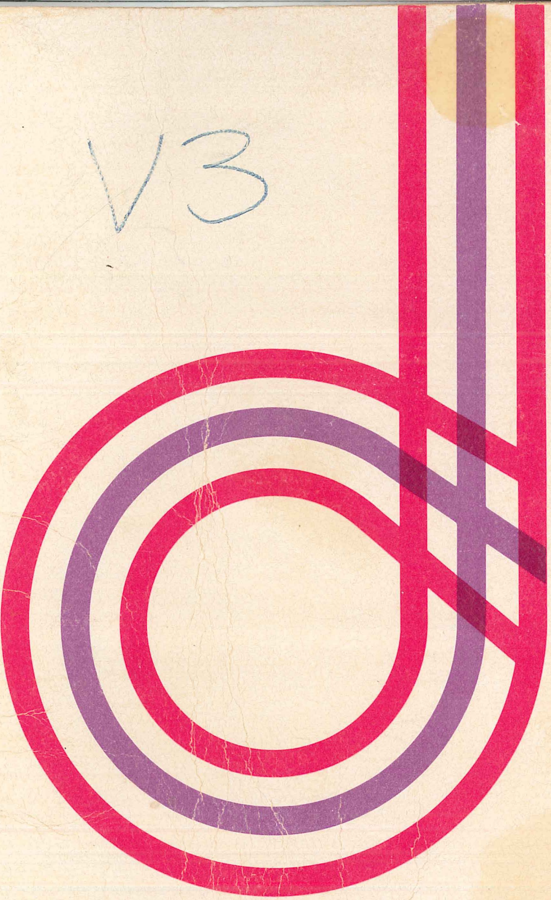
V3



digital

RT-11 pocket guide

RT-11
Pocket Guide

Order No. AV-5287B-TC

CONTENTS

CONTENTS (Cont.)

CONTENTS (Cont.)

SYSCOM LOCATIONS

40,41
Starting address of job

42, 43
Initial value of Stack Pointer (SP)

44,45
Job Status Word (JSW)

Bit	Meaning
15	USR swap bit (unused in FB)
14*	Upper/lower case bit
13*	Reenter bit
12*	TT: special mode bit
11*	Chains to indirect file
10*	Indicates non-privileged job, runs in a virtual memory partition
9	Overlay bit
8*	Indicates job was chained to or is chainable
7*	Error halt bit
6*	Inhibit TT: I/O suspend bit
5*	Filters escape sequences
4*	Processes escape sequences

*Indicates bit that user sets

46,47
USR load address; normally 0

50,51
High address; initially set by linker, modify with .SETTOP

52
EMT error code; address it as a byte

53
User program error status; address it as a byte; never clear it

54,55
Beginning of resident monitor; do not alter it

56
Character requiring fill (7-bit ASCII)

57
Filler count (if 56 and 57 = 0, no fillers are required)

PERMANENT DEVICE NAMES

CR:
Card Reader

CTn:
Cassette (n=0 - 1)

DK:
Default storage device (disk or DECtape)

DKn:
Unit n of same device type as DK:

DLn:
RL01 Disk (n=0 - 3)

DMn:
RK06, RK07 Disk (n=0 - 7)

DPn:
RP02, RP03 Disk (n=0 - 7)

DSn:
RJS03, RJS04 Disk (n=0 - 7)

DTn:
DECtape (n=0 - 7)

DXn:
RX01 Diskette (n=0 - 3)

DYn:
RX02 Diskette (n=0 - 3)

EL:
Error Logging handler

LP:
Line printer

PERMANENT DEVICE NAMES (Cont.)

MMn:
TJU16, TU45 Magtape (n=0 - 7)

MTn:
TM11, TMA11, TS03 Magtape (n=0 - 7)

NL:
Null device handler

PC:
PC11 Combined High Speed Paper Tape Reader and Punch

RF:
RF11 Fixed-Head Disk

RKn:
RK05 Disk (n=0-7)

SY:
System device

SYn:
Unit n of same device type as that from which the system was bootstrapped

TT:
Console terminal

STANDARD FILE TYPES

.BAD
File with bad blocks

.BAK
Editor backup file

.BAS
BASIC source file

.BAT
BATCH command file

.COM
Indirect file

.CTL
BATCH control file (generated by BATCH)

.CTT
BATCH internal temporary file

.DAT
BASIC or FORTRAN data file

.DBL
DIBOL source file

.DIF
SRCCOM output file

.DIR
Directory listing file

.DMP
DUMP output file

.FOR
FORTRAN IV source file

.LDA
Absolute binary file

.LOG
BATCH log file

.LST
Listing file

.MAC
MACRO source file

.MAP
Map file

.OBJ
Relocatable binary file

.REL
Foreground job relocatable image

.SAV
Memory image

.SML
System MACRO library

STANDARD FILE TYPES (Cont.)

.SOU
Temporary source file generated by BATCH

.STB
Symbol table file

.SYS
System files and handlers

SPECIAL FUNCTION KEY COMMANDS

CTRL/A
Displays next page of output; use only after CTRL/S (with GT ON)

CTRL/B
Directs all keyboard input to background job (until CTRL/F)

CTRL/C
Terminates execution; if background, returns to monitor command level

CTRL/E
Displays all I/O on screen and console terminal simultaneously (with GT ON)

CTRL/F
Directs all keyboard input to foreground job (until CTRL/B)

CTRL/O
Inhibits printing on console terminal

CTRL/Q
Resumes console output; use only after CTRL/S

CTRL/S
Temporarily suspends terminal output (until CTRL/A or CTRL/Q)

CTRL/U
Deletes current line being entered

CTRL/X
Ignores entire command string being entered (EDIT only)

CTRL/Z
Terminates TT: input

RETURN
Executes command line (except EDIT)

ESCAPE or ALTMODE
Terminates text string; two ESCAPEs cause command execution (EDIT only)

RUBOUT or DELETE
Deletes last character typed on current line

KEYBOARD MONITOR

General Command Syntax

COMMAND[/options] input-filespec[/options] — output-filespec[/options]
or
COMMAND[/options]
PROMPT1? input-filespec[/options]
PROMPT2? output-filespec[/options]
(filespec represents dev:filnam.typ)

Commands

Command and option summaries start on page 50 at the back of this guide.

APL
Invokes the APL language interpreter

ASSIGN
Associates a logical device name with a physical device

KEYBOARD MONITOR (Cont.)

Commands (Cont.)

B
Sets a relocation base

BASIC
Invokes the BASIC language interpreter

BOOT
Directs a new monitor to take control of the system

CLOSE
Makes permanent all output files that are currently open after the background job terminates

COMPILE
Invokes one or more language processors to assemble or compile the files you specify

COPY
Performs a variety of file transfer and maintenance operations

D
Deposits values in memory beginning at the location you specify

DATE
Sets or displays the current system date

DEASSIGN
Disassociates a logical device name from a physical device

DELETE
Deletes the files you specify

DIBOL
Invokes the DIBOL compiler to compile one or more source programs

DIFFERENCES
Compares two files and lists the differences between them

DIRECTORY
Lists information you request about a device, a file, or a group of files

DUMP
Lists all or any part of a file in octal words, octal bytes, ASCII characters, or Radix-50 characters

E
Prints in octal the contents of a memory address

EDIT
Invokes the text editor

EXECUTE
Invokes one or more language processors to assemble or compile the files you specify; it also links object modules and initiates execution of the resultant program

FOCAL
Invokes the FOCAL language interpreter

FORTRAN
Invokes the FORTRAN IV compiler to compile one or more source programs

FRUN
Initiates execution of foreground jobs

GET
Loads a memory image file into memory

GT
Enables or disables the VT11 or VS60 display hardware

HELP
Lists useful information

INITIALIZE
Clears and initializes a device directory

INSTALL
Installs the device you specify into the system

Commands (Cont.)

LIBRARY
Lets you create, update, modify, list, and maintain library files

LINK
Converts object modules produced by an RT-11 supported language processor into a format suitable for loading and execution

LOAD
Makes a device handler resident in memory for use with BATCH or foreground/background jobs

MACRO
Invokes the MACRO assembler to assemble one or more source files

PRINT
Lists the contents of one or more files on the line printer

R
Loads a memory image file into memory and starts execution

REENTER
Starts the program at its reentry address (the start address minus 2)

REMOVE
Removes a device from the system tables

RENAME
Assigns a new name to an existing file

RESET
Resets several background system tables and does a general clean-up of the background area

RESUME
Continues execution of the foreground job at the point the SUSPEND command was issued

RUN
Loads a memory image file into memory and starts execution

SAVE
Writes memory areas in memory image format to the file and device that you specify

SHOW
Prints on the terminal device assignments, SYSGEN options in effect, hardware configuration, terminal characteristics, and the status of device handlers

SQUEEZE
Consolidates in a single area all unused blocks on the device you specify

START
Initiates execution of the program currently in memory at the address you specify

SUSPEND
Stops execution of the foreground job

TIME
Sets or displays the current time of day

TYPE
Lists the contents of one or more files on the terminal

UNLOAD
Makes handlers that were previously loaded non-resident, thus freeing the memory space they occupied; can also remove an inactive foreground job

The SET Command

SET
Changes device handler characteristics and certain system configuration parameters. Default settings are underlined

The SET Command (Cont.)

CR:

Condition: CODE=n
Action: Modifies the card reader handler to use either the DEC 026 or DEC 029 card codes

Condition: CRLF
Action: Appends a carriage return/line feed combination to each card image

Condition: NOCRLF
Action: Transfers each card image without appending a carriage return/line feed combination

Condition: HANG
Action: Waits for you to make a correction if the reader is not ready at the start of a transfer

Condition: NOHANG
Action: Generates an immediate error if the device is not ready at the start of a transfer

Condition: IMAGE
Action: Causes each card column to be stored as a 12-bit binary number, one column per word

Condition: NOIMAGE
Action: Allows the normal translation (as specified by the CODE option) to take place

Condition: TRIM
Action: Removes trailing blanks from each card that the system reads

Condition: NOTRIM
Action: Transfers a full 80 characters per card

CT:

Condition: RAW
Action: Performs a read-after-write check for every record written

Condition: NORAW
Action: Writes every record directly without reading it back for verification

EDIT

Condition: EDIT
Action: Invokes the text editor EDIT with the keyboard monitor EDIT command

Condition: TECO
Action: Invokes the text editor TECO with the keyboard monitor EDIT command

ERROR

Condition: ERROR
Action: Causes indirect command files and certain keyboard monitor commands to abort if errors or severe errors occur

KEYBOARD MONITOR (Cont.)

The SET Command (Cont.)

Condition: NONE
Action: Allows indirect command files and keyboard monitor commands to continue to execute even though they contain significant errors

Condition: SEVERE
Action: Causes indirect command files and keyboard monitor commands to abort if severe errors occur

Condition: WARNING
Action: Causes indirect command files and keyboard monitor commands to abort if warnings, errors, or severe errors occur

LP:

Condition: CR
Action Sends carriage returns to the printer

Condition: NOCR
Action: Prevents the system from sending carriage returns to the printer

Condition: CTRL
Action: Passes all characters, including non-printing control characters, to the printer

Condition: NOCTRL
Action: Ignores non-printing control characters

Condition: FORM0
Action: Issues a form feed before a request to print block 0

Condition: NOFORM0
Action: Turns off FORM0 mode

Condition: HANG
Action: Waits for you to make a correction if the line printer is not ready or becomes not ready during printing

Condition: NOHANG
Action: Generates an immediate error if the line printer is not ready

Condition: LC
Action: Allows the system to send lower case characters to the printer

Condition: NOLC
Action: Translates lower case characters to upper case before printing

Condition: TAB
Action: Sends TAB characters to the LA180 line printer

Condition: NOTAB
Action: Does not send TAB characters to the line printer; simulates TABs with spaces

The SET Command (Cont.)

Condition: **WIDTH=n**
Action: Sets the line printer width to n (n=30 -255)

MM:

Condition: **DEFALT=9**
Action: Returns to default settings for 9-track tape

Condition: **DENSE=[800 or 809 or 1600]**
Action: Sets density for the 9-track tape handler

Condition: **ODDPAR**
Action: Sets parity to odd for 9-track tape

Condition: **NOODDPAR**
Action: Sets parity to even for 9-track tape

MT:

Condition: **DEFALT=[7 or 9]**
Action: Returns to default settings for 7- or 9-track tape

Condition: **DENSE=[200 or 556 or 807 or 800 or 809]**
Action: Sets density for 7- or 9-track tape

Condtion: **DUMP**
Action: Writes bytes to 7-track tape

Condition: **ODDPAR**
Action: Sets parity to odd for 7- or 9-track tape

Condition: **NOODDPAR**
Action: Sets parity to even for 7- or 9-track tape

TT:

Condition: **CONSOL=n**
Action: Changes background console to the terminal associated with logical unit number n (for multi-terminal systems only). The default is 0.

