0	1	2	3		
ALL	AVAILABLE	BOOTPRIMARY	BOOTSET	PRIMARY	

If a CPU is not specified, the default CPU is used.

/END=address

Specifies the address (hexadecimal) where the comparison ends.

/LOG

/NOLOG

The */LOG* qualifier displays the structure and file names.

/START=address

Specifies the structure starting address (hexadecimal), not the position in the file, where the comparison starts.

Parameters

file-spec

Specifies the file that contains the data to be compared.

WAIT

Pauses the current CLI process for the specified time. The command can be interrupted (the wait canceled) by typing Ctrl/P.

Format

WAIT *time*

Parameters

time

The wait time can be relative or absolute. Relative time is specified as a quoted string in the following format:

```
>>> wait "0 HH:MM:SS"
```

or in seconds as an unquoted integer. For example, a five second wait:

```
>>> wait 5
```

Absolute time is specified as a quoted string as follows:

```
>>> wait "DD-MMM-YYYY HH:MM:SS"
```

Examples

```
❶ >>> WAIT "0 00:00:05"
```

Wait 5 seconds (relative time format).

```
❷ >>> WAIT 5
```

Wait 5 seconds (relative time format).

```
❸ >>> WAIT "12:44:16"
```

Wait until 12:44:16 (absolute time format).

```
❹ >>> WAIT "11-JAN-1990 00:00:15"
```

Wait until January 11, 1990, 00:00:15 (absolute time format).

WRITE

Writes the string-expression to the specified file.

Format

WRITE [*/NOCR LF*] *logical-name string-expression*

Qualifier

/NOCR LF

Allows strings to be appended to each other in the output file record by not terminating the record with a carriage return.

Parameters

logical-name

The logical name assigned by the OPEN command when the file was opened. The predefined logical names STDOUT and SYS\$OUTPUT can also be used.

string-expression

An expression containing any elements that result in a string.

Z

Establishes the XMI XCOM (default) or BI RXCD communication protocol with the specified node. Commands can then be passed to the node and results displayed.

Format

Z *[/qualifier] [node]*

Qualifiers

/BI=node

Specifies the BI RXCD protocol. The node argument specifies the XBI node and the node parameter is not used.

/SPU=node

Specifies an SPU BI node and the BI RXCD protocol (the /BI qualifier should not be used). The node argument specifies the BI node and the node parameter is not used.

Parameters

node

A two-digit number with the most significant digit is the XJA number and the least significant digit is the node number. For example:

```
>>> Z 34
```

specifies XJA3 node 4.

3

Lexical Function Description

The lexical function descriptions in this chapter are nearly identical to the descriptions in the service processor operating system HELP library. In most cases, the two are distinguished only by minor formatting differences to accommodate the different media.

Lexical functions return numeric or string results and can be used anywhere numeric or string expressions are valid. The supported lexical functions are listed below and are described on the following pages:

ASCII	BITVECTOR	CLOCK	CONFIG	CPU
EXTRACT	FIELD	FILL	INFORMATION	INTEGER
LENGTH	LOCATE	LOGICAL	MCU	MEMORY
PARSE	PART	POWER	RADIX	REVISION
SCI	SCU	SEARCH	SERIAL	SID
SIGNAL	STRING	SWITCH	TIME	UPC
VERIFY				

ASCII

Returns the ASCII character for the specified number or keyword. The number is assumed to be in the default radix (usually hexadecimal).

Format

ASCII (*number*
 keyword)

Return Value

The ASCII character for the specified number or keyword.

Arguments

number

A number in the default radix (usually hexadecimal).

keyword

One of the following:

- BACKSPACE
- BELL
- ESCAPE
- FORMFEED
- LINEFEED
- NEWLINE
- RETURN
- TAB
- VERTICAL_TAB

BITVECTOR

Evaluates the specified expression and returns a bitvector data type. The expression must yield a numeric result.

Format

BITVECTOR (*expression* [, *length*])

Return Value

A bitvector data type of the specified length or fewest bytes that can hold the expression result.

Arguments

expression

The expression to be evaluated.

length

The number of bits in the returned bitvector. If the length value exceeds the number of bits in the result, the fewest bytes that can hold the result are returned.

CLOCK

Returns information about the clock subsystem.

Format

CLOCK (*item*)

Return Value

See the item argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
DISABL	DISABL bit state.
EMULATION	TRUE if clock emulation is enabled.
FREQUENCY	Frequency register.
INTERVAL	Interval register.
POSITION	Position (delay) register.
SYNCH	SYNCH bit state.

CONFIG

Returns system configuration information.

Format

CONFIG (*item*)

Return Value

See the item argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
ICU	A mask of ICUs present as a number.
ICU0	TRUE/FALSE.
ICU1	TRUE/FALSE.
MMU	A mask of MMUs present as a number.
MMU0	TRUE/FALSE.
MMU1	TRUE/FALSE.
XJA	A mask of XJAs present as a number.
XJA0	TRUE/FALSE.
XJA1	TRUE/FALSE.
XJA2	TRUE/FALSE.
XJA3	TRUE/FALSE.

CPU

Returns information about the default or specified CPU. See also the SCU lexical.

Format

CPU (*item*, [*cpu-id*])

Return Value

See the *item* argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
ALLOCATED	User who allocated CPU or " " (null) if not allocated.
AVAILABLE	TRUE if this unit is in available set.
BOOT_SET	TRUE if unit is in boot set.
BROKE	TRUE if the unit is broken.
CDB	CDB file name.
CLOCK	TRUE if unit's clock is running.
COLD_START	State of COLD_START flag.
DEFAULT	Default CPU as a string.
HARD	TRUE if the unit has passed the hard-core tests.
INITIALIZED	TRUE if unit is initialized.
POWER	TRUE if the unit has power.
PRESENT	TRUE if unit is present.
REVISION	Revision string for this unit.
RUN	TRUE if the unit is running.

<i>item</i>	Returned Value
SENSE	TRUE if this unit has been SENSED.
STATE	The UCB state bits as a number.
WARM_START	State of WARM_START flag.

cpu-id

Specifies the CPU, where *cpu-id* is one of the following integers or strings:

0 1 2 3 CPU0 CPU1 CPU2 CPU3

If not specified, the default CPU.

EXTRACT

Extracts a substring from a string based on the start position and length information.

Format

EXTRACT (*start*, *length*, *string*)

Return Value

A character substring delimited by the start and length arguments. If the start argument is greater than or equal to the length of the string, a null string (" ") is returned.

Arguments

start

An integer expression representing the offset of the extracted substring. Offset is the position of a string character or a substring relative to the leftmost string character. The leftmost, or first, character in a string is position 0.

length

The number of characters to be extracted. If the length value exceeds the number of characters in the string from the start position to the end of the string, the characters from the start position to the end of the string are returned.

string

The string from which the substring is to be extracted.

Example

```
>>> NAME = "JOHN Q. PUBLIC"  
>>> LAST = EXTRACT (8, 6, NAME)  
>>> SHOW SYMBOL LAST  
LAST = "PUBLIC"
```

The last six characters are extracted from the character string assigned to symbol NAME, and assigned to symbol LAST.

FIELD

Extracts a bitvector from the specified expression based on the specified start and length.

Format

FIELD (*start, length, expression*)

Return Value

A bitvector based on the specified start and length arguments. If the length of the returned data is less than 32 bits, an integer is returned; otherwise, a bitvector is returned.

Arguments

start

An integer value representing the offset of the field. Offset is the position of the field relative to the leftmost bit in the expression.

length

The number of bits in the extracted bitvector.

expression

The bitvector or integer from which the field is to be extracted.

FILL

Inserts the specified fill pattern into an integer or bitvector data type at the specified offset. The pattern is repeated the number of times necessary to fill the length argument.

Format

FILL (*start, length, pattern, size, expression*)

Arguments

start

Specifies the offset of the fill pattern in the expression. Offset is the position of the pattern relative to the leftmost bit in the expression.

length

Specifies the total length of the inserted data.

pattern

The bit pattern to be inserted and repeated as necessary to fill the length argument.

size

The number of bits in the pattern.

expression

The integer or bitvector into which the pattern is to be inserted.

INFORMATION

Returns information about various subsystems.

Format

INFORMATION (*subsystem* [, *item* [, *select*]])

Arguments

subsystem [, *item* [, *select*]]

<i>subsystem</i>	[, <i>item</i>	[, <i>select</i>]]	Returned Value
BOOT_PRIMARY	-	-	The boot primary number as string.
CLOCK	DISABL	-	State of DISABL bit.
CLOCK	FREQUENCY	-	Master clock frequency.
CLOCK	INTERVAL	-	Clock interval count.
CLOCK	POSITION	-	Position register.
CLOCK	SYNCH	-	Synch flag value.
CPU	AVAILABLE	cpu-id or null (D)	TRUE if CPU is in available set.
CPU	BOOTSET	cpu-id or null (D)	TRUE if CPU is in boot set.
CPU	CDB	cpu-id or null (D)	CDB file name.

3-12 Console Lexicals
INFORMATION

<i>subsystem</i>	<i>[, item</i>	<i>[, select]]</i>	Returned Value
CPU	INITIALIZED	cpu-id or null (D)	Initialized state flag.
CPU	PRESENT	cpu-id or null (D)	Present state flag.
CPU	REVISION	cpu-id or null (D)	CPU revision.
CPU	STATE	cpu-id or null (D)	Current CPU state.
DEFAULT	-	-	Default CPU.
ENVIRONMENT	FAULT	monitor-name	Fault flag.
ENVIRONMENT	STATE	monitor-name	Current state.
MODE	-	-	BATCH or INTERACTIVE.
POWER	FAULT	voltage-name	Group fault flag.
POWER	MARGIN	voltage-name	Group margin flag.
POWER	STATE	voltage-name	Group state.
PRIMARY	-	-	Primary unit number as a string.
SCOPE	-	-	Default scope as string.
SIMULATION	-	-	TRUE if the default unit is DECSIM simulation.
SYSTEM	-	-	System type (AQUARIUS, ARIDUS, TESTER, UNKNOWN).
TERMINAL	-	-	TTY name.

INTEGER

Converts a string to an integer based on the optional radix. The default radix is decimal.

Format

INTEGER (*string* [, *radix*])

Return Value

An integer value, in the specified radix, that is equivalent to the specified string expression.

Arguments

string

The string expression to be converted.

radix

Controls integer interpretation, and can be one of the following:

DECIMAL
HEXADECIMAL
OCTAL
BINARY

If not specified, the current default radix.

Example

```
>>> A = "3"  
>>> B = INTEGER("5" + A, HEXADECIMAL)  
>>> SHOW SYMBOL B  
B = 35
```

First, the string literal "5" is concatenated with the string literal "3". (Note that the value of symbol A is automatically substituted in a string expression and the plus (+) is a string concatenation operator.) After the string expression is evaluated, the character string "53" is converted to integer value 35₁₆ and assigned to symbol B.

LENGTH

Returns the length of a string.

