IAS FORTRAN Special Subroutines Reference Manual

Order Number: AA-H001B-TC

Operating System and Version: IAS V3.4

Revision/Update Information: This manual replaces the *IAS FORTRAN Special Subroutines Reference Manual,* Version 3.1, Order Nos. AA-H001A-TC and AD-H001A-T1.

May, 1990

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

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

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

Copyright ©1990 by Digital Equipment Corporation

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

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DDIF
DEC
DEC/CMS
DEC/MMS
DECnet
DECUS
DECwindows
DECwrite
DIBOL
IAS
MASSBUS
PDP
PDT
RSTS
RSX
ULTRIX
UNIBUS
VAX
VAX C
VAXcluster
VAXstation
VMS
VR150/160
VT
digital™

This document was prepared using VAX DOCUMENT, Version 1.2

Contents

PREFACE xi

CHAPTER 1 INTRODUCTION 1–1

1.1 FORTRAN SPECIAL SUBROUTINES 1–1

1.2 SUBROUTINE USAGE 1–1
1.2.1 Optional Arguments 1–2
1.2.2 Task Names 1–2
1.2.3 Integer Arguments 1–2
1.2.4 The GETADR Subroutine 1–3

1.3 ERROR CONDITIONS 1–3

1.4 TASK SCHEDULING 1–3

CHAPTER 2 CONTROL OF TASK EXECUTION 2–1
START 2–2
TRNON 2–3
WAIT 2–4
PROGRAM TERMINATION 2–5

CHAPTER 3 PROCESS INPUT AND OUTPUT SUBROUTINES 3–1

3.1 LUN ASSIGNMENTS 3–2

3.2 AISQ (AISQW) SEQUENTIAL ANALOG INPUT 3–3

3.3 AIRD (AIRDW) ANALOG INPUT IN RANDOM SEQUENCE 3–4

3.4 AI (AIW) SINGLE CHANNEL ANALOG INPUT 3–6

3.5 ANALOG OUTPUT 3–7
3.5.1 AO (AOW) - Output Voltage Via the UDC-11 D/A Converter 3–7
3.5.2 AOSC - Output Voltage Via a Single UDC-11 D/A Channel 3–8

3.6 DIGITAL INPUTS 3–9
3.6.1 DI (DIW) - Read a Series of UDC-11 Contact Sense Digital Input Fields Statically 3–9
3.6.2 RCSPT - Read a Single Contact Sense Point 3–10
3.6.3 RCIPT - Read a Single Contact Interrupt Point 3–12
3.6.4 RBCD - Read a BCD Encoded Contact Sense Digital Input and Return Binary Data 3–14
3.6.5 DIFCS - Read a Contact Sense Digital Input Field 3–15

3.7 DYNAMIC CONTACT INTERRUPT INPUTS 3–15
3.7.1 CTDI - Connect a Buffer for Digital Input 3–16
3.7.2 RDDI - Report the State of a Single Contact Interrupt Point 3–17
3.7.3 DFDI - Disconnect a Buffer from Digital Inputs 3–18

3.8 MOMENTARY DIGITAL OUTPUT 3–18
3.8.1 DOM (DOMW) - Pulse a Series of 1-Shot Digital Output Points 3–19
3.8.2 DOSM - Pulse a Single Momentary Output Point 3–20
3.8.3 DOFM - Pulse a Single 16-Bit Field of Momentary Digital Output Points 3–21

3.9 LATCHING DIGITAL OUTPUT 3–22
3.9.1 DOL (DOLW) - Latch/Unlatch Multiple 16-Point Fields 3–22
3.9.2 DOSL - Latch/Unlatch a Single Point 3–23
3.9.3 DOFL - Latch/Unlatch One 16-point Field 3–24

3.10 TIMER INPUT 3–25
3.10.1 RSTI - Read a Timer Module 3–25
3.10.2 SCTI - Set a Timer Module 3–26

3.11 DYNAMIC TIMER INPUTS 3–27
3.11.1 CTTI - Connect a Circular Buffer for Timer Input 3–27
3.11.2 RDTI - Report Timer Values 3–28
3.11.3 DFTI - Disconnect the Buffer from Timer Inputs 3–29

CHAPTER 4 BIT STRING MANIPULATIONS 4–1

4.1 TYPES OF MANIPULATIONS 4–1

4.2 LOGICAL OPERATIONS 4–1

4.3 SHIFT OPERATIONS 4–2

CHAPTER 5 SYSTEM DIRECTIVES 5–1

5.1 ABORT 5–1

5.2 ALTPRI 5–1

5.3 ASNLUN 5–2

5.4 ATRG 5–2

5.5 CANALL 5–2

5.6 CANMT 5–3

5.7 CANOBY 5–3

5.8 CLREF 5–4

5.9 CRAW 5–4

5.10 CRRG 5–5

5.11 DECLAR 5–5

5.12 DISABL 5–5

5.13 DISCKP 5–6

5.14 DTRG 5–6

5.15 ELAW 5–6

5.16 ENABLE 5–7

5.17 ENACKP 5–7

5.18 ENASTR 5–7

5.19 EXECUT 5–8

5.20 EXITIF 5–8

5.21 EXTTSK 5–9

5.22 FIXMEM 5–9

5.23 GCMX 5–9

5.24 GETADR 5–10

5.25 GETCMN 5–10

5.26 GETLUN 5–11

5.27 GETMCR 5–11

5.28 GETPAR 5–11

5.29 GETREG 5–12

5.30 GETTSK 5–12

5.31 INASTR 5–13

5.32 MAP 5–13

5.33 MARK 5–13

5.34 MNLOAD 5–14

5.35 PWRUP 5–15

5.36 QIO 5–15

5.37 READEF 5–16

5.38 READSW 5–16

5.39 RECEIV AND VRECEV 5–17

5.40 RECOEX AND VRECEX 5–18

5.41 RECOSP AND VRECSP 5–19

5.42 RECOST AND VRECST 5–20

5.43 REQUES 5–21

5.44 RESORU 5–22

5.45 RESUME 5–22

5.46 RREF 5–22

5.47 RUN 5–23

5.48 SCHED 5–24

5.49 SEND AND VSEND 5–25

5.50 SETEF 5–26

5.51 SNDROR AND VSNDRR 5–26

5.52 SPAWN 5–27

5.53 SREF 5–28

5.54 SRRF 5–29

5.55 SSWTCH 5–30

5.56 STOPFR 5–30

5.57 STOPOR 5–30

5.58 STOPTK 5–31

5.59 SUSPND 5–31

5.60 SYNC 5–31

5.61 UNFIX 5–32

5.62 UNMAP 5–32

5.63 UNSTOP 5–33

5.64 WAITFR 5–33

5.65 WFLOR 5–33

5.66 WFSNE 5–34

5.67 WTQIO 5–34

APPENDIX A DSW RETURN CODES A–1

INDEX

TABLES

4–1 Logical Operations 4–1

Preface

Purpose of the Manual

This manual is intended for the FORTRAN programmer who needs to call IAS subroutines from a FORTRAN program.

The subroutines described in this manual can be used with either the FORTRAN-IV system or with the FORTRAN IV-PLUS (F4P) system. The language supported by both of these FORTRAN systems is described in the *PDP-11 FORTRAN Language Reference Manual.*

Document Structure

Chapter 1 provides introductory information about subroutine usage.

Chapter 2 describes the control of task execution.

Chapter 3 details the process input and output subroutines.

Chapter 4 describes bit string manipulations.

Chapter 5 details the IAS system directives.

Appendix A details the DSW return codes.

Associated Documents

The reader should be familiar with the following FORTRAN manuals:

- *PDP-11 FORTRAN Language Reference Manual*
- *FORTRAN-IV User's Guide*
- *FORTRAN-IV-PLUS User's Guide*

The IAS system directives are fully described in the *IAS System Directives Reference Manual.*

Refer to the *IAS Documentation Directory* for a brief description of other related documents in the manual set.

Manual Conventions

Throughout the manual, the phrase "FORTRAN User's Guide" refers to both the *FORTRAN-IV User's Guide* and the *FORTRAN-IV-PLUS User's Guide.*

1 INTRODUCTION

1.1 FORTRAN Special Subroutines

IAS provides many subroutines that enable the FORTRAN programmer to perform process control and system directive operations.

These subroutines fall into three basic groups:

1 Subroutines based on the **Instrument Standard of America (ISA) Standard ISA 62.1**.

   The ISA standard is based on the recommendations of the **Workshop on Standardization of Industrial Computer Languages** (sometimes known as the Purdue Workshop). ISA provides standard FORTRAN subroutine calls for the control of task execution, input/output by means of analog/digital interfaces, and logical operations on binary encoded data. These subroutines are included in the subroutine descriptions in 2, 3 and 4.

2 Digital supplies subroutines that use and monitor specific process control interface devices supported by IAS.

   Subroutines for using the AFC-11 A/D Converter, the AD01-D A/D Converter, and the UDC-11 Universal Digital Controller are described in Chapter 3. Many of the subroutines are based on the ISA standard as described above. Many additional subroutines and optional facilities are provided to take advantage of the full capabilities of these devices.

3 Subroutines for performing system directive operations.

   IAS offers extensive capabilities for control of task execution, inter-task communication, control of system resources, and memory management. FORTRAN subroutine calls (as described in Chapter 5 give the user these capabilities. The memory management directives are also supported by means of a FORTRAN interface. In general, one subroutine is available for each system directive.

   The user can call these subroutines from subroutines written in the MACRO-11 assembly language by using PDP-11 FORTRAN calling sequence conventions described in the *FORTRAN-IV User's Guide*.

1.2 Subroutine Usage

All of the subroutines described in this manual (see 5 are included in the system library, SYSLIB.OLB. This library is automatically searched by the Task Builder when linking a task for execution.

To use one of these routines, include the appropriate CALL statement in the FORTRAN program. TKB selects the subroutine and automatically includes it in the task image during standard task build procedures.

1.2.1 Optional Arguments

Many of the subroutines described in this manual have optional arguments; these arguments do not have to be specified in each call. In the subroutine descriptions, optional arguments are designated by square brackets ([ ]). Omit arguments either by using an empty (or "null" field in the argument list, or by shortening the argument list.

For example, if the format of a call to SUB is

```
CALL SUB ( AA, [BB], [CC], DD, [EE], [FF])
```

then the BB, CC, EE, and FF arguments can be left unspecified, as in

```
CALL SUB (AA,,,DD,,)
     or
CALL SUB (AA,,,DD)
```

In some cases, a subroutine uses a default value for an unspecified optional argument. Such default values are noted in each subroutine description.

