

VAX-11 FORTRAN Language Summary

$$\int \frac{dx}{a + bx^2} = \left\{ \begin{array}{l} \frac{1}{2\sqrt{-ab}} \log \frac{a + x\sqrt{-ab}}{a - x\sqrt{-ab}} \\ \text{or} \\ \frac{1}{\sqrt{-ab}} \tanh^{-1} \frac{x\sqrt{-ab}}{a} \end{array} \right.$$

digital
software



October 1982

VAX-11 FORTRAN

Language Summary

Order No. AV-M763A-TE

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.

Copyright © 1982 by Digital Equipment Corporation.
All Rights Reserved.

Printed in U.S.A.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	RSX
DEC/CMS	Edusystem	UNIBUS
DECnet	IAS	VAX
DECsystem-10	MASSBUS	VMS
DECSYSTEM-20	PDP	VT
DECUS	PDT	
DECwriter	RSTS	

ZK2190

Contents

1.0	VAX-11 FORTRAN COMMAND FORMATS	2
1.1	FORTRAN Command	2
1.2	LINK Command	3
1.3	RUN Command	4
2.0	VAX-11 FORTRAN STATEMENT SUMMARY	5
3.0	VAX-11 SYMBOLIC DEBUGGER COMMAND SUMMARY.	19
4.0	VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS	22
5.0	ASCII CHARACTER SET	38

Symbols and Conventions

- Brackets ([]) enclose optional language elements.
- Braces ({}) enclose lists of items from which one and only one item must be chosen.
- Horizontal ellipses (...) indicate that an item may be repeated one or more times. If the ellipses are preceded with a comma (,...), the items must be separated by commas.
- Text in blue ink describes language features that are VAX-11 extensions to the FORTRAN-77 standard.

Where to Find More Information

The following documents contain more detailed information:

- The *VAX-11 FORTRAN Language Reference Manual* contains reference information on the FORTRAN language elements summarized in this booklet.
- The *VAX-11 FORTRAN User's Guide* describes how to compile, link, debug, and execute programs written in the VAX-11 FORTRAN language using the facilities of the VAX/VMS operating system. It contains other information of interest to FORTRAN programmers, such as FORTRAN input/output error processing, programming efficiency, compatibility between VAX-11 FORTRAN and VAX-11 FORTRAN-66, and compatibility between VAX-11 FORTRAN and PDP-11 FORTRAN.

For a list of other related VAX/VMS documents, see the *VAX-11 Information Directory and Index*.

1.1 FORTRAN Command

FORTRAN [/qualifiers] file-spec-list [/qualifiers]

Qualifiers

/CHECK= { [NO]BOUNDS
[NO]OVERFLOW
[NO]UNDERFLOW
ALL
NONE }

/NOCHECK

Specifies whether the compiler generates code to perform run-time checks for the specified conditions. Default is /CHECK=(NOBOUNDS, OVERFLOW, NOUNDERFLOW).

/CONTINUATIONS = n

Specifies the maximum number of continuation lines allowed in a source statement. Default is n=19.

/[NO]CROSS__REFERENCE

Specifies whether a cross-reference listing is generated as part of the listing output. Default is /NOCROSS__REFERENCE.

/DEBUG= { [NO]SYMBOLS
[NO]TRACEBACK
ALL
NONE }

Specifies whether the compiler provides information for use by the VAX-11 Symbolic Debugger and run-time traceback mechanism. Default is /DEBUG=(NOSYMBOLS, TRACEBACK).

/[NO]D__LINES

Specifies whether the compiler reads and compiles lines that start with a D in column 1 in the source program. Default is /NOD__LINES.

/DML

Specifies that the DML preprocessor is to be invoked.

/[NO]F77

Specifies whether the FORTRAN-77 interpretation rules are used for those statements having a meaning incompatible with FORTRAN-66. Default is /F77.

/[NO]G__FLOATING

Specifies how the compiler interprets REAL*8, COMPLEX*16, DOUBLE PRECISION, and DOUBLE COMPLEX quantities. Default is /NOG__FLOATING.

/[NO]I4

Specifies how the compiler interprets INTEGER and LOGICAL declarations that do not specify a length. Default is /I4.

1.0 VAX-11 FORTRAN COMMAND FORMATS (Cont.)

/LIBRARY

Indicates that the input file is a text library.

/LIST[=file-spec]

/NOLIST

Specifies whether a source listing file is produced. Default is /NOLIST (interactive), /LIST (batch).

/[NO]MACHINE__CODE

Specifies whether the listing file includes a symbolic listing of the machine language code generated by the compiler. Default is /NOMACHINE__CODE.

/OBJECT[=file-spec]

/NOOBJECT

Specifies whether the compiler produces an object module. Default is /OBJECT.

/[NO]OPTIMIZE

Specifies whether the compiler optimizes the compiled program to generate more efficient code. Default is /OPTIMIZE.

/SHOW= { **[NO]INCLUDE**
 [NO]MAP
 [NO]PROCESSOR
 ALL
 NONE }

Specifies whether optionally listed source lines or the entire symbol map appear in the source listing. If /CROSS__REFERENCE is specified, the symbol map is always generated. The default is /SHOW=(NOPREPROCESSOR, NOINCLUDE, MAP).

/STANDARD= { **[NO]SYNTAX**
 [NO]SOURCE__FORM
 ALL
 NONE }

/NOSTANDARD

Specifies whether the compiler generates informational diagnostics for extensions to FORTRAN-77. Default is /NOSTANDARD.

/[NO]WARNINGS

Specifies whether the compiler generates I and W diagnostic messages in response to informational and warning-level errors. Default is /WARNINGS.

1.2 LINK Command

LINK **[/command-qualifiers]** **file-spec** **[/file-qualifiers]**...

Command-qualifiers

/[NO]EXECUTABLE[=file-spec]

Specifies whether the linker produces an executable image. Default is /EXECUTABLE.

/[NO]SHAREABLE[=file-spec]

Specifies whether the image generated by the linker has all its internal references resolved and must be linked with one or more object modules to produce an executable image. Default is /NOSHAREABLE.

/MAP[=file-spec]

Specifies whether a map file is generated. Default is /NOMAP (interactive), /MAP (batch).

/BRIEF

Specifies that the map file contain a summary of the image's characteristics and a list of contributing modules is to be produced.

/FULL

Specifies that the map file contain a summary of the image's characteristics, a list of contributing modules, a list of global symbols and values, and a summary of characteristic of image sections in the linked image.