Format

LENGTH (*string*)

Return Value

An integer value for the length of the string.

Arguments

string

The character string of which the length is to be determined.

Example

```
>>> MESSAGE = "exceeded quota"  
>>> STRING_LENGTH = LENGTH(MESSAGE)  
>>> SHOW SYMBOL STRING_LENGTH  
STRING_LENGTH = 14
```

The LENGTH function returns the length of the character string assigned to the symbol MESSAGE, and assigns the value, 14, to the symbol STRING_LENGTH. Note that quotation marks are not used around symbols (for example, MESSAGE) in character string expressions.

LOCATE

Returns the offset of a substring within a string. Returns the length of the string if the substring is not part of the string.

Format

LOCATE (*substring*, *string*)

Return Value

An integer value representing the offset of the substring argument from the first character in the string. The first character in a string is the leftmost character and is position 0. If the substring argument is not found, the length of the string is returned.

Arguments

substring

The string of characters to be located in the specified string. Specify the substring as a character string expression.

string

The string in which the specified substring is to be found. Specify the string as a character string expression.

Example

```
>>> NAME = "JOHN Q. PUBLIC"  
>>> NO_LAST = LOCATE(".", NAME)  
>>> SHOW SYMBOL NO_LAST  
NO_LAST = 6
```

The LOCATE function returns the position of the period in the string with respect to the beginning of the string. The period is in offset position 6, and that value is assigned to symbol NO_LAST.

The period character is the substring argument and is specified as a string literal (in quotation marks). The string argument NAME is a symbol and is not placed in quotation marks. NAME is automatically replaced by its current value.

LOGICAL

Returns the translation of a logical name.

Format

LOGICAL (*logical-name*)

Return Value

An equivalence string for the specified logical name. The LOGICAL function searches process, job, group, and system logical name tables, in that order, for the first match. If no match is found, LOGICAL returns a null string.

Arguments

logical-name

The logical name is passed as a string expression.

MCU

Returns information about the specified MCU.

Format

MCU (*number*, *item* [, *ring*])

Return Value

See the *item* argument description for returned values.

Arguments

number

The MCU number.

item

<i>item</i>	Returned Value
PRESENT	TRUE if MCU is present in system.
REVISION	The revision field.
RING_LENGTH	Specified CDB ring length or broadcast ring length if ring is not specified.
SERIAL	The serial number (as a number).
STATE	MCU state (GOOD/BROKEN).
TESTED_LENGTH	Specified tested ring length or broadcast ring length if ring is not specified.
TYPE	The type field.
VARIATION	The variation field.

ring

Optionally specifies the ring for the RING_LENGTH and TESTED_LENGTH items. If not specified, the broadcast ring is the default.

MEMORY

Returns information about the memory subsystem.

Format

MEMORY (*item*)

Return Value

See the *item* argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
ALLOCATED	The user name who allocated memory; otherwise, " " (null). Currently always returns " " as there is no way to allocate memory.
INITIALIZED	TRUE if the memory have been initialized; otherwise, FALSE.
SIZE	The highest location in main memory.

PARSE

Parses a file name and extracts a specified field.

Format

PARSE (*file-spec* [, *default-spec*] [, *field*])

Return Value

A character string containing the expanded file specification or the specified field.

Arguments

file-spec

The file specification to be parsed, specified as a character string expression. Wildcard characters can be used; if used, they appear in the returned file specification.

default-spec

A default file specification, specified as a character string expression. Fields in the default file specification are substituted in the output string for missing fields in the *file-spec* argument.

field

The name of a field in a file specification, specified as a character string expression. This argument returns a specific portion of the file specification. The field name can be abbreviated. Valid keywords for the field name are as follows:

Keyword	Field Returned
NODE	Node name
DEVICE	Device name
DIRECTORY	Directory name
NAME	File name
TYPE	File extension
VERSION	File version

NOTE

No punctuation is returned with the type and version fields.

Examples

```
❶ >>> SET DEF DUA0:[CONSOLE]
>>> DIAG = PARSE("TEST1.CMD", "[PROCS]")
>>> SHOW SYMBOL DIAG
>>> DIAG = "DUA0:[PROCS]TEST1.CMD;"
```

The default device and directory are DUA0:[CONSOLE]. Because the directory name [PROCS] is specified as the default-spec argument in the assignment statement, it is used as the directory name in the output string. Note that the default device returned in the output string is DUA0 and the default version number for the file is null. The arguments TEST1.CMD and [PROCS] are placed in quotes because they are string literals.

```
❷ >>> SET DEFAULT DUA0:[CONSOLE]
>>> DIAG = PARSE("TEST1.CMD", , "DIRECTORY")
>>> SHOW SYMBOL DIAG
>>> DIAG = "[CONSOLE]"
```

PARSE returns the directory name of file TEST1.CMD. Note that the default-spec argument's place in the argument list is delimited by a comma when the argument is omitted.

```
❸ >>> DIAG = PARSE("SERE::DUA0:[CONSOLE]TEST1.CMD", , "DEVICE")
>>> SHOW SYMBOL DIAG
>>> DIAG = "DUA0:"
```

A file specification containing a node name is parsed for the DEVICE field and returns DUA0:.

PART

Returns the specified element's part identifier as a string. See also the SERIAL and REVISION lexicals.

Format

PART (*element*
element, arg)

Return Value

The specified element's part identifier as a string.

Arguments

element

<i>element</i>	<i>arg</i>
AIE	None.
AIO	None.
DAC	Specifies which DAC.
MAC	Specifies which MAC.
MCM	None.
MCU	MCU name or number (for example, VAP or 4).
PEM	None.
PLANAR	Specifies CPU _n or SCU. If not specified, default scope unit is used.
RIC	Specifies which RIC, as in 0x52, and so on. This includes CPRIC and IORICs.
SCM	None.
SPM	None.

arg

See the element argument table.

POWER

Returns information about system power.

Format

POWER (*item*, [*bus/cabinet*])

Return Value

See the item argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
ALLOCATED	User who allocated PCS or " " (null) if not allocated.
FAULT	FAULT state.
MARGIN	HI/LO/NOMINAL.
STATE	TRUE/FALSE.

bus/cabinet

The bus argument is one of the following:

A, B, C, D, E, F, G, H, I, or J

Valid bus arguments depend on system configuration.

The cabinet argument is one of the following:

CPA, CPB, SCU, IOA, or IOB

If bus/cabinet are not specified, the returned value is the ORed result of all the buses.

RADIX

Returns the string name of the current radix.

Format

RADIX ()

Return Value

The current radix is returned as one of the following strings:

BINARY
DECIMAL
HEXADECIMAL
OCTAL

REVISION

Returns the specified element's revision string. This is the part of the string following the part number. For example, if the part number is nn-nnnnn-nn.llnn, *ll* is returned. See also the PART lexical.

Format

REVISION (*element*
element, arg)

Return Value

The specified element's revision string.

Arguments

element

<i>element</i>	<i>arg</i>
MCU	MCU name or number (for example, VAP or 4).
KERNEL	None.

arg

Specifies which MCU.

SCI

Returns an integer data type with the value of the specified SCI line.

Format

SCI (*line, port*)

Return Value

The last value set with the SET SCI command on the specified line.

Arguments

line

Specifies one of the following SCI lines:

- BROADCAST
- BYPASS
- CDS
- DATA_IN
- DATA_OUT
- FUNCTION
- SELECT

port

Specifies one of the following SCI ports:

- CPU0
- CPU1
- CPU2
- CPU3
- SCU
- MEM

SCU

Returns information about the SCU. See also the CPU lexical.

Format

SCU (*item*)

Return Value

See the *item* argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
ALLOCATED	User who allocated SCU or " " (null) if not allocated.
BROKE	TRUE if the unit is broken.
CDB	CDB file name.
CLOCK	TRUE if unit's clock is running.
DEFAULT	Default CPU as a string.
HARD	TRUE if the unit has passed the hard-core tests.
INITIALIZED	TRUE if unit is initialized.
POWER	TRUE if the unit has power.
REVISION	Revision string for this unit.
RUN	TRUE if the unit is running.
SENSE	TRUE if this unit has been SENSED.
STATE	The UCB state bits as a number.

SEARCH

Scans a directory and returns expanded file names that match the input file-spec.

Format

SEARCH (*file-spec*)

Return Value

A character string containing the expanded file specification for the file-spec argument. Each time the SEARCH lexical is called with the same file-spec argument, it returns the next expanded file specification that matches the argument. After the last file-spec match or if the file is not found in the directory, a null string is returned.

Arguments

file-spec

The file specification to be searched for, specified as a character string expression. If not specified, default device and directory are used. Defaults are not supplied for file name or type. If the version number is not specified, the file specification with the highest version number is returned. Wildcards can be used.

Example

```
$ START:
$ FILE = SEARCH("SYS$SYSTEM:*.EXE")
$ IF FILE .EQL. " " THEN EXIT
$ SHOW SYMBOL FILE
$ GOTO START
```

This command procedure displays the file-specs of the latest version of all .EXE files in the SYS\$SYSTEM directory. (Only the latest version is returned because a wildcard is not used as the version number.) The file-spec argument SYS\$SYSTEM:*.EXE is a character string expression and is placed in quotation marks.

SERIAL

Returns the specified element's serial number as a string. See also the PART lexical.

Format

SERIAL (*element*
element, arg)

Return Value

The specified element's serial number as a string.

Arguments

element

<i>element</i>	<i>arg</i>
AIE	None.
AIO	None.
DAC	Specifies which DAC.
MAC	Specifies which MAC.
MCM	None.
MCU	MCU name or number (for example, VAP or 4).
PEM	None.
PLANAR	CPU or SCU or null to use default scope unit.
RIC	Specifies which RIC, as in 0x52, and so on. This includes CPRIC and IORICs.
SCM	None.
SPM	None.

arg

See the element argument table.

SID

Returns the system ID.

Format

SID (*[item]*)

Return Value

See the item argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
SIDEX	SPM SIDEX register.
SPM	SPM system ID register.
SYSTEM	System ID (default if item is not specified).

SIGNAL

Returns the value of a signal.

Format

SIGNAL (*signal-name*)

Return Value

The value of the specified signal.

Arguments

signal-name

A string expression that is a valid signal name.

STRING

Converts an integer expression to a string.

Format

STRING (*expression* [, *radix*])

Return Value

A character string that is equivalent to the specified expression.

Arguments

expression

The expression to be evaluated using the specified radix. Leading zeros are dropped, and a minus (-) is placed at the beginning of the string representation of a negative number.

radix

Controls integer interpretation, and can be one of the following:

DECIMAL (D)
 HEXADECIMAL
 OCTAL

Example

```
>>> A = 5
>>> B = STRING(-2 + 'A')
>>> SHOW SYMBOL B
B = "3"
```

First, the expression $(-2 + 'A')$ is evaluated. Note that 5, the value of symbol A, is automatically substituted when the integer expression is evaluated. Next, the resulting integer, 3, is converted to the string "3" and assigned to symbol B.