1.2.2 Task Names

In these subroutines, task names can be up to six characters long. Characters permitted in a task name are the letters A through Z, the numerals 0 through 9, and the special characters dollar sign ($), period (.), and blank ( ). Task names are stored in Radix-50 representation, that permits up to three characters from the above set to be encoded in one PDP-11 storage word. (Radix-50 is described in detail in the *FORTRAN User's Guide*).

In the following subroutine calls, a task name is defined as a variable of type Real that contains the task name in Radix-50 representation. This value may be defined at program compilation by a DATA statement that gives the Real variable an initial value (a Radix-50 constant).

For example, if a task named JOBl is to be used in a system directive call, the task name could be defined and used as follows:

```
DATA JOB1/4RJOB1/
 . . .
CALL REQUES (JOB1)
```

Task names can also be defined during execution by using the IRAD50 subroutine or the RAD50 function. Both of these are described in the *FORTRAN User's Guide*.

1.2.3 Integer Arguments

Many of the arguments used to communicate values to system subroutines or to receive values from these routines are described as integer arguments.

All of the subroutines described in this manual assume that integer arguments are, specifically, Integer*2 type arguments.

Both the FOR and F4P FORTRAN systems normally treat an Integer variable as one PDP-11 storage word, provided that its value is within the range -32768 to +32767. However, if the /I4 compile-time option switch is used, particular care must be taken to ensure that all integer arguments used in these subroutines are explicitly specified to be of type Integer*2.

1.2.4 The GETADR Subroutine

In a few subroutines, notably QIO and WTQIO, an argument in the subroutine call is described as an integer array containing some values that are the addresses of other variables or arrays. The FORTRAN language does not provide a means for assigning such an address as a value. This facility is provided by the GETADR subroutine, which is described in Chapter 5, Section 5.24.

1.3 Error Conditions

In general, each subroutine provides a means by which the calling program can determine whether the requested operation was successfully performed or whether an error condition was detected. Usually, there is an optional argument whose value is set by the subroutine. In the case of an error, the value can also encode the type of error detected. The error conditions, which can be expected for directives, are listed in A of this manual.

In addition, two types of errors are reported by means of the FORTRAN Object Time System diagnostic messages, as described in the *FORTRAN User's Guide*. The error conditions are:

- DIRECTIVE: MISSING ARGUMENT(S)

  One or more necessary arguments were missing from a call to a system directive subroutine. (OTS error number 100)

- DIRECTIVE: INVALID EVENT FLAG NUMBER

  An event flag number in a system directive call was not in the range 1 to 64. (OTS error number 101)

Both of these errors are fatal and result in the termination of the task.

1.4 Task Scheduling

In many of the following subroutine calls, time intervals and the units to measure them must be specified as arguments. Time units can be specified as minutes, seconds, milliseconds, or counts of the system clock.

IAS supports two system clocks for scheduling purposes, the KWll-L and the KWll-P. Although all time intervals can be specified in milliseconds in ISA-standard calls, scheduling precision can be limited by the resolution of the system clock.

The KWll-L is a line frequency clock whose maximum resolution is 16.7 milliseconds at 60 Hz. line frequency, and 20 milliseconds at 50 Hz. line frequency.

The KWll-P is a programmable crystal-controlled clock. The clock rate is selected at system generation time, so scheduling accuracy depends upon the clock rate of the individual installation.

2 Control of Task Execution

Executive interfaces enable the user to control operation of tasks within the system. Given these facilities, the user can start, stop, or delay execution of any task in the system. The argument M, shown below, is equal to or greater than two (2) for all instances in which the request is not accepted by the executive system. Error codes are converted to the above ISA convention by negating the directive status value and adding +1.

The following pages describe the executive interfaces.

START

The START call causes the execution of a designated task after a specified time delay. The actual time delay available is subject to the resolution of the real-time clock. Execution of the designated program starts at its first executable statement.

FORMAT

CALL START (I, J, K, M)

where:

- I = a 2-word integer array or a Real variable that specifies the program to be executed (in Radix-50 representation).
- J = specified the length of the time delay, in units as specified by K, before program execution. If the value is zero or negative, the requested program will run as soon as permissible.
- K = specifies units of time as follows:

  0 - Basic counts of the system's clock
  1 - Milliseconds (Converted to basic counts of the system's clock.)
  2 - Seconds
  3 - Minutes
- M = an integer set on return to the calling program to indicate the diposition if the request as follows:

  1 - Request accepted
  2 or greater - Request not accepted

The START call is implemented by using the RUN$ directive. An error return indicates that the directive was rejected. Subtract one from variable M and negate to obtain the standard system error code.

TRNON

The TRNON call causes a designated task to be executed at a specified time of day. Execution of the designated task starts at its first executable statement.

FORMAT

CALL TRNON (I, J, M)

where:

- I = is a 2-word integer array or a Real variable that specifies the program to be executed (in Radix-50).
- J = designates a 1-dimensional integer array of length three; the array contains the absolute time at which the program is to be executed. The elements of the array are as follows:

  J (1) - Hours using a 24-hour clock
  J (2) - Minutes
  J (3) - Seconds
- M = is an integer set on return to the calling program to indicate the disposition of the request as follows:

  1 - request accepted
  2 or greater - request not accepted.

The subroutine issues a SCHEDULE directive by building the appropriate Directive Parameter Block (DPB) on the stack. An error return indicates that the directive was rejected. Subtract one from variable M and negate to obtain the standard system error code.

WAIT

A task can relinquish control of the system for a specified length of time by means of the WAIT call. After expiration of the specified delay, the subroutine resumes execution of the requesting task at the first executable statement following the WAIT call.

FORMAT

CALL WAIT (J, K, M)

where:

- J = specifies the length of time, in units as specified by K, to delay before continuing execution in the requesting program. If the value is zero or negative no delay occurs.
- K = specifies units of time as follows:

  0 - Basic counts of the system's clock
  1 - Milliseconds (Converted to basic counts of the system's clock.)
  2 - Seconds
  3 - Minutes
  4 - Hours
- M = set on return to the calling program to indicate the disposition of the request as follows:

  The routine executes a MARKTIME directive followed, if successful, by a WAITFOR using Event Flag 29. An error return indicates that the MARKTIME directive failed. Subtract one from the status variable M and negate to obtain the system standard error code.

PROGRAM TERMINATION

Use the STOP statement or CALL EXIT subroutine to terminate a real-time program. (The system continues to perform other tasks in the multiprogramming system.)

3 Process Input and Output Subroutines

Process input and output subroutines enable you to access the following devices:

- AFC-11 A/D Converter
- AD01-D A/D Converter
- UDC-11 Universal Digital Controller

Except for UDC-11 functions, which are serviced with minimum delay, each call is provided in two forms. The first form permits continuation of execution in the requesting program while Process I/O is being accomplished; the other form suspends the requesting task until Process I/O is completed. The latter case is flagged by the addition of a "W" as the last letter in the subroutine name. Both forms of the name are allowed in designated UDC-11 routines for compatibility with the Instrument Society of America (ISA) standard; however, a WAIT(1) is always implied.

The status return argument for subroutines that perform I/O is a 2-word integer array. Word 1 contains the success/fail indication and is set as follows:

- Word 1 =1 Success,
- Word 1 =0 Operation pending,
- Word 1 >3 Operation failure.

The contents of word 2 depend on the individual function code.

To distinguish failures detected by the subroutines from failures detected by the Executive or I/O handler, the following convention has been established:

> Values of Status Word 1 greater than 3 and less than 300 indicate that the QIO request was rejected by the executive. Values of Status Word 1 greater than 300 indicate that an error was detected by the I/O handler. The system standard error code may be obtained by subtracting 3 and negating the result. An error return of (+3) indicates that the subroutine was unable to build the QIO directive.

(1) All UDC-11 functions are serviced immediately. Hence, unless the requesting task is of a higher priority than the Handler Task, the WAIT is effectively a NOP.

Once the QIO has been processed, an Asynchronous System Trap (AST) internal to the called subroutine sets up the first status word. Therefore, as a temporary requirement ASTs should be enabled before the call is executed and until the required status has been returned.

Subroutines with explicit or implied "Waits" (that is, all UDC calls and A/D calls with "W" as the last character in the subroutine name) have status set upon return to the calling program. However, if continuation is permitted, the program must periodically test the state of the request.

The following conventions apply to module number, D/A channel number and point number when specified for the UDC-11:

- Points and D/A channels are numbered sequentially from 0 beginning with the first point or channel on the first module of a given type.
- Modules are numbered sequentially from 0 beginning with the first module of a given type.

3.1 LUN ASSIGNMENTS

Specification of the Logical Unit Number as an optional parameter implies that the calling program has invoked one of the subroutines described below to assign the LUN and record the value for later reference. If the LUN is not specified in this manner, then the argument becomes mandatory and LUN must be assigned via the ASNLUN Executive call (see 5, "ASNLUN".

The following set of subroutines may be called to assign Logical Unit Numbers to process control devices. As stated above, use of these subroutines eliminates the need to specify the LUN in the Process I/O calls.

Assignments are made for the following devices:

- AFC-11 A/D Converter (see "Restrictions" below)
- AD01-D A/D Converter (see "Restrictions" below)
- UDC-11 Universal Digital Controller

CALLING SEQUENCE

CALL SUB (ILUN,[ISW])

Where:

- "SUB" is one of the following subroutine names:
  - — ASAFLN to ASSIGN a LUN to the AFC-11 A/D Converter
  - — ASADLN to ASSIGN a LUN to the AD01-D A/D Converter
  - — ASUDLN to ASSIGN a LUN to the UDC-11 Controller

ARGUMENTS

ILUN
An integer specifying the Logical Unit Number to be assigned.

ISW
An integer to receive the results of the ASSIGN LUN directive.

Each subroutine issues the ASSIGN LUN directive and stores the Logical Unit Number in the task's header for later reference by the appropriate Process I/O calls.

RESTRICTIONS

For A/D devices, only one LUN is saved in the task's header, which is specified in the last call to ASAFLN or ASADLN.

3.2 AISQ (AISQW) SEQUENTIAL ANALOG INPUT

CALLING SEQUENCE

CALL AISQ (INM,ICONT,IDATA,[ISTAT],[ILUN]) *or*

CALL AISQW (INM,ICONT,IDATA,[ISTAT],[ILUN])

ARGUMENTS

INM

An integer variable or constant and designates the number of analog input points to be scanned.