/[NO]CROSS__REFERENCE

Specifies whether cross-reference information for global symbols is produced as part of the map file. Default is /NOCROSS__REFERENCE.

/[NO]DEBUG

Specifies whether the VAX-11 Symbolic Debugger is included in the executable image. Default is /NODEBUG.

/[NO]TRACEBACK

Specifies whether the linker includes traceback information in the image file. Default is /TRACEBACK.

File-qualifiers

/LIBRARY

Specifies that the input file is an object module or shareable image library that is searched to resolve undefined symbols referenced in other input modules.

/INCLUDE=module-name(s)

Specifies that the input file is an object module or shareable image library and only module names specified are explicitly included as input to the linker.

1.3 RUN Command

RUN **/[NO]DEBUG** file-spec

Qualifier

/[NO]DEBUG

Specifies whether the image is run with the VAX-11 Symbolic Debugger, even if /DEBUG was not specified in the previous compile or link commands. Default is /NODEBUG.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY

ACCEPT See **READ**

Arithmetic/Logical/Character Assignment

v = e

- v** is a variable name, an array element name, or a character substring name.
- e** is an expression.

ASSIGN s TO v

- s** is the label of a FORMAT statement or an executable statement.
- v** is an integer variable name.

BACKSPACE ([UNIT=**u**], IOSTAT=**ios**[[, ERR=**s**]) BACKSPACE **u**

- u** is a logical unit specifier.
- ios** is an I/O status specifier.
- s** is the label of an executable statement.

BLOCK DATA [**nam**]

- nam** is a symbolic name.

CALL f([**a**],[**a**][...])

- f** is a subprogram name or entry point.
- a** is an expression, an array name, a procedure name, or an alternate return specifier. An alternate return specifier is *s or &s, where s is the label of an executable statement.

CLOSE ([UNIT=**u**],**p**[[, IOSTAT=**ios**[[, ERR=**s**])

- p** is one of the following parameters:

STATUS	'SAVE'
DISPOSE	= 'KEEP'
DISP	'DELETE'
	'PRINT'
	'SUBMIT'
	'PRINT/DELETE'
	'SUBMIT/DELETE'

- u** is a logical unit specifier.
- s** is the label of an executable statement.
- ios** is an I/O status specifier.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

COMMON [/cb/] nlist [[,]/[cb]/nlist]...

- cb is a common block name.
- nlist is a list of one or more variable names, array names, or array declarators separated by commas.

CONTINUE

DATA nlist/clist[[,] nlist/clist]...

- nlist is a list of one or more variable names, array names, array element names, character substring names, or implied-DO lists, separated by commas. Subscript expressions and substring expressions must be constant.
- clist is a list of one or more constants separated by commas, each optionally preceded by j*, where j is a non-zero, unsigned integer constant.

DECODE (c,f,b[,IOSTAT=ios[,ERR=s)][list]

- c is an integer expression.
- f is a format specifier.
- b is a variable name, array name, array element name, or character substring name.
- ios is an I/O status specifier.
- s is a label of an executable statement.
- list is an I/O list.

DEFINE FILE u(m,n,U,v)[,u(m,n,U,v)]...

- u is a logical unit specifier.
- m is a constant or variable.
- n is a constant or variable.
- U specifies unformatted.
- v is an integer variable name.

DELETE ([UNIT=u[,REC=r[,IOSTAT=ios[,ERR=s])

DELETE (u'r[,IOSTAT=ios[,ERR=s])

- u is a logical unit specifier.
- r is a record specifier.
- ios is an I/O status specifier.
- s is the label of an executable statement.

DIMENSION a(d)[,a(d)]...

- a(d) is an array declarator.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

DO [s|,|| v=e1,e2|,e3]

- s is the label of an executable statement.
- v is a variable name.
- e1,e2,e3 are numeric expressions.

DO [s|,|| WHILE (e)

- s is a statement label.
- e is a logical expression.

ELSE

ELSE IF (e) THEN

- e is a logical expression.

ENCODE (c,f,b|,IOSTAT=ios||,ERR=s|) [list]

- c is an integer expression.
- f is a format specifier.
- b is a variable name, array name, array element name, or substring name.
- ios is an I/O status specifier.
- s is a label of an executable statement.
- list is an I/O list.

END

END DO

ENDFILE ([UNIT=|u|,IOSTAT=ios||,ERR=s|)

ENDFILE u

- u is a logical unit specifier.
- ios is an I/O status specifier.
- s is the label of an executable statement.

END IF

ENTRY nam|(|p|,p|...|)|

- nam is a subprogram name.
- p is a symbolic name or an alternate return specifier (*).

EQUIVALENCE (nlist)|,(nlist)|...

- nlist is a list of two or more variable names, array names, array element names, or character substring names separated by commas. Subscript expressions and substring expressions must be compile-time constant expressions.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

EXTERNAL I[,I]...

I is a subprogram name.

FIND ([UNIT=*u*,REC=*r*,IOSTAT=*ios*][ERR=*s*])

FIND (*u* ,*r*,IOSTAT=*ios*][ERR=*s*])

u is a logical unit specifier.

r is a direct access record number.

ios is an I/O status specifier.

s is the label of an executable statement.

FORMAT (code,...)

Code	Form	Effect
A	A[w]	Transfers alphanumeric
BN	BN	Blanks are ignored
BZ	BZ	Blanks are zeros
D	Dw.d	Transfers real values
E	Ew.d[Ee]	Transfers real values
F	Fw.d	Transfers real values
G	Gw.d[Ee]	Transfers real values
H	nHc...c	Transmits characters
I	Iw[.m]	Transfers decimal values
L	Lw	Transfers logical data
O	Ow[.m]	Transfers octal values
Q	Q	Obtain record size
S	S	Reinvokes optional +
SP	SP	Invokes mandatory +
SS	SS	Suppresses optional +
T	Tn	Specifies positional tabulation
TL	TLn	Specifies relative tabulation (←)
TR	TRn	Specifies relative tabulation (→)
X	nX	n characters skipped
Z	Zw[.m]	Transfers hexadecimal values
\$	\$	Suppresses carriage return
:	:	Terminates format control