NOTE

Assuming the default radix is hexadecimal, symbol A must be enclosed in quotes to distinguish it from A_{16} .

SWITCH

Returns information about the operator control panel switches.

Format

SWITCH (*item*)

Return Value

See the *item* argument description for returned values.

Arguments

item

<i>item</i>	Returned Value
BOOT	Boot switch state.
ACCESS	Access switch state.

TIME

Returns the current date and time string.

Format

TIME ()

Return Value

The current date and time string.

UPC

Returns the current microPC of the specified control store.

Format

UPC (*control-store*)

Return Value

The current microPC of the specified control store.

Arguments

control-store

One of the following:

ECS = EBox control store

JCS = JBox control store

If not specified, the control store of the selected window. (See the SELECT command description in Chapter 2.)

VERIFY

Returns an integer value indicating whether the procedure verification flag is currently on or off. If used with arguments, the procedure verification flag can be turned on or off.

Format

VERIFY [(*expression*)]

Return Value

The integer 0 if the procedure verification flag is off, or the integer 1 if the procedure verification flag is on.

Arguments

expression

An integer expression with a value of 0 to turn the procedure verification flag off, or 1 to turn it on. When specified, the VERIFY function first displays the current flag state, then turns procedure verification on (1) or off (0) as specified by the argument.

When procedure verification is on, each command line in the procedure is displayed.

A

Command Quick Reference

General Syntax: VERB [/qualifiers] object [parameters]
(D) = default

symbol-name :=[=] string

symbol-name =[=] expression

@ file-spec [p1 [p2 [. . . p8]]]

ALLOCATE CPU=cpu-id
MCM
PCS
SCU

BOOT [/BI=node-id
/NODE=node-id
/[NO]PIO_MODE(D)
/R3=register-data
/R5=boot-flags
/[NO]START(D)
/XMI=xmi-id] [device]

CALL label

CLOSE logical-name

CONTINUE [/CPU=cpu-id
/[NO]LOG
/[NO]PIO_MODE(D)
/PIO_PORT={OPA0 | OPA1}]

A-2 Command Quick Reference

COPY [/[NO(D)]CONFIRM
/CONTIGUOUS
/[NO(D)]LOG] input-file-spec output-file-spec

CREATE [/[NO(D)]LOG] file-spec

CREATE/DIRECTORY [/[NO(D)]LOG] directory-spec

CREATE/WINDOW [/CPU=cpu-id
/ECS
/EXAMINE
/JCS
/ODOMETER
/[NO]UPDATE(D)
/VIEWONLY] window-name [AT [W1
H1 ... H2
T1 ... T3
Q1 ... Q4]]

DEALLOCATE CPU=cpu-id
MCM
PCS
SCU

DEASSIGN [/ALL
/PROCESS(D)
/SYSTEM] [logical-name]

DEBUG [/[NO]CONFIRM(D)]

DEFINE [/[NO]LOG(D)
/PROCESS(D)
/SYSTEM] logical-name[:] equivalence-string

DEFINE/KEY [/[NO]ECHO(D)
/LOG
/[NO(D)]TERMINAL] E1 ... E6
F6 ... F14
Help
Do
F17 ... F20
PF1 ... PF4 equivalence-string
KP0 ... KP9
PERIOD
COMMA
MINUS
ENTER

DELETE [/[NO(D)]CONFIRM] file-spec
 [/[NO(D)]LOG]

DELETE/PATTERN [/ALL
 /CPU=cpu-id
 /[NO(D)]LOG] [pattern-name]
 /RESET
 /[NO]SCU]

DELETE/SYMBOL [/ALL
 /GLOBAL
 /LOCAL(D)] [symbol-name]
 /[NO(D)]LOG]

DELETE/TRACE [/ALL
 /CPU=cpu-id
 /[NO(D)]LOG] [tracepoint-name]
 /RESET
 /[NO]SCU]

DELETE/WATCH [/ALL
 /CPU=cpu-id
 /[NO(D)]LOG] [watchpoint-name]
 /RESET
 /[NO]SCU]

DELETE/WINDOW [/ALL
 window-name]

<p>DEPOSIT</p>	<p> /PHYSICAL /SPU /VIRTUAL /ECS /EREG /GENERAL /INTERNAL /JCS /VECTOR=register:element /ASCII /BYTE /D_FLOAT /F_FLOAT /G_FLOAT /LONGWORD(D) /OCTAWORD /QUADWORD /WORD /LENGTH=bits /RING /NEXT[=count] /CPU=cpu-id /[NO(D)]SCU /[NO]LOG /[NO]VERIFY /MCM /PEM /RIC=ric-id /SCC /SJA /CODE /EMEMORY /IMEMORY /PORT_REGISTER /REGISTER </p>	<p>address-expression value-expression</p>
----------------	--	--

DIRECTORY $\left[\begin{array}{l} /DATE \\ /FULL \\ /OWNER \\ /PROTECTION \\ /SIZE \\ /TOTAL \end{array} \right]$ [file-spec]

DISMOUNT [/[NO]UNLOAD] device-name[:]

EDIT $\left[\begin{array}{l} /[NO]COMMAND(D)[=command-file] \\ /[NO]CREATE(D) \\ /[NO]JOURNAL[=journal-file] \\ /[NO]OUTPUT[=output-file] \\ /RECOVER \end{array} \right]$ input-file-spec

EVALUATE $\left[\begin{array}{l} /DISPLAY=radix-spec \\ /RADIX=radix-spec \end{array} \right]$ expression

A-6 Command Quick Reference

EXAMINE

```
/PHYSICAL
/SPU
/VIRTUAL

/ECS
/EREG
/GENERAL
/INTERNAL
/JCS
/VECTOR=register:element

/ASCII[=count]
/BYTE
/D_FLOAT
/F_FLOAT
/G_FLOAT
/INSTRUCTION
/LONGWORD(D)
/OCTAWORD
/QUADWORD
/WORD

/LABEL

/LENGTH=bits
/RING

/NEXT[=count]

/CPU=cpu-id
/[NO(D)]SCU

/[NO]LOG
/SYMBOL=name
/WINDOW[=window-name]

/MCM
/PEM
/RIC=ric-id
/SCC
/SJA

/CODE
/EMEMORY
/IMEMORY
/PORT_REGISTER
/REGISTER
```

address-expression
signal-name
structure-name

EXIT /ATTN [status-code]
 /FAULT

FIND /CPU=cpu-id
 /MEMORY [blocks]
 /RPB

GOTO label

HALT [/CPU=cpu-id]

HELP [topic . . .]

IF expression THEN command

INITIALIZE	<pre> /CLOCK /CPU=cpu-id /IO /KERNEL [/BRIEF] /MEMORY [/BANK_MASK=mask /INTERLEAVE=type /[NO]OUTPUT=file-spec /[NO]RESTORE=file-spec /[NO]TEST(D)] /POWER /SCAN [/FIRMWARE=file-spec /[NO]LOG /[NO]RESET /SCU [/BRIEF]] /SCM [/[NO]DEBUG /FIRMWARE=file-spec /[NO]LOG /SUNDANCE /TIMEOUT=seconds /WPT_AREA=kbytes] /SJA [/[NO]DEBUG /[NO]LOG /[NO]SIMULATION] /VOLUME </pre>
------------	--

A-8 Command Quick Reference

INQUIRE [/[NO]EXPRESSION
/[NO]STRING] symbol-name prompt-string
prompt-symbol

LABEL: [command]

LOAD [/ABS
/CDB
/MAIN_MEMORY(D)
/PEM
/RIC=ric-id
/RING[=ring-id]
/STRUCTURE
/TEXT
/VECTOR] [/CPU=cpu-id
/[NO]DATA(D)
/[NO]LOG
/REVISION=version
/[NO]SCU
/START=address
/[NO]SYMBOL
/[NO]VERIFY] file-spec

LOGOUT [/BRIEF
/FULL(D)]

MAIL [/FILE=file-spec
/[NO]SELF
/SUBJECT="string"] file-spec recipient[, ...]
@recipient-list

MICROSTEP [/[NO]BURST
/CPU=cpu-id
/[NO]SCU
/[NO]SPACEBAR(D)] step-count

MOUNT device volume-name

NEXT [/CPU=cpu-id
/[NO]SPACEBAR(D)
/[NO]VIRTUAL] [step-count]

ATTN
ERROR
ON FAULT THEN command
SEVERE
WARNING

OPEN [/APPEND
/ERROR=label
/[NO]LOG
/READ(D)
/WRITE] logical-name file-spec

PURGE [/[NO(D)]CONFIRM
/KEEP=number
/[NO(D)]LOG] [file-spec[, ...]]

READ [/END_OF_FILE=label
/ERROR=label] logical-name symbol-name

REBOOT /[NO]CONFIRM(D)

RECALL [/ALL | command | index]

RENAME [/[NO(D)]CONFIRM
/[NO(D)]LOG] input-file-spec[, ...] output-file-spec

REPEAT [/COUNT=number] command

RESET [/CPU=cpu-id
/[NO]SCU]

RESTORE /CPU=cpu-id
/[NO]SCU
/[NO]SPU

RETURN

RUN [/[NO]DEBUG
/[NO]DETACHED
/JOB_PRIORITY=level
/KERNEL_STACK=size
/[NO]LOAD
/MAXIMUM_MESSAGES=number
/MODE={KERNEL | USER}
/PARAMETERS="string"
/[NO]POWER_RECOVERY
/PROCESS_PRIORITY=number
/USER_STACK=size] file-spec [command-line]

SAVE /CPU=cpu-id
/[NO]SCU
/[NO]SPU

SCROLL [/CURRENT
/DOWN[=lines](D)
/LEFT[=columns]
/NEXT[=lines]
/PREVIOUS[=lines]
/RIGHT[=columns]
/TO=address
/UP[=lines]] [window-name]

SELECT [/NEXT] [window-name [AT W1
H1 ... H2
T1 ... T3
Q1 ... Q4]]]

SEND [/ALL
/BELL
/OPCOM
/URGENT] [terminal-name] string

CLOCK
CPU=cpu-id
IO
SENSE POWER
SCU
SYSTEM [/COMPARE]
[/[NO]LOG]

SET ATTN_ACTION command

SET AUTOBOOT {ON | OFF}

SET BI_DEVICES [file-spec
device[, ...]]

SET BOOTFLAGS value

SET BOOTSET [/ENABLE=(cpu-id [, ...])
/DISABLE=(cpu-id [, ...])
/PRIMARY=cpu-id] cpu-id [, ...]