Condition: **CRLF**
Action: Issues a carriage return/line feed combination on the console terminal whenever you attempt to type past the right margin

Condition: **NOCRLF**
Action: Takes no special action at the right margin

Condition: **FB**
Action: Treats CTRL/B and CTRL/F as background and foreground program control characters and does not transmit them to your program

Condition: **NOFB**
Action: Causes CTRL/B and CTRL/F to have no special meaning

Condition: **FORM**
Action: Indicates that the console terminal is capable of executing hardware form feeds

KEYBOARD MONITOR (Cont.)

The SET Command (Cont.)

Condition: NOFORM
Action: Simulates form feeds by generating 8 line feeds

Condition: HOLD
Action: Hold screen mode for VT50

Condition: NO HOLD
Action: Disables Hold Screen mode for VT50

Condition: PAGE
Action: Treats CTRL/S and CTRL/Q characters as terminal output hold and unhold flags and does not transmit them to your program

Condition: NOPAGE
Action: Causes CTRL/S and CTRL/Q to have no special meaning

Condition: QUIET
Action: Prevents the system from echoing lines from indirect files

Condition: NOQUIET
Action: Echoes lines from indirect files

Condition: SCOPE
Action: Echoes RUBOUT characters as backspace-space-backspace

Condition: NOSCOPE
Action: Echoes RUBOUT characters as a backslash followed by the character deleted

Condition: TAB
Action: Indicates that the console terminal is capable of executing hardware tabs

Condition: NOTAB
Action: Simulates tab stops every 8 positions

Condition: WIDTH=n
Action: Sets the terminal width to n (n=30 - 255) The default is 72.

WILDCARDS

Condition: EXPLICIT
Action: Causes the system to recognize file specifications exactly as you type them

Condition: IMPLICIT
Action: Causes the system to interpret missing fields in file specifications as asterisks

USR

Condition: SWAP
Action: Allows the background job to place the USR in a swapping state

Condition: NOSWAP
Action: Prevents the background job from placing the USR in a swapping state

GENERAL CSI COMMAND STRING FORMAT

output-filespecs/option=input-filespecs/option

```
output-filespec syntax:
   dev:filnam.typ[n],...dev:filnam.typ[n]
input-filespec syntax:
   dev:filnam.typ,...dev:filnam.typ
/option syntax:
   /o:oval
      or
   /o:dval.
```

dev:	1 to 3 character device name
filnam	1 to 6 character file name
typ	0 to 3 character file type
n	length (decimal) of output file
/o	option character
oval	optional octal or 3 alphabetic character argument
dval.	optional decimal argument
=	delimiter separating output specifications from input specifications

TEXT EDITOR (EDIT)

Command Arguments

n
Any integer in the range - 16383 to +16383

0
Refers to the beginning of a line

/
Refers to the end of text in the current text buffer

=
Represents - n, where n = length of the last text argument used (use with J, D, and C commands only)

Commands

(n = argument, filespec = dev:filnam.typ, $ = ESCAPE, text = text string)

nA
Advances the pointer by n lines

B
Moves the pointer to the beginning of the text buffer

nCtext$
Changes to the indicated text

nD
Deletes n characters from the buffer

EBfilespec$
Opens a file for editing and creates a backup copy (.BAK)

EC
Displays text on a video terminal (use after ED)

ED
Displays text on a video terminal and shows commands below the text window (use as the first command in immediate mode)

EF
Closes the current output file

EL
Echoes characters in upper and lower case

nEM
Executes the macro command string n times

ERfilespec$
Opens a file for input

EU
Echoes characters in upper case only (use after EL)

Commands (Cont.)

EV
Prints the version number of the editor

EWfilespec$
Opens a new file for output

EX
Terminates editing, outputs remainder of the input file, and returns to the monitor

nFtext$
Searches the entire input file for the nth occurrence of the text string

nGtext$
Searches the current text buffer for the nth occurrence of the text string

Itext$
Inserts text into the text buffer

nJ
Moves the pointer by n characters

nK
Deletes n lines from the text buffer

nL
Lists n lines of text on the terminal

Mxcommand stringx$
Inserts the command string into the macro buffer (x represents a delimiter not used in the command string); Mxx deletes the macro buffer; 0M clears the macro buffer

nN
Outputs the current text buffer; reads the next page of input and repeats this sequence n-1 times

nPtext$
Searches the entire input file for the nth occurrence of the text string without producing any output

R
Reads the next page of text into the text buffer

nS
Saves n following lines of text in the save buffer

U
Inserts the contents of the save buffer into the text buffer; 0U deletes the save buffer

V
Prints the current line on the terminal

nW
Writes n following lines of text to the output file

nXtext$
Replaces lines of text with the indicated text

Immediate Mode Commands

CTRL/D
Advances the pointer 1 character

CTRL/G
Moves the pointer to the beginning of the previous line

CTRL/N
Advances the pointer to the beginning of the next line

CTRL/V
Moves the pointer back 1 character

ESCAPE
Use 2 successive ESCAPEs to enter immediate mode; use a single ESCAPE to return to command mode

Immediate Mode Commands (Cont.)

RUBOUT
Deletes the character immediately preceding the pointer

Any character except CTRL/C or those shown above
Inserts the character as text immediately before the pointer

PERIPHERAL INTERCHANGE PROGRAM (PIP)

Options

/A
Copies files in ASCII mode

/B
Copies files in formatted binary mode

/C
Includes only files with current date

/D
Deletes files

/G
Ignores input errors

/K:n
Makes n copies of output file to LP:, TT:, or PC:

/M:n
Indicates file position (n) for cassette or magtape operations

/N
Does not copy or rename file if a file with the same name and type exists on output device

/O
Deletes file on output device if you copy a file with the same name and type to that device

/P
Copies all files except those you specify

/Q
Queries before executing the operation

/R
Renames input file to output file name

/S
Copies file one block at a time (single-block transfer)

/T
Puts current date on files you copy or rename

/U
Copies and concatenates files

/W
Prints a log of files operated upon

/Y
Includes .SYS files in operation

DEVICE UTILITY PROGRAM (DUP)

Options

/B
Writes FILE.BAD entries over bad blocks (use with /Z)

/C:m[:n]
Creates a file; m = starting block number; n = size in blocks

/F
Prints file name of bad block (use with /K)

/H
Reads and writes bad blocks (use with /K)

/I [:rstart:rstop:wstart]
Copies image of one device to another; arguments represent block numbers

Options (Cont.)

/K[:start[:stop]]
Scans a device for bad blocks; arguments represent block numbers

/N:n
Sets number of directory segments; n=1 - 37 (octal); (use with /Z)

/O
Boots device or file

/R[:RET]
Scans an RK06 disk for bad blocks and creates a replacement table (use :RET to keep previous table)

/S
Compresses a device

/T:n
Extends a file by n blocks

/U
Writes bootstrap into blocks 0 and 2-5 of a device

/V[:VOL]
Prints user ID and owner name; use with /Z to write ID on device; use :VOL to change only the ID and owner name

/W
Initiates operation and then pauses

/X
Inhibits automatic rebooting of system device after /S

/Y
Inhibits query messages

/Z[:n]
Initializes device directory; n = number of extra words in each directory segment

DIRECTORY PROGRAM (DIR)

Options

/A
Lists directory alphabetically

/B
Includes starting block numbers in directory listing

/C:n
Lists directory in n columns (n=1-9)

/D[:date]
Includes only files with the date you specify (default is current date)

/E
Lists entire directory, including unused spaces

/F
Lists short directory in 5 columns

/G
Lists the file you specify and all files that follow it in the directory

/J[:date]
Lists files created on or after the date you specify (default is current date)

/K[:date]
Lists files created before the date you specify (default is current date)

/L
Lists normal device directory

/M
Lists unused areas

/N
Lists device summary

DIRECTORY PROGRAM (DIR) (Cont.)

Options (Cont.)

/O
Lists sizes and block numbers in octal

/P
Excludes the files you specify from the listing

/Q
Lists directory of deleted files

/R
Lists directory in reverse order

/S[:xxx]
Sorts listing (xxx=DAT,NAM,POS,SIZ, or TYP)

/V
Lists the volume ID and owner name as part of the directory listing header

LINKER (LINK)

Options

/A
Lists global symbols in program sections in alphabetical order

/B
Changes bottom address of program to n (illegal with /R)

/C
Allows additional lines of command string input

/E:n
Extends a program section to n

/F
Uses the default FORTRAN library on system device

/H:n
Specifies program's highest address (n)

/I
Includes global symbols from library in load module

/K:n
Inserts value (n=1-28) into word 56 of block 0

/L
Produces output file in LDA format (illegal with /R)

/M[:n]
Allows console specification of stack address (n=6-digit unsigned octal number); linker requests global symbol if you omit n

/O:n
Program is an overlay structure; n specifies the overlay region to which the module is assigned

/P:n
Changes the amount of space the linker uses for library routines list

/R[:n]
Produces output in REL format (n=stack size); REL files can be linked in either SJ or FB but can be run only in foreground with FB

/S
Allows maximum memory space for linker symbol table

/T[:n]
Allows console specification of program's starting (transfer) address (n=6-digit unsigned octal number); linker requests global symbol if you omit n

/U:n
Rounds up a program section (n=power of 2)

/W
Produces wide load map listing

LINKER (LINK) Cont.)

Options (Cont.)

/X
Does not output bitmap if code is below 400

/Y:n
Starts a program section on address boundary n

/Z:n
Sets unused locations in load module to n

//
Allows additional lines of command string input

LIBRARIAN (LIBR)

Options

/C
Allows additional lines of command string input

/D
Deletes module from library file

/E
Extracts module from library file and stores it in an OBJ file

/G
Deletes global from library directory

/N
Includes module names in library directory

/P
Includes p-section names in library directory

/R
Replaces modules in library file

/U
Updates (inserts and replaces) modules in library file

/W
Produces a wide library directory listing

//
Allows additional lines of command string input

Default
The default operation (if you specify no option) is insertion

DUMP

Options

/B
Outputs octal bytes

/E:n
Ends output at block number n

/G
Ignores input errors

/N
Suppresses ASCII output

/O:n
Outputs only block number n

/S:n
Starts output with block number n

/T
Defines a tape as non-RT-11 file-structured

/W
Outputs octal words (the default mode)

/X
Outputs Radix-50 characters

FILEX

Options

/A
Performs a character-by-character ASCII transfer; deletes rubouts and nulls (^Z terminates the transfer)

/D
Deletes a file from the device directory (valid for DOS/BATCH, RSTS DECtape, interchange diskette)

/F
Lists short device directory (file names and types)

/I
Performs image mode transfer

/L
Lists entire device directory

/P
Performs packed image mode transfer

/S
Indicates that the device is DOS/BATCH or RSTS block-replaceable device

/T
Indicates that the device is DECsystem-10 DECtape

/U[:n]
Indicates that the device is interchange diskette (n=length of each output record in characters)

/Y
Suppresses query message

/Z
Zeroes device directory in proper format (valid for DOS/BATCH, RSTS DECtape, interchange diskette)

SRCCOM

Options

/B
Compares blank lines

/C
Ignores comments and spacing

/F
Sends form feeds to the output file

/H
Prints a list of available options

/L:n
Specifies the number of lines that must agree to constitute a match (n=1-310); default value is 3

/S
Ignores spaces and tabs

ON-LINE DEBUGGING TECHNIQUE (ODT)

Commands

RETURN
Closes open location and accepts next command

LINE FEED
Closes current location and opens next sequential location

↑ or ^
Opens previous location

← or _
Indexes contents of opened location by contents of PC and opens resulting location

>
Uses contents of open location as a relative branch and opens the referenced location

<
Returns to sequence prior to @, > or ← and opens next location

Commands (Cont.)