Default Field Descriptor Values

Field Descriptor	List Element	w	d	e
I, O, Z	BYTE	7		
I, O, Z	INTEGER*2, LOGICAL*2	7		
I, O, Z	INTEGER*4, LOGICAL*4	12		
O, Z	REAL*4	12		
O, Z	REAL*8	23		
O, Z	REAL*16	44		
L	LOGICAL	2		
F, E, G, D	REAL, COMPLEX*8	15	7	2
F, E, G, D	REAL*8, COMPLEX*16	25	16	2
F, E, G, D	REAL*16	42	33	3
A	LOGICAL*1	1		
A	LOGICAL*2, INTEGER*2	2		
A	LOGICAL*4, INTEGER*4	4		
A	REAL*4, COMPLEX*8	4		
A	REAL*8, COMPLEX*16	8		
A	REAL*16	16		
A	CHARACTER*n	n		

Effect of Data Magnitude on G Format Conversions

Data Magnitude	Effective Conversion
$m < 0.1$	Ew.d[Ee]
$0.1 \leq m < 1.0$	F(w-4).d, n(' ')
$1.0 \leq m < 10.0$	F(w-4).(d-1), n(' ')
.	.
.	.
.	.
$10^{**}d-2 \leq m < 10^{**}d-1$	F(w-4).1, n(' ')
$10^{**}d-1 \leq m < 10^{**}d$	F(w-4).0, n(' ')
$m \geq 10^{**}d$	Ew.d[Ee]

Carriage Control

Character	Meaning
'+'	Overprinting: starts output at the beginning of the current line and returns to the left margin after printing
'Δ'	Single spacing: starts output at the beginning of the next line
'0'	Double spacing: skips a line before starting output
'1'	Paging: starts output at the top of a new page
'\$'	Prompting: starts output at the beginning of the next line, and suppresses carriage return at the end of the line
ASCII NUL	Overprinting with no advance: starts output at the beginning of the current line and does not return to the left margin after printing

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

[typ] FUNCTION nam[*n]([p],p[...])

typ is a data type specifier.

nam is a symbolic name.

*n is a data type length specifier.

p is a symbolic name.

GO TO s

s is a label of an executable statement.

GO TO (slist), e

slist is a list of one or more statement labels separated by commas.

e is an integer expression.

GO TO v|,|(slist)

v is an integer variable name.

slist is a list of one or more statement labels separated by commas.

IF (e) s1,s2,s3

e is an expression.

s1,s2,s3 are labels of executable statements.

IF (e) st

e is an expression.

st is any executable statement except a DO, END DO, END, block IF, or logical IF.

IF (e1) THEN

block

ELSE IF (e2) THEN

block

ELSE

block

END IF

e1,e2 are logical expressions.

block is a series of zero or more FORTRAN statements.

IMPLICIT typ (a[,a,...]),typ(a[,a,...])...

IMPLICIT NONE

typ is a data type specifier.

a is either a single letter, or two letters in alphabetical order separated by a hyphen (that is, X-Y).

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

INCLUDE 'file specification[/[NO]LIST]'
INCLUDE 'file specification'(module-name)/[NO]LIST'

- file specification
is a character constant that specifies the file to be included.
- module-name
is the name of a text module located in a text library.
- /[NO]LIST
indicates that the statements in the specified file are to be in the source listing.

INQUIRE (par[,par]...)

- par
is a keyword specification having the form:
key = value
- key
is a keyword as described below.
- value
depends on the keyword.

Keyword	Values
Inputs	
FILE	fin
UNIT	e
DEFAULTFILE	fin
Outputs	
ACCESS	cv
BLANK	cv
CARRIAGECONTROL	cv
DIRECT	cv
ERR	s
EXIST	lv
FORM	cv
FORMATTED	cv
IOSTAT	v
KEYED	cv
NAME	cv
NAMED	lv
NEXTREC	v
NUMBER	v
OPENED	lv
ORGANIZATION	cv
RECL	v
RECORDTYPE	cv
SEQUENTIAL	cv
UNFORMATTED	cv

- e
is a numeric expression.
- fin
is a character expression.
- v
is an integer variable or integer array element.
- lv
is a logical variable or array element.
- cv
is a character variable, array element, or substring reference.
- s
is the label of an executable statement.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

INTRINSIC v[,v]...

v is an intrinsic function name.

NAMelist /group-name/ namelist[,] /group-name/ namelist]...

group-name is a symbolic name.

namelist is a list of one or more variables or array names.

OPEN (par[,par]...)

par is a keyword specification in one of the following forms:

key
key = value

key is a keyword, as described below.

value depends on the keyword.

Keyword	Values
ACCESS	'SEQUENTIAL' 'DIRECT' 'KEYED' 'APPEND'
ASSOCIATEVARIABLE	v
BLOCKSIZE	e
BLANK	'NULL' 'ZERO'
BUFFERCOUNT	e
CARRIAGECONTROL	'FORTRAN' 'LIST' 'NONE'
• DEFAULTFILE	c
DISP	(same as DISPOSE
DISPOSE	'KEEP' or 'SAVE' 'PRINT' 'DELETE' 'SUBMIT' 'SUBMIT/DELETE' 'PRINT/DELETE'
ERR	s
EXTENDSIZE	e
FILE	c
FORM	'FORMATTED' 'UNFORMATTED'
INITIALSIZE	e
IOSTAT	v
KEY	keyspec
MAXREC	e
NAME	(same as FILE)
NOSPANBLOCKS	—
ORGANIZATION	'SEQUENTIAL' 'RELATIVE' 'INDEXED'
READONLY	—
RECL	e
RECORDSIZE	(same as RECL)

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

OPEN (par[,par]...) (Cont.)

RECORDTYPE	'FIXED' 'VARIABLE' 'SEGMENTED'
SHARED STATUS	— 'OLD' 'NEW' 'SCRATCH' 'UNKNOWN'
TYPE	(same as STATUS)
UNIT	e
USEROPEN	p

Keyword	Values
---------	--------

c	is a character expression, numeric array name , numeric variable name , numeric array element name , or Hollerith constant .
e	is a numeric expression.
p	is a program unit name.
s	is a statement label.
v	is an integer variable name.
keyspec	is (e1:e2[:type]) where: e1 is the beginning byte of the key field. e2 is the ending byte of the key field. type is either INTEGER or CHARACTER.

OPTIONS qualifier [,qualifier...]

qualifier	is one of the following: /[NO]G__FLOATING /[NO]I4 /[NO]F77 /[NO]CHECK /CHECK= { ALL [NO]OVERFLOW [NO]BOUNDS [NO]UNDERFLOW NONE }
-----------	---

PARAMETER (p=c[,p=c]...)

p	is a symbolic name.
c	is a constant or compile-time constant expression.