SET CLOCK [/[NO]ATTENTION(D)=attention-name
/CPU=cpu-id
/[NO]EMULATION
/FREQUENCY=value
/INTERVAL=cycles
/[NO]LOG
/POSITION=value
/SAMPLE_RATE=hertz
/[NO]SCU
/SYNCH=synch-option] [ON(D) | OFF]

SET COLD_START [/CPU=cpu-id] {ON | OFF}

SET COMMAND [/CLEAR
/[NO(D)]LOG] file-spec

SET CPU *0 ... 3* **[/[NO]LOG(D)] BOOTPRIMARY
PRIMARY**

SET CYCLE $\left[\begin{array}{l} /CPU=cpu-id \\ /INTERVAL=value \\ /[NO]SCU \end{array} \right]$ *count*

SET DEFAULT *directory-spec[:]*

SET ERROR_HANDLING $\left[\begin{array}{l} /[NO]MATCH \\ /[NO]RECOVERY \\ /[NO]REPORTING \end{array} \right]$ *{ON | OFF}*

SET FAULT_ACTION *command*

SET ISOLATION $\left[\begin{array}{l} /CPU=cpu-id \\ /LOG \\ /SYBIL \end{array} \right]$ *file-spec*

SET KEEP_ALIVE *[/CPU=cpu-id] {ON | OFF | MANUAL}*

SET LABELS *{ON | OFF}*

SET LOGGING $\left[\begin{array}{l} /ALL \\ /DISABLE \\ /ENABLE \\ /FILE=[file-spec] \end{array} \right]$ *{ON | OFF}*

SET MESSAGE $\left[\begin{array}{l} /[NO]FACILITY \\ /[NO]IDENT \\ /[NO]SEVERITY \\ /[NO]TEXT \end{array} \right]$ *[file-spec]*

SET PATTERN $\left[\begin{array}{l} /ABSOLUTE \\ /ALL \\ /CPU=cpu-id \\ /DISABLE \\ /ENABLE \\ /FILE=file-name \\ /[NO]LOG \\ /NAME=pattern-name \\ /ODOMETER[=window-name] \\ /RELATIVE \\ /[NO]SCU \\ /SYBIL \\ /[NO]VERIFY \\ /SYBIL pattern-name \end{array} \right]$ *@file-name
pattern-name
signal-list*

A-12 Command Quick Reference

SET PERSONAL_NAME "string"

SET POWER [/[NO]ATTENTION=class[, ...]
/BUS=bus-name
/CABINET=cabinet-id
/COUNTERS
/IO
/[NO]MARGIN=margin-keyword] [ON(D) | OFF]

SET PROMPT [prompt-string]

SET RADIX HEXADECIMAL
DECIMAL
OCTAL
BINARY

SET REMOTE [/[NO]PASSWORD=string] {ON | OFF}

SET REVISION [/OS
/PLANAR [/CPU=cpu-id
/SCU]] revision-string

SET SCI [/[NO]BROADCAST={0 | 1}
/[NO]BYPASS={0 | 1}
/[NO]CDS={0 | 1}
/CLOCK=state
/DATA={0 | 1}
/DEFAULT
/FUNCTION=state
/PORT=port-id
/SELECT={0 | 1}]

SET SCM [/[NO]ATTENTIONS=mask
/[NO]BI_VERIFY
/[NO]BYPASS
/[NO]CACHE
/RATE=scan-rate
/[NO]SCAN_VERIFY
/[NO]VECTOR_PROCESSOR=mask
/[NO]VERIFY]

SET SCOPE [/[NO]LOG] label-spec

SET SCREEN {ON(D) | OFF}

```

/AIE
/AIO
SET SERIAL /DAC [/IDENTIFIER=dac-number] 10-digit-serial-number
/PLANAR [ /CPU=cpu-id ]
/SCU
/SCM

```

```

SET SNAPSHOT [ /CPU=cpu-id
/FILENAME=file-spec ] {ON | OFF | TRIGGER}
/[NO]SCU

```

SET SOURCE directory

SET STEP {INSTRUCTION(D) | NOINSTRUCTION}

```

SET TERMINAL [ /[NO]BROADCAST(D)
/CPU=cpu-id
/DEVICE={LA100 | VT200}
/[NO]ECHO(D)
/[NO]EIGHTBIT
/[NO]ESCAPE
/KEYPAD={APPLICATION | NUMERIC}
/PAGE=lines
/PIO_PORT={OPA0 | OPA1}
/[NO]PROGRAM
/[NO]TALK_MODE
/WIDTH=characters ] [terminal-name]

```

SET TIME time-string

```

SET TRACE [ /ABSOLUTE
/ALL
/COMPARE
/CPU=cpu-id
/DISABLE
/ENABLE
/FILE=file-spec
/FROM=cycle
/[NO]LOG
/NAME=tracepoint-name
/ODOMETER[=odometer-window-name]
/RELATIVE
/[NO]SCU
/TO=cycle
/VECTOR ] signal-list
@signal-file

```

SET VERIFY {ON(D) | OFF}

SET WARM_START [/CPU=cpu-id] {ON | OFF}

A-14 Command Quick Reference

SET WATCH [/ALL
/CPU=cpu-id
/DISABLE
/ENABLE
/FROM=cycle
/[NO]LOG
/NAME=watchpoint-name
/[NO]SCU
/TO=cycle] signal-list DO command
@signal-file

SET XMI_DEVICES [file-spec]

SHOW ATTN_ACTION

SHOW AUTOBOOT

SHOW AVAILABLE

SHOW BATCH

SHOW BBU

SHOW BI_DEVICES

SHOW BOOTFLAGS

SHOW BOOTSET

SHOW CLOCK [/FULL]

SHOW CONFIGURATION [/CLOCK
/CPU=cpu-id
/DATE=date
/KERNEL
/MCU=mcu-id
/MEMORY
/OUTPUT=file-spec
/POWER
/RINGS
/SCU
/SPU
/SYSTEM
/XMI=xmi-id]

```

SHOW CPU [ /ALL
           /FULL ] [cpu-id]

SHOW CYCLE [ /CPU=cpu-id
            /[/NO]SCU ]

SHOW DEFAULT

SHOW DEVICE [device-name]

SHOW ENVIRONMENT [ /CONTINUOUS
                  /INTERVAL=seconds
                  /OUTPUT=file-spec ]

SHOW ERROR_HANDLING

SHOW FAULT_ACTION

SHOW FLAGS [/CPU=cpu-id]

SHOW HISTORY [ /[/NO]BINARY
              /CPU=cpu-id
              /[/NO]INSTRUCTION
              /MAXIMUM=locations
              /OUTPUT=file-spec
              /PHYSICAL
              /[/NO]SCU
              /VIRTUAL
              /WINDOW[=window-name] ]

SHOW ISOLATION [ /CPU=cpu-id
                /MCU=mcu-id ] [broadcast-ring-bit-number]
                /SYBIL

SHOW KEEP_ALIVE

SHOW KEY /ALL
        E1 ... E6
        F6 ... F14
        Help
        Do
        F17 ... F20
        PF1 ... PF4
        KP0 ... KP9
        PERIOD
        COMMA
        MINUS
        ENTER
    
```

A-16 Command Quick Reference

SHOW LOGGING

SHOW LOGICAL [/ALL
/PROCESS] [logical-name]
/SYSTEM

SHOW MEMORY

SHOW MESSAGE [message-id]

SHOW MODE

SHOW NODE

SHOW PATTERN [/CPU=cpu-id
/NODE=node-id] [pattern-name]
/REMOTE
/[NO]SCU
/SYBIL

SHOW PERSONAL

SHOW POWER [/BUS=bus-name
/CABINET=cabinet-id
/CONTINUOUS
/COUNTERS
/INTERVAL=seconds
/IO
/OUTPUT=file-spec]

SHOW PROCESS

SHOW RADIX

SHOW REMOTE

SHOW SCI [/[NO]DATA_ONLY
/PORT=port-id]

SHOW SCM [/BAD_PAGE_MAP
/ENTRY_BLOCK
/ERROR_BLOCK
/MEMORY]

SHOW SCOPE

SHOW SCU

SHOW SJA

SHOW SOURCE

SHOW STEP

	BP	
	CACHE0	
	CACHE1	
	ECC0	
	ECC1	
	ECS	
	EREG	
	FRAM	
	IPAMM	
	JCS	
SHOW STRUCTURE	MPAMM	[/CPU=cpu-id]
	NPAMM	[/[NO]SCU]
	PCHB	
	TAG0	
	TAG1	
	TAGRM	
	TBRAMS	
	VALIDRAMS	
	VIC	
	VICA	
	VICB	
	VREG	
SHOW SWITCHES		

A-18 Command Quick Reference

SHOW SYMBOL [/ALL(D)
/GLOBAL
/LOCAL

/BP
/CACHE0
/CACHE1
/ECC0
/ECC1
/ECS
/EREG
/FRAM
/IPAMM
/JCS
/MPAMM
/NPAMM
/PCHB
/TAG0
/TAG1
/TAGRM
/TBRAMS
/VALIDRAMS
/VIC
/VICA
/VICB
/VREG] [symbol-name]

SHOW SYSTEM [/[NO]PROCESS]

SHOW TERMINAL [terminal-name:]

SHOW THRESHOLD

SHOW TIME

SHOW TRACE [/CPU=cpu-id] [tracepoint-name]
/[NO]SCU

SHOW USERS

SHOW VERSION [/ALL
/CLI(D) [/FACILITY=name]
/CLOCK
/RTL [/FACILITY=name]
/SCM]

SHOW WATCH [/CPU=cpu-id] [watchpoint-name]
/[NO]SCU]

SHOW WINDOW [window-name]

SHOW XMI_DEVICES

SHOW ZONE

START [/CPU=cpu-id
/[NO]LOG
/[NO]PIO_MODE(D)
/PIO_PORT={OPA0 | OPA1}] address-expression

STOP /JOB=port-id
/PROCESS=process-id process-name
/SCM

SUBMIT [/LOG] command-procedure-file

TALK terminal-name

TYPE [/[NO(D)]CONFIRM] file-spec[, ...]
/[NO]PAGED

UNJAM [/XJA=xja-id]

VERIFY [/MAIN_MEMORY] [/CPU=cpu-id
/PEM /END=address
/RIC=ric-id /[NO]LOG
/STRUCTURE] [/START=address] file-spec

WAIT time

WRITE [/NOCTRLF] logical-name string-expression

X address count

Z [/BI=node
/SPU=node
node]

B

Lexical Function Quick Reference

VERB (argument [, optional-argument], ...)

(D) = default

General Syntax: $\left\{ \begin{array}{l} \text{ARG1} \\ \dots \\ \text{ARGn} \end{array} \right\}$ = Select one required argument.

$\left[\begin{array}{l} \text{ARG1} \\ \dots \\ \text{ARGn} \end{array} \right]$ = Select one optional argument.