@
Uses contents of open location as absolute address and opens the referenced location

r/
Opens word at location r

/
Reopens last opened location

r\
Opens byte at location r

\
Reopens last opened location as byte

!
Prints address of opened location relative to the relocation register whose contents are closest

n!
Prints address of opened location relative to relocation register n

$n/
Opens general register n

$B/
Opens first word of breakpoint table

$C/
Opens constant register

$F/
Opens format register

$M/
Opens first mask register

$P/
Opens priority register

$R/
Opens first relocation register

$S/
Opens status register

r;nA
Prints n bytes in ASCII format starting at location r, then allows input of n bytes from the terminal

;B
Removes all breakpoints

r;nB
Sets breakpoint n at location r

;nB
Removes breakpoint n

r;C
Stores r in constant register

r;E
Searches for instructions that reference effective address r

;F
Fills memory words with the contents of the constant register

r;G
Starts execution of program at location r

;I
Fills memory bytes with low-order 8 bits of the constant register

r;O
Calculates offset from current location to r

;P
Execution proceeds from breakpoint

n;P
Execution proceeds for the next n instructions

k;P
Execution proceeds from breakpoint; stops after encountering the breakpoint k times

Commands (Cont.)

;R
Sets all relocation registers to -1

;nR
Sets relocation register n to -1

r;nR
Sets relocation register n to the value of r (default n=0)

R
Subtracts the relocation register whose contents are closest to but <= contents of open location from contents of open location and prints the result

nR
Subtracts contents of relocation register n from contents of the opened word and prints the result

;S
Disables single-instruction mode

;nS
Enables single-instruction mode; disables breakpoints

r;W
Searches for words with bit patterns matching r

X
Performs a Radix-50 unpack of opened location; permits storage of new Radix-50 binary number

PATCH

Options

/A
Patches directory, source file, or monitor file

/C
Requests checksum

/D
Prints checksum

/O
Patches overlay-structured file

Commands

v;nR
Sets relocation register n to v

x;B
Sets bottom address of overlay file to x

r,o/
Opens word location indicated by relocation register r + offset o

r,o
Opens byte location indicated by relocation register r + offset o

s:r,o/
Opens word location indicated by relocation register r + offset o in overlay segment s

s:r,o
Opens byte location indicated by relocation register r + offset o in overlay segment s

RETURN
Closes open word or byte

LINE FEED
Closes open word or byte and opens next word or byte

^
Closes open word or byte and opens previous word or byte

@
Closes open word and opens the word it addresses

PATCH (Cont.)

Commands (Cont.)

F
  Closes open file and requests new file

E
  Closes file and returns to the monitor

x;O
  Modifies overlay value

←
  Adds next locations to checksum value until next ←

A
  Prints contents as ASCII characters

X
  Prints contents as Radix-50 characters

C(x[x])
  Resets contents to ASCII character

P(xxx)
  Resets contents to Radix-50 value of ASCII characters

PAT

General Command Syntax

[output-filespec=] input-filespec[/C[:n]] — ,correct-filespec[/C[:n]]

BATCH COMMAND LANGUAGE

Command Field Options

/BAN[NER]
  Prints the header of a job on the log file

/NOBAN[NER]
  Does not print a job header

/CRE[F]
  Produces a cross-reference listing during assembly (use with $MACRO only)

/NOCRE[F]
  Does not produce a cross-reference listing

/DEL[ETE]
  Deletes input files after the operation completes

/NODEL[ETE]
  Does not delete input files after the operation completes

/DOL[LARS]
  Following data can have a $ in the first character position of a line (use with $CREATE, $DATA, $FORTRAN, $MACRO); reading of the data is terminated by $JOB, $SEQUENCE, $EOD, $EOJ, or by physical end-of-file

/NODOL[LARS]
  Following data cannot have a $ in the first character position ($ in the first character position signifies a BATCH control command)

/LIB[RARY]
  Includes the default library in the link operation (use with $LINK or $MACRO)

/NOLIB[RARY]
  Does not include the default library in the link operation

/LIS[T]
  Produces a temporary listing file on the listing device or writes data image on the log device (use with $BASIC, $CREATE, $DATA, $FORTRAN, $JOB, $MACRO)

/NOLIS[T]
  Does not produce a temporary listing file

Command Field Options (Cont.)

/MAP
Produces a temporary link map on the listing device (use with $FORTRAN, $LINK, $MACRO)

/NOMAP
Does not produce a link map

/OBJ[ECT]
Produces a temporary object file as output of compilation or assembly or includes temporary object files in the link (use with $FORTRAN, $LINK, $MACRO)

/NOOBJ[ECT]
Does not produce a temporary object file or does not include temporary object files in link

/RT11
Sets BATCH to operate in RT-11 mode (use only with $JOB)

/NORT11
Does not set BATCH to operate in RT-11 mode

/RUN
Links and executes programs compiled since the last 'link-and-go' operation or start of job (use with $BASIC, $FORTRAN, $LINK, $MACRO)

/NORUN
Does not link and execute the program after performing a command

/TIM[E]
Writes time of day to the log file when BATCH executes (use with $JOB)

/NOTIM[E]
Does not write time of day to the log file

/UNI[QUE]
Checks for unique spelling of options and keynames (use with $JOB)

/NOUNI[QUE]
Does not check for unique spelling

/WAI[T]
Pauses for operator action; type RETURN to continue (use with $DISMOUNT, $MESSAGE, $MOUNT)

/NOWAI[T]
Does not pause for operator action

/WRI[TE]
Indicates that the operator is to WRITE-ENABLE a device or volume (use with $MOUNT)

/NOWRI[TE]
Indicates that no writes are allowed or that the volume is read-only; informs the operator

Specification Field Options

/BAS[IC]
BASIC source file

/EXE[CUTABLE]
Indicates the executable program image file to be created as the result of a link operation

/FOR[TRAN]
FORTRAN source file

/INP[UT]
Input file (default if you do not specify any other option)

/LIB[RARY]
Indicates a library file to be included in the link operation (prior to the default library)

/LIS[T]
Listing file

Specification Field Options (Cont.)

/LOG[ICAL]
Indicates that the device is a logical device name (use with $DISMOUNT, $MOUNT)

/MAC[RO]
MACRO source file

/MAP
Linker map file

/OBJ[ECT]
Object file (output of assembly or compilation)

/OUT[PUT]
Output file

/PHY[SICAL]
Indicates physical device name

/SOU[RCE]
Indicates source file

/VID
Volume identification

Commands

$BAS[IC] [/option] [filespec/op] [!comments]
Compiles a BASIC source program; /option can be /RUN, /NORUN, /LIS, /NOLIS; /op can be /BAS, /SOU, /INP

$CAL[L] filespec [!comments]
Transfers control to another BATCH file, executes that BATCH file, and returns to calling BATCH stream

$CHA[IN] filespec [!comments]
Passes control to another BATCH file

$COP[Y] [/option] filespec/[OUTPUT] — filespec[/INPUT] [!comments]
Copies files; /option can be /DEL, /NODEL

$CRE[ATE] [/option] filespec [!comments]
Creates new files from data included in BATCH stream; /option can be /DOL, /NODOL, /LIS, /NOLIS

$DAT[A] [/option] [!comments]
Indicates the start of data; /option can be /DOL, /NODOL, /LIS, /NOLIS

$DEL[ETE] filespecs [!comments]
Deletes files

$DIR[ECTORY] [filespec/LIST] [filespec/INPUT] — [!comments]
Provides a device directory

$DIS[MOUNT] [/option] ldn:[/LOGICAL] [!comments]
Signals the operator to dismount a volume from a device; deassigns logical device name; /option can be /WAI, /NOWAI

$EOD [!comments]
Indicates the end of data

$EOJ [!comments]
Indicates the end of a job

$FOR[TRAN] [/option] [filespec1/op] — [filespec2/OBJECT] [filespec3/LIST] — [filespec4/EXECUTE] [filespec5/MAP] — [filespec6/LIBRARY] [!comments]
Compiles a FORTRAN source program; /option can be /RUN, /NORUN, /OBJ, /NOOBJ, /LIS, /NOLIS, /MAP, /NOMAP, /DOL, /NODOL; /op can be /FOR, /SOU, /INP

$JOB[/options] [!comments]
Indicates the start of a job; /option can be /BAN, /NOBAN, /LIS, /NOLIS, /RT11, /NORT11, /TIM, /NOTIM, /UNI, /NOUNI

Commands (Cont.)

$LIB[RARY] mylib [!comments] or
$LIB[RARY] mylib+SYSLIB [!comments]

Specifies libraries that BATCH should use in link operations

$LIN[K] [/option] [filespec1/OBJECT] –
[filespec2/LIBRARY] [filespec3/MAP] –
[filespec4/EXECUTE] [!comments]

Links modules for execution; /option can be /LIB, /NOLIB, /MAP, /NOMAP, /OBJ, /NOOBJ, /RUN, /NORUN

$MAC[RO] [/option] [filespec1/op] [filespec2/OBJECT] –
[filespec3/LIST] [filespec4/MAP] [filespec5/LIBRARY] –
[filespec6/EXECUTE] [!comments]

Assembles MACRO source programs; /option can be /RUN, /NORUN, /OBJ, /NOOBJ, /LIS, /NOLIS, /CRE, /NOCRE, /MAP, /NOMAP, /DOL, /NODOL, /LIB, /NOLIB; /op can be /MAC, /SOU, /INP

$MES[SAGE] [/option] message [!comments]

Issues a message to the operator; /option can be /WAI, /NOWAI

$MOU[NT] [/option] dev: [/PHYSICAL] [/VID=x] –
[ldn:/LOGICAL] [!comments]

Signals the operator to mount a volume on a device; optionally assigns a logical device name; /option can be /WAI, /NOWAI, /WRI, /NOWRI

$PRI[NT] [/option] filespec[/INPUT] [,filespec] –
[!comments]

Prints files on device LST:; /option can be /DEL, /NODEL

$RT11 [!comments]

Specifies that the following lines are RT-11 mode command lines; $ causes exit from RT-11 mode

$RUN filespec [!comments]

Causes a program to execute

$SEQ[UENCE] id [!comments]

Assigns an arbitrary identification number to a job

RT-11 Mode Control Commands

TAB NOTTY

Does not write terminal input and output to the log file

TAB TTYIN

Writes only terminal input to the log file

TAB TTYIO

Writes terminal input and output to the log file

TAB TTYOUT

Writes only terminal output to the log file (default)

'text'

Command to BATCH run-time handler; text can be:

CTY	Accepts input from console terminal
FF	Outputs current log buffer
NL	Inserts new line (LINE FEED) in BATCH stream
x	Inserts contents of variable x
"message"	Directs message to console terminal

Operating Options

/N

Compiles but does not execute

/T:n

Sets the /NOTIME option as default on $JOB (if n = 0); sets /TIME as default (if n = 1)

BATCH COMMAND LANGUAGE (Cont.)

Operating Options (Cont.)

/U

Detaches BATCH run-time handler from RT-11 monitor so the operator can unload it

/X

Indicates that input is a precompiled BATCH program

RETURN

Prints BATCH compiler version number

Operator Directives

\A

Changes input source to be console terminal

\B

Changes input source to be BATCH stream

\C

Sends following characters to log device

\D

Considers following characters as user data

\E

Sends following characters to RT-11 monitor

\F

Forces the output of the current log block; follow it immediately by another directive or BATCH terminates with forced end condition

\Hn

Help function to change the logging mode; n can be:

0	Logs only .TTYOUT and .PRINT
1	Logs .TTYOUT, .PRINT, and .TTYIN
2	Does not log .TTYOUT, .PRINT, and .TTYIN
3	Logs only .TTYIN

FORMAT

Options

/S

Single density, single surface diskette format

/W

Waits for volume substitution

/Y

Suppresses the automatic query message

Default

The default operations (if you specify no option) are as follows:

for device RK:	RK05 format
for device DY:	double density, single surface diskette format

MACRO

Listing Control Options

/L:arg

Enables listing argument

/N:arg

Disables listing argument

Arguments are as follows:

BEX	Binary extensions
BIN	Generated binary code
CND	Unsatisfied conditions and all .IF and .ENDC statements
COM	Comments

MACRO (Cont.)

Listing Control Options (Cont.)

