REAL TIME INPUT/OUTPUT SYSTEM (RTIOS)

User's Manual

093–000148–00

Ordering No. 093-000148

Printed in the United States of America
Rev. 00, May 1975

This document is intended for ECLIPSE™ * system users. It was derived from 093-000095, which is primarily for NOVA® * system users. There may still be references to NOVA equipment, documents, and processes within this manual; if so, be assured that the references also apply to ECLIPSE systems.

Revision History:

093-000095

Original Release - September 1973
First Revision - February 1975

093-000148

Original Release - May 1975

This revision of the Real Time Input/Output System (RTIOS) 093-000148-00 is a major revision. A vertical bar or an asterisk in the margin of a page indicates substantive change or deletion, respectively, from 093-000095 revision 01.

PREFACE

The Real Time Input/Output System (RTIOS) User's Manual includes discussions of the following topics.

Chapter 1 is an introduction to RTIOS.

Chapter 2 is a detailed discussion of the FORTRAN IV calls associated with RTIOS.

Chapter 3 is a detailed discussion of the Assembly Language calls associated with RTIOS.

Chapter 4 discusses RTIOS system generation.

Appendix A is a summary of error messages.

Appendix B provides consideration for special products

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

CHAPTER 1

Introduction

The Real Time Input Output System (RTIOS) provides software support for Data General Corporation's process input/output devices. These devices come under the main classes:

Digital I/O (4066, 5602)
A/D input (4120 series, 4055, 4202-4085)
Interrupt input (4067, 5602-J)
miscellaneous (console switches, power fail restart, I/O bus switch)

The Real-Time Input/Output System (RTIOS) is a library of device drivers and subroutines that work with RDOS, MRDOS or RTOS operating systems in a multitasking environment to control input/output transfers between a user program and analog and digital sensor devices.

Additionally RTIOS supports high-speed data channel operations and analog/digital conversion with programmable gain, and provides automatic scanning of multiple input lines for external interrupts. The automatic scanning capability is particularly important in reducing processor overhead in an interrupt-driven multitasking environment.

COMMANDS

The RTIOS software supports Assembly Language calls as well as FORTRAN IV calls. Chapter 2 discusses the FORTRAN IV calls and Chapter 3 discusses the Assembly Language calls. User programs written in either FORTRAN or Assembly Language, with appropriate calls to RTIOS, can directly utilize the RTIOS device-independent assembly language interface.

The RTIOS FORTRAN IV interfaces are consistent with the standards set by the International Purdue Workshop on Industrial Computer Systems, and include all calls specified by ISA (Instrument Society of America) standard S61.1, FORTRAN Procedures for Executive Functions and Process Input/Output.

SOFTWARE ENVIRONMENT

RTIOS operates with Data General's Real-time Disk Operating System (RDOS), Mapped Real-time Disk Operating System (MRDOS), and the main memory-based Real-Time Operating System (RTOS). Whenever an I/O request for an analog/digital process device is made, the operating system refers it to RTIOS for handling. The operating system continues to perform its supervisory system functions, while delegating the interrupt and device handling functions for process I/O devices to RTIOS.

REAL TIME INTERFACE ROUTINES

Process input/output interfaces allows the user to read and set analog and digital devices. Two modes of control of these interfaces are provided by RTIOS. The first mode suspends the requesting task while the input or output is accomplished. The second mode permits the task to continue while the input or output is in progress. RTIOS statements which cause the requesting task to wait for I/O completion include the letter W. For example, the DI statement permits digital input without suspending the caller. The DIW statement permits digital inputs and suspends the caller until completion of the I/O.

HARDWARE REQUIREMENTS

The minimum hardware configuration for using the RTIOS Run-time Library, under RTOS, is an ECLIPSE computer with 16K bytes of read/write memory, real-time clock, and either a Teletype console or video display. The read/write memory required under RDOS is 32K bytes, and 48K bytes under MRDOS. The configuration with both RDOS and MRDOS must also include a 512K-byte fixed-or moving-head disk drive.

SYSTEM GENERATION

RTIOS system generation, discussed in Chapter 4, lets the user specify, and load into memory, only those portions of RTIOS needed to support the particular hardware configuration. In addition, the modular design of RTIOS permits the user to write assembly language device drivers for special-purpose hardware interfaces, and include them in the RTIOS library.

SPECIAL PRODUCTS

The users of special products 5602 are directed to Appendix B for special procedures.

END OF CHAPTER

CHAPTER 2

FORTRAN Calls

INTRODUCTION

This chapter discusses all of the available FORTRAN IV calls associated with RTIOS. All calls are consistent with standards set up by the International Purdue Workshop on Standardization of Industrial Computer Systems as adopted by the Instrument Society of America (ISA standard S61.1, FORTRAN Procedures for Executive Functions and Process I/O). Table 2-1 lists the FORTRAN calls and indicates their relationship to the hardware.

GENERAL

The calls to the RTIOS subroutines include an argument MSTAT which returns an integer indicating the status of the request. If the call is to a routine that suspends execution of the requesting program, MSTAT will be set upon return. If the call is to a routine that does not suspend execution of the requesting program, the requesting program must provide for periodic testing of the status of the request.

The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
$\geq$3 - error (=RDOS error +3)

When the user requires the Process I/O routines, the RTIOS library must be loaded before all other FORTRAN IV libraries, i.e. It must precede FMT.LB.

Each of the calls are discussed in the following pages and are of the following format:

CALL name (argument list)

where: name is the name of the call being used
argument list is all the required arguments

Table 2-1 FORTRAN Calls and Definitions

FORTRAN CALLS	DEFINITION	ASSOCIATED* HARDWARE
DI	Digital Input	Console Switches, 4066,4068,5602(1)
DIW	Digital Input with wait	Console Switches, 4066,4068,5602(1)
DOL	Digital Output	4066,4068,5602(0)
DOLW	Digital Output with wait	4066,4068,5602(0)
DOM	Momentary Digital Output	4066,4068,5602(0)
DOMW	Momentary Digital Output with wait	4066,4068,5602(0)
DIRA	Starts Input Scan and Return	5602 with Data Channel option
DIRAW	Start Input Scan and Wait for completion	
DORA	Start Output Scan and Return	
DORAW	Start Output Scan and Wait for completion	
DIRCW	Data Input with compare	
ISCAN	External Interrupt Input Scan	4067,5602(3,5,6)
AISQ	Sequential Scan	4055,4120,5602(7)
AISQW	Sequential Scan with wait	4055,5120,5602(7)
AIRD	Random Scan	4055,4120,5602(7)
AIRDW	Random Scan with wait	4055,4120,5602(7)
AISA	Sequential Scan with Data Channel	4055,4120 with DCH
AISAW	Sequential Scan with Data Channel and wait	4055,4120 with DCH
AISG	Sequential Gain	4055,4120,5602(7) with gain option
AISGW	Sequential Gain with wait	4055,4120,5602(7) with gain option
AIRG	Random Gain	4055,4120,5602(7) with gain option
AIRGW	Random Gain with wait	4055,4120,5602(7) with gain option
AIRB	Random Gain Data Channel	5602-X-DCH
AIRBW	Random Gain Data Channel with wait	5602-X-DCH
AISB	Sequential Gain Data Channel	4120,4055
AISBW	Sequential Gain Data Channel with wait	4120,4055
AOS	Analog Output in Sequential Order	4056,4180 series
AOSW	Analog Output in Sequential Order with wait	4056,4180 series
AO	Analog Output in Any Sequence	4056,4180 series
AOW	Analog Output in Any Sequence and wait	4056,4180 series
SCOPE	Scope Control	4120 with 4183 option
IOBUS	I/O Bus Switch Control	5470/8080 Bus Switch

* Where several hardware modes are available, the mode is indicated in parenthesis and corresponds to the mode indicated in the sysgen dialogue. See Chapter 4.