ASCII ($\left\{ \begin{array}{l} \text{number} \\ \text{BACKSPACE} \\ \text{BELL} \\ \text{ESCAPE} \\ \text{FORMFEED} \\ \text{LINEFEED} \\ \text{NEWLINE} \\ \text{RETURN} \\ \text{TAB} \\ \text{VERTICAL_TAB} \end{array} \right\}$)

BITVECTOR (expression [, length])

CLOCK ($\left\{ \begin{array}{l} \text{DISABL} \\ \text{EMULATION} \\ \text{FREQUENCY} \\ \text{INTERNAL} \\ \text{POSITION} \\ \text{SYNCH} \end{array} \right\}$)

B-2 Lexical Function Quick Reference

CONFIG ({
 ICU
 ICU0
 ICU1
 MMU
 MMU0
 MMU1
 XJA
 XJA0
 XJA1
 XJA2
 XJA3
})

CPU ({
 ALLOCATED
 AVAILABLE
 BOOT_SET
 BROKE
 CDB
 CLOCK
 COLD_START
 DEFAULT
 HARD
 INITIALIZED
 POWER
 PRESENT
 REVISION
 RUN
 SENSE
 STATE
 WARM_START
} [, cpu-id])

EXTRACT (start, length, string)

FIELD (start, length, expression)

FILL (start, length, pattern, size, expression)

INFORMATION ({
 BOOT_PRIMARY
 CLOCK [, [DISABLE
 FREQUENCY
 INTERVAL
 POSITION
 SYNCH]]
 CPU [, [AVAILABLE
 BOOTSET
 CDB
 INITIALIZED
 PRESENT
 REVISION
 STATE]] [, cpu-id
 null(D)]
 DEFAULT
 ENVIRONMENT [, [FAULT
 STATE]] [, monitor-name]
 MODE
 POWER [, [FAULT
 MARGIN
 STATE]] [, voltage-name]
 PRIMARY
 SCOPE
 SIMULATION
 SYSTEM
 TERMINAL
)

INTEGER (string [, [HEXADECIMAL
 DECIMAL
 OCTAL
 BINARY]])

LENGTH (string)

LOCATE (substring, string)

LOGICAL (logical-name)

B-4 Lexical Function Quick Reference

MCU (number, { PRESENT
REVISION
RING_LENGTH [, ring]
SERIAL
STATE
TESTED_LENGTH [, ring]
TYPE
VARIATION })

MEMORY ({ ALLOCATED
INITIALIZED
SIZE })

PARSE (file-spec [, default-spec] [, DEVICE
DIRECTORY
NAME
NODE
TYPE
VERSION])

PART ({ AIE
AIO
DAC, dac-id
MAC, mac-id
MCM
MCU, mcu-id
PEM
PLANAR [, CPUⁿ
SCU]
RIC, ric-id
SCM
SPM })

POWER ({ ALLOCATED
FAULT
MARGIN
STATE [A
B
C
D
E
F / CPA
G
H
I
J
CPB
IOA
IOB
SCU] })

RADIX ()

REVISION ({ KERNEL
MCU, mcu-id })

SCI ({ BROADCAST
BYPASS
CDS
DATA_IN
DATA_OUT
FUNCTION
SELECT } , { CPU0
CPU1
CPU2
CPU3
MEM
SCU })

SCU ({ ALLOCATED
BROKE
CDB
CLOCK
DEFAULT
HARD
INITIALIZED
POWER
REVISION
RUN
SENSE
STATE })

SEARCH (file-spec)

SERIAL ({ AIE
AIO
DAC, dac-id
MAC, mac-id
MCM
MCU, mcu-id
PEM
PLANAR [, CPUⁿ
SCU]
RIC, ric-id
SCM
SPM })

B-6 Lexical Function Quick Reference

SID ({ SIDEX
SPM
SYSTEM })

SIGNAL (signal-name)

STRING (expression [, HEXADECIMAL
DECIMAL(D)
OCTAL])

SWITCH ({ ACCESS
BOOT })

TIME ()

UPC ({ null
ECS
JCS })

VERIFY [(expression)]

Index

:=, 2-3
=, 2-5
@, 2-7

A

Address

mnemonics

DEPOSIT, 2-50

EXAMINE, 2-70

space qualifier, 2-46, 2-47, 2-65,
2-67

/CODE, 2-46, 2-47, 2-65,
2-67

/EMEMORY, 2-46, 2-47,
2-65, 2-67

external memory, 2-46,
2-47, 2-65, 2-67

/IMEMORY, 2-46, 2-47,
2-65, 2-67

internal

memory, 2-46, 2-47,
2-65, 2-67

register, 2-46, 2-47,
2-65, 2-67

physical, 2-46, 2-66

port register, 2-46, 2-47,
2-65, 2-67

/PORT_REGISTER, 2-46,
2-47, 2-65, 2-67

program, 2-46, 2-47, 2-65,
2-67

DEPOSIT, 2-43

EXAMINE, 2-62

/REGISTER, 2-46, 2-47,
2-65, 2-67

virtual, 2-48, 2-68

AIE, SET SERIAL, 2-173

AIO, SET SERIAL, 2-173

ALLOCATE, 2-2

and DEALLOCATE, 2-24

ALL qualifier, 1-5

Arithmetic operators, 1-11

ASCII

ASCII lexical, 3-2

string

DEPOSIT, 2-43

EXAMINE, 2-62

Assign

string (:=), 2-3

symbol (=), 2-5

and DELETE/SYMBOL,
2-36

At (@, execute procedure), 2-7

Available set, 1-4

SHOW AVAILABLE, 2-196

B

Batch, SHOW BATCH, 2-197

BBU (battery backup unit), SHOW

BBU, 2-198

BI

BOOT, 2-9

device

SET BI_DEVICES, 2-131

SHOW BI_DEVICES, 2-199

node, Z, 2-277

protocol, Z, 2-277

Bitvector

BITVECTOR lexical, 3-3

data type, 1-12

FIELD lexical, 3-9

2 Index

Boolean operators, 1-11

Boot

BOOT, 2-9
and REBOOT, 2-104

primary

CPU, 1-4
SET CPU, 2-141

set, 1-4

SET

AUTOBOOT, 2-130
BOOTFLAGS, 2-132
BOOTSET, 2-133

SHOW

AUTOBOOT, 2-195
BOOTFLAGS, 2-200
BOOTSET, 2-201

Bus

I/O, table, 2-159
power, table, 2-158

C

Cabinet. *See* System, cabinet power

Cache threshold

SHOW THRESHOLD, 2-255

CALL, 2-11

CDB (configuration database), 2-44

LOAD, 2-88

CIO (console I/O), xi, 2-265

CLI (command language
interpreter), 1-9

Clock

CLOCK lexical, 3-4

cycle count

SET CYCLE, 2-142

SHOW CYCLE, 2-208

emulation, SET CLOCK, 2-136

frequency, SET CLOCK, 2-136

INITIALIZE, 2-81

interval

SET CLOCK, 2-136

SET CYCLE, 2-142

MCM

ALLOCATE, 2-2

Clock

MCM (cont'd.)

DEALLOCATE, 2-24

DEPOSIT, 2-45

EXAMINE, 2-65

register mnemonics, 2-45,
2-65

OFF, SET CLOCK, 2-137

ON, SET CLOCK, 2-137

position, SET CLOCK, 2-136

sample rate, SET CLOCK, 2-137

SCI, SET SCI, 2-166

SCU

MICROSTEP, 2-94

SET CLOCK, 2-137

SENSE CLOCK, 2-122

SET CLOCK, 2-135

SHOW

CLOCK, 2-202

CONFIGURATION, 2-203

synch, SET CLOCK, 2-137

CLOSE, 2-12

and OPEN, 2-99

Cold-start

flags, SHOW FLAGS, 2-216

SET COLD_START, 2-139

Command

language interpreter. *See* CLI

confirmation, 1-5

responses, 1-5

format, 1-1

object, 1-1

parameter, 1-1

qualifier, 1-1, 1-3

See also Qualifier

/ALL, 1-5

confirmation, 1-5

/CPU, 1-4

See also CPU, qualifier

entry, 1-3

/SCU, 1-4

See also SCU, qualifier

values, 1-4

quick reference, A-1

- Command (cont'd.)
 - RECALL, 2-105
 - REPEAT, 2-109
 - SET COMMAND, 2-140
 - syntax, 1-1
 - verb, 1-1
- Configuration
 - database. *See* CDB
 - CONFIG lexical, 3-5
 - SHOW CONFIGURATION, 2-203
- Confirmation
 - qualifier, 1-5
 - responses, 1-5
- Console, xi
 - I/O mode. *See* CIO
- CONTINUE, 2-14
- Control store
 - ECS
 - CREATE/WINDOW, 2-21
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
 - JCS
 - CREATE/WINDOW, 2-21
 - DEPOSIT, 2-44
 - EXAMINE, 2-64
 - microPC, UPC lexical, 3-34
 - symbol, LOAD, 2-90
- COPY, 2-16
- CPU
 - ALLOCATE, 2-2
 - available set, 1-4
 - SHOW AVAILABLE, 2-196
 - boot
 - primary, SET CPU, 2-141
 - set, 1-4
 - SET BOOTFLAGS, 2-132
 - SET BOOTSET, 2-133
 - SHOW BOOTSET, 2-201
 - bootprimary, 1-4
 - cpu-id, 1-4
 - CPU lexical, 3-6
- CPU (cont'd.)
 - DEALLOCATE, 2-24
 - default, SET CPU, 2-141
 - id, 1-4
 - primary, 1-4
 - SET BOOTSET, 2-133
 - SET CPU, 2-141
 - qualifier, 1-4
 - arguments, 1-4
 - CONTINUE, 2-14
 - CREATE/WINDOW, 2-21
 - DELETE/PATTERN, 2-34
 - DELETE/TRACE, 2-38
 - DELETE/WATCH, 2-40
 - DEPOSIT, 2-43
 - EXAMINE, 2-63
 - FIND, 2-74
 - HALT, 2-77
 - INITIALIZE, 2-81
 - LOAD, 2-89
 - MICROSTEP, 2-93
 - NEXT, 2-96
 - RESET, 2-110
 - RESTORE, 2-111
 - SAVE, 2-115
 - SET
 - CLOCK, 2-135
 - COLD_START, 2-139
 - CYCLE, 2-142
 - ISOLATION, 2-147
 - KEEP_ALIVE, 2-148
 - PATTERN, 2-154
 - REVISION, 2-164
 - SERIAL, 2-173
 - SNAPSHOT, 2-175
 - TERMINAL, 2-179
 - TRACE, 2-183
 - WARM_START, 2-188
 - WATCH, 2-189
 - SHOW
 - CONFIGURATION, 2-203
 - CYCLE, 2-209
 - FLAGS, 2-216

4 Index

CPU

qualifier

SHOW (cont'd.)