PAUSE [disp]

disp	is a decimal digit string containing 1 to 5 digits or a character constant.
------	---

PRINT

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

PROGRAM nam

nam A symbolic name.

READ ([UNIT=]u,[FMT=]f[,IOSTAT=ios][,END=s][,ERR=s]) [list]
READ f[,list]

ACCEPT f[,list]

u is a logical unit specifier.
f is a format specifier.
ios is an I/O status specifier.
s is a label of an executable statement.
list is an I/O list.

READ ([UNIT=]u,[FMT=]*[,IOSTAT=ios][,END=s][,ERR=s]) [list]
READ *[,list]

ACCEPT *[,list]

u is a logical unit specifier.
* denotes list-directed formatting.
ios is an I/O status specifier.
s is a label of an executable statement.
list is an I/O list.

READ ([UNIT=]u[,NML=]nl[,IOSTAT=ios][,END=s][,ERR=s])

READ nl

ACCEPT nl

u is a logical unit specifier.
nl is a namelist group-name.
ios is an I/O status specifier.
s is a label of an executable statement.

READ ([UNIT=]u[,IOSTAT=ios][,END=s][,ERR=s]) [list]

u is a logical unit specifier.
ios is an I/O status specifier.
s is a label of an executable statement.
list is an I/O list.

READ ([UNIT=]u,[FMT=]f,REC=r[,IOSTAT=ios][,ERR=s]) [list]

READ (u' r,[FMT=]f[,IOSTAT=ios][,ERR=s]) [list]

u is a logical unit specifier.
r is a record specifier.
f is a format specifier.
ios is an I/O status specifier.
s is a label of an executable statement.
list is an I/O list.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

READ ((UNIT=*u*,REC=*r*,IOSTAT=*ios*||,ERR=*s*) |list)

READ (*u* *r*,IOSTAT=*ios*||,ERR=*s*) |list|

u is a logical unit specifier.

r is a record specifier.

ios is an I/O status specifier.

s is a label of an executable statement.

list is an I/O list.

READ ((UNIT=*u*,[FMT=*f*,keyspec[,KEYID=*kn*||,IOSTAT=*ios*]
[,ERR=*s*]) |list|

READ ((UNIT=*u*,keyspec[,KEYID=*kn*||,IOSTAT=*ios*]
[,ERR=*s*]) |list|

u is a logical unit specifier.

f is a format specifier.

keyspec is a key specifier (see Section 7.2.1.5).

kn is a key-of-reference specifier.

ios is an I/O status specifier.

s is the label of an executable statement.

list is an I/O list.

READ ((UNIT=*c*,[FMT=*f*,IOSTAT=*ios*||,ERR=*s*||,END=*s*]) |list|

c is an internal file specifier.

f is a format specifier.

ios is an I/O status specifier.

s is the label of an executable statement.

list is an I/O list.

RETURN [*i*]

i is an integer value that indicates which alternate return is to be taken.

REWIND ((UNIT=*u*,IOSTAT=*ios*||,ERR=*s*)

REWIND *u*

u is a logical unit specifier.

ios is an I/O status specifier.

s is the label of an executable statement.

REWRITE ((UNIT=*u*,[FMT=*f*,IOSTAT=*ios*||,ERR=*s*]) |list|

REWRITE ((UNIT=*u*,IOSTAT=*ios*||,ERR=*s*) |list|

u is a logical unit specifier.

f is a format specifier.

ios is an I/O status specifier.

s is the label of an executable statement.

list is an I/O list.

SAVE [a[,a]...]

a is the name of a variable, an array, or a named common block enclosed in slashes.

Statement Function

$f([p],p[\dots]) = e$

f is a symbolic name.

p is a symbolic name.

e is an expression.

STOP [disp]

disp is a decimal digit string containing 1 to 5 digits or a character constant.

SUBROUTINE nam([p],p[\dots])

nam is a symbolic name.

p is a symbolic name or an alternate return specifier (*).

TYPE See WRITE

Type Declaration

type v[/clist/],v[/clist/]...

type is one of the following data type specifiers:

BYTE
LOGICAL
LOGICAL*1
LOGICAL*2
LOGICAL*4
INTEGER
INTEGER*2
INTEGER*4
REAL
REAL*4
REAL*8
REAL*16
DOUBLE PRECISION
COMPLEX
COMPLEX*8
COMPLEX*16
DOUBLE COMPLEX
CHARACTER*len
CHARACTER*(*)

v is a variable name, array name, function or function entry name, or an array declarator. The name can optionally be followed by a data type length specifier (*n). For character entities, the length specifier can be *len or *(*).

clist is an initial value or values to be assigned to the immediately preceding variable or array element.

UNLOCK ([UNIT=*u*], IOSTAT=*ios*], ERR=*s*)**UNLOCK *u***

- u* is a logical unit specifier.
- ios* is an I/O status specifier.
- s* is the label of an executable statement.

VIRTUAL *a(d)*, *a(d)*...

- a(d)* is an array declarator.

WRITE ([UNIT=*u*], FMT=*f*], IOSTAT=*ios*], ERR=*s*) [*list*]**PRINT *f*], *list*]****TYPE *f*], *list*]**

- u* is a logical unit specifier.
- f* is a format specifier.
- ios* is an I/O status specifier.
- s* is a label of an executable statement.
- list* is an I/O list.

WRITE ([UNIT=*u*], FMT=*], IOSTAT=*ios*], ERR=*s*) [*list*]****PRINT ***], *list*]****TYPE ***], *list*]**

- u* is a logical unit specifier.
- ** denotes list-directed formatting.
- ios* is an I/O status specifier.
- s* is a label of an executable statement.
- list* is an I/O list.

WRITE ([UNIT=*u*], NML=*nl*], IOSTAT=*ios*], ERR=*s*)**PRINT *nl*****TYPE *nl***

- u* is a logical unit specifier.
- nl* is a namelist group-name.
- ios* is an I/O status specifier.
- s* is a label of an executable statement.

WRITE ([UNIT=*u*], IOSTAT=*ios*], ERR=*s*) [*list*]

- u* is a logical unit specifier.
- s* is a label of an executable statement label.
- ios* is an I/O status specifier.
- list* is an I/O list.

2.0 VAX-11 FORTRAN STATEMENT SUMMARY (Cont.)

WRITE ([UNIT=]u,REC=r,[FMT=]f,[IOSTAT=ios],[ERR=s]) [list]

WRITE (u ,r,f,[ERR=s]) [list]

- u is a logical unit specifier.
- r is a record specifier.
- f is a format specifier.
- ios is an I/O status specifier.
- s is a label of an executable statement.
- list is an I/O list.