DIGITAL INPUT

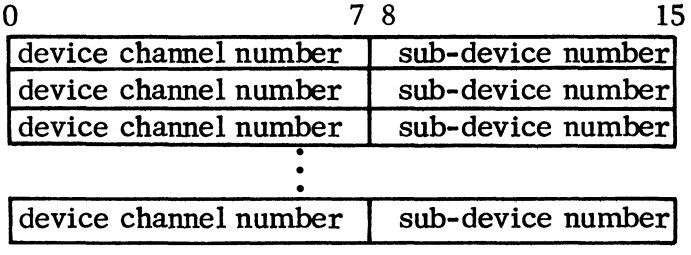
Subroutines DI and DIW allow the input of data from digital registers in a sequence which is independent of the input hardware (random order). Array IDEV controls the input sequence and allows the acquisition on an individual word basis. The format of the calls to these routines are:

```
CALL DI (NDEV, IDEV, IVAL, MSTAT)
CALL DIW (NDEV, IDEV, IVAL, MSTAT)
```

where: NDEV is an integer variable, integer constant, or integer array element specifying the number of devices to be accessed.

IDEV is an integer array containing the device identifiers; the device channel number *256 + sub device number (if any). There must be NDEV entries in the array. An entry may appear more than once.

The array IDEV is formatted as follows:



IVAL is an integer array in which the data value for the corresponding device identifier will be stored.

MSTAT is an integer variable or an integer array element that will return the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥ 3 - error (=RDOS error + 3)

DIGITAL OUTPUT

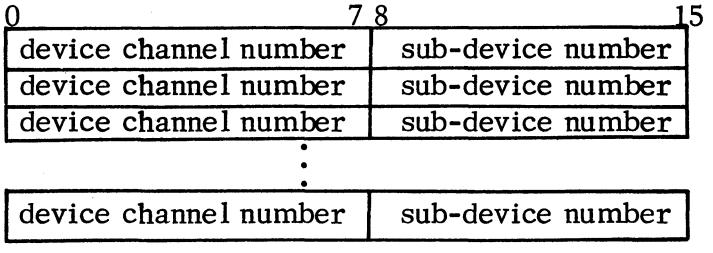
Subroutines DOL and DOLW allow the output of digital signals which can be latched in either the set state or the reset state. This type of output takes up to a full external digital word. Individual outputs are set when a corresponding bit in the output data is set ("1" state) and are reset when a corresponding bit is reset ("0" state). The formats of the calls to these routines are:

```
CALL DOL (NDEV, IDEV, IVAL, MASK, MSTAT)
CALL DOLW (NDEV, IDEV, IVAL, MASK, MSTAT)
```

where: NDEV is an integer variable, integer constant, or integer array element specifying the number of devices to be accessed.

IDEV is an integer array containing the device identifiers; device channel number *256 + sub device number (if any). There must be NDEV entries in the array. An entry may appear more than once.

The array IDEV is formatted as follows:



IVAL is an integer array from which data values are taken to be output to the device specified by the corresponding entry in IDEV.

MASK is an integer array of masks. The elements of the data array IVAL are masked (ANDed with) by corresponding mask array elements and output to corresponding devices.

MSTAT is an integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥3 - error (=RDOS error + 3)

MOMENTARY DIGITAL OUTPUT (WITH AND WITHOUT WAIT)

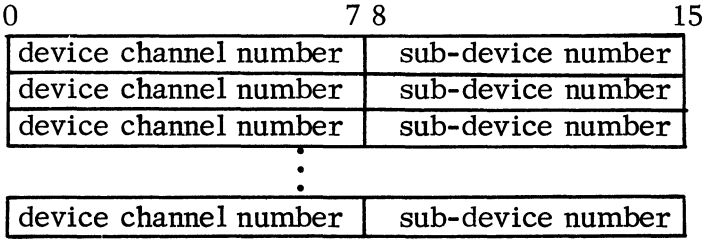
These subroutines DOM and DOMW allow momentary digital output. The format of the calls to these routines are:

CALL DOM (NDEV, IDEV, IVAL, ITICK, MSTAT)
CALL DOMW (NDEV, IDEV, IVAL, ITICK, MSTAT)

where: NDEV is an integer variable, integer constant, or integer array specifying the device count - (the number of output points).

IDEV is an integer array containing the device identifier; device number *256 plus sub device number (if any). There must be NDEV entries in the array. An entry may appear more than once.

The array IDEV is formatted as follows:



IVAL is the name of an integer array containing the data to be output to corresponding devices.

ITICK is an integer variable, integer constant or integer array element indicating the required number of clock ticks that the digital outputs will be held on.

MSTAT is the integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT.

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

DATA CHANNEL DIGITAL INPUT/OUTPUT SCAN

Data channel digital I/O scan supports the data channel extension (device code 64) associated with custom products in the 5602 series. The following are the currently available subroutines for data channel digital I/O scan:

DIRA - starts input scan and returns.
DIRAW - starts input scan and waits for transfer completion before return.
DORA - starts output scan and return.
DORAW - starts output scan and waits for transfer completion before return.
DIRCW - The data input is compared (ANDed) with the corresponding word in the mask array. The results are compared with the data array, and if the values are not equal, the unmasked data value and the changed device identifier are returned.

If asynchronous transfers take place or other tasks are alive the user should insure that arguments storage assignments are not released or over-written.

The formats of the subroutine calls are:

```
CALL DIRA    (NDEV, IDEV, IVAL, MSTAT)
CALL DIRAW   (NDEV, IDEV, IVAL, MSTAT)
CALL DORA    (NDEV, IDEV, IVAL, MSTAT)
CALL DORAW   (NDEV, IDEV, IVAL, MSTAT)
CALL DIRCW   (NDEV, IDEV, IVAL, MASK, CDEV, NVAL, MSTAT)
```

where: NDEV is an integer variable, integer constant, or integer array element specifying the number of digital registers to be read.

IDEV is an integer array containing the sub-device numbers) of the digital registers to be scanned.

IVAL is an integer variable or an integer array, into which the data is stored on input, from which it is taken for output, and with which it is compared for the changed data scan. The order of elements in IVAL should correspond to the order of elements in IDEV.

MASK is the mask array. Each mask corresponds to an entry in the IDEV array. The mask will be ANDed with data from the corresponding device before it is compared to the element of the IVAL array.

CDEV is an integer variable or integer array in which the device number of the changed device will be returned.

NVAL is the integer variable or array that will contain the unmasked data value from the changed device.