ICONT

A 1-dimensional integer array designating terminal connection data in the following format:

Bit	Meaning
0 - 11	Channel number (right justified), must be specified in ICONT (1) only
12 - 15	Channel gain value for this sample, expressed in the bit patterns shown as follows:

Bits				Gain
15	14	13	12	
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	10
0	1	0	1	20
1	0	0	0	50
1	0	0	1	100
1	1	0	0	200
1	1	0	1	1000

Permissible values for the terminal connection array depend on hardware type and configuration.

For sequential analog input, the channel number is computed in steps of one, beginning with the value specified in the first element of ICONT. The channel number field is ignored in all other elements.

ARGUMENTS

IDATA

An integer array to receive the points converted by the hardware.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

In the event of an error the contents of the data array are undefined.

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASADLN or ASAFLN as described above.

3.3 AIRD (AIRDW) ANALOG INPUT IN RANDOM SEQUENCE

CALLING SEQUENCE

CALL AIRD (INM,ICONT,IDATA,[ISTAT],[ILUN]) *or*

CALL AIRDW (INM,ICONT,IDATA,[ISTAT],[ILUN])

ARGUMENTS

INM

An integer variable/constant that designates the number of analog input points to be scanned.

ICONT

A 1-dimensional integer array designating terminal connection data in the following format:

Bit	Meaning
0 - 11	Channel number (right justified), must be specified in ICONT (1) only
12 - 15	Channel gain value for this sample, expressed in the bit patterns shown as follows:

Bits				Gain
15	14	13	12	
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	10
0	1	0	1	20
1	0	0	0	50
1	0	0	1	100
1	1	0	0	200
1	1	0	1	1000

Permissible values for the terminal connection array depend on hardware type and configuration.

IDATA

An integer array to receive the points converted by the hardware.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

In the event of an error the contents of the data array are undefined.

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASADLN or ASAFLN as described above.

3.4 AI (AIW) Single Channel Analog Input

CALLING SEQUENCE

CALL AI (ICHN,ISTAT,[ILUN]) *or*

CALL AIW (ICHN,ISTAT,[ILUN])

ARGUMENTS

ICHN

An integer specifying channel number and gain in the following format:

Bit	Meaning
0 - 11	Channel number (right justified), must be specified in ICONT (1) only
12 - 15	Channel gain value for this sample, expressed in the bit patterns shown as follows:

Bits 15	 14	 13	 12	Gain
0	0	0	0	1
0	0	0	1	2
0	0	1	0	4
0	0	1	1	8
0	1	0	0	10
0	1	0	1	20
1	0	0	0	50
1	0	0	1	100
1	1	0	0	200
1	1	0	1	1000

Permissible values for ICHN depend on hardware type and configuration.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

Parameter	Status Value
	In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN
An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASADLN or ASAFLN as described above.

3.5 Analog Output

You can call the following subroutines to perform analog output via the UDC-11.

- AO(AOW)—To output voltages from several UDC-11 D/A channels.
- AOSC —To output a voltage via a single UDC-11 D/A channel.

3.5.1 AO (AOW) - Output Voltage Via the UDC-11 D/A Converter

CALLING SEQUENCE

CALL AO (INM,ICONT,IDATA,[ISTAT],[ILUN]) *or*

CALL AOW (INM,ICONT,IDATA,[ISTAT],[ILUN])

Calls to AO and AOW are interchangeable.

ARGUMENTS

INM
An integer variable/constant that designates the number of analog channels to be output.

ICONT
A 1-dimensional integer array containing the UDC-11 D/A channel numbers.

IDATA
An integer array containing the output voltage setting. Output voltage is specified as an integer quantity between 0 and 1023 where:

0	=	0 volts and
1023	=	+10 volts

ISTAT
A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and <300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN
An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.5.2 AOSC - Output Voltage Via a Single UDC-11 D/A Channel

CALLING SEQUENCE

CALL AOSC (ICHN,IVOLTS,[ISTAT],[ILUN])

ARGUMENTS

ICHN
An integer specifying UDC-11 channel number.

IVOLTS
An integer of value 0 to 1023 specifying output voltage where:

0 =	0 volts.
1023 =	+10 volts.

ISTAT
A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASADLN or ASAFLN as described above.

3.6 Digital Inputs

The following subroutines allow the state of a series of points or a single point to be read:

- DI(DIW) - To return contact sense data from multiple 16-bit fields. A field consists of 16 points beginning at point n and extending to point n+15. Fields can cross module boundaries.
- RBCD - to return 16 bits of BCD encoded contact sense data as binary data.
- DIFCS - to return one field (16 bits) of contact sense data in binary form.
- RCSPT - to return the state of a single contact sense point as a logical value.
- RCIPT - To return the state of a single contact interrupt point as a logical value.

3.6.1 DI (DIW) - Read a Series of UDC-11 Contact Sense Digital Input Fields Statically

CALLING SEQUENCE

CALL DI(INM,ICONT,IDATA,ISTAT,[ILUN]) *or*

CALL DIW(INM,ICONT,IDATA,ISTAT,[ILUN])

Calls to DI and DIW are interchangeable.

ARGUMENTS

INM

An integer for the number of fields to be read.

ICONT

A 1-dimensional integer array containing the initial point in a field of 16 points.

IDATA

A 1-dimensional integer array to receive the input data, 16 bits of contact data are retuned for each field read.

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and <300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.6.2 RCSPT - Read a Single Contact Sense Point

CALLING SEQUENCE

CALL RCSPT (IPT,ISTAT,[ILUN])

ARGUMENTS

IPT

An integer specifying the point number to be read. Points are numbered sequentially, with the first point on the first module as point 0.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and <300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.6.3 RCIPT - Read a Single Contact Interrupt Point

CALLING SEQUENCE

CALL RCIPT (IPT,ISTAT,[ILUN])

ARGUMENTS

IPT

An integer specifying the point number to be read. Points are numbered sequentially starting with the first point on the first module as 0.

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
</td></tr>
</table>

Parameter	Status Value
	In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.6.4 RBCD - Read a BCD Encoded Contact Sense Digital Input and Return Binary Data

CALLING SEQUENCE

CALL RBCD (IPT,IMSK,ISTAT,[ILUN])

ARGUMENTS

IPT
An integer specifying the initial point to read in a field of 16 points.

IMSK
An integer specifying a 16-bit mask to be applied to the input data. Points not masked are set to zero. The result, after masking, is converted to binary. An AND is performed on the mask and the input data prior to conversion.

ISTAT
A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value	
ISTAT(1)	Set to one of the following values to indicate status:	
	Value	**Status**
	=1	The request succeeded.
	>3	The request failed.
	In the event of failure, ISTAT(1) assumes one of the following values:	
	ISTAT(1)	**Meaning**
	>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
	>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

In the event of an error the contents of the data array are undefined.

ILUN
An integer specifying the Logical Unit Number. If this parameter is not specified, the LUN must be assigned via a call to ASADLN or ASAFLN as described above.

RESTRICTIONS

For A/D devices, only one LUN is saved in the task's header, which is specified in the last call to ASAFLN or ASADLN.

3.6.5 DIFCS - Read a Contact Sense Digital Input Field

CALLING SEQUENCE

CALL DIFCS (IPT,IMSK,ISTAT,[ILUN])

ARGUMENTS

IPT

An integer specifying the initial point to be read, in a field of 16 points. Bit N in ISTAT(2) represents the point IPT+N.

IMSK

An integer specifying a 16-bit mask to be applied to the input data. Points not masked are set to zero. An AND is performed on the mask and the module data.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified the LUN must be assigned via a call to ASUDLN.

3.7 Dynamic Contact Interrupt Inputs

Dynamic contact interrupt inputs are read directly into a user-supplied circular buffer by the UDC-11 device handler. FORTRAN- callable subroutines allow the user to perform the following functions:

- CTDI = Connect a user-supplied buffer for digital input.
- RDDI = Report the state of a single contact interrupt point.

- DFDI = Disconnect the buffer from digital inputs.

3.7.1 CTDI - Connect a Buffer for Digital Input

CALLING SEQUENCE

CALL CTDI (IBUF,ISZ,IEFN,[ISTAT],[ILUN])

ARGUMENTS

IBUF

An integer array to receive contact interrupt data.

ISZ

An integer specifying the length of the array. Array size must not be less than 15. See below.

IEFN

An integer specifying the Trigger Event Flag number. This flag will be set each time the handler attempts to make an entry into the buffer.

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and <300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN

An integer specifying the Logical Unit Number. If this parameter is not specified the LUN must be assigned via a call to ASUDLN.

The handler generates a 5-word entry; each time a change of state is detected. One entry is created for each interrupting module. A buffer of minimum size accomodates a single entry plus the additional impure storage required by the handler and the subroutine RDDI (see Section 3.7.2).

The following considerations should be used in establishing the size of the buffer:

1 The rate at which contact interrupts occur.

2 The number of modules that can interrupt simultaneously.

3 The rate at which the circular buffer is to be read.

3.7.2 RDDI - Report the State of a Single Contact Interrupt Point

RDDI does no actual I/O but performs a point-by-point scan of the input buffer, returning the state of each point as a logical value. The Trigger Event Flag is cleared on every call and the point to be examined is advanced by one unless an invalid entry is detected.

CALLING SEQUENCE

CALL RDDI (IPT,IVAL,[ICT])

ARGUMENTS

IPT

An integer to receive the digital input point number. IPT may be set to one of the following values:

- IPT<0 - No valid entry found.
- IPT>0 - The value indicated is the point number; the state is in IVAL.

IVAL

An integer to receive the state of the point, IVAL is set .TRUE. "-1" if the point is closed and .FALSE. "0" if the point is open.

ICT

An integer to receive the overrun count. The count is supplied as a positive value greater than 0 and indicates that the handler was unable to store one or more entries.

In all other cases ICT is set to 0.

The initial call to RDDI causes the data word and module number for an entry to be read from the circular buffer and stored within the 10-word buffer header. The subroutine then reads the state of data bit 0 and converts the data bit number "n" to a point number using the following formula:

POINT = module number x 16 + n (15 <n <0)

Each subsequent call causes n to be increased by one; that is, the next bit in sequence is examined. When 16 is reached, n is reset to zero and an attempt is made to fetch another entry from the buffer. If a valid entry is not present, IPT is set negative and ICT, if specified, contains either 0 or an overrun count supplied by the handler. A value of 0 for ICT indicates that no further entries remain. A nonzero value indicates that the handler has received more data than can be stored in the buffer. The latter count represents the number of entries missed.