WRITE ([UNIT=]u,REC=r,[IOSTAT=ios],[ERR=s]) [list]

WRITE (u ,r,[IOSTAT=ios],[ERR=s]) [list]

- u is a logical unit specifier.
- r is a record specifier.
- ios is an I/O status specifier.
- s is a label of an executable statement label.
- list is an I/O list.

WRITE ([UNIT=]c,[FMT=]f,[IOSTAT=ios],[ERR=s]) [list]

- c is an internal file specifier.
- f is a format specifier.
- s is the label of an executable statement.
- ios is an I/O status specifier.
- list is an I/O list.

3.0 VAX-11 SYMBOLIC DEBUGGER COMMAND SUMMARY

CALL NAME [(argument-list)]

CANCEL ALL

CANCEL BREAK [/qualifier] [address-expression]
/ALL

CANCEL EXCEPTION BREAK

CANCEL MODE

CANCEL MODULE [/qualifier] module [,module...]
/ALL

CANCEL SCOPE

CANCEL SOURCE

CANCEL TRACE [/qualifier] [address-expression]
/ALL
/BRANCH
/CALL

CANCEL TYPE OVERRIDE

CANCEL WATCH [/qualifier] [address-expression]
/ALL

<CTRL/C>

<CTRL/Y>

<CTRL/Z>

DEFINE symbol=expression [,symbol=expression...]

DEPOSIT [/qualifier]...address-expression = data
[,data ...]

/ASCII:length

/BYTE

/DECIMAL

/D__FLOAT

/FLOAT

/G__FLOAT

/HEXADECIMAL

/H__FLOAT

/INSTRUCTION

/LONG

/OCTAL

/OCTAWORD

/QUADWORD

/WORD

EVALUATE[/qualifier]...expression [,expression...]
/ADDRESS

3.0 VAX-11 SYMBOLIC DEBUGGER COMMAND SUMMARY (Cont.)

EXAMINE [/qualifier]...addr-expr[:addr-expr]

[,addr-expr[:addr-expr]...]

/ASCII:length

/BYTE

/DECIMAL

/D_FLOAT

/FLOAT

/G_FLOAT

/HEXADECIMAL

/H_FLOAT

/INSTRUCTION

/LONG

/NOSYMBOLIC

/OCTAL

/OCTAWORD

/QUADWORD

/SYMBOLIC

/WORD

EXIT

@file-spec

GO [address-expression]

HELP topic [subtopic ...]

SEARCH[/qualifier [/qualifier]] range string

/ALL

/NEXT

/IDENTIFIER

/STRING

SET keyword [/qualifier] parameter

SET **BREAK**[/qualifier] address-expression

[**DO** (cmd[:cmd...])]

/AFTER

SET **EXCEPTION** **BREAK**

SET **LANGUAGE** language-name

SET **LOG**

SET **MARGIN** $\left\{ \begin{array}{l} \text{rm} \\ \text{lm :rm} \\ \text{lm :} \\ \text{:rm} \end{array} \right\}$

SET **MAX_SOURCE_FILES**

SET **MODE** modekeyword [,modekeyword...]

SET **MODULE** [/qualifier] [module-name [module-name] ...]

/ALL

3.0 VAX-11 SYMBOLIC DEBUGGER COMMAND SUMMARY (Cont.)

SET OUTPUT parameter[,parameter...]

SET SCOPE scope [,scope ...]

SET SEARCH parameter[,parameter]

SET SOURCE [/MODULE=modname] dirname[,dirname...]

SET STEP parameter[,parameter...]

SET TRACE [/qualifier] [address-expression]
 /BRANCH
 /CALL

SET TYPE [/qualifier] [address-expression]
 OVERRIDE

SET WATCH address-expression

SHOW BREAK

SHOW CALLS [integer]

SHOW LANGUAGE

SHOW LOG

SHOW MARGIN

SHOW MAX__SOURCE__FILES

SHOW MODE

SHOW MODULE

SHOW OUTPUT

SHOW SCOPE

SHOW SEARCH

SHOW SOURCE

SHOW STEP

SHOW TRACE

SHOW TYPE

SHOW WATCH

STEP [/qualifier] [integer]
 /INSTRUCTION
 /LINE
 /INTO
 /OVER
 /NOSOURCE
 /NOSYSTEM
 /SOURCE
 /SYSTEM

TYPE [[modname\] line-number[:line-number]]

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Square Root ¹ $a^{1/2}$	1	SQRT	SQRT DSQRT QSQRT CSQRT CDSQRT	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16
Natural Logarithm ² $\log_e a$	1	LOG	ALOG DLOG QLOG CLOG CDLOG	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16
Common Logarithm ² $\log_{10} a$	1	LOG10	ALOG10 DLOG10 QLOG10	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Exponential e^a	1	EXP	EXP DEXP QEXP CEXP CDEXP	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16

Sine ³ Sin a	1	SIN	SIN DSIN Q ¹⁶ SIN CSIN C ¹⁶ SIN	REAL*4 REAL*8 REAL* ¹⁶ COMPLEX*8 COMPLEX* ¹⁶
Sine ³ (degree) Sin a	1	SIND	SIND DSIND Q ¹⁶ SIND	REAL*4 REAL*8 REAL* ¹⁶
Cosine ³ Cos a	1	COS	COS DCOS Q ¹⁶ COS CCOS C ¹⁶ COS	REAL*4 REAL*8 REAL* ¹⁶ COMPLEX*8 COMPLEX* ¹⁶
Cosine ³ (degree) Cos a	1	COSD	COSD DCOSD Q ¹⁶ COSD	REAL*4 REAL*8 REAL* ¹⁶
Tangent ³ Tan a	1	TAN	TAN DTAN Q ¹⁶ TAN	REAL*4 REAL*8 REAL* ¹⁶
Tangent ³ (degree) Tan a	1	TAND	TAND DTAND Q ¹⁶ TAND	REAL*4 REAL*8 REAL* ¹⁶

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Arc Sine ^{4,5} Arc Sin a	1	ASIN	ASIN DASIN QASIN	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Arc Sine (degree) Arc Sin a	1	ASIND	ASIND DASIND QASIND	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Arc Cosine ^{4,5} Arc Cos a	1	ACOS	ACOS DACOS QACOS	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Arc Cosine (degree) Arc Cos a	1	ACOSD	ACOSD DACOSD QACOSD	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Arc Tangent ⁵ Arc Tan a	1	ATAN	ATAN DATAN QATAN	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16
Arc Tangent ^{5,7} (degree) Arc Tan a	1	ATAND	ATAND DATAND QATAND	REAL*4 REAL*8 REAL*16	REAL*4 REAL*8 REAL*16