MSTAT is an integer variable or an integer array element that will return the status of the request. The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

EXTERNAL INTERRUPTS INPUT SCAN

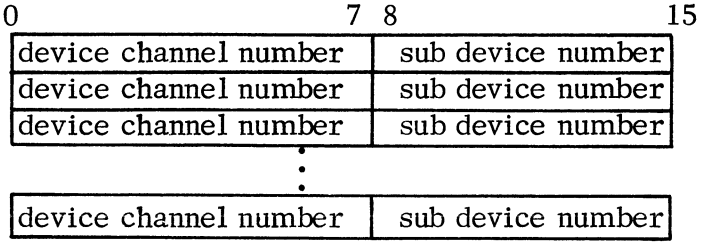
The subroutine ISCAN provides for automated scanning of external interrupt devices. The call scans the devices in the device array IDEV for bits of interrupt input. The interrupt bits of a device are ANDed with the complement of the MASK. If the result is zero, the scan continues. If the result is non-zero, the index into IDEV of the device currently being tested will be returned in IELEM and the interrupt bits for that device will be returned in IVAL. If the scan is completed with no interrupt found, the task waits for an interrupt on the first device. The scan may be performed repeatedly by periodically generating an interrupt on the first device in the device array. The format of the call to this routine is:

CALL ISCAN (NDEV, IDEV, MASK, IELEM, IVAL, MSTAT)

where: NDEV is an integer variable, integer constant or integer array specifying number of the device to be input.

IDEV is an integer array containing the device identifiers; device number *256 plus sub device (if any). There must be NDEV entries in the array.

The array IDEV is formatted as follows:



MASK is an integer array of masks. A one in any bit position in a MASK will cause input on the interrupt line for the corresponding device in the IDEV array, to be ignored.

IELEM is an integer variable or integer array element in which the index into the device array (IDEV) of the first interrupting device is returned.

IVAL is an integer variable or integer array element in which data is returned. The data returned is the inclusive OR of the mask and a word containing ones in the bit positions corresponding to the interrupting lines.

MSTAT is an integer variable or integer array that returns the status of the request.

The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

ANALOG TO DIGITAL CONVERSION WITHOUT GAIN PROGRAMMING

There are six modes of A-D conversion without gain programming. Calls can be: with or without wait, sequential-random, and data channel. The calls are defined as follows: AISQ sequential scan, AISQW sequential scan with wait, AIRD random scan, AIRDW random scan with wait, AISA sequential scan with data channel, AISAW sequential scan with data channel and wait. The calls are defined as follows:

AISQ - Start to scan a number of sequential analog input (points) and continue with the next program statement.

AISQW - Scan a number of sequential analog inputs and wait for completion before continuing with the next program statement.

AIRD - Start to scan a number of random analog inputs (points) and continue with the next program statement.

AIRDW - Scan a number of random analog inputs (points) and wait for completion before continuing with the next program statement.

AISA - Start to scan a number of sequential analog inputs using the data channel and continue with the next program statement.

AISAW - Scan a number of sequential analog inputs with data channel and wait for completion before continuing with the next program statement.

The format of these calls are:

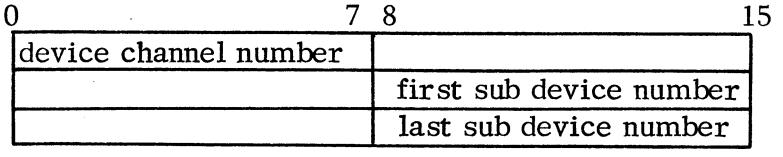
```
CALL AISQ    (NDEV, IDEV, IVAL, MSTAT)
CALL AISQW   (NDEV, IDEV, IVAL, MSTAT)
CALL AIRD    (NDEV, IDEV, IVAL, MSTAT)
CALL AIRDW   (NDEV, IDEV, IVAL, MSTAT)
CALL AISA    (NDEV, IDEV, IVAL, MSTAT)
CALL AISAW   (NDEV, IDEV, IVAL, MSTAT)
```

where: NDEV is an integer variable, integer constant, integer array element specifying the number of analog points to be input.

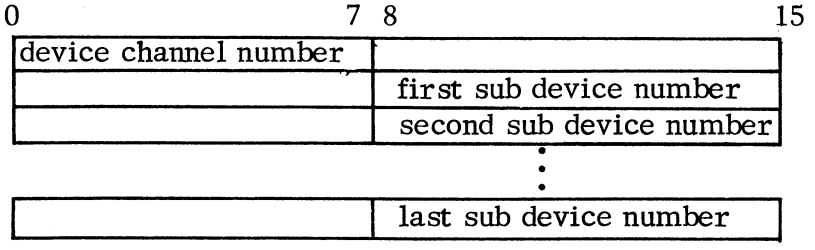
IDEV is an integer array specifying the points.

The first word of the point array has device number in the left byte. For sequential scans the next two elements of point array should have the starting and final A-D point numbers (sub-dev #). For random scans the sequence of points to be read start from the second element of point array (sub device #). All points should belong to this device.

The array IDEV is structured as follows for sequential calls:



The array IDEV is structured as follows for random calls:



IVAL is an integer array element in which the data samples will be stored.

MSTAT is an integer variable or integer array element that will return the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥ 3 - error (=RDOS error + 3)

ANALOG TO DIGITAL CONVERSION WITH GAIN PROGRAMMING

There are eight modes for A-D conversion with gain programming. Calls can be: with or without wait, sequential-random, and data channel or not. The calls are defined as follows: AISG - sequential gain, AISGW - sequential gain with wait, AIRG - random gain, AIRGW - random gain with wait, AIRB - random gain data channel, AIRBW - random gain data channel with wait, AISB - sequential gain data channel, AISBW - sequential gain data channel and wait. The calls are defined as follows:

AISG - Start to scan a number of sequential analog inputs (points) with gain and continue with the next program statement.

AISGW - Scan a number of sequential analog inputs (points) with gain and wait for completion before continuing with the next program statement.

AIRG - Start to scan a number of random analog inputs (points) with gain and continue with the next program statement.

AIRGW - Scan a number of random analog inputs (points) and wait for completion before continuing with the next program statement.

AIRB - Start to scan a number of random analog inputs (points) with gain and data channel, and continue with the next program statement.

AIRBW - Scan a number of random analog inputs (points) with gain and data channel, and wait for completion before continuing with the next program statement.

AISB - Start to scan a number of sequential analog inputs (points) with gain and data channel and continue with the next program instruction.

AISBW - Scan a number of sequential analog inputs (points) with gain and data channel, and wait for completion before continuing with the next program statement.

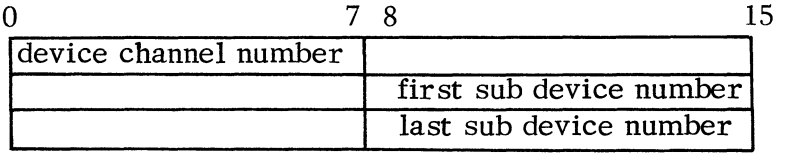
The format of these calls are:

```
CALL AISG    (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AISGW   (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AIRG    (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AIRGW   (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AIRB    (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AIRBW   (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AISB    (NDEV, IDEV, IVAL, IGAIN, MSTAT)
CALL AISBW   (NDEV, IDEV, IVAL, IGAIN, MSTAT)
```

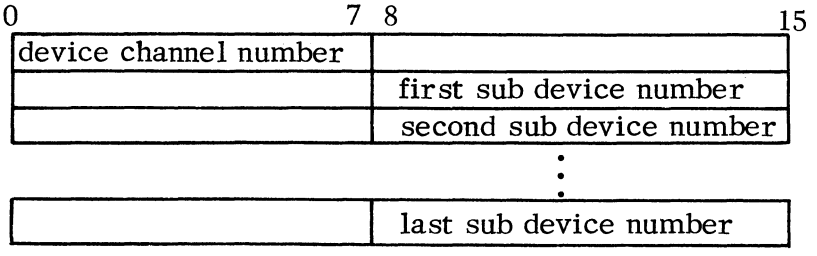
where: NDEV is an integer variable, integer constant, or integer array element specifying the number of analog points to be input.