LD	Listing directives having no arguments
LOC	Location counter
MC	Macro calls and repeat range expansions
MD	Macro definitions and repeat range expansions
ME	Macro expansions
MEB	Macro expansion binary code
SEQ	Source line sequence numbers
SPC	Source code
SYM	Symbol table
TOC	Table of contents
TTM	Listing output format in terminal mode

Function Control Options

/D:arg
Disables (.DSABL) certain functions in source input files

/E:arg
Enables (.ENABL) certain functions in source input files

Arguments are as follows:

ABS	Absolute binary output
AMA	Assembles all relative addresses as absolute addresses
CDR	Treats source columns >= 73 as comments
FPT	Floating point truncation
LC	Accepts lower case ASCII input
LSB	Local symbol block
PNC	Binary output
REG	Mnemonic definitions of registers

Cross-reference Options

/C
Equivalent to /C:S:M:E

/C:C
Control and program sections

/C:E
Error code grouping

/C:M
MACRO symbolic names

/C:P
Permanent symbols

/C:R
Register symbols

/C:S
User-defined symbols

BASIC/RT-11

Statements

CALL ["] routine name["] [(argument list)]
Calls assembly language routines from a BASIC program

CHAIN string [LINE expression]
Terminates execution of the program, loads the program specified by string, and begins execution at the lowest line number or at the line number specified by expression

CLOSE [[#] expr1,[#] expr2,[#] expr3, . . .]
Closes the files associated with the channel numbers and virtual channel numbers specified. If you do not specify a channel number, it closes all open files

Statements (Cont.)

COMMON list
Preserves values and names of specified variables and arrays when the CHAIN statement executes. The general format for list is:
var1[(expr[,expr])] [,var2[(expr[,expr])] , . . .]

DATA list
Use with READ to input listed data into an executing program

DEF FNletter $\left[\begin{Bmatrix}\$\\ \%\end{Bmatrix}\right]$ **(var1[,var2, . . . ,var5])=expression**
Defines a user function

DIM list
Reserves space in memory for arrays according to the subscripts specified after the variable name. The general format of list is:
var1(expr[,expr]) [,var2(expr[,expr]), . . .]

DIM #integer1,variable(integer2[,integer3]) [=integer4]
Dimensions the virtual array file associated with the channel number specified by integer1. Integer4 specifies the string size for string virtual arrays

END
Optional. Place this at the physical end of the program to terminate execution

FOR var=expr1 TO expr2 [STEP expr3]
Sets up a loop to be executed the specified number of times

GOSUB line number
Unconditionally transfers control to specified line of subroutine

GO TO line number
Unconditionally transfers control to specified line number

IF relational expression $\begin{Bmatrix}\textbf{THEN statement}\\ \textbf{THEN line number}\\ \textbf{GO TO line number}\end{Bmatrix}$
Conditionally executes the specified statement or transfers control to the specified line number

IF END expr $\begin{Bmatrix}\textbf{THEN statement}\\ \textbf{THEN line number}\\ \textbf{GO TO line number}\end{Bmatrix}$
Tests for end-of-file condition of input sequential file associated with channel number specified by expression

INPUT [#expr,] variable1[,variable2, . . .]
Inputs data from your terminal or from the file associated with the channel number specified by expression

KILL string
Deletes file specified by string

[LET] variable=expression
Assign value of expression to the specified variable

LINPUT [#expr,] string var1[,string var 2, . . .]
Inputs string data from the terminal or from the file associated with the channel number specified by expression

NAME string1 TO string2
Renames file specified by string 1 to name specified by string2

NEXT variable
Place at end of FOR loop to return control to FOR statement

ON expression GOSUB line number1 [,line number2 – ,line number 3, . . .]

Conditionally transfers control to subroutine at one line number specified in list

ON expression GO TO line number1 [,line number2 – ,line number 3, . . .]

Conditionally transfers control to one line number in the list

ON expression THEN line number1 [,line number2, . . .]

Equivalent to ON GO TO

OPEN string $\left\{\begin{array}{l}\text{FOR INPUT}\\ \text{FOR OUTPUT}\end{array}\right\}$ **AS FILE [#] expr – [DOUBLE BUF] [,RECORDSIZE expr] [,MODE expr] – [,FILESIZE expr]**

Opens a file specified by string for input or output as specified (default is input) and associates the file with the channel number specified by expr1

OVERLAY string [LINE expression]

Overlays or merges the program currently in memory with the program in the file specified by string; when overlay is completed, transfers control to either the next sequential BASIC line number or the line number specified by expression

PRINT [#expr,] [list]

Prints items in list on the terminal or to the file associated with channel number specified by expression

PRINT [#expr,] USING string, list

Prints items in list on the terminal or to the file associated with channel number specified by expr in the format determined by string

RANDOMIZE

Causes the random number generator (RND function) to produce different random numbers

READ variable [,variable2, . . .]

Assigns values listed in DATA statements to specified variables

REM comment

Contains explanatory comments about the BASIC program; has no effect on execution of program

RESET [#expr]

Equivalent to RESTORE

RESTORE [#expr]

Resets either the data pointer or, when specified, the input file associated with the specified channel number to the beginning

RETURN

Terminates a subroutine and returns control to the statement following the last executed GOSUB statement

STOP

Terminates execution of the program; place it at the logical end of the program

Key Commands

CTRL/C

Interrupts execution of a command or program and causes BASIC to print the READY message

CTRL/O

Causes all further terminal output to be discarded

CTRL/Q

Continues output to the terminal; cancels effect of CTRL/S

Key Commands (Cont.)

CTRL/S
Temporarily suspends all output to terminal until CTRL/Q is typed

CTRL/U
Deletes the entire current input line

RUBOUT
Deletes the last character typed

Commands

APPEND [file specification]
Merges the program in your area in memory with the program specified by the file specification

BYE
Returns control to the RT-11 monitor

CLEAR
Initializes all variables to 0 and all string variables to nulls; deletes arrays

COMPILE [file specification]
Saves a compiled version of the program

DEL line specification [,line specification, . . .]
Deletes specified lines

LENGTH
Prints on your terminal the size of the program in memory and the size of the remaining free memory

LIST[NH] [line specification1,line specification2, . . .]
Prints on the terminal the specified lines of the program currently in memory; NH suppresses the printing of the header line

NEW [program name]
Erases your storage area and sets the current program name to the one specified

OLD [file specification]
Erases your storage area and inputs the program from the specified file

RENAME program name
Changes the current program name to the one specified

REPLACE [file specification]
Replaces the specified file with the current program

RESEQ[[new line number],[old line number] – [-old line number2],[increment]]
Resequences program as specified

RUN[NH]
Executes the program in memory; NH suppresses the printing of the header line

RUN[NH] file specification
Erases your storage area, inputs the program from the specified file, and then executes the program; does not print header line in any case

SAVE [file specification]
Outputs the program in memory to the specified file

SCR
Erases your storage area and changes the program name to NONAME

SUB line numberxstring1xstring2[xinteger]
Substitutes the integer occurrence of string1 with string2 on the line specified; x is a delimiter and can be any character

UNSAVE file specification
Deletes the specified file

Arithmetic Functions

ABS(expr)
Returns the absolute value of the expression

ATN(expr)
Returns the arctangent of the expression as an angle in radians in the range + or – pi/2

COS(expr)
Returns the cosine of the angle specified by the expression in radians

EXP(expr)
Returns the value of e raised to the expression power; e is approximately 2.71828

INT(expr)
Returns the greatest integer that is less than or equal to the expression

LOG(expr)
Returns the natural logarithm of the expression

LOG10(expr)
Returns the base 10 logarithm of the expression

PI
Returns the value of pi (3.141593)

RND[(expr)]
Returns a random number between 0 and 1

SGN(expr)
Returns a value indicating the sign of expression

SIN(expr)
Returns the sine of the angle specified by expression in radians

SQR(expr)
Returns the square root of the expression

TAB(expr)
Causes the terminal type head to tab to the column number specified by the expression (valid only in the PRINT statement)

String Functions

ASC(string)
Returns as a decimal number the 8-bit internal code (ASCII value) for the 1-character string expression

BIN(string)
Converts a string expression containing a binary number to a decimal value; ignores blanks

CHR$(expr)
Generates a 1-character string whose ASCII value is the low-order 8 bits of the integer value of the expression

CLK$
Returns the time as a string in the form hh:mm:ss

DAT$
Returns the date as a string in the form dd-mon-yr

LEN(string)
Returns the number of characters in the string

OCT(string)
Converts a string expression containing an octal number to a decimal value; ignores blanks

POS(string1,string2,expr)
Searches for and returns the position of the first occurrence of string2 in string1

SEG$(string,expr1,expr2)
Returns the string of characters in position specified by expression1 through the position specified by expression2

STR$(expr)
Returns the string that represents the numeric value of the expression

BASIC/RT-11 (Cont.)

String Functions (Cont.)

TRM$(expr)
Returns string without trailing blanks

VAL(string)
Returns the value of the decimal number contained in the string

Utility Functions

ABORT LET variable=ABORT(expression)
Program deletes itself when it terminates

CTRLC LET variable=CTRLC
Enables the CTRL/C command

RCTRLC LET variable=RCTRLC
Disables CTRL/C and prevents it from interrupting the program

RCTRLO LET variable=RCTRLO
BASIC prints on the terminal even though a CTRL/O is in effect

SYS LET variable=SYS(expression1 [,expression2])
Performs system-dependent operations

TTYSET LET variable=TTYSET(255,expression)
Sets your terminal's margin

FORTRAN IV

Statements

Arithmetic or Logical Assignment
a = b
Assigns value of expression b to variable a

Arithmetic Statement Function
name (a1, . . .)=x
Creates user-defined function having variables a1, . . . as dummy arguments

ACCEPT f,list
Reads input from logical unit 5 (default TT:); f is format statement label; list is optional data list

ASSIGN n TO ivar
Associates statement label n with integer variable ivar

BACKSPACE u
Backspaces one record in file currently open on logical unit number u

BLOCK DATA [nam]
Specifies subprogram that follows as BLOCK DATA subprogram; nam = symbolic name; default name is .DATA

CALL $\left\{\begin{array}{l}\textbf{name}\\ \textbf{name (arg1,arg2, . . .)}\end{array}\right\}$
Calls SUBROUTINE subprogram with name specified, passing actual arguments (arg) to replace dummy arguments in SUBROUTINE definition

CLOSE (UNIT=u[,DISPOSE=p] [,ERR=s])
Disconnects a file from a unit

COMMON/name1/var1, . . . , /name2/var,var
Reserves one or more blocks of storage space under name specified to contain variables (var) associated with that block name

CONTINUE
Causes no processing

DATA var1,var2, . . . /val1,val2, . . . /
Causes elements in list of values (val) to be initially stored in corresponding elements of list of variable names

Statements (Cont.)

DECODE (c,f,v[,ERR=s]) [list]
Changes elements in list of variables from ASCII into desired internal format (c = number of characters, f = format specifier, v = source of characters to be converted, s = statement label for transfer of control on error conditions)

DEFINE FILE u (m,n,U,ivar), . . .
Defines record structure of direct access file

DIMENSION s1,s2, . . . ,sk
Reserves storage space for specified arrays

DO n ivar = e1,e2[,e3]
(1) Sets integer variable (ivar) equal to the expression (e1); (2) executes statements through statement label n; (3) increments ivar = ivar+e3; (4) if e3 > 0 and ivar <= e2, or e3 < 0 and ivar >= e2, goes back to 2 above, or else falls through to statement following statement n

ENCODE (c,f,v[,ERR=s]) list
Changes elements in list of variables into ASCII format (c = number of characters, f = format specifier, v = target of converted ASCII data, s = statement label for transfer of control on error conditions)

END
Delimits a program unit

END FILE u
Writes EOF record in file currently open on logical unit number u

EQUIVALENCE (var1,var2, . . .),(varj,vark, . . .)
Assigns same storage location to each variable within set of parentheses

EXTERNAL name1,name2, . . .
Informs system that names specified are those of FUNCTION or SUBROUTINE subprograms

FIND (u'r)
Positions direct access file on logical unit number u to record r and sets associated variable to record number r

FORMAT (field specification, . . .)
Describes format in which one or more records are to be transmitted

[type] FUNCTION name[*len] [(var1,var2, . . .)]
Begins FUNCTION subprogram, indicating subprogram name and any dummy variable names (var); optional type specifications can be included (*len = data type length specifier)

Unconditional GOTO
GOTOn
Transfers control to statement label n

Computed GOTO
GOTO (k1,k2, . . . kn),e
Transfers control to statement label ki where i = value of expression e; if e < 1 or e > n no transfer takes place

Assigned GOTO
GOTO ivar[,(k1,k2, . . . kn)]
Transfers control to statement most recently associated with ivar by an ASSIGN statement

Arithmetic IF
IF (expression) n1,n2,n3
Transfers control to statement label n depending upon value of the expression

Statements (Cont.)