Arc Tangent ^{5,6} Arc Tan a_1/a_2	2	ATAN2	ATAN2 DATAN2 QATAN2	REAL*4 REAL*8 REAL*16
Arc Tangent ^{5,7} (degree) Arc Tan a_1/a_2	2	ATAN2D	ATAN2D DATAN2D QATAN2D	REAL*4 REAL*8 REAL*16
Hyperbolic Sine Sinh a	1	SINH	SINH DSINH QSINH	REAL*4 REAL*8 REAL*16
Hyperbolic Cosine Cosh a	1	COSH	COSH DCOSH QCOSH	REAL*4 REAL*8 REAL*16
Hyperbolic Tangent Tanh a	1	TANH	TANH DTANH QTANH	REAL*4 REAL*8 REAL*16
Absolute Value ⁸ a	1	ABS	IIABS JIABS ABS DABS QABS CABS CDABS	INTEGER*2 INTEGER*4 REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16
		IABS	IIABS JIABS	INTEGER*2 INTEGER*4

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Truncation ^{9,12} a	1	INT	IINT	REAL*4	INTEGER*2
			JINT	REAL*4	INTEGER*4
			IIDINT	REAL*8	INTEGER*2
			JIDINT	REAL*8	INTEGER*4
			IIQINT	REAL*16	INTEGER*2
			JIQINT	REAL*16	INTEGER*4
			—	COMPLEX*8	INTEGER*2
			—	COMPLEX*8	INTEGER*4
			—	COMPLEX*16	INTEGER*2
			—	COMPLEX*16	INTEGER*4
			IDINT	REAL*8	INTEGER*2
			JIDINT	REAL*8	INTEGER*4
			IQINT	REAL*16	INTEGER*2
			JIQINT	REAL*16	INTEGER*4
		AINT	AIN	REAL*4	REAL*4
			DINT	REAL*8	REAL*8
			QINT	REAL*16	REAL*16

Nearest Integer^{9,12}
[a+.5*sign(a)]

1

ININT

ININT
JNINT
IINNT
JIDNNT
IINNT
JINNT

REAL*4
REAL*4
REAL*8
REAL*8
REAL*16
REAL*16

INTEGER*2
INTEGER*4
INTEGER*2
INTEGER*4
INTEGER*2
INTEGER*4

IDNINT

IINNT
JIDNNT

REAL*8
REAL*8

INTEGER*2
INTEGER*4

IQNINT

IINNT
JINNT

REAL*16
REAL*16

INTEGER*2
INTEGER*4

ANINT

ANINT
DNINT
QNINT

REAL*4
REAL*8
REAL*16

REAL*4
REAL*8
REAL*16

Zero-Extend Functions

1

ZEXT

IZEXT
—
—
JZEXT
—
—
—
—

LOGICAL*1
LOGICAL*2
INTEGER*2
LOGICAL*1
LOGICAL*2
LOGICAL*4
INTEGER*2
INTEGER*4

INTEGER*2

INTEGER*4

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Conversion to ¹⁰ REAL*4	1	REAL	FLOATI	INTEGER*2	REAL*4
			FLOATJ	INTEGER*4	REAL*4
			—	REAL*4	REAL*4
			SNGL	REAL*8	REAL*4
			SNGLQ	REAL*16	REAL*4
			—	COMPLEX*8	REAL*4
			—	COMPLEX*16	REAL*4
Conversion to ¹⁰ REAL*8	1	DBLE	—	INTEGER*2	REAL*8
			—	INTEGER*4	REAL*8
			DBLE	REAL*4	REAL*8
			DBLEQ	REAL*8	REAL*8
			—	REAL*16	REAL*8
			—	COMPLEX*8	REAL*8
			—	COMPLEX*16	REAL*8

Conversion to REAL*16	1	QEXT	—	INTEGER*2 INTEGER*4 REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	REAL*16 REAL*16 REAL*16 REAL*16 REAL*16 REAL*16 REAL*16
Fix ^{10,12} (REAL*4-to-integer conversion)	1	IFIX	IIFIX JIFIX	REAL*4 REAL*4	INTEGER*2 INTEGER*4
Float ¹⁰ (Integer-to-REAL*4 conversion)	1	FLOAT	FLOATI FLOATJ	INTEGER*2 INTEGER*4	REAL*4 REAL*4
REAL*8 Float ¹⁰ (Integer-to-REAL*8 conversion)	1	DFLOAT	DFLOATI DFLOATJ	INTEGER*2 INTEGER*4	REAL*8 REAL*8
REAL*16 Float (Integer-to-REAL*16 conversion)	1	QFLOAT	—	INTEGER*2 INTEGER*4	REAL*16 REAL*16
Conversion to COMPLEX*8, or COMPLEX*8 from Two Arguments	1,2 ¹² 1,2 1,2 1,2 1,2 1 1	CMPLX	— — — — — — —	INTEGER*2 INTEGER*4 REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	COMPLEX*8 COMPLEX*8 COMPLEX*8 COMPLEX*8 COMPLEX*8 COMPLEX*8 COMPLEX*8

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Conversion to COMPLEX*16 or COMPLEX*16 from Two Arguments	1,2 ¹² 1,2 1,2 1,2 1,2 1 1	DCMPLX	— — — — — — —	INTEGER*2 INTEGER*4 REAL*4 REAL*8 REAL*16 COMPLEX*8 COMPLEX*16	COMPLEX*16 COMPLEX*16 COMPLEX*16 COMPLEX*16 COMPLEX*16 COMPLEX*16 COMPLEX*16
Real Part of Complex	1	—	REAL DREAL	COMPLEX*8 COMPLEX*16	REAL*4 REAL*8
Imaginary Part of Complex	1	—	AIMAG DIMAG	COMPLEX*8 COMPLEX*16	REAL*4 REAL*8
Complex from Two Arguments	(See Conversion to COMPLEX*8 and Conversion to COMPLEX*16)				
Complex Conjugate (if a=(X,Y) CONJG (a)=(X,-Y))	1	CONJG	CONJG DCONJG	COMPLEX*8 COMPLEX*16	COMPLEX*8 COMPLEX*16
REAL*8 product of REAL*4's a1*a2	2	—	DPROD	REAL*4	REAL*8