IDEV is an integer array element of points to be scanned. The first word of the point array has device number in the left byte. For sequential scans the next two elements of point array should have the starting and final A-D point numbers (sub device #). For random scans the sequence of points to be read start with the second element of point array (sub device #). All points (sub device #'s) should belong to the same device.

The array IDEV is structured as follows for sequential calls:



The array IDEV is structured as follows for random calls:



IVAL is an integer array in which data samples will be stored.

IGAIN is an integer array whose elements specify the gain for each corresponding sub device in IDEV.

MSTAT is an integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

NOTES: AIRB(W) is supported only by 5602 -X-DCH. The gain array in this case will be modified by the routine to set up the channel gain word (auto gain not supported).

In AISG(W) and AIRG(W) only the right most 4 bits of the gain word will be used. In the case of 5602 however, -1 means auto-gain and the gain value will be returned in gain array.

AISB(W) is supported by 4120 and 4055 with gain and data channel. The first value of gain will be used for all points.

All points should belong to the same device number.

ANALOG OUTPUT IN SEQUENTIAL ORDER

Subroutines AOS and AOSW allow the output of high and low speed analog signals is a sequential order which is specified by the subroutine arguments and the output hardware. These calls support hardware types 4056 and 4080.

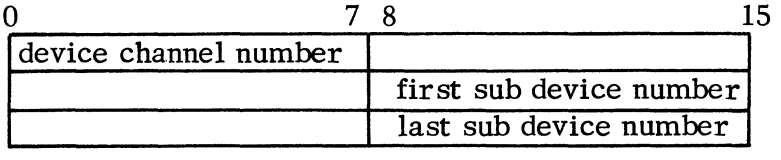
The formats of the calls to these routines are:

```
CALL AOS (NDEV, IDEV, IVAL, MSTAT)
CALL AOSW (NDEV, IDEV, IVAL, MSTAT)
```

where: NDEV is an integer variable, integer constant, or integer array element specifying the number of analog points to be output.

IDEV is an integer array name. All points should belong to the same device. The first element of the point array has the device number in the left byte. The second and third elements of point array are the starting and final point numbers (sub device #).

The array IDEV is structured as follows:



IVAL is an integer array name, designating the array from which analog values are taken. The array has NDEV elements. Data from integer array is output sequentially.

MSTAT is an integer variable or integer array element that will return the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥3 - error (=RDOS error + 3)

ANALOG OUTPUT IN ANY SEQUENCE

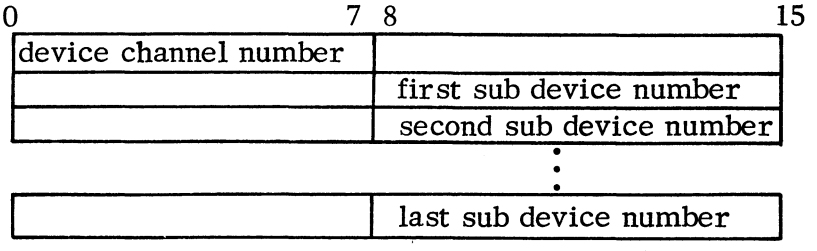
Subroutines AO and AOW allow the output of high and low speed analog signals in a random sequence specified by the subroutine arguments. These calls support hardware types 4056 and 4080. The formats of the calls to these subroutines are:

```
CALL AO (NDEV, IDEV, IVAL, MSTAT)
CALL AOW (NDEV, IDEV, IVAL, MSTAT)
```

where: NDEV is an integer variable, integer constant, or integer array element specifying the number of analog points to be output.

IDEV is an integer array name. The first element of the array has the device number in the left byte. The second element begins the point number sequence. All points in a single call refer to the same device. There are NDEV elements in the array.

The array IDEV is structured as follows:



IVAL is an integer array name, designating the array from which the analog values are taken. There are NDEV elements in the array.

MSTAT is an integer variable or integer array element that will return the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥3 - error (=RDOS error + 3)

SCOPE CONTROL

Scope control provides all necessary control signals for a special application requiring two D/A converters and a 4183 Scope Control. One converter is connected to the "X" input of an oscilloscope; the other converter drives the "Y" input. The resulting display will contain a 4096 by 4096 dot matrix. The subroutine SCOPE is used to control a scope (select-start-erase) and the call is formatted as follows:

CALL SCOPE (IDEV, IMODE, IARG, MSTAT)

where: IDEV is an integer variable or integer constant specifying the device number of this scope. The device number is formatted with the device number in the left byte.

IMODE is an integer variable, integer constant, or integer array element which specifies one of the following modes:

mode 0 - start, erase select
mode 1 - D-A channel 1 displacement
mode 2 - D-A channel 2 displacement
mode 3 - get status

IARG is an integer variable, integer constant or integer array element which specifies the functions which the scope is to perform. The function depends upon the mode:

If mode is 0 and the argument is equal to 1, send start pulse to scope, intensifing the beam.

If mode is 0 and the argument is equal to 2, erase the scope.

If the mode is 0 and the argument is equal to 3, erase the scope.

If the mode is 1, the argument equals the channel 1 displacement.

If the mode is 2, the argument equals the channel 2 displacement.

If the mode is 3, the status is returned in argument.

SCOPE CONTROL (Continued)

MSTAT is an integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

I/O BUS SWITCH CONTROL

The Input/Output Bus Switch (type 5470 or 8080) allows a common set of peripherals to be switched between two Central Processor Units (CPU's). This switching provides an economical means of sharing peripherals between CPU's. The subroutine IOBUS is used to control the I/O Bus Switch and is formatted as follows:

CALL IOBUS (IDEV, IMODES, ISTAT, MSTAT)

where: IDEV is an integer variable, integer constant, or integer array element specifying the logical device number of the I/O bus switch. The device number is formatted with the device number in the left byte.

IMODES is an integer variable, integer constant, or integer array element which specifies one of the following operation codes:

0 - Read I/O bus switch status
1 - Select calling CPU if bus is neutral
2 - Clear if calling CPU is master (bus switched to neutral)
3 - Deselect calling CPU (bus reset to neutral)

ISTAT is an integer argument used to return the status when IMODE = 0. The following are the status codes:

bit 15 - SW bus busy (selected by this or the other CPU)
bit 14 - this CPU selected
bit 13 - this CPU is master
bit 12 - this switch is manually selected

The preceding codes are bit settings and hence additive.

MSTAT is an integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT:

0 - undefined (initialization or hardware error)
1 - successful return.
2 - call in progress (calls without W option)
≥3 - error (=RDOS error + 3)

POWER FAIL

The user has the facility to write a routine or task that will receive control when power is restored after a power failure. To use the feature, a call formatted as follows, is required:

CALL POWER (TASKNAME, TID, TPRI, MSTAT)

where: TASKNAME represents the name of a task to be run when power is restored.

TID represents the task identification for TASKNAME.

TPRI represents the priority assigned to TASKNAME.