Logical IF
IF (expression) statement
Executes statement if logical expression tests true

IMPLICIT type (a1,b2-b3, . . .), . . .
Elements a and b represent single (or a range of) letter(s) whose presence as the initial letter of a variable specifies variable to be of that type, if that variable is not explicitly given a type

OPEN (p[,p] . . .)
Connects an existing file to a logical unit, or creates a new file and connects it to a logical unit

PAUSE
PAUSE display
Suspends program execution and prints octal constant, decimal constant, or alphanumeric literal display, if one is specified; resume execution by typing carriage return

PRINT f,list
Writes output on logical unit 6 (default LP:); f is format statement label, list is optional data list

PROGRAM name
Specifies object module name for main program unit

Formatted READ
READ (u,f[,END=s] [,ERR=s2]) [list]
READ f,list
Reads at least one logical record from device u (default 1) according to format specification f and assigns values to the variables in the list

List Directed READ
READ (u,*[,END=s1] [,ERR=s2]) [list]
READ *,list
Reads at least one logical record from device u (default 1) and assigns values to the variables in the list

Unformatted READ
READ (u[,END=s1] [,ERR=s2]) [list]
Reads one logical record from device u, assigning values to variables in list

Direct Access READ
READ (u'r[,END=s1] [,ERR=s2]) [list]
Reads from logical unit number u, record number r, and assigns values to variables in list

Transfer of Control on Error
END=s1
ERR=s2
END=s1,ERR=s2
Optional elements in READ statement allowing control transfer on end of file conditions or error

RETURN
Returns control to calling program from current subprogram

REWIND u
Repositions logical unit number u to beginning of currently opened file

STOP
STOP display
Terminates program execution and prints octal constant, decimal constant, or alphanumeric literal display, if one is specified

SUBROUTINE name
SUBROUTINE name (var1,var2, . . .)
Begins a SUBROUTINE subprogram, indicating subprogram name and any dummy variable names (var)

Statements (Cont.)

TYPE f,list
Writes output on logical unit 7 (default TT:); f is format statement label and list is optional data list

Type Declarations

type var1,var2, . . . vark
Variable names (var) are assigned specified data type in program unit; type = REAL(*4,*8), INTEGER (*2,*4), DOUBLE PRECISION, LOGICAL (*4,*1), or COMPLEX (*8); an optional byte length may be given after the type and value as listed above (e.g., INTEGER*2)

VIRTUAL s1,s2 . . . , sk
Restores storage space in extended memory for arrays s1, s2, through sk

Formatted WRITE

WRITE (u,f[,END=s1] [,ERR=s2])[list]
Causes one or more logical records containing values of variables in list to be written onto logical unit u according to format specification f

List Directed WRITE

WRITE (u,*[,END=s1] [,ERR=s2])[list]
Causes one or more logical records containing values of variables in list to be written onto logical unit u

Unformatted WRITE

WRITE (u[,END=s1] [,ERR=s2])[list]
Causes one or more logical records containing values of variables in list to be written onto logical unit u

Direct Access WRITE

WRITE (u'r[,END=s1] [,ERR=s2])[list]
Causes one logical record containing values of variables in list to be written onto record r of logical unit number u

Transfer of Control on Error

END=s1
ERR=s2
END=s1,ERR=s2
Optional elements in WRITE statement list allowing control transfer on end-of-file conditions or error

Library Functions

(arg type → result type in all cases)

ABS(X)	Real absolute value	Real	→ Real
IABS(I)	Integer absolute value	Integer	→ Integer
DABS(X)	Double precision absolute value	Double	→ Double
CABS(Z)	Complex to Real, absolute value where Z=(x,y); CABS(Z)=$(x^2+y^2)^{1/2}$	Complex	→ Real
FLOAT(I)	Integer to Real conversion	Integer	→ Real
IFIX(X)	Real to Integer conversion IFIX(X) is equivalent to INT(X)	Real	→ Integer
SNGL(X)	Double to Real conversion	Double	→ Real
DBLE(X)	Real to Double conversion	Real	→ Double
REAL(Z)	Complex to Real conversion, obtain real part	Complex	→ Real

Library Functions (Cont.)

AIMAG(Z) Complex to Real conversion, obtain imaginary part	Complex	→ Real
CMPLX(X,Y) Real to Complex conversion CMPLX(X,Y)=X+i*Y	Real	→ Complex

Truncation functions return the sign of the argument * largest integer ⩽ | arg |

AINT(X) Real to Real truncation	Real	→ Real
INT(X) Real to Integer truncation	Real	→ Integer
IDINT(X) Double to Integer truncation	Double	→ Integer

Remainder functions return the remainder when the first argument is divided by the second

AMOD(X,Y) Real remainder	Real	→ Real
MOD(I,J) Integer remainder	Integer	→ Integer
DMOD(X,Y) Double precision remainder	Double	→ Double

Maximum value functions return the largest value from among the argument list; ⩾2 arguments

AMAX0(I,J, . . .) Real maximum from Integer list	Integer	→ Real
AMAX1(X,Y, . . .) Real maximum from Real list	Real	→ Real
MAX0(I,J, . . .) Integer maximum from Integer list	Integer	→ Integer
MAX1(X,Y, . . .) Integer maximum from Real list	Real	→ Integer
DMAX1(X,Y, . . .) Double maximum from Double list	Double	→ Double

Minimum value functions return the smallest value from among the argument list; ⩾2 arguments

AMIN0(I,J, . . .) Real minimum of Integer list	Integer	→ Real
AMIN1(X,Y, . . .) Real minimum of Real list	Real	→ Real
MIN0(I,J, . . .) Integer minimum of Integer list	Integer	→ Integer
MIN1(X,Y, . . .) Integer minimum of Real list	Real	→ Integer
DMIN1(X,Y, . . .) Double minimum of Double list	Double	→ Double

The transfer of sign functions return (sign of the second argument) * (absolute value of first argument)

SIGN(X,Y) Real transfer of sign	Real	→ Real
ISIGN(I,J) Integer transfer of sign	Integer	→ Integer
DSIGN(X,Y) Double precision transfer of sign	Double	→ Double

Positive difference functions return the first argument minus the minimum of the two arguments

DIM(X,Y) Real positive difference	Real	→ Real

Library Functions (Cont.)

IDIM(I,J) Integer positive difference	Integer	→	Integer

Exponential functions return the value of e raised to the argument power

EXP(X) e^X	Real	→	Real
DEXP(X) e^X	Double	→	Double
CEXP(Z) e^Z	Complex	→	Complex
ALOG(X) Returns log$_e$(X)	Real	→	Real
ALOG10(X) Returns log$_{10}$(X)	Real	→	Real
DLOG(X) Returns log$_e$(X)	Double	→	Double
DLOG10(X) Returns log$_{10}$(X)	Double	→	Double
CLOG(Z) Returns log$_e$ of complex argument	Complex	→	Complex
SQRT(X) Square root of Real argument	Real	→	Real
DSQRT(X) Square root of Double precision argument	Double	→	Double
CSQRT(Z) Square root of Complex argument	Complex	→	Complex
SIN(X) Real sine	Real	→	Real
DSIN(X) Double precision sine	Double	→	Double
CSIN(Z) Complex sine	Complex	→	Complex
COS(X) Real cosine	Real	→	Real
DCOS(X) Double precision cosine	Double	→	Double
CCOS(Z) Complex cosine	Complex	→	Complex
TANH(X) Hyperbolic tangent	Real	→	Real
ATAN(X) Real arc tangent	Real	→	Real
DATAN(X) Double precision arc tangent	Double	→	Double
ATAN2(X,Y) Real arc tangent of (X/Y)	Real	→	Real
DATAN2(X,Y) Double precision arc tangent of (X/Y)	Double	→	Double
CONJG(Z) Complex conjugate, if Z=X+i*Y; COMJG(Z)=Z-i+y	Complex	→	Complex

Library Functions (Cont.)

RAN(I,J)
Returns a random number of uniform distribution over the range 0 to 1. I and J must be integer variables and should be set initially to 0. Resetting I and J to 0 regenerates the random number sequence. Alternate starting values for I and J will generate different random number sequences. Integer → Real

System Subroutines

CALL ASSIGN (lun,name,icnt,mode,control,numbuf)
Allows the association of device and/or file name information with a logical unit number

CALL CLOSE (lun)
Closes the file currently open on logical unit lun and releases the unit for subsequent use with other files

CALL DATE (array)
Returns the current date in array as nine ASCII characters

CALL ERRTST (ierr,ires)
Allows the user program to monitor the types of errors detected during program execution

CALL ERRSNS (ires,iunit)
Allows the user program to obtain information about the most recent error that has occurred during program execution

CALL EXIT
Equivalent to the STOP statement without message

CALL IDATE (mon,iday,iyear)
Returns three integer values representing the current month, day, and year

RAN (i1,i2)
A function call to the random number generator

CALL RANDU(i1,i2,x)
A subroutine call to the random number generator

CALL SETERR (number,ncount)
Allows you to specify the disposition of OTS-detected errors

CALL USEREX (name)
Passes control to routine name as part of program termination

Logical Device Assignments

Logical Unit Number	Default Device
1	SY:
2	DK:
3	DK:
4	DK:
5	TT:(input)
6	LP:
7	TT:(output)
8	PC:
9	PC:

Options

/A
Includes compilation statistics

/B
Enables expanded listings of compiler diagnostic information

/D
Compiles lines with D in column one

/E
Reads a full 80 columns of each source program line

/H
Prints a list of compiler options

/I:cod
Selects type of object code to be generated; cod = EAE, EIS, FIS, or THR

/L:n
Specifies listing options; n = any one or sum of the following:

0	lists diagnostics only
1	lists source program only (/L:SRC)
2	lists storage map only (/L:MAP)
4	lists generated code only (/L:COD)

/M:xxx
Disables certain optimizations; xxx = SPD, CSE, STR, BND

/N:m
Specifies maximum number of logical units that can be concurrently open (m = octal constant between 1 and 17)

/O
Includes options-in-effect

/P:xxx
Selects optimization level; xxx = SPD, CSE, STR, or BND

/Q
Disables logging

/R:m
Specifies maximum record size allowed (m = octal constant between 4 and 7777) at execution time

/S
Suppresses internal statement numbers

/T
Allocates two words for default length of integer variables

/U
Disables USR swapping at run time

/V
Disables vectoring of arrays

/W
Enables compiler warning diagnostics

/X:sys
Indicates cross-compilation for the target environment (sys = RT, RST, RSX)

/Z
Causes pure code and data sections to take RO (readonly) attribute

FORMAT OF PROGRAMMED REQUESTS

.CDFN area, addr, num
Defines new I/O channels

.CHAIN
Allows background program to transfer control to another background program that is specified in locations 500-507(RAD50) without operator intervention; saves words 500-777

.CHCOPY area, chan, ochan
Opens a channel for input and logically connects it to a channel on another job open for either input or output

.CLOSE chan
Terminates activity on the specified channel and frees it for use in another operation

.CMKT area, id, time
Cancels one or more outstanding mark time requests; id is the .MRKT id to be cancelled and time is a pointer to two words where monitor is to store number of ticks remaining in the request

.CNTXSW area, addr
Specifies locations to be included in context switch; addr is terminated with a 0 word; valid locations are 2-476, user job area, and 160000-177776

.CRAW area ,addr
Defines a virtual address window and optionally maps it into a physical memory region

.CRRG area ,addr
Allocates a dynamic region in physical memory for use by the current requesting program

.CSIGEN devspc, defext, cstring [,linbuf]
Calls CSI in general mode and loads handlers in devspc; if cstring is 0, gets command string from terminal; returns command line to you if you specify linbuf

.CSISPC outspc, defext, cstring [,linbuf]
Calls CSI in special mode and returns file specifications in outspc; if cstring is 0, gets string from terminal; returns command line to you if you specify linbuf

.CSTAT area, chan, addr
Furnishes 6 information words about channel:

word 1	channel status
word 2	file starting block
word 3	file length
word 4	highest block written
word 5	device unit number
word 6	device name (RAD50)

.DATE
Moves current date word into R0:

bits 14-10	month (1-12.)
bits 9-5	day (1-31.)
bits 4-0	year (72.-99.)