HISTORY, 2-217

ISOLATION, 2-219

PATTERN, 2-229

STRUCTURE, 2-248

TRACE, 2-257

WATCH, 2-261

START, 2-265

VERIFY, 2-273

SENSE CPU, 2-123

symbolic names, 1-4

threshold

SHOW THRESHOLD, 2-255

CREATE, 2-18

/DIRECTORY, 2-20

/WINDOW, 2-21

and DELETE/WINDOW,
2-42

Cycle, clock

SET CYCLE, 2-142

SHOW CYCLE, 2-208

D

D_FLOAT

DEPOSIT, 2-44

EXAMINE, 2-63

DAC (daughter array card), SET

SERIAL, 2-173

Data types, 1-12

bitvector, 1-12

BITVECTOR lexical, 3-3

integer, 1-12

string, 1-12

Date and time. *See* Time and date

DCL (Digital command language),

xi, 1-1

DEALLOCATE, 2-24

and ALLOCATE, 2-2

DEASSIGN, 2-25

and DEFINE, 2-28

and SHOW LOGICAL, 2-224

Default

CPU, SET CPU, 2-141

device

SET DEFAULT, 2-144

SHOW DEFAULT, 2-210

directory

SET DEFAULT, 2-144

SHOW DEFAULT, 2-210

disk

SET DEFAULT, 2-144

SHOW DEFAULT, 2-210

DEFINE, 2-28

and DEASSIGN, 2-25

and SHOW LOGICAL, 2-224

/KEY, 2-30

and SHOW KEY, 2-221

Defining keys, 1-7

DELETE, 2-32

/PATTERN, 2-34

and SET PATTERN, 2-154

and SHOW PATTERN,
2-229

/SYMBOL, 2-36

and = (assign symbol), 2-5

/TRACE, 2-38

and SET TRACE, 2-183

and SHOW TRACE, 2-257

/WATCH, 2-40

and SET WATCH, 2-189

and SHOW WATCH, 2-261

/WINDOW, 2-42

and CREATE/WINDOW,

2-21

DEPOSIT, 2-43

and EXAMINE, 2-62

Device

interrupt level, SHOW DEVICE,
2-211

LA100, SET TERMINAL, 2-179

SET DEFAULT, 2-144

SHOW DEFAULT, 2-210

status, SHOW DEVICE, 2-211

VT200, SET TERMINAL, 2-179

XMI

- Device
 - XMI (cont'd.)
 - SET XMI_DEVICES, 2-192
 - SHOW XMI_DEVICES, 2-263
- Digital command language. *See* DCL
- Directory
 - CREATE/DIRECTORY, 2-20
 - DIRECTORY, 2-53
 - SET DEFAULT, 2-144
 - SHOW DEFAULT, 2-210
- Disk
 - SET DEFAULT, 2-144
 - SHOW DEFAULT, 2-210
 - volume
 - DISMOUNT, 2-56
 - INITIALIZE, 2-83
 - MOUNT, 2-95
- DISMOUNT, 2-56
 - and MOUNT, 2-95

- E**
- EBox
 - control store. *See* ECS
 - register. *See* EREG
- ECS (EBox control store), 2-21
 - CREATE/WINDOW, 2-21
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
- EDIT, 2-57
- EMEMORY
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
- Emulation, clock, SET CLOCK, 2-136
- End_of_file qualifier, READ, 2-102
- Equals
 - See* := (assign string)
 - See* = (assign symbol)
- EREG (EBox register)
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
- Error
 - Error (cont'd.)
 - handling
 - SET ERROR_HANDLING, 2-145
 - SHOW ERROR_HANDLING, 2-214
 - EVALUATE, 2-60
 - Examine
 - EXAMINE, 2-62
 - and DEPOSIT, 2-43
 - window, CREATE/WINDOW, 2-21
 - Execute procedure (@), 2-7
 - EXIT, 2-73
 - Expressions, 1-10
 - address mnemonics
 - DEPOSIT, 2-50
 - EXAMINE, 2-70
 - arithmetic operators, 1-11
 - Boolean operators, 1-11
 - DEPOSIT, 2-49
 - EVALUATE, 2-61
 - EXAMINE, 2-69
 - lexical functions, 1-11
 - mixed, 1-10
 - numeric, 1-10
 - literals
 - DEPOSIT, 2-49
 - EXAMINE, 2-69
 - relational operators, 1-11
 - special operators
 - DEPOSIT, 2-50
 - EXAMINE, 2-70
 - string, 1-10
 - External memory address space
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
 - qualifier, 2-46, 2-47, 2-65, 2-67
 - EXTRACT lexical, 3-8

F

F_FLOAT

- DEPOSIT, 2-44
- EXAMINE, 2-63

Field

- See also* Message, fields
- FIELD lexical, 3-9

FILL lexical, 3-10

FIND, 2-74

Firmware

PEM

- LOAD, 2-88
- VERIFY, 2-273

RIC

- LOAD, 2-89
- VERIFY, 2-273

SCAN, INITIALIZE, 2-81

SCM, INITIALIZE, 2-81

Flags

- cold start, SHOW FLAGS, 2-216
- warm start, SHOW FLAGS, 2-216

Floating point

D_FLOAT

- DEPOSIT, 2-44
- EXAMINE, 2-63

F_FLOAT

- DEPOSIT, 2-44
- EXAMINE, 2-63

G_FLOAT

- DEPOSIT, 2-44
- EXAMINE, 2-63

Format

- command, 1-1
- message, 1-8
- SET MESSAGE, 2-152

Frequency, clock, SET CLOCK, 2-136

G

G_FLOAT

- DEPOSIT, 2-44
- EXAMINE, 2-63

General

- purpose register. *See* GPR
- DEPOSIT, 2-44
- EXAMINE, 2-63

Global

- DELETE/SYMBOL, 2-36
- SHOW SYMBOL, 2-251

GOTO, 2-76

GPR (general-purpose register), 2-50, 2-70

- DEPOSIT, 2-43
- EXAMINE, 2-63

register

- address, 2-50, 2-70
- mnemonics, 2-50, 2-70

H

HALT, 2-77

Help

- HELP, 2-78
- in-line help, 1-5

I

I/O

- ALLOCATE, 2-2
- bus table, 2-159
- DEALLOCATE, 2-24
- INITIALIZE, 2-82
- SENSE IO, 2-124
- SET POWER, 2-159
- SHOW CONFIGURATION, 2-204
- SHOW POWER, 2-233

IF, 2-80

- with expression THEN, 2-80

IMEMORY

- DEPOSIT, 2-44
- EXAMINE, 2-63
- INFORMATION lexical, 3-11
- INITIALIZE, 2-81
- In-line help, 1-5
- INQUIRE, 2-85
- Instruction, EXAMINE, 2-63
- Integer
 - conversion, STRING lexical, 3-31
 - data type, 1-12
 - INTEGER lexical, 3-13
- Internal
 - processor register. *See* IPR
 - memory address space
 - DEPOSIT, 2-44
 - EXAMINE, 2-63
 - qualifier, 2-46, 2-47, 2-65, 2-67
 - register
 - address space
 - DEPOSIT, 2-46
 - EXAMINE, 2-66
 - qualifier, 2-46, 2-47, 2-65, 2-67
 - DEPOSIT, 2-44
 - EXAMINE, 2-64
- Interrupt level, device, SHOW DEVICE, 2-211
- Interval, clock, SET CLOCK, 2-136
- IO. *See* I/O
- IPAMM (JBox I/O physical address memory map), 2-82
- IPR (internal processor register), 2-43, 2-50, 2-70
 - register
 - address, 2-50, 2-71
 - mnemonics, 2-50, 2-71

J

- JCS (JBox control store), 2-21
 - CREATE/WINDOW, 2-21
 - DEPOSIT, 2-44
 - EXAMINE, 2-64

K

- Kernel
 - INITIALIZE, 2-82
 - SHOW CONFIGURATION, 2-204
- Key
 - DEFINE/KEY, 2-30
 - definitions, 1-7
 - key name table, 1-8, 2-31, 2-222
 - predefined, 1-7
 - SHOW KEY, 2-221
 - special, 1-7
 - user-defined, 1-8
- Keypad mode, 2-31
 - SET TERMINAL, 2-180

L

- LA100, SET TERMINAL, 2-179
- Label
 - EXAMINE, 2-64
 - LABEL, 2-87
 - SET SCOPE, 2-171
- LENGTH lexical, 3-14
- Lexical functions, 1-11, 3-1
 - ASCII, 3-2
 - BITVECTOR, 3-3
 - CLOCK, 3-4
 - CONFIG, 3-5
 - CPU, 3-6
 - EXTRACT, 3-8
 - FIELD, 3-9
 - FILL, 3-10
 - INFORMATION, 3-11

Lexical functions (cont'd.)

INTEGER, 3-13
 LENGTH, 3-14
 LOCATE, 3-15
 LOGICAL, 3-16
 MCU, 3-17
 MEMORY, 3-18
 PARSE, 3-19
 PART, 3-21
 POWER, 3-22
 quick reference, B-1
 RADIX, 3-23
 REVISION, 3-24
 SCI, 3-25
 SCU, 3-26
 SEARCH, 3-27
 SERIAL, 3-28
 SID, 3-29
 SIGNAL, 3-30
 STRING, 3-31
 SWITCH, 3-32
 TIME, 3-33
 UPC, 3-34
 VERIFY, 3-35
 LOAD, 2-88
 Local
 DELETE/SYMBOL, 2-36
 SHOW SYMBOL, 2-251
 LOCATE lexical, 3-15
 Logical
 LOGICAL lexical, 3-16
 name
 DEASSIGN, 2-25
 DEFINE, 2-28
 SHOW LOGICAL, 2-224
 translation, LOGICAL
 lexical, 3-16
 LOGOUT, 2-91

M

Margin, power, SET POWER, 2-160
 Master clock module. *See* MCM
 MCM (master clock module)
 ALLOCATE, 2-2
 DEALLOCATE, 2-24
 DEPOSIT, 2-45
 EXAMINE, 2-65
 register mnemonics, 2-45, 2-65
 MCU (multichip unit)
 MCU lexical, 3-17
 qualifier
 SHOW
 CONFIGURATION, 2-204
 ISOLATION, 2-219
 Memory
 address space
 DEPOSIT, 2-44
 EXAMINE, 2-63
 qualifier, 2-46, 2-47, 2-65, 2-67
 ALLOCATE, 2-2
 bad block list, 2-82, 2-83
 DEALLOCATE, 2-24
 INITIALIZE, 2-82
 LOAD, 2-88
 MEMORY lexical, 3-18
 qualifier, FIND, 2-75
 SHOW
 CONFIGURATION, 2-204
 MEMORY, 2-225
 SPU
 DEPOSIT, 2-48
 EXAMINE, 2-68
 VERIFY, 2-273
 Message
 fields, 1-8
 format, 1-8
 SET MESSAGE, 2-152
 SHOW MESSAGE, 2-226
 Microcode. *See*:

Microcode. See: (cont'd.)