MSTAT is an integer variable or integer array element that returns the status of the request. The following is a list of the values returned in MSTAT:

- 0 - undefined (initialization or hardware error)
- 1 - successful return.
- 2 - call in progress (calls without W option)
- ≥ 3 - error (=RDOS error + 3)

CTLA1

Subroutine CTLA1 sets up USTIT in the Users Status Table (UST) to point to a cleanup routine which calls XRTIO when the user types console interrupt (CTRL A) from the background console. UST is a 24 octal word table which records information pertinent to the execution of a program level.

XRTIO is an assembly language subroutine which examines the device table from RTIOS System Generation (RTIOG) and if a 5602 device has been sysgened, sets the sub device number to 377_8 and disables interrupts of all priorities within the 5602 chassis. The format of this call is:

CALL CTLA1

NOTE: There are no arguments associated with this call.

CTLA2

Subroutine CTLA2 calls XRTIO. XRTIO is an assembly language subroutine which examines the device table from RTIOS System Generation (RTIOG) and if a 5602 device has been sysgened, sets the sub device number to 377_8 and disables interrupts of all priorities within the 5602 chassis. The format of the call is:

CALL CTLA2

NOTE: There are no arguments associated with this call.

END OF CHAPTER

CHAPTER 3

Assembly Language Calls

INTRODUCTION

The RTIOS assembly language calls available to the user support the following:

- Digital Inputs and Outputs
- Interrupt Inputs
- Analog to Digital Inputs
- Digital to Analog Inputs

Table 3-1 is a list of the calls and indicates their associated hardware.

GENERAL INFORMATION

The following is the general information necessary for using the RTIOS assembly language calls:

- Accumulators AC0, AC1, and AC2 are preserved unless data is expected in AC0 or AC1 or an error code is returned in AC2. AC3 is always returned with a USP value.
- The required save area is set up during system generation. There should be at least as many save frames as the maximum number of parallel tasks. Each frame is 5 words long with the following displacements:

  STATS - Status
  AC0 - Accumulator 0
  AC1 - Accumulator 1
  AC2 - Accumulator 2
  AC3 - Accumulator 3

  The save area ends with a -1 in status word .STATS word is available as a temporary for data other than 0 or -1.
- Interrupt enable/disable depends on macro definitions of ENABL and DSABL which should leave the accumulators intact.

  The device table (DTABL) is set up at system generation time and has three words per device arranged contiguously. A -1 indicates the end of the table.

GENERAL INFORMATION (Continued)

- The first call of RTIOS checks the interrupting device table (IDTBL) also set up during system generation and defines all the interrupts to the system. Autonomous devices 4066, if in that mode, 4067, 5602-J and any others if present will be started. Subsequent RTIOS calls will bypass this section waiting if necessary for the first call to finish the definitions. The first call should have the highest priority.

Calling Sequence

All user calls at assembly level in RTIOS are similar to the RDOS system calls with arguments passed in AC0, and AC1. Channel numbers may be passed in AC2 with 77 in the command line.

CALL FORMAT

```
JSR @ .RTIOS
command line
Error Return
Normal Return
```

where: command line consists of one of the RTIOS commands in the left byte and either a 77 or the logical channel number in the right byte. If 77 is used, AC2 must contain the logical channel number.

Each of the RTIOS calls are discussed individually in the following text and the calls are summarized in Table 3-2 at the end of this Chapter.

Logical channel numbers must be associated with the correct inputting or outputting device at system generation time. Chapter 4 is a detailed discussion of system generation.

Table 3-1 RTIOS Assembly Language Commands and Associated Hardware

Hardware Type	Associated Commands
Console Switches	DI.
4066	DI., DO. IW.
4067	II. IW. IM.
4068	DO. DI.
4055 or 4120 Series	AI. SA, AI. SB, AI. S, AI. R
4056 or 4180 D/A	AO.
Bus Switch 5470/8080	BS.
4183	SC.
5602 (Each of the following represent a DIOS mode entered in response to SYSGEN question 22 in Chapter 4)	
0	DO.
1	DI.
2	None Available
3	II.
4	None Available
5	IW.
6	IM. II. IW.
7	AI. R, AI. S
8	None Available
9	AI. RB
Data Channel extension to DIOS (Question 24 of SYSGEN in Chapter 4)	DI. A, DO. A, DC. A

Digital Input (DI.)

The Digital Input call (DI.) samples 16 lines of external input and transfers them into the address contained in AC0.

CALL FORMAT

```
JSR  @  .RTIOS
DI.  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - contains the address where input data is to be stored.

AC1 - contains the device and sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is completed.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Digital Output (DO.)

The Digital Output call (DO.) transfers the contents of the address in AC0 to sixteen output lines.

CALL FORMAT

```
JSR @ .RTIOS
DO. logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - contains the address of the data word to be output.

AC1 - contains the device and sub-device number. (If bit 0 is set, transfer is started and control will be returned to the caller before the transfer is completed).

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Interrupt Input (II.)

The Interrupt Input (II.) call is used to input the current interrupt bits which are inclusively ORed with the mask in AC0. The transferred bits are cleared in the device interrupt buffer (offset DATAW of the device control table extension).

CALL FORMAT

```
JSR @ .RTIOS
II. logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - contains the mask value (a one in a mask bit will cause an interrupt on the corresponding line to be ignored).

AC1 - contains the device and sub-device number.

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

On return AC0 contains the returned data (the inclusive OR of the mask and the contents of the device interrupt buffer).

Example

```
(CALL)
        DATAW (DEVICES WHICH HAVE INTERRUPTED)   1010
        AC0(MASK)   0011

(RETURN)

        DATAW (DEVICES WHICH HAVE INTERRUPTED)   0010
        AC0  (DATA RETURNED)     1011
```

As an example of what occurs when a call with the II. Command is issued, consider the case where two sub-devices have interrupted. The bits corresponding to the interrupting sub-devices will be set in the device interrupt buffer (DATAW) of the corresponding device control table extension. DATAW = 1010, with the ones indicating the interrupting devices. If AC0 = 0011 when the call occurs, after the return, DATAW will be 0010 and AC0 will be 1011.

Wait for an Interrupt (IW.)

The Wait for an Interrupt call (IW.) causes a task to be suspended until an interrupt occurs on one of the interrupting inputs. (Interrupts may be disabled by the IM. command.)

CALL FORMAT

```
JSR  @  .RTIOS
IW.  logical channel number or 77
Error Return
Normal return
```

Accumulator Contents

AC1 - contains a sub device number (if any).

Set Interrupt Mask (IM.)

The Set Interrupt Mask call (IM.) allows the user to enable or disable selected interrupt lines by loading a register with a mask.

CALL FORMAT

```
JSR  @  .RTIOS
IM.  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - bits 0-7 contain the mask for 4067 and bits 8-15 contain the mask for 5602J

AC1 - contains the device and sub-device number.

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Scope Control (SC.)

This call provides all necessary control signals for a special application requiring two D/A Converters and a 4183 Scope Control. One converter is connected to the "X" input of an oscilloscope; the other converter drives the "y" input. The resulting display will contain a 4096 by 4096 dot matrix. This call controls the scope; select, start and stop.