The input circular buffer can be scanned using the following procedures:

1 Execute a "Wait For" on the event flag used by CTDI. See Chapter 5, Section 5.62.

2 Repeatedly call RDDI until all valid points are read; that is, until IPT is less than zero.

3 Perform any required processing and return to step 1, above.

Step 2 should be performed at CPU speed, retaining the state of significant points for processing after all entries have been scanned.

3.7.3 DFDI - Disconnect a Buffer from Digital Inputs

CALLING SEQUENCE

CALL DFDI ([ISTAT],[ILUN])

ARGUMENTS

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

3.8 Momentary Digital Output

The following subroutines are provided to pulse a single point or a series of points:

- DOM(DOMW) = Pulses a series of 1-shot (momentary) digital output points.
- DOSM = Pulses a single point.
- DOFM = Pulses a single 16-bit field.

3.8.1 DOM (DOMW) - Pulse a Series of 1-Shot Digital Output Points

CALLING SEQUENCE

CALL DOM (INM,ICONT,IDATA,[IDX],[ISTAT],[ILUN]) *or*

CALL DOMW(INM,ICONT,IDATA,[IDX],[ISTAT],[ILUN])

Both forms of the call are interchangeable.

ARGUMENTS

INM
An integer variable and designates the number of entries in the terminal connection and data arrays.

ICONT
An integer array specifying terminal connection data in the format described below

Entries in ICONT consist of digital output point numbers.

Each entry in ICONT is paired with a word in the data array. Bit "N" in the data array corresponds to the point number +N "N = 0 -15". One point is pulsed for each bit set in the data array entry.

IDATA
An integer array specifying the points to be pulsed.

IDX
A dummy variable retained for compatibility with existing FORTRAN Process Control calls.

ISTAT
A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value	
ISTAT(1)	Set to one of the following values to indicate status:	
	Value	**Status**
	=1	The request succeeded.
	>3	The request failed.

Parameter	Status Value
	In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN
An integer specifying the Logical Unit Number. If this parameter is not specified the LUN must be assigned via a call to ASUDLN.

ITM
An integer to receive timer values.

IVRN
An integer to receive the overrun count. The count is supplied as a positive value and is set to zero when no overrun condition exists.

NOTE: On the occurrence of a time module interrupt, the handler resets the count to an initial value, normally that specified in array IV of subroutine CTTI (Section 3.11.1). The initial value can be modified for a specific module by calling subroutine SCTI (Section 3.10.2).

3.8.2 DOSM - Pulse a Single Momentary Output Point

CALLING SEQUENCE

CALL DOSM (IPT,[ISTAT],[ILUN])

ARGUMENTS

IPT
An integer variable specifying the point number.

ISTAT
A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

.8.3 DOFM - Pulse a Single 16-Bit Field of Momentary Digital Output Points

CALLING SEQUENCE

CALL DOFM (IPT,IDATA,[ISTAT],[ILUN])

ARGUMENTS

IPT

An integer specifying the initial point in a field of 16 points.

IDATA

Specifies the points to be pulsed. A point is pulsed for each bit set in the data word. If bit N is set then point "IPT+N" is triggered.

ISTAT

A 2-word integer array to receive the results of the call as follows.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

Parameter	Status Value
	In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN

An integer specifying the UDC-11 logical unit number. A previous call to ASUDLN is required if this value is not specified in the calling sequence.

3.9 Latching Digital Output

The following subroutines are provided to activate the latching outputs:

- DOL(DOLW) = Latch/unlatch multiple 16-point fields.
- DOSL = Latch/unlatch a single point.
- DOFL = Latch/unlatch one 16-point field.

3.9.1 DOL (DOLW) - Latch/Unlatch Multiple 16-Point Fields

CALLING SEQUENCE

CALL DOL (INM,ICONT,IDATA,IMSK,[ISTAT,][ILUN]) *or*

CALL DOLW (INM,ICONT,IDATA,IMSK,[ISTAT,][ILUN])

Both forms of the call are interchangeable.

ARGUMENTS

INM

An integer specifying the number of entries in the output arrays described below.

ICONT

An integer array. Each entry specifies an initial point in a string of 16 points.

IDATA

An integer array. Each entry in the array specifies a string of 16 points where bit N corresponds to the point number +N (0<N<15).

If the corresponding mask bit in IMSK is set, then the point is unlatched if bit N is 0, and latched if bit N is 1. The point is not touched if the mask bit is zero.

IMSK

An integer array. Each entry specifies a 16-bit mask word. To change the state of a point the corresponding mask bit must be set.

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and <300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN

An integer specifying the Logical Unit Number; if this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.9.2 DOSL - Latch/Unlatch a Single Point

CALLING SEQUENCE

CALL DOSL (IPT,IVAL,[ISTAT],[ILUN])

ARGUMENTS

IPT

An integer specifying the output point.

IVAL

A logical (integer) variable which will cause the point to be latched if .TRUE. (-1) or unlatched if .FALSE. (0).

ISTAT

A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and
<300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN
An integer specifying the logical unit number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.9.3 DOFL - Latch/Unlatch One 16-point Field

CALLING SEQUENCE

CALL DOFL (IPT,IDATA,IMSK,[ISTAT],[ILUN])

ARGUMENTS

IPT
An integer specifying the initial point in a field of 16 points.

IDATA
An integer specifying the points to be latched/ unlatched. Bit N designates the point "IPT+N". A value of “1” specifies latching, “0” specifies unlatching.

IMSK
An integer designating the points that are allowed to change state.

ISTAT
A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>>3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>>3 and
<300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>>300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN

An integer specifying the logical unit number; if this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.10 Timer Input

- RSTI = Read a timer module.
- SCTI = Initialize a timer module.

3.10.1 RSTI - Read a Timer Module

CALLING SEQUENCE

CALL RSTI (IMOD,ISTAT,[ILUN])

ARGUMENTS

IMOD

An integer specifying the timer module number.

ISTAT

A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN
An integer specifying the logical unit number; if this parameter is not specified the LUN must be assigned via a call to ASUDLN.

3.10.2 SCTI - Set a Timer Module

CALLING SEQUENCE

CALL SCTI (IMOD,IVAL,[ISTAT],ILUN])

ARGUMENTS

IMOD
An integer containing the timer module number.

IVAL
An integer specifying the timer initial value.

ISTAT
A 2-word integer array to receive the results of the call as follows:

Parameter	Status Value
ISTAT(1)	Set to one of the following values to indicate status:

Value	Status
=1	The request succeeded.
>3	The request failed.

Parameter	Status Value
	In the event of failure, ISTAT(1) assumes one of the following values:

ISTAT(1)	Meaning
>3 and <300	QIO directive failed. Subtract 3 and negate the result to obtain the system error code.
>300	Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.

ILUN

An integer specifying the logical unit number; if this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

3.11 Dynamic Timer Inputs

Timer input may be read dynamically in a manner similar to contact interrupts. These subroutines are provided to perform the following functions:

- CTTI = To connect a buffer to receiver timer inputs.
- RDTI = To report timer values.
- DFTI = To disconnect a timer input buffer.

3.11.1 CTTI - Connect a Circular Buffer for Timer Input

CALLING SEQUENCE

CALL CTTI (IBUF,ISZ,IEFN,IV,[ISTAT],[ILUN])

ARGUMENTS

IBUF

An integer array to receive timer data.

ISZ

An integer specifying the length of the array (IBUF). ISZ must not be less than 12.

IEFN

An integer specifying the Trigger Event Flag number. The specified Event Flag will be set each time the handler attempts to make an entry in the data buffer.

IV

A table of initial timer module values. One entry is required for each timer module in the device. Entry N corresponds to timer module number (N-1) where N is a FORTRAN index into the array. See the note below.

ISTAT
A 2-word integer array to receive the results of the call as follows:

<table>
<tr><th>Parameter</th><th>Status Value</th></tr>
<tr><td>ISTAT(1)</td><td>Set to one of the following values to indicate status:
<table>
<tr><th>Value</th><th>Status</th></tr>
<tr><td>=1</td><td>The request succeeded.</td></tr>
<tr><td>&gt;3</td><td>The request failed.</td></tr>
</table>
In the event of failure, ISTAT(1) assumes one of the following values:
<table>
<tr><th>ISTAT(1)</th><th>Meaning</th></tr>
<tr><td>&gt;3 and &lt;300</td><td>QIO directive failed. Subtract 3 and negate the result to obtain the system error code.</td></tr>
<tr><td>&gt;300</td><td>Handler detected an error. Subtract 300 and negate the result to obtain the system error code. In this case, ISTAT(2) contains the second word of I/O status.</td></tr>
</table>
</td></tr>
</table>

ILUN
An integer specifying the logical unit number. If this parameter is not specified, the LUN must be assigned via a call to ASUDLN.

The handler generates a 4-word entry for each interrupting timer module. A buffer of minumum size, accomodates a single entry plus the additional impure storage required by the handler and the subroutine RDTI (see Section 3.11.2).

The following considerations should be used in establishing the size of the buffer:

1 The rate at which timer module interrupts occur.

2 The number of modules that can interrupt simultaneously.

3 The rate at which the circular buffer is to be read.

NOTE: When a timer module interrupt occurs, the handler resets the count to an initial value, normally that specified in array IV. The initial value can be modified for a specific module by calling the subroutine SCTI, described in Section 3.10.2.

3.11.2 RDTI - Report Timer Values

RDTI does no actual I/O but executes a scan of each entry in the buffer reporting module number and time value. One entry is examined for each call and the specified Trigger Event Flag is cleared. New entries are examined until all valid data has been reported.

CALLING SEQUENCE

CALL RDTI (INM,ITM,[IVRN])

ARGUMENTS

INM

An integer to receive the module number. This variable may be set to one of the following values:

- INM<0 - Entry not valid.
- INM>0 - Timer number. The value of the timer is in ITM.

ITM

An integer to receive timer values.

IVRN

An integer to receive the overrun count. The count is supplied as a positive value and is set to zero when no overrun condition exists.

NOTE: On the occurence of a timer module interrupt, the handler resets the count to an initial value, normally that specified in array IV of subroutine CTTI (Section 3.11.1). The initial value can be modified for a specific module by calling subroutine SCTI (Section 3.10.2).

.11.3 DFTI - Disconnect the Buffer from Timer Inputs

CALLING SEQUENCE

CALL DFTI ([ISTAT],[ILUN])

ARGUMENTS

ISTAT

A 2-word integer array to receive status as described for subroutine CTDI (Section 3.7.1).

ILUN

An integer specifying Logical Unit Number as described previously.