.DELETE area, chan, dblk, seqnum
Deletes named file from indicated device

.DEVICE area, addr [,link]
Sets up list of addresses to be loaded with specified values upon program termination; creates a linked list if you specify link

.DSTATUS retspc, dnam
Provides information about device characteristics:

word 1	device status
word 2	handler size
word 3	handler entry point
word 4	device size

.ELAW area ,addr
Cancels a defined window and permits redefinition

FORMAT OF PROGRAMMED REQUESTS (Cont.)

.ELRG area ,addr
Deallocates a dynamic memory region and returns it to the free list

.ENTER area, chan, dblk, len, seqnum
Allocates space on specified device and creates tentative directory entry for the named file

.EXIT
Terminates user background program and returns control to the monitor, when used from a background program under FB; when used under SJ, causes KMON to run in background area

.FETCH addr, dnam
Loads device handler into memory from the system device

.GMCX area ,addr
Returns the mapping status of a specified window

.GTIM area, addr
Returns current time of day in ticks past midnight

.GTJB area, addr
Passes certain job parameters back to the user program:

word 1	job number (0=B; 2=F)
word 2	high memory limit
word 3	low memory limit
word 4	start of I/O channel space
word 5	address of job's impure area with FB and XM monitors
words 6-8	reserved

.GTLIN linbuf, prompt
Obtains command information other than CSI format strings and allows the program to operate through indirect files

.GVAL area, offse
Returns a monitor fixed offset value in R0

.HERR
Disables user error interception and allows the monitor to detect and act on fatal errors

.HRESET
Resets channels, releases device handlers, and stops all I/O transfers in progress

.INTEN prio [,pic]
Notifies monitor that an interrupt occurred and switches to "system state"; sets processor priority to the correct level

.LOCK
Locks USR in memory

.LOOKUP area, chan, dblk, seqnum
Associates specified channel with a device and existing file

.MAP area ,addr
Maps a previously defined address window into a region of extended memory

.MFPS addr
Reads priority bits from the processor word

.MRKT area, time, crtn, id
Schedules completion routine to be entered after a specified time interval (number of ticks)

.MTATCH area, addr, unit
Attaches a terminal for exclusive use by the requesting job

.MTDTCH area, unit
Detaches a terminal from one job and makes it available for other jobs

.MTGET area, addr, unit
Returns the status of the specified terminal to the caller

FORMAT OF PROGRAMMED REQUESTS (Cont.)

.MTIN area, addr, unit[,chrcnt]
A multi-terminal form of .TTYIN; transfers one character to your buffer

.MTOUT area, addr, unit[,chrcnt]
A multi-terminal form of .TTYOUT; prints one character from your buffer

.MTPRN area, addr, unit
A multi-terminal form of .PRINT

.MTPS addr
Sets priority bits, condition codes, and T bit in the processor status word

.MTRCTO area, unit
Resets CTRL/O for the specified terminal

.MTSET area, addr, unit
Allows the user program to set terminal and line characteristics

.MWAIT
Suspends execution until all messages are transmitted or received

.PRINT addr
Outputs an ASCII string to the terminal

.PROTECT area, addr
Used by a job to obtain exclusive control of a vector pair in the range 0-476

.PURGE chan
Deactivates a channel without closing the file (tentative output file is lost)

.QSET addr, len
Enlarges I/O queue for the monitor

.RCTRLO
Enables console terminal printing (resets CTRL/O)

.RCVD area, buf, wcnt
Posts a request to receive message and continues execution

.RCVDC area, buf, wcnt, crtn
Posts a request to receive message and enters completion routine when message is received

.RCVDW area, buf, wcnt
Posts a request to receive message and waits until it is received

.READ area, chan, buf, wcnt, blk
Initiates transfer of words from specified channel into memory and continues execution

.READC area, chan, buf, wcnt, crtn, blk
Initiates transfer from channel to memory; continues executing user program; enters specified routine when transfer completes

.READW area, chan, buf, wcnt, blk
Transfers words from specified channel into memory; returns control to user program when the transfer completes or when an error is detected

.RELEAS dnam
Removes a device handler from memory

.RENAME area, chan, dblk
Changes a file name

.REOPEN area, chan, cblk
Reassociates a channel with a file on which a SAVESTATUS was performed

.RSUM
Resumes execution of a foreground job after it was suspended

.SAVESTATUS area, chan, cblk

Stores 5 words (containing data concerning file definition) into memory; frees channel for use:

word 1	channel status
word 2	starting block of file
word 3	length of file
word 4	reserved
word 5	even byte — I/O count; odd byte — device unit number

.SCCA area, addr

Inhibits CTRL/C abort; indicates that CTRL/C was initiated at the keyboard; distinguishes between single and double CTRL/C

.SDAT area, buf, wcnt

Initiates message transfer; returns control to user program immediately

.SDATC area, buf, wcnt, crtn

Initiates message transfers: transfers control to specified routine when message is received

.SDATW area, buf, wcnt

Initiates message transfer; returns control to user program when message is received

.SERR

Inhibits monitor from aborting jobs after fatal errors

.SETTOP addr

Requests additional memory for program and returns the highest address available in R0

.SFPA area, addr

Sets user interrupt for floating point processor exceptions

.SPFUN area, chan, func, buf, wcnt, blk,[,crtn]

Provides special device-dependent functions to magtape, cassette, and diskette

.SPND

Suspends a foreground job

.SRESET

Resets certain memory areas, dismisses device handlers, purges currently open files, resets to 16 channels, resets I/O queue to one element

.SYNCH area[,pic]

Enables the user program to perform monitor programmed requests from within an interrupt service routine

.TLOCK

Attempts to gain ownership of USR; if unsuccessful, returns control with C bit set

.TRPSET area, addr

Allows user job to intercept traps to 4 and 10

.TTINR

Inputs a character from the terminal; returns if none available

.TTOUTR

Outputs a character to the terminal; returns if no room in buffers

.TTYIN char

Inputs a character from the terminal and waits until operation is done

.TTYOUT char

Outputs a character to the terminal and waits until operation is done

.TWAIT area, time

Suspends the running job for the specified amount of time (number of ticks); requires queue element

.UNLOCK

Releases USR from memory

FORMAT OF PROGRAMMED REQUESTS (Cont.)

.UNMAP area ,addr
Unmaps a window and flags that portion of PVAS as being inaccessible

.UNPROTECT area, addr
Cancels any protected vectors in the range 0-476

.WAIT chan
Suspends program execution until channel I/O completes

.WRITC area, chan, buf, wcnt, crtn, blk
Transfers words from memory to specified channel; when complete, passes control to specified routine

.WRITE area, chan, buf, wcnt, blk
Initiates transfer from memory to channel; returns control to user program immediately

.WRITW area, chan, buf, wcnt, blk
Transfers words from memory to channel; when transfer is complete, returns control to user program

BOOTSTRAPS

Unit 0 Bootstraps

		Contents	
		RP02/RP03	RJS03/4
Address	RX01 Disk	Disk	Disk
001000	012702	012705	012705
001002	1002n7*	176716	172044
001004	012701	012715	012745
001006	177170	177400	177400
001010	130211	012745	012745
001012	001776	000005	000071
001014	112703	105715	032715
001016	000007	100376	100200
001020	010100	005007	001775
001022	010220		100762
001024	000402		005007
001026	012710		
001030	000001		
001032	006203		
001034	103402		
001036	112711		
001040	111023		
001042	030211		
001044	001776		
001046	100756		
001050	103766		
001052	105711		
001054	100771		
001056	005000		
001060	022710		
001062	000240		
001064	001347		
001066	122702		
001070	000247		
001072	005500		
001074	005007		

*n = 4 for unit 0
6 for unit 1

BOOTSTRAPS (Cont.)

Unit 0 Bootstraps (Cont.)

	Contents	
Address	TJU16 Magtape	TM11 Magtape
010000	012700	012700
010002	172440	172524
010004	012710	005310
010006	000021	012740
010010	012760	060011
010012	001300	105710
010014	000032	100376
010016	012760	005710
010020	177777	100767
010022	000006	012710
010024	012720	060003
010026	000031	105710
010030	105760	100376
010032	000010	005710
010034	100375	100777
010036	012710	005007
010040	177000	
010042	012740	
010044	000071	
010046	032710	
010050	100200	
010052	001775	
010054	100007	
010056	022760	
010060	001000	
010062	000014	
010064	001403	
010066	000005	
010070	000167	
010072	177704	
010074	005007	

		Contents	
Address	DECtape	RK11 Disk	RF11 Disk
001000	012700	012700	012700
001002	177344	177406	177466
001004	012710	012710	005010
001006	177400	177400	005040
001010	012740	012740	012740
001012	004002	000005	177400
001014	005710	105710	012740
001016	100376	100376	000005
001020	012710	005007	105710
001022	000003		100376
001024	105710		005007
001026	100376		
001030	012710		
001032	000005		
001034	105710		
001036	100376		
001040	005007		

BOOTSTRAPS (Cont.)

Unit n Bootstrap RK11 Disk

Address	Contents
001000	012700
001002	177406
001004	012760
001006	xxxxxx
001010	000004
001012	012700
001014	177406
001016	012710
001020	177400
001022	012740
001024	000005
001026	105710
001030	100376
001032	005007

xxxxxx=	020000	unit 1
	040000	unit 2
	060000	unit 3
	100000	unit 4
	120000	unit 5
	140000	unit 6
	160000	unit 7

1. Enter 001000 in switch register
2. Press LOAD ADRS
3. Deposit the appropriate bootstrap from the preceding tables
4. Enter 001000 in switch register and examine each location for correctness
5. Ensure that device is ready
6. Enter 001000 in switch register
7. Press LOAD ADRS and START

RK06 Bootstrap Loader

Address	Contents
001000	012701
001002	177440
001004	012711
001006	000003
001010	032711
001012	100200
001014	001775
001016	012761
001020	177400
001022	000002
001024	012711
001026	000021
001030	032711
001032	100200
001034	001775
001036	005007

Hardware Bootstrap

If you have a hardware bootstrap, write its load address and switch setting (if any) here

LOAD ADRS:
SWITCH SETTING:
START

FORMAT OF SYSTEM SUBROUTINE LIBRARY CALLS

File-Oriented Operations

CLOSEC CALL CLOSEC (chan)
Closes the specified channel

IDELET i=IDELET (chan,dblk[,seqnum])
Deletes the file from the specified device

IENTER i=IENTER (chan,dblk,length[,seqnum])
Creates a new file for output

IRENAM i=IRENAM (chan,dblk)
Changes the name of the indicated file to a new name

LOOKUP i=LOOKUP (chan,dblk[,count])
Opens an existing file for I/O with the specified channel

Data Transfer Functions

IRCVD i=IRCVD (buff,wcnt)
Receives data and returns to user program immediately

IRCVDC i=IRCVDC (buff,wcnt,crtn)
Receives data and enters an assembly language completion routine

IRCVDF i=IRCVDF (buff,wcnt,area,crtn)
Receives data and enters a FORTRAN subprogram

IRCVDW i=IRCVDW (buff,wcnt)
Receives data, waits for completion, and returns to the user program

IREAD i=IREAD (wcnt,buff,blk,chan)
Transfers data on the specified channel to a memory buffer and returns control to the user program

IREADC i=IREADC (wcnt,buff,blk,chan,crtn)
Transfers data on the specified channel to a memory buffer and enters an assembly language completion routine

IREADF i=IREADF (wcnt,buff,blk,chan,area,crtn)
Transfers data on the specified channel to a memory buffer and enters a FORTRAN subprogram

IREADW i=IREADW (wcnt,buff,blk,chan)
Transfers data on the specified channel to a memory buffer and returns control to the user program only after the transfer is complete

ISDAT i=ISDAT (buff,wcnt)
Transfers the specified number of words from one job to the other and returns control immediately to the user program

ISDATC i=ISDATC (buff,wcnt,crtn)
Transfers the specified number of words from one job to the other and returns control to the user program immediately

ISDATF i=ISDATF (buff,wcnt,area,crtn)
Transfers the specified number of words from one job to the other and enters a FORTRAN subprogram

ISDATW i=ISDATW (buff,wcnt)
Transfers the specified number of words from one job to the other and returns control to the user program only after the transfer is complete

ITTINR i=ITTINR ( )
Inputs one character from the console terminal

ITTOUR i=ITTOUR (char)
Transfers one character to the console terminal

IWAIT i=IWAIT (chan)
Waits for completion of all I/O on the specified channel; completion routines continue to run

FORMAT OF SYSTEM SUBROUTINE LIBRARY CALLS (Cont.)