CALL FORMAT

```
JSR  @  .RTIOS
SC.  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - contains one of the following:
- the scope mode data if the mode is equal to 00.
- the x or y displacement data if the mode is equal to 01 or 10.
- the status return if the mode was 11.

AC1 - contains in bits 14 and 15 one of the following control codes:
- 00 - scope mode set
- 01 - displacement to channel 1 (x axis)
- 10 - displacement to channel 2 (y axis)
- 11 - get status

If bit 06 is set, scope erase will occur.
If bit 07 is set, scope start will occur.

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Analog Output in any Sequence (AO.)

The Analog Output in any Sequence call (AO.) outputs the contents of AC0 to a Analog to Digital converter channel contained in AC1.

CALL FORMAT

```
JSR  @  .RTIOS
AO.  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - contains the data to be output.

AC1 - contains the device or sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Sequential Analog Input (AI.S)

The Sequential Analog Input call (AI.S) allows for the synchronous or asynchronous sampling of analog inputs from a specified number of sub devices. The scan begins at the starting channel and terminates with the ending channel. The scan takes samples on channels in numeric order.

CALL FORMAT

```
JSR @ .RTIOS
AI.S  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a three word array containing:
- word 1 - pointer to the number of channels (sub devices) to be read.
- word 2 - pointer to an array which contains the starting and ending sub device to be read.
- word 3 - pointer to the starting address where the analog inputs will be stored.

AC1 - contains the device or sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Sequential with Variable Gain (AI.SG)

The A/D Sequential with Variable Gain call (AI.SG) allows for the synchronous or asynchronous sampling of analog inputs, with variable gain, from a specified number of sub devices. The scan begins at the starting channel and terminates with the ending channel. The scan takes samples on channels in numeric order.

CALL FORMAT

```
JSR @ .RTIOS
AI.SG  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to an array which contains the starting and ending sub device to be read.
- word 3 - pointer to the starting address where the analog inputs will be stored.
- word 4 - pointer to the gain array (1 entry per channel).

AC1 - contains the device or sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Sequential Data Channel (AI.SA)

The A/D Sequential Channel call (AI.SA) allows for the synchronous or asynchronous sampling of analog inputs using the data channel. The scan begins at the starting channel and terminates with the ending channel. The scan takes samples on channels in numeric order.

CALL FORMAT

```
JSR  @  .RTIOS
AI.SA  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to an array which contains the starting and ending sub device to be read.
- word 3 - pointer to the starting address where the analog inputs will be stored.
- word 4 - pointer to the gain array (1 entry per channel).

AC1 - contains the device or sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Sequential Program Gain Data Channel (AI.SB)

The A/D Sequential Program Gain Data Channel call (AI.SB) allows for the synchronous or asynchronous sampling of analog inputs with sequential program gain using the data channel. The scan begins at the starting channel and terminates with the ending channel. The scan takes samples on channels in numeric order.

CALL FORMAT

```
JSR @ .RTIOS
AI.SB logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to an array which contains the starting and ending sub device to be read.
- word 3 - pointer to the starting address where the analog inputs will be stored.
- word 4 - pointer to the gain array.

AC1 - contains the device or sub-device number. (If bit 0 is set transfer is started and control will be returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Random Analog Inputs (AI.R)

The Random Analog Inputs call (AI.R) allows for the synchronous or asychronous sampling of analog inputs from a list of sub devices.

CALL FORMAT

```
JSR @ .RTIOS
AI.R  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a three word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to the array which contains a list of sub devices to be sampled.
- word 3 - pointer to the starting address where the analog inputs will be stored.

AC1 - contains the device or sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Random Variable Gain (AI.RG)

The A/D Random Variable Gain call (AI.RG) allows for the synchronous or asynchronous sampling of analog inputs, with variable gain, from a list of sub devices.

CALL FORMAT

```
JSR @ .RTIOS
AI.RG  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to the array which contains a list of sub devices to be sampled.
- word 3 - pointer to the starting address where the analog inputs will be stored.

AC1 - contains the device or sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Random Data Channel (AI. RA)

The A/D Random Data Channel call (AI. RA) allows for the synchronous or asynchronous sampling of analog inputs using the data channel. Such devices are sampled in listed order.

CALL FORMAT

```
JSR @ .RTIOS
AI.RA  logical channel number of 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to the array which contains a list of sub devices to be sampled.
- word 3 - pointer to the starting address where the analog inputs will be stored.
- word 4 - pointer to the gain array.

AC1 - contains the device or sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

A/D Random Program Gain Data Channel (AI.RB)

The A/D Random Program Gain Data Channel allows for the synchronous or asynchronous sampling of analog inputs using the data channel. Sub devices are sampled in the listed order.

CALL FORMAT

```
JSR  @  .RTIOS
AI.RB  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array containing:
- word 1 - pointer to the number of sub devices to be read.
- word 2 - pointer to the array which contains a list of sub devices to be sampled.
- word 3 - pointer to the starting address where the analog inputs will be stored.
- word 4 - pointer to the gain array.

AC1 - contains the device or sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Digital Input with Data Channel Scan (DI. A)

The Digital Input with Data Channel Scan call (DI. A) reads inputs from channels specified in channel array and stores the data for each channel in the corresponding position in the data array. This call uses the data channel option which transfers the data directly into core.

CALL FORMAT

```
JSR @ .RTIOS
DI. A  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a three word array which contains:
word 1 - pointer to the number of channels to be read.
word 2 - pointer to the channel array.
word 3 - pointer to the data array.

AC1 - contains the device and sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Digital Output with Data Channel Scan (DO. A)

The Digital Output with Data Channel Scan call (DO. A) outputs data from the data array to the channels listed in the channel array. Each channel will receive data from the corresponding positions in the data array.

CALL FORMAT

```
JSR  @  .RTIOS
DO. A  logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a three word array which contains:
word 1 - pointer to the number of channels.
word 2 - pointer to the channel array.
word 3 - pointer to the data array.

AC1 - contains the device and sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Digital Data Channel Compare (DC. A)

The Digital Data Channel Compare call (DC. A) compares the input data with the corresponding data in the mask array. If they are the same, a 0 is stored in the data array. Otherwise, the new input data is stored in the data array. This call uses the data channel option which transfers the data directly into core.

CALL FORMAT

```
JSR @ .RTIOS
DC. A logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - is a pointer to a four word array which contains:
word 1 - pointer to the number of channels.
word 2 - pointer to the channel array.
word 3 - pointer to the data array.
word 4 - pointer to the mask array.

AC1 - contains the device and sub-device number. (If bit 0 is set, transfer is started and control is returned to the caller before the transfer is complete.)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

Input/Output Bus Switch (BS.)

The Input/Output Bus Switch call (BS.) allows a common set of peripherals to be switched between two Central Processing Units. This switching provides an economical means of sharing peripherals between processors.

CALL FORMAT

```
JSR @ .RTIOS
BS. logical channel number or 77
Error Return
Normal Return
```

Accumulator Contents

AC0 - One of the following statuses is returned in AC0:

- bit 15 - Bus Switch busy (selected by this or the other CPU)
- bit 14 - this CPU selected
- bit 13 - this CPU is master
- bit 12 - this CPU is manually selected

AC1 - Enter one of the following modes:

- 0 - read I/O Bus Switch code
- 1 - select calling CPU if the Bus is in neutral
- 2 - clear if calling CPU is master (leave switch in neutral)
- 3 - deselect calling CPU (reset switch to neutral)

AC2 - If 77 is entered in the command line, AC2 must contain the logical device channel number.

CTRLA

CTRL A is an assembly language subroutine which sets up USTIT in the Users Status Table (UST) to point to a cleanup routine which calls XRTIO when the user types console interrupt (CTRL A) from the background console. UST is a 24 octal word table which records information pertinent to the execution of a program level.

XRTIO is an assembly language subroutine which examines the device table from RTIOS System Generation (RTIOG) and if a 5602 device has been sysgened, sets the sub device number to 377_8 and disables interrupts of all priorities within the 5602 chassis.

CALL FORMAT