4 Bit String Manipulations

4.1 Types of Manipulations

In order to process words from arrays it is necessary to be able to manipulate information on a bit-by-bit basis. Two types of bit string manipulations are provided, logical and shifting. These are available to support the previously specified calls.

4.2 Logical Operations

These operations are implemented as function subprograms. In the following functions, m and n specify integer variables or array elements. Operations are performed on a full word, bit-by-bit.

The logical operations described below are included for compatibility reasons and are equivalent to the following FORTRAN logical operators:

.OR. .AND. .NOT .XOR.

When operating on integer expressions, the FORTRAN operators should be used. The equivalent FORTRAN operators follow the word "use" in each of the operators described below:

Table 4–1 Logical Operations

Operation	Notation	Result	Explanation
Inclusive OR	IOR (m,n)	use m .OR. n	Where m and n designate arguments which are logically added.
Logical Product	IAND (m,n)	use m .AND. n	Where m and n designate arguments which are logically multiplied.
Logical Complement	NOT (m)	use .NOT. m	Where m designates the argument which is logically complemented.
Exclusive OR	IEOR (m,n)	use m .XOR. n	Where m and n designate arguments which are exclusively added.

4.3 Shift Operations

The Logical shift is implemented as a function subprogram. A right or left shift can be specified. Zeroes are propagated following the shifted value and the argument's sign is not extended.

- ISHFT (m,n)
  - m = the argument to be shifted.
  - n = specifies the number of positions to be shifted and the direction of the shift thus:
    - n < 0 shift right
    - n > 0 shift left
    - n = 0 no shift

The absolute value of n should not exceed fifteen positions. If it does, the value of the function will be zero.

5 SYSTEM DIRECTIVES

This chapter describes FORTRAN callable subroutines for effecting the IAS System Directives. Only the formats of the directives are described. The functions of the directives are described more fully in the IAS System Directives Reference Manual.

5.1 ABORT

Subroutine to issue an ABORT directive.

CALLING SEQUENCE

CALL ABORT (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be aborted (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.2 ALTPRI

Subroutine to instruct the system to alter the priority of a specified task to a new priority indicated in the call.

CALLING SEQUENCE

CALL ALTPRI ([TSK],[IPRI],[IDS])

ARGUMENTS

TSK
A 2-word Radix-50 name of the task whose priority is to be altered. If this argument is omitted, the default is the calling task.

IPRI
A 1-word integer value for the new priority. If this argument is omitted, a priority of 50 (decimal) is used.

IDS
A 1-word integer variable for the directive status. If this argument is omitted, status information is not provided.

5.3 ASNLUN

Subroutine to issue an ASSIGN LUN directive to assign a Logical Unit Number (LUN) to a physical device.

CALLING SEQUENCE

CALL ASNLUN (ILUN,IDEV,IUNT,[IDS])

ARGUMENTS

ILUN
An integer containing a Logical Unit Number.

IDEV
An integer (format: 1A2) containing a device name.

IUNT
An integer containing a device unit number.

IDS
An integer variable to receive the Directive Status Word.

5.4 ATRG

Subroutine to issue an ATTACH REGION directive to attach the issuing task to a static common region or to a named dynamic region.

CALLING SEQUENCE

CALL ATRG (IRDB,[IDS])

ARGUMENTS

IRDB
An 8-word integer array containing a region definition block.

IDS
An integer variable to receive the Directive Status Word.

5.5 CANALL

Subroutine to issue a CANCEL SCHEDULED REQUESTS directive to cancel all scheduled requests for an indicated task.

CALLING SEQUENCE

CALL CANALL (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character task name in Radix-50.

IDS
An integer variable to receive the Directive Status Word.

5.6 CANMT

Subroutine to issue a CANCEL MARK TIME directive.

CALLING SEQUENCE

CALL CANMT ([IEFN],[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag number.

IDS
An integer variable to receive the Directive Status Word.

When an Event Flag Number is specified, only MARK TIME requests (made by the issuing task) that would set that event flag are cancelled.

When an Event Flag is not specified, all MARK TIME requests (made by the issuing task) are cancelled.

5.7 CANOBY

Subroutine to issue a CANCEL SCHEDULED REQUESTS directive to cancel only scheduled requests for an indicated task made by another indicated task.

CALLING SEQUENCE

CALL CANOBY (SCHLED,[SCHLER],[IDS])

ARGUMENTS

SCHLED
The task name (Radix-50) of the scheduled task.

SCHLER
The task name (Radix-50) of the scheduler task.

IDS
An integer variable to receive the Directive Status Word.

When a Scheduler Task is not specified, the issuing task is taken as the Scheduler Task.

5.8 CLREF

Subroutine to issue a CLEAR EVENT FLAG directive.

CALLING SEQUENCE

CALL CLREF (IEFN,[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag number.

IDS
An integer variable to receive the Directive Status Word.

5.9 CRAW

Subroutine to issue a CREATE ADDRESS WINDOW directive to create a new virtual address window by establishing its virtual address base and size.

CALLING SEQUENCE

CALL CRAW (IWDB[,IDS])

ARGUMENTS

IWDB
An 8-word integer array containing a window definition block.

IDS
An integer variable to receive the Directive Status Word.

.10 CRRG

Subroutine to issue a CREATE REGION directive to create a dynamic region in a system-controlled partition and optionally attach the region to the issuing task.

CALLING SEQUENCE

CALL CRRG (IRDB[,IDS])

ARGUMENTS

IRDB
An 8-word integer array containing a region definition block.

IDS
An integer variable to receive the Directive Status Word.

.11 DECLAR

Subroutine to issue a DECLARE SIGNIFICANT EVENT directive.

CALLING SEQUENCE

CALL DECLAR ([IEFN],[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag Number.

IDS
An integer variable to receive the Directive Status Word.

.12 DISABL

Subroutine to issue a DISABLE directive.

CALLING SEQUENCE

CALL DISABL (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2 word 1- to 6-character name of the task be be disabled (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.13 DISCKP

Subroutine to issue a DISABLE CHECKPOINTING directive.

CALLING SEQUENCE

CALL DISCKP

5.14 DTRG

Subroutine to issue a DETACH REGION directive to detach the issuing task from a specified, previously attached region.

CALLING SEQUENCE

CALL DTRG (IRDB [,IDS])

ARGUMENTS

IRDB
An 8-word integer array containing a region definition block.

IDS
An integer variable to receive the Directive Status Word.

5.15 ELAW

Subroutine to issue an ELIMINATE ADDRESS WINDOW directive to delete an existing address window, unmapping it first if necessary.

CALLING SEQUENCE

CALL ELAW (IWDB [,IDS])

ARGUMENTS

IWDB
An 8-word integer array containing a window definition block.

IDS
An integer variable to receive the Directive Status Word.

5.16 ENABLE

Subroutine to issue an ENABLE directive.

CALLING SEQUENCE

CALL ENABLE (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be enabled (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.17 ENACKP

Subroutine to issue an ENABLE CHECKPOINTING directive.

CALLING SEQUENCE

CALL ENACKP

5.18 ENASTR

Subroutine to issue an ENABLE AST RECOGNITION directive.

CALLING SEQUENCE

CALL ENASTR

5.19 EXECUT

Subroutine to issue an EXECUTE directive.

CALLING SEQUENCE

CALL EXECUT (TSKNAM,[IOP],[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be executed (in Radix-50 form).

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1) –Radix-50 partition name (1st Half)**
- IOP(2) –Radix-50 partition name (2nd Half)**
- IOP(3) –Run priority**
- IOP(4) –UIC (User Identification Code)**

  –Byte 0 - member code; Byte 1 - group code.
- **Zero when none specified

IDS
An integer to receive the Directive Status Word.

5.20 EXITIF

Subroutine to issue an EXITIF event flag set directive.

CALLING SEQUENCE

CALL EXITIF (IEFN,[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag Number.

IDS
An integer variable to receive the Directive Status Word.

The (unconditional) EXIT routine is described in the *FORTRAN User's Guide.*

5.21 EXTTSK

Subroutine to issue an EXTEND TASK directive.

CALLING SEQUENCE

CALL EXTTSK ([INC][,IDS])

ARGUMENTS

INC
A positive or negative number equal to the number of 32-word blocks by which a task is to be extended or reduced.

IDS
An integer variable to receive the Directive Status Word.

When INC is not specified, the task's size is made equal to its installed size.

5.22 FIXMEM

Subroutine to issue a FIX-IN-MEMORY directive.

CALLING SEQUENCE

CALL FIXMEM (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character name of the task to be fixed (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.23 GCMX

Subroutine to issue a GET MAPPING CONTEXT directive to cause the Executive to return a description of the current window-to-region mapping assignments.

CALLING SEQUENCE

CALL GCMX (IMCX [,IDS])

ARGUMENTS

IMCX
An integer array to receive the mapping context.

IDS
An integer variable to receive the Directive Status Word.

5.24 GETADR

Subroutine to insert argument addresses in an integer array.

CALLING SEQUENCE

CALL GETADR (IPRM,[ARG1],[ARG2],...,[ARGn])

ARGUMENTS

IPRM
An integer array.

- ARG1,...ARGn = Arguments whose addresses are to be inserted in IPRM. Arguments are inserted in the order specified. If an argument is not specified then the corresponding entry in IPRM is left unchanged.

The primary intent of this call is to facilitate the construction of the parameter array for the QIO directive subroutine.

5.25 GETCMN

Subroutine to issue a GET COMMON BLOCK PARAMETERS directive.

CALLING SEQUENCE

CALL GETCMN (CMNNAM,IBUF,[ISW])

ARGUMENTS

CMNNAM
A 2-word, 1- to 6-character Common Block name in Radix-50 form.

IBUF
An 8-word integer array to receive Common Block Parameters.

ISW
An integer variable to receive the Directive Status Word.

5.26 GETLUN

Subroutine to issue a GET LUN INFORMATION directive.

CALLING SEQUENCE

CALL GETLUN (ILUN,IDATA,[IDS])

ARGUMENTS

ILUN
An integer containing a Logical Unit Number.

IDATA
A 6-word integer array to receive LUN information.

IDS
An integer variable to receive the Directive Status Word.

5.27 GETMCR

Subroutine to issue a GET MCR COMMAND LINE directive.

CALLING SEQUENCE

CALL GETMCR (IBUF,[ISW])

ARGUMENTS

IBUF
An 80-byte array to receive the Command line and a terminating carriage return character.

ISW
An integer variable to receive the Directive Status Word.