Data Transfer Functions (Cont.)

IWRITC i=IWRITC (wcnt,buff,blk,chan,crtn)
Transfers data on the specified channel to a device and enters an assembly language completion routine

IWRITE i=IWRITE (wcnt,buff,blk,chan)
Transfers data on the specified channel to a device and returns control to the user program immediately

IWRITF i=IWRITF (wcnt,buff,blk,chan,area,crtn)
Transfers data on the specified channel to a device and enters a FORTRAN subprogram

IWRITW i=IWRITW (wcnt,buff,blk,chan)
Transfers data on the specified channel to a device and returns control to the user program only after the transfer is complete

MWAIT CALL MWAIT
Waits for messages to be processed

PRINT CALL PRINT (string)
Outputs an ASCII string to the terminal

Channel-Oriented Operations

ICDFN i=ICDFN (num)
Defines additional channels for I/O

ICHCPY i=ICHCPY (chan,ochan)
Allows access to files currently open in the other job's environment

ICSTAT i=ICSTAT (chan,addr)
Returns the status of a specified channel

IFREEC i=IFREEC (chan)
Returns the specified RT-11 channel to the available pool of channels for the FORTRAN I/O system

IGETC i=IGETC ( )
Allocates an RT-11 channel and marks it in use for the FORTRAN I/O system

ILUN i=ILUN (lun)
Returns the RT-11 channel number with which the FORTRAN logical unit is associated

IREOPN i=IREOPN (chan,cblk)
Restores the parameters stored with an ISAVES function and reopens the channel for I/O

ISAVES i=ISAVES (chan,cblk)
Stores five words of channel status information into an array

PURGE CALL PURGE (chan)
Deactivates a channel without performing an ISAVES or CLOSEC; tentative files are lost

Device and File Specifications

IASIGN i=IASIGN (lun,idev[,ifilex[,isize[,itype]]])
Sets information in the FORTRAN logical unit table

ICSI i=ICSI (filspc,deftyp[,cstring] ,[option] ,n)
Calls the RT-11 CSI in special mode to decode file specifications and options

Timer Support Operations

CVTTIM CALL CVTTIM (time,hrs,min,sec,tick)
Converts a 2-word internal format time to hours, minutes, seconds, and ticks

GTIM CALL GTIM (itime)
Gets time of day

ICMKT i=ICMKT (id,time)
Cancels an unexpired mark time request

Timer Support Operations (Cont.)

ISCHED i=ISCHED (hrs,min,sec,tick,area,id,crtn)
Schedules the specified FORTRAN subroutine to be entered at the specified time of day as an asynchronous completion routine

ISLEEP i=ISLEEP (hrs,min,sec,tick)
Suspends mainline execution of the running job for a specified amount of time; completion routines continue to run

ITIMER i=ITIMER (hrs,min,sec,tick,area,id,crtn)
Schedules the specified FORTRAN subroutine to be entered as an asynchronous completion routine when the interval specified has elapsed

ITWAIT i=ITWAIT (itime)
Suspends the running job for a specified amount of time; completion routines continue to run

IUNTIL i=IUNTIL (hrs,min,sec,tick)
Suspends the mainline execution of the running job until a specified time of day; completion routines continue to run

JTIME CALL JTIME (hrs,min,sec,tick,time)
Converts hours, minutes, seconds, and ticks into 2-word internal format time

MRKT i=MRKT (id,crtn,time)
Mark time — i.e., schedules an asynchronous routine to be entered after a specified interval

SECNDS a=SECNDS (atime)
Returns the current system time in seconds past midnight minus a specified time

TIMASC CALL TIMASC (itime,strng)
Converts a specified 2-word internal format time into an 8-character ASCII string

TIME CALL TIME (strng)
Returns the current system time of day as an 8-character ASCII string

RT-11 Services

CHAIN CALL CHAIN (dblk,var,wcnt)
Chains to another program (in the background job only)

DEVICE CALL DEVICE (ilist[,link])
Specifies actions to be taken on normal or abnormal program termination

GTJB CALL GTJB (addr)
Returns the parameters of this job

IDSTAT i=IDSTAT (devnam,cblk)
Returns the status of the specified device

IFETCH i=IFETCH (devnam)
Loads device handlers into memory

IQSET i=IQSET (qleng)
Expands the size of the RT-11 monitor queue from the free space managed by the FORTRAN system

ISPFN i=ISPFN (code,chan[,wcnt,buff,blk])
Queues the specified operation and returns control to the user program immediately

ISPFNC i=ISPFNC (code,chan,wcnt,buff,blk,crtn)
Queues the specified operation and enters an assembly language completion routine

ISPFNF i=ISPFNF (code,chan,wcnt,buff,blk,area,crtn)
Queues the specified operation and enters a FORTRAN subprogram

FORMAT OF SYSTEM SUBROUTINE LIBRARY CALLS (Cont.)

RT-11 Services (Cont.)

ISPFNW i=ISPFNW (code,chan[,wcnt,buff,blk])
Queues the specified operation and returns control to the user program only after the operation completes

ITLOCK i=ITLOCK ( )
Indicates whether the USR is currently in use by another job and performs a LOCK if possible

LOCK CALL LOCK
Makes the RT-11 monitor USR permanently resident until an UNLOCK is executed

RCHAIN CALL RCHAIN (flag,var,wcnt)
Allows a program to access variables passed across a chain

RCTRLO CALL RCTRLO
Enables output to the terminal by cancelling the effect of a previously typed CTRL/O, if any

RESUME CALL RESUME
Causes the mainline execution of a job to resume after it was suspended with a SUSPND call

SCCA CALL SCCA [(iflag)]
Inhibits a CTRL/C abort; indicates that a CTRL/C is active; distinguishes between a single and double CTRL/C

SETCMD CALL SETCMD (string)
Allows a user program to pass a command line to the keyboard monitor to be executed after the program exits

SUSPND CALL SUSPND
Suspends mainline execution of the running job; completion routines continue to run

UNLOCK CALL UNLOCK
Releases the USR if a LOCK was performed

*INTEGER*4 Support Functions*

AJFLT a=AJFLT (jsrc)
Converts the specified INTEGER*4 value to REAL*4 and returns the result as a function value

DJFLT d=DJFLT (jsrc)
Converts the specified INTEGER*4 value to REAL*8 and returns the result as a function value

IAJFLT i=IAJFLT (jsrc,ares)
Converts the specified INTEGER*4 value to REAL*4 and stores the result

IDJFLT i=IDJFLT (jsrc,dres)
Converts the specified INTEGER*4 value to REAL*8 and stores the result

IJCVT i=IJCVT (jsrc[,ires])
Converts the specified INTEGER*4 value to INTEGER*2

JADD i=JADD (jopr1,jopr2,jres)
Computes the sum of two INTEGER*4 values

JAFIX i=JAFIX (asrc,jres)
Converts a REAL*4 value to INTEGER*4

JCMP i=JCMP (jopr1,jopr2)
Compares two INTEGER*4 values and returns an INTEGER*2 value that reflects the signed comparison result

JDFIX i=JDFIX (dsrc,jres)
Converts a REAL*8 value to INTEGER*4

JDIV i=JDIV (jopr1,jopr2,jres[,jrem])
Computes the quotient of two INTEGER*4 values

FORMAT OF SYSTEM SUBROUTINE LIBRARY CALLS (Cont.)

*INTEGER*4 Support Functions (Cont.)*

JICVT i=JICVT (isrc,jres)
Converts an INTEGER*2 value to INTEGER*4

JJCVT CALL JJCVT (jsrc)
Converts 2-word internal time formats to INTEGER*4 format, and vice versa

JMOV i=JMOV (jsrc,jdest)
Assigns an INTEGER*4 value to a variable

JMUL i=JMUL (jopr1,jopr2,jres)
Computes the product of two INTEGER*4 values

JSUB i=JSUB (jopr1,jopr2,jres)
Computes the difference between two INTEGER*4 values

Character String Functions

CONCAT CALL CONCAT (a,b,out[,len[,err]])
Concatenates two variable-length strings

GETSTR CALL GETSTR (lun,out,len,err)
Reads a character string from a specified FORTRAN logical unit

GTLIN CALL GTLIN (result[,prompt])
Transfers a line of input from the terminal or an indirect file to the user program

INDEX CALL INDEX (a,pattrn[,i],m) or m=INDEX (a,pattrn[,i])
Returns in m the starting location of string pattern occurring in string source, following location i

INSERT CALL INSERT (in,out,i[,m])
Inserts a string at a specified position in another string

LEN i=LEN (a)
Returns the number of characters in string a

PUTSTR CALL PUTSTR (lun,in,char,err)
Writes a variable-length character string to a specified FORTRAN logical unit

REPEAT CALL REPEAT (in,out,i[,len[,err]])
Concatenates a specified string with itself to produce an indicated number of copies, and stores the resultant string in an array

SCOMP CALL SCOMP (a,b,i) or i=ISCOMP (a,b)
Compares two character strings and returns the integer result of the comparison

SCOPY CALL SCOPY (in,out[,len[,err]])
Copies a character string from one array to another

STRPAD CALL STRPAD (a,i[,err])
Pads a variable-length string on the right with blanks to create a new string of a specified length

SUBSTR CALL SUBSTR (in,out,i[,len])
Copies a substring from a specified string

TRANSL CALL TRANSL (in,out,r[,p])
Replaces one string with another after performing character modification

TRIM CALL TRIM (a)
Removes trailing blanks from a character string

VERIFY CALL VERIFY (a,b,i) or i=IVERIF (a,b)
Determines whether each character of a specified string occurs anywhere in another string

FORMAT OF SYSTEM SUBROUTINE LIBRARY CALLS (Cont.)

Radix-50 Conversion Operations

IRAD50 n=IRAD50 (icnt,input,output)
Converts ASCII characters to Radix-50 and returns the number of characters converted

R50ASC CALL R50ASC (icnt,input,output)
Converts Radix-50 characters to ASCII

RAD50 a=RAD50 (input)
Converts six ASCII characters and returns a REAL*4 result that is the 2 word Radix-50 value

Multi-terminal Operations

MTATCH (unit,addr)
Attaches a specific terminal

MTDTCH (unit)
Detaches a specific terminal

MTGET i=MTGET (unit,addr)
Gets status information about a specific terminal in a multi-terminal system

MTIN i=MTIN (unit,char[,chrcnt])
Transfers a character from a specific terminal to the user program

MTOUT i=MTOUT (unit,char[,chrcnt])
Transfers a character to a specific terminal

MTPRNT (unit,string)
Outputs an ASCII string to a specific terminal

MTRCTO (unit)
Resets CTRL/O for a specific terminal

MTSET i=MTSET (unit,addr)
Sets status information for a specific terminal

Miscellaneous Services

IADDR i=IADDR (arg)
Obtains the 16-bit absolute memory address of the specified argument

IGETSP i=IGETSP (min,max,iaddr)
Gets the address and size of some free space from the FORTRAN system

INTSET i=INTSET (vect,pri,id,crtn)
Establishes a specific FORTRAN subroutine as an interrupt completion routine at a specified priority

IPEEK i=IPEEK (iaddr)
Returns the contents of the word located at the specified absolute 16-bit memory address

IPEEKB i=IPEEKB (iaddr)
Returns the contents of the byte located at a specified absolute memory byte address

IPOKE CALL IPOKE (iaddr,ivalue)
Stores a specified 16-bit integer value into a specified absolute memory location

IPOKEB i=IPOKEB (iaddr,ivalue)
Stores a specified 8-bit integer value into a specified byte location

ISPY i=ISPY (ioff)
Returns the integer value of the word at a specified offset from the RT-11 resident monitor

```
APL
```

```
ASSIGN (SP) physical-device-name (SP) logical-device-name
```

```
B[ (SP) address]
```