- Control store
- ECS
- Firmware
- JCS
- PEM
- Structure
- MICROSTEP, 2-93
 - and NEXT, 2-96
- Mixed expressions, 1-10
- Mode
 - console I/O. *See* CIO
 - program I/O mode. *See* PIO
 - space bar step mode. *See* SBSM
 - step. *See* SBSM
 - keypad, SET TERMINAL, 2-180
 - SHOW MODE, 2-227
- MOUNT, 2-95
 - and DISMOUNT, 2-56
- MPAMM (JBox memory physical address memory map), 2-82

N

- Name
 - SET PERSONAL_NAME, 2-157
 - SHOW PERSONAL, 2-231
- NEXT, 2-96
 - and MICROSTEP, 2-93
- Node
 - BOOT, 2-9
 - SHOW NODE, 2-228
- NPAMM (JBox nonexistent physical address memory map), 2-82
- Number radix, 1-9
- Numeric
 - expressions, 1-10
 - literal expressions
 - DEPOSIT, 2-49
 - EXAMINE, 2-69

O

- Object, command, 1-1
- Odometer, CREATE/WINDOW, 2-22
- ON, 2-98
 - with condition THEN, 2-98
- OPCOM (operator communications manager) qualifier, SEND, 2-120
- OPEN, 2-99
 - and CLOSE, 2-12
- Operating system, SET REVISION, 2-164
- Operator control panel switches, SHOW SWITCHES, 2-250
- OS. *See* Operating system

P

- Parameter, command, 1-1
- PARSE lexical, 3-19
- PART lexical, 3-21
- Pattern
 - DELETE
 - /PATTERN, 2-34
 - SET PATTERN, 2-154
- PCS (power control subsystem)
 - ALLOCATE, 2-2
 - DEALLOCATE, 2-24
- PEM (power and environmental monitor)
 - DEPOSIT, 2-45
 - EXAMINE, 2-65
 - firmware
 - LOAD, 2-88
 - VERIFY, 2-273
 - register mnemonics, 2-45, 2-65
- Physical address space
 - DEPOSIT, 2-46
 - EXAMINE, 2-66
- PIO (program I/O)
 - mode, 2-179

PIO (program I/O) (cont'd.)

mode qualifier

BOOT, 2-9
 CONTINUE, 2-14
 SET TERMINAL, 2-180
 START, 2-265

PIO_PORT qualifier

CONTINUE, 2-14
 START, 2-265

/PROGRAM

SET TERMINAL, 2-180

Planar

SET

REVISION, 2-164
 SERIAL, 2-173

PLL (phase-locked loop), 2-202

Port

id, SHOW USERS, 2-258

PIO_PORT qualifier

CONTINUE, 2-14
 START, 2-265

register address space

DEPOSIT, 2-46
 EXAMINE, 2-66
 qualifier, 2-46, 2-47, 2-65,
 2-67

Position, clock, SET CLOCK, 2-136

Power

and environmental monitor. *See*
 PEM

control subsystem. *See* PCS

bus

SET POWER, 2-158
 SHOW POWER, 2-232
 table, 2-158, 2-232

margin, SET POWER, 2-160

POWER lexical, 3-22

SENSE POWER, 2-125

subsystem

INITIALIZE, 2-82
 margin, SET POWER, 2-160
 SHOW CONFIGURATION,
 2-205

state

SET POWER, 2-158

Power

subsystem

state (cont'd.)

SHOW POWER, 2-232

system cabinet

SET POWER, 2-159
 SHOW POWER, 2-233
 table, 2-159, 2-233

Predefined key table, 1-7

Primary

CPU, 1-4

SET BOOTSET, 2-133

SET CPU, 2-141

Process

DEASSIGN, 2-25
 DEFINE, 2-28
 SHOW PROCESS, 2-235

Processor

status longword. *See* PSL
 INITIALIZE, 2-82

Program

I/O mode. *See* PIO

address space

DEPOSIT, 2-43
 EXAMINE, 2-62
 qualifier, 2-46, 2-47, 2-65,
 2-67

Protocol

BI, Z, 2-277
 XMI, Z, 2-277

PSL (processor status longword),
2-50, 2-70

register

address, 2-50, 2-70
 mnemonics, 2-50, 2-70

PURGE, 2-100

Q

Qualifier

/SCU. *See* SCU, qualifier

address space, 2-46, 2-47, 2-65,
 2-67

Qualifier**address space (cont'd.)**

/CODE, 2-46, 2-47, 2-65,
2-67

/EMEMORY, 2-46, 2-47,
2-65, 2-67

external memory, 2-46,
2-47, 2-65, 2-67

/IMEMORY, 2-46, 2-47,
2-65, 2-67

internal

memory, 2-46, 2-47,
2-65, 2-67

register, 2-46, 2-47,
2-65, 2-67

/PHYSICAL, 2-46, 2-66

port register, 2-46, 2-47,
2-65, 2-67

/PORT_REGISTER, 2-46,
2-47, 2-65, 2-67

program, 2-46, 2-47, 2-65,
2-67

/REGISTER, 2-46, 2-47,
2-65, 2-67

/VIRTUAL, 2-48, 2-68

/ALL, 1-5

command, 1-1

confirmation, 1-5

entry, 1-3

values, 1-4

Quick reference

command, A-1

lexical functions, B-1

R

R3 qualifier, BOOT, 2-9

R5 qualifier, BOOT, 2-9

Radix, 1-9

name string, RADIX lexical,
3-23

RADIX lexical, 3-23

SET RADIX, 2-162

SHOW RADIX, 2-236

READ, 2-102

READ (cont'd.)

and WRITE, 2-276

REBOOT, 2-104

and BOOT, 2-9

RECALL, 2-105

Register**address**

GPR, 2-50, 2-70

IPR, 2-50, 2-71

PSL, 2-50, 2-70

space qualifier, 2-46, 2-47,
2-65, 2-67

clock

frequency, SET CLOCK,
2-136

interval

SET CLOCK, 2-136

SET CYCLE, 2-142

position, SET CLOCK,
2-136

EREG

DEPOSIT, 2-44

EXAMINE, 2-63

GPR

DEPOSIT, 2-43

EXAMINE, 2-63

internal

DEPOSIT, 2-44, 2-46

EXAMINE, 2-64, 2-66

IPR

DEPOSIT, 2-43

mnemonics

GPR, 2-50, 2-70

IPR, 2-50, 2-71

MCM, 2-45, 2-65

PEM, 2-45, 2-65

PSL, 2-50, 2-70

RIC, 2-46, 2-66

SCC, 2-47, 2-67

SJA, 2-48, 2-68

port

DEPOSIT, 2-46

EXAMINE, 2-66

/R3, BOOT, 2-9

/R5, BOOT, 2-9

Register (cont'd.)

RIC

DEPOSIT, 2-46
EXAMINE, 2-66

SCU

DEPOSIT, 2-47
EXAMINE, 2-67

SJA

DEPOSIT, 2-48
EXAMINE, 2-68

vector

DEPOSIT, 2-48
EXAMINE, 2-68
LOAD, 2-89

Regulator intelligence card. *See* RIC

Relational operators, 1-11

RENAME, 2-107

REPEAT, 2-109

Reset

DELETE

/PATTERN, 2-35
/TRACE, 2-39
/WATCH, 2-41

RESET, 2-110

Responses, command confirmation,
1-5

Restart parameter block. *See* RPB

RESTORE, 2-111

and SAVE, 2-115

RETURN, 2-112

Revision

REVISION lexical, 3-24

SET REVISION, 2-164

string, SET REVISION, 2-164

RIC (regulator intelligence card)

DEPOSIT, 2-46

EXAMINE, 2-66

firmware

LOAD, 2-89

VERIFY, 2-273

register mnemonics, 2-46, 2-66

Ring

DEPOSIT, 2-47

EXAMINE, 2-67

Ring (cont'd.)

pattern, LOAD, 2-89

RPB (restart parameter block),
FIND, 2-75

RUN, 2-113

S

Sample rate, clock, SET CLOCK,
2-137

SAVE, 2-115

and RESTORE, 2-111

SBSM (space bar step mode), 2-23

MICROSTEP, 2-93

NEXT, 2-96

Scan

control

chip. *See* SCC

interconnect. *See* SCI

module. *See* SCM

latch, RESET, 2-110

ring

DEPOSIT, 2-47

EXAMINE, 2-67

pattern, LOAD, 2-89

subsystem, INITIALIZE, 2-83

SCC (scan control chip)

DEPOSIT, 2-47

EXAMINE, 2-67

register mnemonics, 2-47, 2-67

SCI (scan control interconnect),
2-165

port

CPU0, 3-25

CPU1, 3-25

CPU2, 3-25

CPU3, 3-25

MEM, 3-25

SCU, 3-25

SCI lexical, 3-25

SCM (scan control module), 2-169

firmware, INITIALIZE, 2-81

INITIALIZE, 2-83

qualifier

- SCM (scan control module)
 - qualifier (cont'd.)
 - SET SERIAL, 2-174
 - STOP, 2-267
- Screen
 - SCROLL, 2-116
 - segment keyword table, 2-22, 2-119
 - SET SCREEN, 2-172
- SCROLL, 2-116
- SCU (system control unit)
 - ALLOCATE, 2-2
 - clock
 - MICROSTEP, 2-94
 - SET CLOCK, 2-137
 - DEALLOCATE, 2-24
 - qualifier, 1-4
 - DELETE
 - /PATTERN, 2-35
 - /TRACE, 2-39
 - /WATCH, 2-41
 - DEPOSIT, 2-47
 - EXAMINE, 2-67
 - INITIALIZE, 2-83
 - LOAD, 2-90
 - MICROSTEP, 2-94
 - SET
 - CLOCK, 2-137
 - CYCLE, 2-142
 - PATTERN, 2-155
 - REVISION, 2-164
 - SERIAL, 2-174
 - SNAPSHOT, 2-175
 - TRACE, 2-185
 - WATCH, 2-190
 - SHOW
 - CONFIGURATION, 2-205
 - CYCLE, 2-209
 - HISTORY, 2-218
 - PATTERN, 2-229
 - STRUCTURE, 2-249
 - TRACE, 2-257
 - WATCH, 2-261
- SCU lexical, 3-26
- SCU (system control unit) (cont'd.)
 - SENSE SCU, 2-126
 - SHOW SCU, 2-244
 - SEARCH lexical, 3-27
 - SELECT, 2-118
 - SEND, 2-120
 - and TALK, 2-269
 - SENSE, 2-121
 - CLOCK, 2-122
 - CPU, 2-123
 - IO, 2-124
 - POWER, 2-125
 - SCU, 2-126
 - SYSTEM, 2-127
 - Serial number
 - SERIAL lexical, 3-28
 - SET SERIAL, 2-174
 - Service processor unit. *See* SPU
 - SET, 2-128
 - and SHOW, 2-193
 - ATTN_ACTION, 2-129
 - and SHOW ATTN_ACTION, 2-194
 - AUTOBOOT, 2-130
 - and SHOW AUTOBOOT, 2-195
 - BI_DEVICES, 2-131
 - and SHOW BI_DEVICES, 2-199
 - BOOTFLAGS, 2-132
 - and SHOW BOOTFLAGS, 2-200
 - BOOTSET, 2-133
 - and SHOW BOOTSET, 2-201
 - CLOCK, 2-135
 - and SHOW CLOCK, 2-202
 - COLD_START, 2-139
 - and SET WARM_START, 2-188
 - and SHOW FLAGS, 2-216
 - COMMAND, 2-140
 - CPU, 2-141
 - and SHOW CPU, 2-207
 - CYCLE, 2-142