5.28 GETPAR

Subroutine to issue a GET PARTITION PARAMETERS directive.

CALLING SEQUENCE

CALL GETPAR ([PARNAM],IBUF,[IDS])

ARGUMENTS

PARNAM
A 2-word 1- to 6-character partition name in Radix-50 form. If this value is omitted, the system defaults to the partition in which the task is currently running.

IBUF
A 3-word integer array to receive partition parameters.

IDS
An integer variable to receive the Directive Status Word.

5.29 GETREG

Subroutine to issue a GET REGION PARAMETERS directive to cause an indicated 3-word buffer to be filled with information regarding the specified region to which the task is attached.

CALLING SEQUENCE

CALL GETREG (IRID,IBUF [,IDS])

ARGUMENTS

IRID
An integer which contains the region ID.

IBUF
A 3-word integer array to receive the region information.

IDS
An integer variable to receive the Directive Status Word.

5.30 GETTSK

Subroutine to issue a GET TASK PARAMETERS directive.

CALLING SEQUENCE

CALL GETTSK (IDATA,[IDS])

ARGUMENTS

IDATA
A 16-word integer array to receive task parameters.

IDS
An integer variable to receive the Directive Status Word.

).31 INASTR

Subroutine to issue an INHIBIT AST RECOGNITION directive.

CALLING SEQUENCE

CALL INASTR

).32 MAP

Subroutine to issue a MAP ADDRESS WINDOW directive to map an existing window onto an attached region.

CALLING SEQUENCE

CALL MAP (IWDB [,IDS])

ARGUMENTS

IWDB
An 8-word integer array containing a window definition block.

IDS
An integer variable to receive the Directive Status Word.

).33 MARK

Subroutine to issue a MARK TIME directive.

CALLING SEQUENCE

CALL MARK (IEFN,IDM,IDU,[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag Number.

IDM
An integer containing the time delta magnitude.

IDU
An integer containing the time delta units (1-4).

IDS
An integer variable to receive the Directive Status Word.

5.34 MNLOAD

Subroutine to issue a MANUAL-LOAD request for an overlay segment.

CALLING SEQUENCE

[EXTERNAL TRPSUB]

CALL MNLOAD (SEGNAM,[IEFN],[IOS],[TRPSUB],[ISW])

ARGUMENTS

SEGNAM
A 2-word (real) variable containing the segment name in Radix-50 format.

IEFN
An optional Event Flag Number to be used for an asynchronous load request.

IOS
An optional 2-word integer array to contain the I/O status as described for the QIO directive.

TRPSUB
An optional AST trap routine to be entered at the completion of the load request (See NOTE.)

ISW
An optional integer variable to receive the results of the request. One of the following values will be returned.

- +1—Request was successfully executed.
- -1—Unsuccessful load request or bad parameters.

NOTE: The Asynchronous System Trap (AST) subroutine can be supplied with the contents of the I/O Status Block if specified in the main program, by defining the appropriate dummy variable as:

```
SUBROUTINE TRPSUB(IOS)
DIMENSION IOS(2)
```

In this case, the Status Block must be specified in the call to "MNLOAD".

Subroutine Function

MNLOAD sets up an argument parameter block from the list of calling arguments, then transfers control to "$LOAD" to get the segment.
The reader is referred to the *IAS Task Builder Reference Manual* for a complete description of the MANUAL LOAD feature.

5.35 PWRUP

Task-resident subroutine to cause another subroutine to be entered upon recovery from a power failure, or remove a power fail AST.

CALLING SEQUENCE (TO ESTABLISH A POWERFAIL AST)

EXTERNAL SUBNAM

CALL PWRUP (SUBNAM)

CALLING SEQUENCE (TO REMOVE A POWERFAIL AST)

CALL PWRUP

ARGUMENTS

SUBNAM
The name of a subroutine to be executed upon power recovery. The PWRUP subroutine will effect a CALL SUBNAM (no arguments). SUBNAM is called as a result of a power recovery AST (Asynchronous System Trap), and therefore can be controlled at critical points by using INHIBIT and ENABLE AST recognition directives.

5.36 QIO

Subroutine to issue a QIO directive. This subroutine is provided to supplement the Process Control I/O calls described in 3. See also Chapter 5, Section 5.67 for details of the WTQIO directive.

CALLING SEQUENCE

CALL QIO (IFNC,ILUN,[IEFN],[IPRI],[ISTAT],[IPRM],[ISW])

ARGUMENTS

IFNC
An integer specifying the device function code.

ILUN
An integer specifying the Logical Unit Number.

IEFN
An integer specifying Event Flag Number.

IPRI
An integer specifying QIO request priority.

ISTAT

A 2-word integer array to receive device status.

IPRM

A 6-word integer array containing device dependent parameters to be placed in parameter words 1-6 of the

Directive Parameter Block (DPB). See the *IAS Device Handlers Reference Manual* for parameters required by specific devices.

ISW

An integer variable to receive the Directive Status Word.

Subroutine "GETADR" can be called to insert addresses in selected elements of the IPRM array. (See Section 5.24.)

For example:

```
      INTEGER*2 ISTAT(2), IPRM(6),IBUF(40)
      DATA IWVB/"11000/    !I/O FUNCTION CODE IO.WVB WRITE VIRTUAL
                !BLOCK
      CALL GETADR(IPRM,IBUF) !QIO REQUIRES ADDRESS OF BUFFER
      IPRM(2)=80        !AND LENGTH OF BUFFER (DO NOT PUT LENGTH
                !IN GETADR CALL)
C     WRITE OUT BUFFER
      CALL QIO(IWVB,2,1,,ISTAT,IPRM,ISW)
```

5.37 READEF

Subroutine to issue a READ EVENT FLAG directive.

CALLING SEQUENCE

CALL READEF (IEFN,[IDS])

ARGUMENTS

IEFN

An integer containing an Event Flag Number.

IDS

An integer variable to receive the Directive Status Word.

5.38 READSW

Subroutine to read the Sense Switches.

CALLING SEQUENCE

CALL READSW (ISW)

ARGUMENTS

ISW
An integer variable to contain the console Sense Switch settings.

Specific bits can be tested by using the functions described in 4.

5.39 RECEIV AND VRECEV

Subroutine to issue a RECEIVE DATA directive to receive a 15-word block of data.

CALLING SEQUENCE

CALL RECEIV ([TSKNAM],IDATA,[IPRV],[IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character sender task name in Radix-50 form. This parameter allows the user to specify receipt of messages from specific tasks.

IDATA
A 15-word integer array for data received.

IPRV
A full-word logical variable set as follows:

Directive failed:	IPRV unchanged.
Receiving task suspended:	IPRV unchanged.
Sender task privileged:	IPRV set to .TRUE. (-1).
Sender task not privileged:	IPRV set to .FALSE. (0).

IDS
An integer variable to receive the Directive Status Word.

When a sender task is not specified, the first data sent by any task is received.

Subroutine to issue a RECEIVE DATA to receive an n-word block of data.

CALLING SEQUENCE

CALL VRECEV ([TSKNAM],IDATA,[IBUFSZ],[ITI],[IPRV],[IDS])

Only those arguments that differ from the RECEIV call are described.

ARGUMENTS

IDATA

An integer array for data received. It must be able to accommodate the number of words specified by IBUFSZ.

IBUFSZ

The size in words of the data to be received. It can be up to 255 words. If omitted, 13 words are used.

ITI

An integer variable for the TI of the sender.

When a sender task is not specified, the first data sent by any task is received.

5.40 RECOEX AND VRECEX

Subroutine to issue a RECEIVE DATA OR EXIT directive to receive a block of data, or exit if no data is queued.

CALLING SEQUENCE

CALL RECOEX ([TSKNAM],IDATA,[IPRV],[IDS])

ARGUMENTS

TSKNAM

A 2-word 1- to 6-character sender task name in Radix-50 form.

IDATA

A 15-word integer array for data received. The first two words are the name of the sending task in Radix-50 format.

IPRV

A full-word logical variable set as follows:

Directive failed: IPRV unchanged Receiving task suspended: IPRV unchanged Sender task privileged: IPRV set to .TRUE. (-1) Sender task not privileged: IPRV set to .FALSE. (0)

IDS

An integer variable to receive the Directive Status Word.

When a sender task is not specified, the first data sent by any task is received.

Subroutine to issue a RECEIVE DATA OR EXIT directive to receive an n-word data block.

CALLING SEQUENCE

CALL VRECEX ([TSKNAM],IDATA,[IBUFSZ],[ITI],[IPRV],IDS])

Only those arguments that differ from the RECOEX call are described.

ARGUMENTS

IDATA
An integer array for data received. It must be able to accommodate the number of words specified by IBUFSZ plus two. The first two words are the name of the sending task in Radix-50 format.

IBUFSZ
The size in words of the data to be received. It can be up to 255 words. If omitted, the buffer size is 13 words.

ITI
An integer variable in which to receive the TI of the sender.

When a sender task is not specified, the first data sent by any task is received.

.41 RECOSP AND VRECSP

Subroutine to issue a RECEIVE DATA OR SUSPEND directive to receive a block of data, or suspend the issuing task if none has been queued.

CALLING SEQUENCE

CALL RECOSP ([TSKNAM],IDATA,[IPRV],[IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character sender task name in Radix-50 form.

IDATA
A 15-word integer array for data received. The first two words are the name of the sending task in Radix-50 format.

IPRV
A full-word logical variable, set as follows:

Directive failed:	IPRV unchanged.
Receiving task suspended:	IPRV unchanged.
Sender task privileged:	IPRV set to .TRUE. (-1).
Sender task not privileged:	IPRV set to .FALSE. (0).

IDS
An integer variable to receive the Directive Status Word.

When a sender task is not specified, the first data sent by any task is received.

Subroutine to issue a RECEIVE DATA OR SUSPEND directive to receive an n-word data block.

CALLING SEQUENCE

CALL VRECSP ([TSKNAM],IDATA,[IBUFSZ],[ITI],[IPRV],[IDS])

Only those arguments that differ from the RECOSP call are described.

ARGUMENTS

IDATA
An integer array for data received. It must be able to accommodate the number of words specified by IBUFSZ plus two. The first two words are the name of the sending task in Radix-50 format.

IBUFSZ
The size in words of the data to be received. It can be up to 255 words. If omitted, the buffer size is 13 words.

ITI
An integer variable in which to receive the TI of the sender.

When a sender task is not specified, the first data sent by any task is received.

5.42 RECOST AND VRECST

Subroutine to issue a RECEIVE DATA OR STOP directive to receive a 15-word block of data, or stop the issuing task if none has been queued.