BASIC

BOOT (SP) filespec

CLOSE

```
COMPILE [   /LIST[:filespec] [/ALLOCATE:size]                ] (SP) filespecs [/LIBRARY]
        [   /[NO] OBJECT[:filespec] [/ALLOCATE:size]         ]                [/PASS:1 ]
        [                                                    ]
        [ ( /DIBOL                                        )  ]
        [ (  [/ALPHABETIZE       ]                        )  ]
        [ (  [/CROSSREFERENCE    ]                        )  ]
        [ (  [/[NO] LINENUMBERS  ]                        )  ]
        [ (  [/ONDEBUG           ]                        )  ]
        [ (  [/[NO] WARNINGS     ]                        )  ]
        [ ( /FORTRAN                                      )  ]
        [ (  [/CODE:type         ]                        )  ]
        [ (  [/DIAGNOSE          ]                        )  ]
        [ (  [/EXTEND            ]                        )  ]
        [ (  [/HEADER            ]                        )  ]
        [ (  [/I4                ]                        )  ]
```

```
/[NO] LINENUMBERS
/ONDEBUG
/[NO] OPTIMIZE [:type]
/RECORD:length
/SHOW [:value]
/STATISTICS
/[NO] SWAP
/UNITS:n
/[NO] VECTORS
/WARNINGS
/MACRO
 /CROSSREFERENCE [:type [...:type] ]
 /DISABLE:value [...:value]
 /ENABLE:value [...:value]
 /[NO] SHOW:value
```

```
COPY [ /BOOT          ] (SP) input-filespecs [ (/DOS[/OWNER:[nnn,nnn]] ) ] (SP) output-filespec [ (/ALLOCATE:size      ) ]
     [                ]                      [ (/INTERCHANGE          ) ]                       [ (/DOS               ) ]
     [ /DEVICE        ]                      [ (/POSITION:n           ) ]                       [ (/INTERCHANGE[:size]) ]
     [                ]                      [ (/TOPS                 ) ]                       [ (/POSITION:n        ) ]
     [ {/ASCII  }     ]
     [ {/BINARY }     ]
     [ {/IMAGE  }     ]
     [ /CONCATENATE   ]
     [ /EXCLUDE       ]
     [ /IGNORE        ]
     [ /[NO] LOG      ]
     [ /NEWFILES      ]
```

```
        /PACKED
        /PREDELETE
        /[NO] QUERY
        /[NO] REPLACE
        /SETDATE
        /SLOWLY
        /SYSTEM
```

```
D (SP) address=value[, . . . value]
```

DATE [SP dd-mmm-yy]

DEASSIGN [SP logical-device-name]

```
DELETE ⎡/DOS           ⎤ (SP) filespecs
       ⎢               ⎥
       ⎢/INTERCHANGE   ⎥
       ⎢               ⎥
       ⎢/EXCLUDE       ⎥
       ⎢/LOG           ⎥
       ⎢/NEWFILES      ⎥
       ⎢/POSITION:n    ⎥
       ⎢/[NO] QUERY    ⎥
       ⎣/SYSTEM        ⎦
```

```
DIBOL ┌/LIST[:filespec] [/ALLOCATE:size]                   ┐ (SP) filespecs
      │/[NO] OBJECT[:filespec] [/ALLOCATE:size]             │
      │                                                     │
      │/ALPHABETIZE                                         │
      │/CROSSREFERENCE                                      │
      │/[NO] LINENUMBERS                                    │
      │/ONDEBUG                                             │
      └/[NO] WARNINGS                                       ┘
```

```
DIFFERENCES [ { /OUTPUT:filespec[/ALLOCATE:size] }  ] (SP) filespec 1,filespec 2
              { /PRINTER                         }
              { /TERMINAL                        }

                /BLANKLINES
                /[NO] COMMENTS
                /FORMFEED
                /MATCH:n
                /[NO] SPACES
```

```
DIRECTORY [ (/OUTPUT:filespec[/ALLOCATE:size])  ] [(SP) filespecs[/BEGIN]]
            {/PRINTER                         }
            (/TERMINAL                         )

             /BADBLOCKS[/FILES]

             /DOS[/OWNER:[nnn,nnn]]

             /INTERCHANGE

             /TOPS

             /VOLUMEID

            (/BEFORE[date])
            {/DATE[date]  }
            {/NEWFILES    }
            (/SINCE[date] )
```

```
[ ( /ALPHABETIZE[/REVERSE]            )
  { /ORDER[:category] [/REVERSE]      }
  ( /SORT[:category] [/REVERSE]       )
    /BLOCKS
    /BRIEF
    /COLUMNS:n
    /DELETED
    /EXCLUDE
    /FAST
    /FREE
    /FULL
    /OCTAL
    /POSITION
    /SUMMARY                           ]
```

```
DUMP [ (/OUTPUT:filespec[/ALLOCATE:size])  ] (SP) filespec
     [ {/PRINTER                         }  ]
     [ (/TERMINAL                        )  ]
     [                                      ]
     [  /[NO] ASCII                         ]
     [  /BYTES                              ]
     [  /IGNORE                             ]
     [  /ONLY:block                         ]
     [  /RAD50                              ]
     [  [/START:block] [/END:block]         ]
     [  /WORDS                              ]
```

```
E (SP) address[-address]
```

```
EDIT [ ( /CREATE                                    ) ] (SP) filespec[/ALLOCATE:size]
     [ { /INSPECT                                   } ]
     [ ( /OUTPUT:filespec[/ALLOCATE:size]           ) ]
```

```
EXECUTE ┌  /EXECUTE[:filespec] [/ALLOCATE:size]           ┐ (SP) filespecs ┌/LIBRARY┐
        │  /LIST[:filespec] [/ALLOCATE:size]              │                └/PASS:1 ┘
        │  /MAP[:filespec] [/ALLOCATE:size] [/WIDE]       │
        │  /OBJECT[:filespec] [/ALLOCATE:size]            │
        │                                                 │
        │  /BOTTOM:n                                      │
        │  /DEBUG[:filespec]                              │
        │  /LINKLIBRARY[:filespec]                        │
        │  /[NO] RUN                                      │
        │                                                 │
        │ ┌/DIBOL                                   ┐     │
        │ │┌/ALPHABETIZE              ┐           │     │
        │ ││/CROSSREFERENCE           │           │     │
        │ ││/[NO] LINENUMBERS         │           │     │
        │ ││/ONDEBUG                  │           │     │
        │ │└/[NO] WARNINGS            ┘           │     │
        │ │/FORTRAN                               │     │
        │ │┌/CODE:type                ┐           │     │
        │ ││/DIAGNOSE                 │           │     │
```

```
       ⎡ /EXTEND                             ⎤
       ⎢ /HEADER                             ⎥
       ⎢ /I4                                 ⎥
       ⎢ /[NO] LINENUMBERS                   ⎥
       ⎢ /ONDEBUG                            ⎥
       ⎢ /[NO] OPTIMIZE [:type]              ⎥
       ⎢ /RECORD:length                      ⎥
       ⎢ /SHOW [:value]                      ⎥
       ⎢ /STATISTICS                         ⎥
       ⎢ /[NO] SWAP                          ⎥
       ⎢ /UNITS:n                            ⎥
       ⎢ /[NO] VECTORS                       ⎥
       ⎣ /WARNINGS                           ⎦
     /MACRO
       ⎡ /CROSSREFERENCE [:type [...:type]   ⎤
       ⎢ /DISABLE:value [...:value]          ⎥
       ⎢ /ENABLE:value [...:value]           ⎥
       ⎣ /[NO] SHOW:value                    ⎦
```

```
FOCAL
```

```
FORTRAN ⎡/LIST[:filespec] [/ALLOCATE:size]            ⎤ (SP) filespecs
        ⎢/[NO] OBJECT[:filespec] [/ALLOCATE:size]     ⎥
        ⎢                                             ⎥
        ⎢/CODE:type                                   ⎥
        ⎢/DIAGNOSE                                    ⎥
        ⎢/EXTEND                                      ⎥
```

/HEADER
/I4
/[NO] LINENUMBERS
/ONDEBUG
/[NO] OPTIMIZE[:type]
/RECORD:length
/SHOW[:value]
/STATISTICS
/[NO] SWAP
/UNITS:n
/[NO] VECTORS
/WARNINGS

FRUN (SP) filespec [/N:n
/P
/T:n]

GET (SP) filespec

```
GT (SP) ⎧ OFF          ⎫
        ⎨              ⎬
        ⎩ ON [ /L:n ]  ⎭
             [ /T:n ]
```

```
HELP [ ⎧ /PRINTER  ⎫ ] [ (SP) topic[ (SP) subtopic[:item] ] ]
       ⎩ /TERMINAL ⎭
```

```
INITIALIZE [ /DOS[/[NO] QUERY]                   ] (SP) device
           [                                     ]
           [ /FILE:filespec                      ]
           [                                     ]
           [ /INTERCHANGE[/[NO] QUERY]           ]
           [                                     ]
           [ /[NO] QUERY                         ]
           [ /VOLUMEID[:ONLY]                    ]
           [ /SEGMENTS:n                         ]
           [{/REPLACE[:RETAIN]}                  ]
           [{/BADBLOCKS       }                  ]
```

```
INSTALL (SP) device[ , . . . device]
```

```
LIBRARY[ /LIST[:filespec] [/ALLOCATE:size]            ] (SP) library[ (SP) filespecs [ { /REPLACE } ] ]
       [ /[NO] OBJECT[:filespec] [/ALLOCATE:size]     ]             [                 [ {          } ] ]
       [                                              ]             [                 [ { /UPDATE  } ] ]
       [ [ { /CREATE  } ]                             ]
       [ [ { /EXTRACT } ]                             ]
       [ [ { /INSERT  } ]                             ]
       [ [ { /MACRO   } ]                             ]
       [   /DELETE                                    ]
       [   /PROMPT                                    ]
       [   /REMOVE                                    ]
```

```
LINK [ /[NO]EXECUTE[:filespec]                             ] (SP) filespecs
     [ /MAP[:filespec] [/ALLOCATE:size] [/WIDE]            ]
     [                                                     ]
     [ { /LDA                              }               ]
     [ {                                   }               ]
     [ { /FOREGROUND[:stacksize]           }               ]
     [ {  [/FILL:n]                        }               ]
     [ {                                   }               ]
     [ {  [/BOTTOM:n ]                     }               ]
     [ {  [/FILL:n   ]                     }               ]
     [ {  [/RUN      ]                     }               ]
     [ {  [/STACK[:n]]                     }               ]
```

```
/BOUNDARY:value
/DEBUG[:filespec]
/EXTEND:n
/INCLUDE
/LIBRARY:filespec
/LINKLIBRARY:filespec
/PROMPT
/ROUND:n
/SLOWLY
/TRANSFER[:n]
```

```
LOAD (SP) device[=jobtype] [ , . . . device[=jobtype] ]
```

```
MACRO ┌/LIST[:filespec] [/ALLOCATE:size]            ┐ (SP) filespecs ┌/LIBRARY┐
      │/[NO] OBJECT[:filespec] [/ALLOCATE:size]      │                └/PASS:1 ┘
      │                                              │
      │/CROSSREFERENCE[:type[...:type] ]             │
      │/DISABLE:value[...:value]                     │
      │/ENABLE:value[...:value]                      │
      └/[NO] SHOW[:value]                            ┘
```

```
PRINT ⎡/COPIES:n ⎤ (SP) filespecs
      ⎢/DELETE   ⎥
      ⎢/[NO] LOG ⎥
      ⎢/NEWFILES ⎥
      ⎣/QUERY    ⎦
```

```
R (SP) filespec
```

```
REENTER
```

```
REMOVE (SP) device[ , . . . device]
```

```
RENAME ⎡/[NO] LOG       ⎤ (SP) input-filespecs (SP) output-filespec
       ⎢/NEWFILES       ⎥
       ⎢/QUERY          ⎥
       ⎢/[NO] REPLACE   ⎥
       ⎢/SETDATE        ⎥
       ⎣/SYSTEM         ⎦
```

RESET

RESUME

```
RUN (SP) filespec [ { (SP) input-list [ (SP) output-list] } ]
                    { (SP) argument                       }
```

```
SAVE (SP) filespec[ (SP) parameters]
```

```
SET (SP) {physical-device-name} (SP) condition
         {item                }
```

```
SHOW [ (SP)   CONFIGURATION ]
     [        DEVICES       ]
     [        TERMINALS     ]
```

```
SQUEEZE ┌/OUTPUT:device ┐ (SP) device
        └/[NO] QUERY    ┘
```

```
START[ (SP) address]
```

```
SUSPEND
```

```
TIME [ (SP) hh:mm:ss]
```

```
TYPE ┌/COPIES:n  ┐ (SP) filespecs
     │/DELETE    │
     │/[NO] LOG  │
     │/NEWFILES  │
     └/QUERY     ┘
```

```
UNLOAD (SP) device [ , . . . device]
```