<p>Maximum¹² $\max(a_1, a_2, \dots, a_n)$ (returns the maximum value from among the argument list; there must be at least two arguments)</p>	n	MAX	IMAX0	INTEGER*2	INTEGER*2
			JMAX0	INTEGER*4	INTEGER*4
			AMAX1	REAL*4	REAL*4
			DMAX1	REAL*8	REAL*8
			QMAX1	REAL*16	REAL*16
		MAX0	IMAX0	INTEGER*2	INTEGER*2
			JMAX0	INTEGER*4	INTEGER*4
		MAX1	IMAX1	REAL*4	INTEGER*2
			JMAX1	REAL*4	INTEGER*4
		AMAX0	AIMAX0 AJMAX0	INTEGER*2 INTEGER*4	REAL*4 REAL*4
<p>Minimum¹² $\min(a_1, a_2, \dots, a_n)$ (returns the minimum value among the argument list; there must be at least two arguments)</p>	n	MIN	IMIN0	INTEGER*2	INTEGER*2
			JMIN0	INTEGER*4	INTEGER*4
			AMIN1	REAL*4	REAL*4
			DMIN1	REAL*8	REAL*8
			QMIN1	REAL*16	REAL*16
		MIN0	IMIN0	INTEGER*2	INTEGER*2
			JMIN0	INTEGER*4	INTEGER*4
		MIN1	IMIN1	REAL*4	INTEGER*2
			JMIN1	REAL*4	INTEGER*4
		AMIN0	AIMIN0 AJMIN0	INTEGER*2 INTEGER*4	REAL*4 REAL*4

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Positive Difference $a_1 - \min(a_1, a_2)$ (returns the first argument minus the minimum of the two arguments)	2	DIM	IIDIM	INTEGER*2	INTEGER*2
			JIDIM	INTEGER*4	INTEGER*4
			DIM	REAL*4	REAL*4
			DDIM	REAL*8	REAL*8
			QDIM	REAL*16	REAL*16
Remainder $a_1 - a_2 * [a_1 / a_2]$ (returns the remainder when the first argument is divided by the second)	2	MOD	IIDIM	INTEGER*2	INTEGER*2
			JIDIM	INTEGER*4	INTEGER*4
			IMOD	INTEGER*2	INTEGER*2
			JMOD	INTEGER*4	INTEGER*4
			AMOD	REAL*4	REAL*4
Transfer of Sign $ a_1 \text{ Sign } a_2$	2	SIGN	DMOD	REAL*8	REAL*8
			QMOD	REAL*16	REAL*16
			IISIGN	INTEGER*2	INTEGER*2
			JISIGN	INTEGER*4	INTEGER*4
			SIGN	REAL*4	REAL*4
			DSIGN	REAL*8	REAL*8
			QSIGN	REAL*16	REAL*16
			IISIGN	INTEGER*2	INTEGER*2
			JISIGN	INTEGER*4	INTEGER*4

Bitwise AND (performs a logical AND on corresponding bits)	2	IAND	IAND JIAND	INTEGER*2 INTEGER*4
Bitwise OR (performs an inclusive OR on corresponding bits)	2	IOR	IIR JIOR	INTEGER*2 INTEGER*4
Bitwise Exclusive OR (performs an exclusive OR on corresponding bits)	2	IEOR	IIEOR JIEOR	INTEGER*2 INTEGER*4
Bitwise Complement (complements each bit)	1	NOT	INOT JNOT	INTEGER*2 INTEGER*4
Bitwise Shift (a ₁ logically shifted left a ₂ bits)	2	ISHFT	IISHFT JISHFT	INTEGER*2 INTEGER*4
Bit Extraction (extracts bits a ₂ through a ₂ +a ₃ -1 from a ₁); see also MVBITS system subroutine	3	IBITS	IIBITS JIBITS	INTEGER*2 INTEGER*4
Bit Set (returns the value of a ₁ with bit a ₂ of a ₁ set to 1)	2	IBSET	IIBSET JIBSET	INTEGER*2 INTEGER*4
Bit Test (returns .TRUE. if bit a ₂ of argument a ₁ equals 1)	2	BTEST	BITEST BJTEST	LOGICAL*2 LOGICAL*4

(continued on next page)

4.0 VAX-11 FORTRAN GENERIC AND INTRINSIC FUNCTIONS (Cont.)

Functions	Number of Arguments	Generic Name	Specific Name	Type of Argument	Type of Result
Bit Clear (returns the value of a_1 with bit a_2 of a_1 set to 0)	2	IBCLR	IIBCLR JIBCLR	INTEGER*2 INTEGER*4	INTEGER*2 INTEGER*4
Bitwise Circular Shift (circularly shifts rightmost a_3 bits of argument a_1 by a_2 places)	3	ISHFTC	IISHFTC JISHFTC	INTEGER*2 INTEGER*4	INTEGER*2 INTEGER*4
Length ¹² (returns length of the character expression)	1	—	LEN	CHARACTER	INTEGER*4
Index c_1, c_2 ¹² (returns the position of the substring c_2 in the character expression c_1)	2	—	INDEX	CHARACTER	INTEGER*4
Character ¹² (returns a character that has the ASCII value specified by the argument)	1	—	CHAR	LOGICAL*1 INTEGER*2 INTEGER*4	CHARACTER

ASCII Value ¹¹ (returns the ASCII value of the argument; the argument must be a character expression that has a length of 1)	1	—	ICHAR	CHARACTER	INTEGER*4
Character relationals (ASCII collating sequence)	2	—	LLT	CHARACTER	LOGICAL*4
	2	—	LLE	CHARACTER	LOGICAL*4
	2	—	LGT	CHARACTER	LOGICAL*4
	2	—	LGE	CHARACTER	LOGICAL*4

NOTES

1. The argument of SQRT, DSQRT, or QSQRT must be greater than or equal to zero. The result of CSQRT or CDSQRT is the principal value with the real part greater than or equal to zero. When the real part is zero, the result is the principal value with the imaginary part greater than or equal to zero.
2. The argument of ALOG, DLOG, QSQRT, ALOG10, DLOG10, QLOG10, ATAND, ATAN2D, ASIND, DASIND, ACOSD, DACOSD, or QACOSD must be greater than zero. The argument of CLOG or CDLOG must not be (0.,0.).
3. The argument of SIN, DSIN, QSIN, COS, DCOS, QCOS, TAN, DTAN, or QTAN must be in radians. The argument is treated modulo 2π . The argument of SIND, COSD, or TAND must be in degrees. The argument is treated modulo 360.
4. The absolute value of the argument of ASIN, DASIN, QASIN, ACOS, DACOS, QACOS, ASIND, DASIND, QASIND, ACOSD, DACOSD, or QACOSD must be less than or equal to 1.