CALLING SEQUENCE

CALL RECOST ([TSKNAM],IDATA,[IPRV],IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character sender task name in Radix-50 form.

IDATA
A 15-word integer array for data received. The first two words are the name of the sending task in Radix-50 format.

IPRV
A full-word logical variable, set as follows:

Directive failed:	IPRV unchanged.
Receiving task suspended:	IPRV unchanged.

Sender task privileged:	IPRV set to .TRUE. (-1).
Sender task not privileged:	IPRV set to .FALSE. (0).

IDS

An integer variable to receive the Directive Status Word.

When a sender task is not specified, the first data sent by any task is received.

Subroutine to issue a RECEIVE DATA or STOP directive to receive an n-word block of data, or stop the issuing task if none has been queued.

CALLING SEQUENCE

CALL VRECST ([TSKNAM],IDATA,[IBUFSZ],[ITI],[IPRV],[IDS])

Only those arguments that differ from the RECOST call are described.

ARGUMENTS

IDATA

An integer array for data received. It must be able to accomodate the number of words specified by IBUFSZ plus two. The first two words are the name of the sending task in Radix-50 format.

IBUFSZ

The size in words of the data to be received. It can be up to 255 words. If omitted, the buffer size is 13 words.

ITI

An integer variable in which to receive the TI of the sender.

5.43 REQUES

Subroutine to issue a REQUEST directive.

CALLING SEQUENCE

CALL REQUES (TSKNAM,[IOP],[IDS])

ARGUMENTS

TSKNAM

A 2-word 1- to 6-character name of the task being requested (in Radix-50 form).

IOP

A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority**

- IOP(4)—UIC (User Identification Code)**
  - Byte 0 - member code; Byte 1 - group code
- **zero when none specified

IDS
An integer variable to receive the Directive Status Word.

5.44 RESORU

Subroutine to issue a RESUME OR UNSTOP directive.

CALLING SEQUENCE

CALL RESORU (TSKNAM[,IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be resumed or unstopped (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.45 RESUME

Subroutine to issue a RESUME directive.

CALLING SEQUENCE

CALL RESUME (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be resumed (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.46 RREF

Subroutine to issue a RECEIVE BY REFERENCE directive to cause the next packet in the receive-by-reference queue of the issuing (receiver) task to be dequeued.

CALLING SEQUENCE

CALL RREF (IWDB,[ISRB],[ITIBUF],[IDS])

ARGUMENTS

IWDB
An integer array containing a window definition block.

ISRB
A 10-word integer array to be used as the receive buffer.

ITIBUF
An 8-word integer array with the first word filled with the terminal identification of the sending task.

IDS
An integer variable to receive the Directive Status Word.

5.47 RUN

Subroutine to issue a RUN directive.

CALLING SEQUENCE

CALL RUN (TSKNAM,[IOP],ISD,ISU,[IRI],[IRU],[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be run (in Radix-50 form).

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority**
- IOP(4)—UIC (User Identification Code)**
  - Byte 0 - member code; Byte 1 - group code
- **Zero when none specified

ISD
A schedule delta magnitude.

ISU
A schedule delta unit (1-4).

IRI
A ıeschedule interval.

IRU
A reschedule unit (1-4).

IDS
An integer variable to receive the Directive Status Word.

5.48 SCHED

Subroutine to issue a SCHEDULE directive.

CALLING SEQUENCE

CALL SCHED (TSKNAM,[IOP],IH,IM,IS,IT,[IRI],[IRU],[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be scheduled (in Radix-50 form).

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority**
- IOP(4)—UIC (User Identification Code)**
  - Byte 0 - member code; Byte 1 - group code
- **Zero when none specified

IH
The scheduled hours.

IM
The scheduled minutes.

IS
The scheduled seconds.

IT
The scheduled ticks.

IRI
The rescheduled interval.

IRU
The rescheduled unit (1-4).

IDS
An integer variable to receive the Directive Status Word.

5.49 SEND AND VSEND

Subroutine to issue a SEND DATA directive for a 13-word data block.

CALLING SEQUENCE

CALL SEND (TSKNAM,IDATA,[IEFN],IDS])

ARGUMENTS

TSKNAM
A 2-word 1- to 6-character receiver task name in Radix-50 form.

IDATA
A 13-word integer array of data to be sent.

IEFN
The number of an event flag to be set.

IDS
An integer variable to receive the Directive Status Word.

Subroutine to issue a SEND DATA directive for an n-word data block.

CALLING SEQUENCE

CALL VSEND (TSKNAM,IDATA,[IEFN],[IBUFSZ],[IPRI],[ITI],[IDS])

Only those arguments that differ from the SEND call are described.

ARGUMENTS

IDATA
An integer array of data to be sent. It must be able to accommodate the number of words specified by IBUFSZ.

IBUFSZ
The number of words to send. Up to 255 words can be sent. If IBUFSZ is omitted, 13 words are sent. See note below.

IPRI

The priority of the send (0 through 250). If IPRI is omitted, the priority of the sending task is used. See note below.

ITI

The TI of the task to which the date is to be sent. If ITI is omitted, the TI of the sending task is used.

NOTE: IEFN, IBUFSZ, IPRI, and ITI can be either constants or integer variables.

5.50 SETEF

Subroutine to issue a SET EVENT FLAG directive.

CALLING SEQUENCE

CALL SETEF (IEFN,[IDS])

ARGUMENTS

IEFN

An integer containing an Event Flag Number.

IDS

An integer variable to receive the Directive Status Word.

5.51 SNDROR AND VSNDRR

Subroutine to issue a SEND DATA AND REQUEST OR RESUME directive for a 13-word data block.

CALLING SEQUENCE

CALL SNDROR (TSKNAM,[IOP],IDATA,[IEFN],[IDS])

ARGUMENTS

TSKNAM

A 2-word 1- to 6-character receiver task name in Radix-50 form.

IOP

A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority**
- IOP(4)—UIC (User Identification Code)**

- **Zero when none specified.

IDATA
A 13-word integer array of data to be sent.

IEFN
The number of an event flag to be set.

IDS
An integer variable to receive the Directive Status Word.

Subroutine to issue a SEND DATA AND REQUEST OR RESUME directive for an n-word data block.

CALLING SEQUENCE

CALL VSNDRR (TSKNAM,[IOP],IDATA,[IEFN],[IBUFSZ],[IPRI],[ITI],[IDS])

Only those arguments that differ from the SNDROR call are described.

ARGUMENTS

IDATA
An integer array of data to be sent. It must be able to accommodate the number of words specified by IBUFSZ.

IBUFSZ
The number of words to send. Up to 255 words can be sent. If IBUFSZ is omitted, 13 words are sent. See note below.

IPRI
The priority of the send (0 through 250). If IPRI is omitted, the priority of the sending task is used. See note below.

ITI
The ITI of the task to which the data is sent. If ITI is omitted, the TI of the sending task is used.

NOTE: IEFN, IBUFSZ, IPRI and ITI can be either constants or integer variables.

;.52 SPAWN

Subroutine to issue a SPAWN directive.

CALLING SEQUENCE

CALL SPAWN (TSKNAM,[IOP],[IEFN],[IESB],[ICMD],[ICMDL],[IVTUN],[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be spawned (in Radix-50 form).

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority**
- IOP(4)—UIC (User Identification Code)**
- **Zero when none specified

IEFN
The number of the event flag to be set when the spawned task exits.

IESB
An 8-element integer array for the exit status block.

ICMD
A byte or integer array containing a Command line to be passed to the spawned task.

ICMDL
The integer length in bytes of the Command line.

IVTUN
Reserved for compatability with RSX-11M systems.

IDS
An integer variable to receive the Directive Status Word.

5.53 SREF

Subroutine to issue a SEND BY REFERENCE directive to insert a packet containing a reference to a region into the receive-by-reference queue of a specified (receiver) task.

CALLING SEQUENCE

CALL SREF (TSKNAM,[IEFN],IWDB,[ISRB],[ITI],[IDS])

ARGUMENTS

TSKNAM
A 2-word, 1-to 6-character variable containing the name of the receiving task in Radix-50 format.

IEFN
An integer variable containing an event flag number.

IWDB
An 8-word integer array contaning a window definition block.

ISRB
An 8-word integer array containing additional information.

ITI
An integer array containing the terminal identification of the task to which the reference is to be sent.

IDS
An integer variable to receive the Directive Status Word.

5.54 SRRF

Subroutine to issue a SEND BY REFERENCE AND REQUEST OR RESUME directive to insert a packet containing a reference to a region into request, resume or unstop the execution of the receiver task.

CALLING SEQUENCE

CALL SRRF (TSKNAM,[IOP],[IEFN],IWDB,[ISRB][,IDS])

ARGUMENTS

TSKNAM
A 2-word, 1- to 6- character variable containing the name of the receiving task in Radix-50 format.

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix 50 partition name (1st half)**
- IOP(2)—Radix 50 partition name (2nd half)**
- IOP(3)—Run priority**
- IOP(4)—UIC (User Identification Code)**
- **Zero when none specified

IEFN
An integer containing an event flag number.

IWDB
An 8-word integer array containing the window definition block.

ISRB
An 8-word integer array containing additional information.

IDS
An integer variable to receive the Directive Status Word.

5.55 SSWTCH

Subroutine to test a Sense Switch.

CALLING SEQUENCE

CALL SSWTCH (I,J)

ARGUMENTS

I
An integer specifying the switch to be tested (0 to 15).

J
An integer variable to receive the results of the test.

- J=2 if switch was off.
- J=1 if switch was on.

5.56 STOPFR

Subroutine to issue a STOP FOR SINGLE EVENT FLAG directive.

CALLING SEQUENCE

CALL STOPFR (IEFN,[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag Number.

IDS
An integer variable to receive the Directive Status Word.

5.57 STOPOR

Subroutine to issue a STOP FOR LOGICAL OR OF EVENT FLAGS directive.

CALLING SEQUENCE

CALL STOPOR (IEF1,IEF2,...IEFn)

ARGUMENTS

IEF1...IEFN
A list of Event Flag Numbers to be taken as the set of event flags to be specified in the directive.

5.58 STOPTK

Subroutine to issue a STOP directive.

CALLING SEQUENCE

CALL STOPTK

5.59 SUSPND

Subroutine to issue a SUSPND directive.

CALLING SEQUENCE

CALL SUSPND

5.60 SYNC

Subroutine to issue a SYNC directive.