SET

- CYCLE (cont'd.)
 - and SHOW CYCLE, 2-208
- DEFAULT, 2-144
 - and SHOW DEFAULT, 2-210
- ERROR_HANDLING, 2-145
 - and SHOW ERROR_HANDLING, 2-214
 - and SHOW THRESHOLD, 2-255
- FAULT_ACTION, 2-146
 - and SHOW FAULT_ACTION, 2-215
- ISOLATION, 2-147
 - and SHOW ISOLATION, 2-219
- KEEP_ALIVE, 2-148
 - and SHOW KEEP_ALIVE, 2-220
- LABELS, 2-149
- LOGGING, 2-150
 - and SHOW LOGGING, 2-223
- MESSAGE, 2-152
 - and SHOW MESSAGE, 2-226
- PATTERN, 2-154
 - and DELETE/PATTERN, 2-34
 - and SHOW PATTERN, 2-229
- PERSONAL_NAME, 2-157
 - and SHOW PERSONAL, 2-231
- POWER, 2-158
 - and SHOW POWER, 2-232
- PROMPT, 2-161
- RADIX, 2-162
 - and SHOW RADIX, 2-236
- REMOTE, 2-163
 - and SHOW REMOTE, 2-237
- SCI, 2-165
 - and SHOW SCI, 2-238
- SCM, 2-169

SET

- SCM (cont'd.)
 - and SHOW SCM, 2-239
- SCOPE, 2-171
 - and SHOW SCOPE, 2-243
- SCREEN, 2-172
- SERIAL, 2-173
- SNAPSHOT, 2-175
- SOURCE, 2-177
 - and SHOW SOURCE, 2-246
- STEP, 2-178
 - and SHOW STEP, 2-247
- TERMINAL, 2-179
 - and SHOW TERMINAL, 2-253
- TIME, 2-182
 - and SHOW TIME, 2-256
- TRACE, 2-183
 - and DELETE/TRACE, 2-38
 - and SHOW TRACE, 2-257
- VERIFY, 2-187
- WARM_START, 2-188
 - and SET COLD_START, 2-139
 - and SHOW FLAGS, 2-216
- WATCH, 2-189
 - and DELETE/WATCH, 2-40
 - and SHOW WATCH, 2-261
- XMI_DEVICES, 2-192
 - and SHOW XMI_DEVICES, 2-263
- SHOW, 2-193
 - and SET, 2-128
 - ATTN_ACTION, 2-194
 - and SET ATTN_ACTION, 2-129
 - AUTOBOOT, 2-195
 - and SET AUTOBOOT, 2-130
 - AVAILABLE, 2-196
 - BATCH, 2-197
 - BBU, 2-198
 - BI_DEVICES, 2-199
 - and SET BI_DEVICES, 2-131

SHOW (cont'd.)

- BOOTFLAGS, 2-200
 - and SET BOOTFLAGS, 2-132
- BOOTSET, 2-201
 - and SET BOOTSET, 2-133
- CLOCK, 2-202
 - and SET CLOCK, 2-135
- CONFIGURATION, 2-203
- CPU, 2-207
 - and SET CPU, 2-141
- CYCLE, 2-208
 - and SET CYCLE, 2-142
- DEFAULT, 2-210
 - and SET DEFAULT, 2-144
- DEVICE, 2-211
- ENVIRONMENT, 2-213
- ERROR_HANDLING, 2-214
 - and SET ERROR_HANDLING, 2-145
- FAULT_ACTION, 2-215
 - and SET FAULT_ACTION, 2-146
- FLAGS, 2-216
 - and SET COLD_START, 2-139
 - and SET WARM_START, 2-188
- HISTORY, 2-217
- ISOLATION, 2-219
- KEEP_ALIVE, 2-220
 - and SET KEEP_ALIVE, 2-148
- KEY, 2-221
 - and DEFINE/KEY, 2-30
- LOGGING, 2-223
 - and SET LOGGING, 2-150
- LOGICAL, 2-224
 - and DEASSIGN, 2-25
 - and DEFINE, 2-28
- MEMORY, 2-225
- MESSAGE, 2-226
 - and SET MESSAGE, 2-152
- MODE, 2-227
- NODE, 2-228

SHOW (cont'd.)

- PATTERN, 2-229
 - and DELETE/PATTERN, 2-34
 - and SET PATTERN, 2-154
- PERSONAL, 2-231
 - and SET PERSONAL_NAME, 2-157
- POWER, 2-232
 - and SET POWER, 2-158
- PROCESS, 2-235
- RADIX, 2-236
 - and SET RADIX, 2-162
- REMOTE, 2-237
 - and SET REMOTE, 2-163
- SCI, 2-238
 - and SET SCI, 2-165
- SCM, 2-239
 - and SET SCM, 2-169
- SCOPE, 2-243
 - and SET SCOPE, 2-171
- SCU, 2-244
- SJA, 2-245
- SOURCE, 2-246
 - and SET SOURCE, 2-177
- STEP, 2-247
 - and SET STEP, 2-178
- STRUCTURE, 2-248
- SWITCHES, 2-250
- SYMBOL, 2-251
- SYSTEM, 2-252
- TERMINAL, 2-253
 - and SET TERMINAL, 2-179
- THRESHOLD, 2-255
- TIME, 2-256
 - and SET TIME, 2-182
- TRACE, 2-257
 - and DELETE/TRACE, 2-38
 - and SET TRACE, 2-183
- USERS, 2-258
- VERSION, 2-259
- WATCH, 2-261
 - and DELETE/WATCH, 2-40
 - and SET WATCH, 2-189
- WINDOW, 2-262

SHOW (cont'd.)

XMI_DEVICES, 2-263
and SET XMI_DEVICES,
2-192

ZONE, 2-264

SID lexical, 3-29

SIGNAL lexical, 3-30

SJA

SHOW SJA, 2-245

SJA (SPU-to-JBox adapter)

DEPOSIT, 2-48

EXAMINE, 2-68

register mnemonics, 2-48, 2-68

SMP (symmetrical multiprocessing,
symmetrical multiprocessor),
1-4, 2-133, 2-196

Space bar step mode. *See* SBSM

Special

keys, 1-7

operators

DEPOSIT, 2-50

EXAMINE, 2-70

SPU (service processor unit), xi

to JBox adapter. *See* SJA

BI node, Z, 2-277

DEPOSIT, 2-48

EXAMINE, 2-68

REBOOT, 2-104

SHOW CONFIGURATION,
2-205

START, 2-265

Step

space bar step mode. *See* SBSM

SET STEP, 2-178

SHOW STEP, 2-247

STOP, 2-267

String

assign. *See* Assign, string

ASCII

DEPOSIT, 2-43

EXAMINE, 2-62

conversion, INTEGER lexical,
3-13

data type, 1-12

String (cont'd.)

date and time, TIME lexical,
3-33

expressions, 1-10

EXTRACT lexical, 3-8

length, LENGTH lexical, 3-14

offset, LOCATE lexical, 3-15

STRING lexical, 3-31

Structure

LOAD, 2-89

microcode, VERIFY, 2-273

SHOW

STRUCTURE, 2-248

SYMBOL, 2-251

Switches, operator control panel,
SHOW SWITCHES, 2-250

SWITCH lexical, 3-32

SYBIL

SET

ISOLATION, 2-147

PATTERN, 2-155

Symbol

assign

See also Assign, symbol

INQUIRE, 2-85

control store, LOAD, 2-90

CPU symbolic names, 1-4

EXAMINE, 2-68

global

DELETE/SYMBOL, 2-36

SHOW SYMBOL, 2-251

local

DELETE/SYMBOL, 2-36

SHOW SYMBOL, 2-251

Synch, clock, SET CLOCK, 2-137

Syntax, command, 1-1

System

control unit. *See* SCU

date. *See* Time and date

time. *See* Time and date

cabinet power

SET POWER, 2-159

SHOW POWER, 2-233

table, 2-159, 2-233

System (cont'd.)

- DEASSIGN, 2-25
- DEFINE, 2-28
- I/O bus table, 2-159
- ID, SID lexical, 3-29
- power bus
 - SET POWER, 2-158
 - SHOW POWER, 2-232
 - table, 2-158, 2-232
- SENSE SYSTEM, 2-127
- SHOW CONFIGURATION, 2-206

T

- TALK, 2-269
 - and SEND, 2-120
 - mode, SET TERMINAL, 2-180
- Tape volume
 - DISMOUNT, 2-56
 - INITIALIZE, 2-83
 - MOUNT, 2-95
- Terminal
 - SET TERMINAL, 2-179
 - SHOW TERMINAL, 2-253
- THEN
 - with IF expression, 2-80
 - with ON condition, 2-98
- Threshold
 - SHOW THRESHOLD, 2-255
 - cache, 2-255
 - CPU, 2-255
 - VBox, 2-255
- Time and date
 - SET TIME, 2-182
 - SHOW TIME, 2-256
 - TIME lexical, 3-33
- Tracepoint
 - DELETE/TRACE, 2-38
 - SET TRACE, 2-183
 - SHOW TRACE, 2-257
- TYPE, 2-270

U

- UIC (user identification code), 2-53
- UNJAM, 2-272
- UPC (utility port conditioner) lexical, 3-34
- User-defined keys, 1-8

V

- VBox threshold
 - SHOW THRESHOLD, 2-255
- Vector
 - DEPOSIT, 2-48
 - EXAMINE, 2-68
 - register, LOAD, 2-89
- Verb, command, 1-1
- VERIFY, 2-273
 - lexical, 3-35
- Virtual address space
 - DEPOSIT, 2-48
 - EXAMINE, 2-68
- VMS instruction
 - EXAMINE, 2-63
- Volume
 - DISMOUNT, 2-56
 - INITIALIZE, 2-83
 - MOUNT, 2-95
- VT200, SET TERMINAL, 2-179

W

- WAIT, 2-275
- Warm-start
 - flags, SHOW FLAGS, 2-216
 - SET WARM_START, 2-188
- Watchpoint
 - DELETE/WATCH, 2-40
 - SET WATCH, 2-189
 - SHOW WATCH, 2-261
- WHILE, 2-80
- Window

Window (cont'd.)

CREATE/WINDOW, 2-21
ECS, CREATE/WINDOW, 2-21
examine, CREATE/WINDOW,
2-21
JCS, CREATE/WINDOW, 2-21
keyword table, 2-22, 2-119
odometer, CREATE/WINDOW,
2-22
SCROLL, 2-116
SELECT, 2-118
SHOW WINDOW, 2-262
WRITE, 2-276
and READ, 2-102

X

XBI (XMI-to-BI adapter)
node, Z, 2-277
XJA (XMI-to-SCU adapter), 2-82
qualifier
UNJAM, 2-272
XMI
BOOT, 2-10
protocol, Z, 2-277
qualifier
SHOW CONFIGURATION,
2-206

Z

Z, 2-277