5. The result of ASIN, DASIN, QASIN, ACOS, QACOS, ATAN, DATAN, QATAN, ATAN2, DATAN2, or QATAN2 is in radians. The result of ASIND, DASIND, QASIND, ACOSD, QACOSD, ATAND, DATAND, QATAND, ATAN2D, DATAN2D, or QATAN2D is in degrees.
6. If the value of the first argument of ATAN2, DATAN2, or QATAN2 is positive, the result is positive. When the value of the first argument is zero the result is zero if the second argument is positive and π if the second argument is negative. If the value of the first argument is negative the result is negative. If the value of the second argument is zero, the absolute value of the result is $\pi/2$. Both arguments must not have the value zero. The range of the result for ATAN2, DATAN2, and QATAN2 is: $-\pi < \text{result} \leq \pi$.
7. If the value of the first argument of ATAN2D, DATAN2D, or QATAN2D is positive, the result is positive. When the value of the first argument is zero the result will be zero if the second argument is positive and 180 degrees if the second argument is negative. If the value of the first argument is negative the result is negative. If the value of the second argument is zero, the absolute value of the result is 90 degrees. Both arguments must not have the value zero. The range of the result for ATAN2, DATAN2, QATAN2D is: -180 degrees $< \text{result} \leq$ 180 degrees.
8. The absolute value of a complex number, (X,Y), is the real value:

$$(X^2 + Y^2)^{1/2}$$
9. [x] is defined as the largest integer whose magnitude does not exceed the magnitude of x and whose sign is the same as that of x. For example [5.7] equals 5, and [-5.7] equals -5.

10. Functions that cause conversion of one data type to another type provide the same effect as the implied conversion in assignment statements. The function REAL with a real argument, the function DBLE with a double precision argument, the function INT with an integer argument, and the function QEXT with a REAL*16 argument return the value of the argument without conversion.
11. See Chapter 6 of the VAX-11 FORTRAN Language Reference Manual for additional information on character functions.
12. The functions INT, IDINT, IQINT, NINT, IDNINT, IQNINT, IFIX, MAX1, MIN1, and ZEXT return INTEGER*4 values if the /I4 command qualifier is in effect; INTEGER*2 values if the /NOI4 qualifier is in effect.
13. When CMPLX and DCMPLX have only one argument, this argument is converted into the real part of a complex value, and zero is assigned to the imaginary part; when there are two arguments (not complex), a complex value is produced by conversion of the first argument into the real part of the value; the second argument into the imaginary part.

5.0 ASCII CHARACTER SET

ASCII Decimal Number	Character	Meaning
0	NUL	Null
1	SOH	Start of heading
2	STX	Start of text
3	ETX	End of text
4	EOT	End of transmission
5	ENQ	Enquiry
6	ACK	Acknowledgement
7	BEL	Bell
8	BS	Backspace
9	HT	Horizontal tab
10	LF	Line feed
11	VT	Vertical tab
12	FF	Form feed
13	CR	Carriage return
14	SO	Shift out
15	SI	Shift in
16	DLE	Data link escape
17	DC1	Device control 1
18	DC2	Device control 2
19	DC3	Device control 3
20	DC4	Device control 4
21	NAK	Negative acknowledgement
22	SYN	Synchronous idle
23	ETB	End of transmission block
24	CAN	Cancel
25	EM	End of medium
26	SUB	Substitute
27	ESC	Escape
28	FS	File separator
29	GS	Group separator
30	RS	Record separator
31	US	Unit separator
32	SP	Space or blank
33	!	Exclamation mark
34	"	Quotation mark
35	#	Number sign
36	\$	Dollar sign
37	%	Percent sign
38	&	Ampersand
39	'	Apostrophe
40	(Left Parenthesis
41)	Right parenthesis
42	*	Asterisk
43	+	Plus sign
44	,	Comma
45	-	Minus sign or hyphen
46	.	Period or decimal point
47	/	Slash
48	0	Zero
49	1	One
50	2	Two
51	3	Three
52	4	Four
53	5	Five
54	6	Six

5.0 ASCII CHARACTER SET (Cont.)

ASCII Decimal Number	Character	Meaning
55	7	Seven
56	8	Eight
57	9	Nine
58	:	Colon
59	;	Semicolon
60	<	Left angle bracket
61	=	Equal sign
62	>	Right angle bracket
63	?	Question mark
64	@	At sign
65	A	Uppercase A
66	B	Uppercase B
67	C	Uppercase C
68	D	Uppercase D
69	E	Uppercase E
70	F	Uppercase F
71	G	Uppercase G
72	H	Uppercase H
73	I	Uppercase I
74	J	Uppercase J
75	K	Uppercase K
76	L	Uppercase L
77	M	Uppercase M
78	N	Uppercase N
79	O	Uppercase O
80	P	Uppercase P
81	Q	Uppercase Q
82	R	Uppercase R
83	S	Uppercase S
84	T	Uppercase T
85	U	Uppercase U
86	V	Uppercase V
87	W	Uppercase W
88	X	Uppercase X
89	Y	Uppercase Y
90	Z	Uppercase Z
91	[Left square bracket
92	\	Back slash
93]	Right square bracket
94	^ or †	Circumflex or up arrow
95	← or —	Back arrow or underscore
96	`	Grave accent
97	a	Lowercase a
98	b	Lowercase b
99	c	Lowercase c
100	d	Lowercase d
101	e	Lowercase e
102	f	Lowercase f
103	g	Lowercase g
104	h	Lowercase h
105	i	Lowercase i
106	j	Lowercase j
107	k	Lowercase k
108	l	Lowercase l
109	m	Lowercase m

5.0 ASCII CHARACTER SET (Cont.)

ASCII Decimal Number	Character	Meaning
110	n	Lowercase n
111	o	Lowercase o
112	p	Lowercase p
113	q	Lowercase q
114	r	Lowercase r
115	s	Lowercase s
116	t	Lowercase t
117	u	Lowercase u
118	v	Lowercase v
119	w	Lowercase w
120	x	Lowercase x
121	y	Lowercase y
122	z	Lowercase z
123	{	Left brace
124		Vertical line
125	}	Right brace
126	~	Tilde
127	DEL	Delete



AV-M763A-TE