CALLING SEQUENCE

CALL SYNC (TSKNAM,[IOP],IOM,IOU,ISYU,[IRI],[IRU],[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be synchronized (in Radix-50 form)

IOP
A 4-word integer array containing the following optional parameters:

- IOP(1)—Radix-50 partition name (1st half)**
- IOP(2)—Radix-50 partition name (2nd half)**
- IOP(3)—Run priority
- IOP(4)—(User Identification Code)**
  - Byte 0 - member code; Byte 1 - group code

- **Zero when none specified

IOM
A sync offset magnitude.

IOU
A sync offset unit (1-4).

ISYU
A synchronization unit (1-4).

IRI
A reschedule interval.

IRU
A reschedule unit (1-4).

IDS
An integer variable to receive the Directive Status Word.

5.61 UNFIX

Subroutine to issue an UNFIX directive.

CALLING SEQUENCE

CALL UNFIX (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2 word 1- to 6-character name of the task to be unfixed (in Radix-50 form)

IDS
An integer variable to receive the Directive Status Word.

5.62 UNMAP

Subroutine to issue an UNMAP ADDRESS WINDOW directive to unmap a specified window.

CALLING SEQUENCE

CALL UNMAP (IWDB [,IDS])

ARGUMENTS

IWDB
An 8-word integer array containing a window definition block.

IDS
An integer variable to receive the Directive Status Word.

5.63 UNSTOP

Subroutine to issue an UNSTOP directive.

CALLING SEQUENCE

CALL UNSTOP (TSKNAM,[IDS])

ARGUMENTS

TSKNAM
The 2-word 1- to 6-character name of the task to be unstopped (in Radix-50 form).

IDS
An integer variable to receive the Directive Status Word.

5.64 WAITFR

Subroutine to issue a WAIT FOR SINGLE EVENT FLAG directive.

CALLING SEQUENCE

CALL WAITFR (IEFN,[IDS])

ARGUMENTS

IEFN
An integer containing an Event Flag Number.

IDS
An integer variable to receive the Directive Status Word.

5.65 WFLOR

Subroutine to issue a WAITFOR LOGICAL OR directive.

CALLING SEQUENCE

CALL WFLOR (IEF1,IEF2,...IEFn)

ARGUMENTS

IEF1,,,IEFN
A list of Event Flag Numbers to be taken as the set of event flags to be specified in the directive.

5.66 WFSNE

Subroutine to issue a WAIT FOR SIGNIFICANT EVENT directive.

CALLING SEQUENCE

CALL WFSNE

5.67 WTQIO

Subroutine to perform a Queue I/O and then wait for the specified event flag. All other actions are the same as for both QIO and WAITFR.

CALLING SEQUENCE

CALL WTQIO (same parameters as QIO)

ARGUMENTS

See QIO descriptions.

A DSW Return Codes

The symbols listed below are associated with the directive status codes returned by the Executive.

```
.MCALL DRERR$
DRERR$
```

```
;
; STANDARD ERROR CODES RETURNED BY DIRECTIVES IN THE DIRECTIVE STATUS
; WORD
;
        IE.UPN   -01.   INSUFFICIENT DYNAMIC STORAGE
        IE.INS   -02.   SPECIFIED TASK NOT INSTALLED
        IE.PTS   -03.   PARTITION TOO SMALL FOR TASK
        IE.UNS   -04.   INSUFFICIENT DYNAMIC STORAGE FOR SEND
        IE.ULN   -05.   UNASSIGNED LUN
        IE.HWR   -06.   DEVICE HANDLER NOT RESIDENT
        IE.ACT   -07.   TASK NOT ACTIVE
        IE.ITS   -08.   DIRECTIVE INCONSISTENT WITH TASK STATE
        IE.FIX   -09.   TASK ALREADY FIXED/UNFIXED
        IE.CKP   -10.   ISSUING TASK NOT CHECKPOINTABLE
        IE.TCH   -11.   TASK IS CHECKPOINTABLE
        IE.RBS   -15.   RECEIVE BUFFER IS TOO SMALL
        IE.PRI   -16.   PRIVILEGE VIOLATION
        IE.RSU   -17.   RESOURCE IN USE
        IE.NSW   -18.   NO SWAP SPACE AVAILABLE
        IE.ILV   -19.   ILLEGAL VECTOR SPECIFIED
;
;
;
        IE.AST   -80.   DIRECTIVE ISSUED/NOT ISSUED FROM AST
        IE.MAP   -81.   ILLEGAL MAPPING SPECIFIED
        IE.IOP   -83.   WINDOW HAS I/O IN PROGRESS
        IE.ALG   -84.   ALIGNMENT ERROR
        IE.WOV   -85.   ADDRESS WINDOW ALLOCATION OVERFLOW
        IE.NVR   -86.   INVALID REGION ID
        IE.NVW   -87.   INVALID ADDRESS WINDOW ID
        IE.ITP   -88.   INVALID TI PARAMETER
        IE.IBS   -89.   INVALID SEND BUFFER SIZE
        IE.LNL   -90.   LUN LOCKED IN USE
        IE.IUI   -91.   INVALID UIC
        IE.IDU   -92.   INVALID DEVICE OR UNIT
        IE.ITI   -93.   INVALID TIME PARAMETERS
        IE.PNS   -94.   PARTITION/REGION NOT IN SYSTEM
        IE.IPR   -95.   INVALID PRIORITY
        IE.ILU   -96.   INVALID LUN
        IE.IEF   -97.   INVALID EVENT FLAG
        IE.ADP   -98.   PART OF DPB OUT OF USER'S SPACE
        IE.SDP   -99.   DIC OR DPB SIZE INVALID
```

Index

A

ABORT • 5–1
AI (AIW) • 3–6, 3–7, 3–8, 3–9
AIRD (AIRDW) • 3–4, 3–5
AISQ (AISQW) • 3–3, 3–4
ALTPRI • 5–1
AO (AOW) • 3–7, 3–8
AOSC • 3–8
AOSCO • 3–9
Arguments
  integer • 1–2
  omitting • 1–2
  optional • 1–2
ARTG • 5–2
ASNLUN • 5–2

C

CALL EXIT • 2–5
CANALL • 5–2, 5–3
CANMT • 5–3
CANOBY • 5–3, 5–4
CLREF • 5–4
CRAW • 5–4
CRRG • 5–5
CTDI • 3–15, 3–16, 3–17
CTTI • 3–27, 3–28

D

DECLAR • 5–5
DFDI • 3–15, 3–18
DFTI • 3–27, 3–29
DI (DIW) • 3–9, 3–10
DIFCS • 3–9, 3–15
DISABL • 5–5, 5–6
DISCKP • 5–6
DOFL • 3–22, 3–24, 3–25
DOFM • 3–18, 3–21, 3–22
DOL (DOLW) • 3–22, 3–23
DOM (DOMW) • 3–18, 3–19, 3–20
DOSL • 3–22, 3–23, 3–24
DOSM • 3–18, 3–20, 3–21
DTRG • 5–6

E

ELAW • 5–6, 5–7
ENABLE • 5–7
ENACKP • 5–7
ENASTR • 5–7
Error conditions • 1–3
EXECUT • 5–8
Executive interfaces • 2–1
  START
  TRNON
  WAIT
EXITIF • 5–8
EXITTSK • 5–9

F

FIXMEM • 5–9

G

GCMX • 5–9, 5–10
GETADR • 1–3, 5–10
GETCMN • 5–10
GETLUN • 5–11
GETMCR • 5–11
GETPAR • 5–11, 5–12
GETREG • 5–12
GETTSK • 5–12

I

IAS
  See Instrument Standard of America
INASTR • 5–13
Instrument Standard of America • 1–1

Integer arguments • 1–2

K

KWII-L • 1–3
KWII-P • 1–3

L

LUN assignments • 3–2

M

MAP • 5–13
MARK • 5–13, 5–14
MNLOAD • 5–14

O

Optional arguments • 1–2

P

PWRUP • 5–15

Q

QIO • 1–3, 3–1, 5–15, 5–16

R

RBCD • 3–9, 3–14
RCIPT • 3–9, 3–12, 3–13
RCSPT • 3–9, 3–10, 3–12
RDDI • 3–15, 3–17, 3–18
RDTI • 3–27, 3–28, 3–29
READEF • 5–16
READSW • 5–16, 5–17
RECEIV • 5–17
RECOEX • 5–18
RECOSP • 5–19
RECOST • 5–20, 5–21
REQUES • 5–21, 5–22
RESORU • 5–22
RESUME • 5–22
RREF • 5–22, 5–23
RSTI • 3–25, 3–26
RUN • 5–23, 5–24

S

SAITFR • 5–33
SCHED • 5–24, 5–25
SCTI • 3–25, 3–26, 3–27
SEND • 5–25
SETEF • 5–26
SFLOR • 5–33, 5–34
Shift Operations • 4–2
SNDROR • 5–26, 5–27
SPAWN • 5–27, 5–28
SREF • 5–28, 5–29
SRRF • 5–29
SSWTCH • 5–30
START • 2–2
STOP • 2–5
STOPOR • 5–30, 5–31
STOPRF • 5–30
STOPTK • 5–31
STQIO • 5–34
SUB • 3–2
SUSPND • 5–31
SYNC • 5–31, 5–32
SYSLIB.OLB • 1–1
System clocks
  KWII-L
  KWII-P

T

Task names • 1–2
Task scheduling • 1–3
TRNON • 2–3

U

UNFIX • 5–32
UNMAP • 5–32, 5–33
UNSTOP • 5–33

V

VRECEV • 5–17, 5–18
VRECEX • 5–18, 5–19
VRECSP • 5–19, 5–20
VRECST • 5–20, 5–21
VSEND • 5–25, 5–26
VSNDRR • 5–26, 5–27

W

WAIT • 2–4, 3–1
WFSNE • 5–34
WTQIO • 1–3

IAS
FORTRAN Special Subroutines Reference Manual
AA-H001B-TC

Reader's Comments

This form is for document comments only. Digital will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

□ Assembly language programmer
□ Higher-level language programmer
□ Occasional programmer (experienced)
□ User with little programming experience
□ Student programmer
□ Other (please specify)____________________

Name____________________Date__________
Organization____________________
Street____________________
City____________________State__________Zip Code__________
or Country

Do Not Tear - Fold Here and Tape

digital™

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO 33 MAYNARD MASS

POSTAGE WILL BE PAID BY ADDRESSEE

IAS Engineering/Documentation
Digital Equipment Corporation
5 Wentworth Drive GSF/L20
Hudson, NH 03051-4929

Do Not Tear - Fold Here