```
JSR  CTRL A
Normal Return
```

Accumulator Contents

Not Applicable

XRTIO

XRTIO is an assembly language subroutine which examines the device table from RTIOS System Generation (RTIOG) and if a 5602 device has been sysgened , sets the sub device number to 377_8 and disables interrupts of all priorities within the 5602 chassis.

CALL FORMAT

```
JSR  XRTIO
Normal Return
```

Accumulator Contents

Not Applicable

TABLE 3-2 ASSEMBLY LANGUAGE CALLS

COMMAND	AC0	AC1	WAIT	DESCRIPTION
DI.	DATA ADDR.	SUB DEV #	*	ONE WORD DIGITAL INPUT.
DO.	DATA ADDR.	SUB DEV #	*	ONE WORD DIGITAL OUTPUT.
II.	MASK WORD DATA RETURNED.	SUB DEV #		INTERRUPT INPUT
IW.	—	—		WAIT FOR AN INTERRUPT
IM.	MASK WORD	SUB DEV #		SET INTERRUPT MASK
AI. S	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY	SUB DEV #	*	A-D SEQUENTIAL
AI. R	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY	SUB DEV #	*	A-D RANDOM
AO.	DATA	POINT #	*	D-A
SC.	DATA	CALLING MODE	*	SCOPE CONTROL
AI. SG	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. GAIN ARRAY	SUB DEV #	*	A-D SEQUENTIAL WITH VARIABLE GAIN.
AI. RG	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. GAIN ARRAY	SUB DEV #	*	A-D RANDOM VARIABLE GAIN.
AI. RA	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. GAIN ARRAY	SUB DEV #	*	A-D RANDOM DATA CHANNEL.
AI. SA	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. GAIN ARRAY	SUB DEV #	*	A-D SEQUENTIAL DATA CHANNEL.
AI. RB	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. GAIN ARRAY	SUB DEV #	*	A-D RANDOM PROG. GAIN-DATA CH.
AI. SB	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY D. GAIN ARRAY		*	A-D SEQUENTIAL PROG GAIN-DATA CH.
DI. A	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY	SUB DEV #	*	DIGITAL INPUT DCH SCAN.
DO. A	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY	SUB DEV #	*	DIGITAL OUTPUT DCH SCAN
DC. A	POINTER TO A. # OF CH. S; B. CH. ARRAY; C. DATA ARRAY; D. MASK ARRAY	SUB DEV #	*	DIGITAL INPUT DCH COMPARE
BS.	STATUS RET	OPERATION CODE		INPUT/OUTPUT BUS
CTRL A	—	—		Set USTIT
XRTIO	—	—		Examine device table for 5602 device.

Calls with an asterisk (*) in the WAIT column in Table 3-2 wait for the transfer to be complete before returning to the caller unless the left most bit in AC0 is set in which case the transfer is asynchronous.

END OF CHAPTER

CHAPTER 4

System Generation

INTRODUCTION

This chapter discusses system generation, loading and initialization procedures for RTIOS.

SYSTEM GENERATION

To request the system generation program (RTIOG) which is provided to the user as a save file tape, the user enters the following on the system console:

```
RTIOG)
```

The system generator program then initializes a questionaire on the system console to which the user should respond. The questionaire consists of a series of questions regarding the hardware configuration.

User responses are used to generate a source file which is a device table (DTABL) parameterizing the user's hardware configuration. When assembled with parameter table PARU.SR and RTPAR.SR and loaded with user programs and the RTIOS libraries, the device table external references cause modules containing device drivers and interrupt handlers to be loaded from the libraries. The libraries may contain several device drivers and interrupt handlers for each piece of hardware. The appropriate device routines will be selected for the hardware configuration and mode of operation indicated during the sysgen dialogue.

The questions and their responses are discussed in the following text.

ANSWER 1 OR 0 FOR YES OR NO

1) ARE CONSOLE SWITCHES READ?

A response of 1 (yes) indicates that the console switches are to be read and the user continues to question 2. A 0 response (no) indicates that the switches are not read and the user continues with question3.

2) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to the switch.

3) I/O BUS SWITCH - DEV 74,75 PRESENT?

A response of 1 indicates that the I/O Bus Switch is being used and the user continues with question 4. A 0 response indicates that the switch is not being used and the user continues with question 5.

4) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value must be unique to the switch.

5) IS DIO - 4066 USED?

A yes response (1) indicates that 4066 is being used and the user continues with question 6. A 0 response (no) indicates that 4066 is not being used and the user continues with question 8.

6) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to this device.

7) MODE OF 4066 - DEV 42 ARE:

- 1 INTERRUPTS NOT USED
0 EXTERNAL INTERRUPT INPUT
1 INPUT AT INTERRUPT TIME
2 OUTPUT AT INTERRUPT TIME

TYPE IN MODE # FOR DEVICE?

A -1 is applicable for DI and DO calls. 0 is applicable for II and IW calls. 1 is applicable for DI only. 2 is applicable for DO only.

RTIOS DEVICE SPECIFICATION (Continued)

8) IS XI -4067 -DEV 43 USED?

A 1 response (yes) indicates that 4067 is being used and the user is directed to question 10. A 0 response (no) indicates that the device is not being used and the user is directed to question 9. (A yes response causes questions associated with 4068 to be omitted.)

9) IS PIT 4068 - DEV 43 USED?

A 1 response indicates that 4068 is being used and the user is directed to question 10. A 0 response (no) indicates that the device is not being used and the user is directed to question 11.

10) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to this device.

11) IS A-D 4055 OR 4120 DEV 21 USED?

A 1 response indicates that 4055 is being used and the user is directed to question 12. A zero response (no) indicates that the device is not being used and the user is directed to question 13.

12) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to this device.

13) IS THERE HARDWARE GAIN PROGRAMMING?

A 1 response (yes) indicates that the hardware for gain programming is included. A 0 response (no) indicates that the hardware is not available. The user is directed to question 14 for either a 0 or 1 response.

14) IS THERE HARDWARE FOR AUTO SCAN?

A 1 response (yes) indicates that the hardware for auto scan is included. A 0 response (no) indicates that the hardware is not available. The user is directed to question 15 for either a 0 or 1 response.

RTIOS DEVICE SPECIFICATION (Continued)

15) IS D-A - 4056 OR 4180 - DEV 23 USED?

A 1 response (yes) indicates that either 4056 or 4180 is being used and the user is directed to question 16. A 0 response indicates that the devices are not being used and the user is directed to question 17.

16) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to this device.

17) IS SCOPE CONTROL 4183 DEV 23 USED?

A 1 response (yes) indicates that 4183 is being used and the user is directed to question 18. A 0 response indicates that the device is not being used and the user is directed to question 19.

18) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

A response between 0 and 255 is entered. The value entered must be unique to this device.

19) IS DIOS -5602- DEV 65 USED?

A 1 response (yes) indicates that 5602 is being used and the user is directed to question 20. A 0 response indicates that the device is not used and the user is directed to question 25.

20) TYPE IN LOGICAL CHANNEL # FOR DEVICE?

In response, a value between 0 and 255 is entered. The value entered must be unique to this device.

21) NEXT 5602 SUB-DEV #? (-1 MEANS NO MORE)

The user enters a value between 0 and 253. The value entered must be unique to the sub device number. Questions 21, 22 and 23 are repeated until a -1 is entered to this question (21). After the -1 is entered, the user continues with question 24.

RTIOS DEVICE SPECIFICATION (Continued)

22) MODES OF DIOS -5602- DEV 65

0-DIGITAL OUTPUT AND A PULSE-INTERRUPT UNUSED
1-DIGITAL INPUT-INTERRUPTS UNUSED
2-NOT USED
3-DIGITAL INPUT AT INTERRUPT TIME
4-NOT USED
5-UNTIED DIG. INPUT AND ONE EXT. INTERRUPT INPUT
6-EXTERNAL INTERRUPT INPUT 5602-J
7-WIDE RANGE A-D
8-NOT USED
9-DCH SCAN WIDE RANGE A-D

In response the user enters one of the applicable digits 0-9. (DCH-Wide Channel A/D in 5602 requires an associated 4 channel number. 4 consecutive sub-device numbers are taken up by this device 21, 23, 24 and 25. Question 22 is output following each occurrence of question 21. If a response of 7 is entered to the question, the user is directed to question 23.

23) ASSOCIATED WIDE RANGE A/D SUB DEV #?

This question is output when a 7 is entered in response to question 22. The user enters the applicable sub-device number. (The sub device number is based upon the applicable hardware assignment. Upon completion, the user is directed to question 21.

24) IS DATA CHANNEL EXT TO DIOS-EXT-DEV 64 USED?

A 1 response (yes) indicates that device 64 is used. A 0 response (no) indicates that it is not. This question determines whether or not the user desires to have data channel scan of digital I/O. The user continues with question 25.

25) USER POWER UP?

A 1 response (yes) indicates the power up option is being used. A 0 response (no) indicates the power-up option is not being used. The user continues with question 26. If the power-up option is used, the user must provide his own power-up routine which will receive control on a power-up situation.

RTIOS DEVICE SPECIFICATION (Continued)

26) TYPE IN OUTPUT DEVICE OR FILE NAME?

In response the user enters the output device or file name and continues to question 27.

27) DEFAULT A-D GAIN FOR WRAIS?

In response, the user enters the applicable default value. The value represents the hardware range of gains. Continue with question 28.

28) MAXIMUM NUMBER OF PARALLEL TASKS?

In response, the user enters the maximum number of parallel tasks and continues with question 29.

29) STOP

This message indicates that the system generation procedure is complete.

Device Control Table (DTABL) Description

As explained above the table output by the sysgen procedure when assembled and loaded provides the mechanism for extracting the appropriate device routines from the RTIOS libraries. At execution time, the table serves as a data base and temporary area for RTIOS routines.

The table is divided into several sections in order to simplify access to it. The first section contains the Device Control Table (DCT's) for all RTIOS devices sysgened. There is one three word DCT for each device. The three words of data in the DCT and the symbolic offsets to each word are:

DEVCE - logical channel number assigned by the user at sysgen time.

Device Control Table (DTABL) Description (Continued)

PROCD - procedure address, a pointer to the device handler to handle I/O (including interrupts) for this device. The symbolic name of the procedure is generated by RTIOG at sysgen time, depending on the mode of operation indicated by the user. If the user writes his own device handler, he can cause it to be loaded from a library and called at execution time by editing the symbolic entry point to the routine into the device table source in place of the sysgened routine name. (For devices that support sub devices see the following text.)

DEVCT - pointer to the DCT extension if one exists, -1 otherwise. (See descriptions of DCT extensions below.)

The second section of the table contains the DCT extensions. There are two different types of extensions, those for devices which do not support sub-devices, and those for devices which do.

For devices which do not support sub devices, an extension of up to eight words is provided. The extension serves as a work area for the device driver and interrupt handler. Offsets into the extension are described in RTPAR.SR. Use of the extensions varies from driver to driver.

For devices which do support sub device numbers the DCT extension is a sub DCT. Its offsets are the same as for the DCT but the use of each offset is slightly different. In the sub device DCT, use of each word and offset to it is:

DEVCE - sub device code entered by user during sysgen.

PROCD - procedures address - pointer to procedure for this sub device. (Note that for a device with sub devices, the procedure in the main DCT is a dispatch routine that searches the table of sub device DCT's for an entry with sub device numbers that matches the sub device from the call and dispatches to the procedure indicated in the sub-DCT). (Note also that a user written sub device driver may be incorporated in a manner analogous to that described for the procedure in the main DCT.)

Device Control Table (DTABL) Description (Continued)

DEVCT - pointer to a further extension to the DCT. Use and format of the further extension is the same as that for devices which do not support sub devices. If no further extension is used, a -1 is entered.

The third section of the device table (DTABL) contains a list of sysgened device codes. This list is used in MRDOS systems to enable these devices. (Entry point of the list is DVTBL.)

The fourth section of the device table is a list of interrupting devices and a three word system DCT for each. This section of the table is used to IDEF the interrupting devices. A detailed description of system DCT's and their function is provided in the RDOS Reference Manual (093-000075) and the RTOS Reference Manual (093-000056).

The final section of the device table contains a save area and constants used in the power fail option.

ASSEMBLY PROCEDURES

The device table (DTABL) output by the system generator is assembled with the RTIOS parameters (PARU.SR and RTPAR.SR) and resulting relocatable binary should be loaded with user programs calling RTIOS modules. The relocatable binary is identified as DTABL.RB and is created by the following command line:

MAC PARU/S RTPAR/S filename $LPT/L)

where: filename represents the filename used in response to system generation question 26.

LOADING PROCEDURES

RTIOS is supplied as two separate libraries.

RTIOS.LB - Assembly language calls + Interrupt device handlers

F4RTIOS.LB - FORTRAN IV interface to RTIOS.LB

LOADING PROCEDURES

Typical ECLIPSE load commands for FORTRAN IV are:

UNMAPPED RDOS:

```
RLDR 5/K 10/C $LPT/L F4MAINPROG (SUBPROG1...SUBPROGn) ↑
DTABLE F4RTIOS.LB BRTIOS.LB BFMT.LB FORT.LB
```

MAPPED RDOS:

```
RLDR 5/K 10/C $LPT/L F4MAINPROG (SUBPROG1...SUBPROGn) ↑
DTABLE F4RTIOS.LB ARTIOS.LB AFMT.LB FORT.LB
```

RTOS:

```
RLDR/C $LPT/L F4MAINPROG (SUBPROG1...SUBPROGn) ↑
BRTOS.RB DTABLE F4RTIOS.LB BRTIOS.LB ↑
BRTOSFMT.LB FORT.LB BRTOS.LB
```

Typical ECLIPSE load commands for Assembly Language are:

UNMAPPED RDOS:

```
RLDR 5/K 10/C $LPT/L MAINPROG (SUBPROG1...SUBPROGn) ↑
DTABLE BRTIOS.LB
```

MAPPED RDOS:

```
RLDR 5/K 10/C $LPT/L MAINPROG (SUBPROG1...SUBPROGn) ↑
DTABLE ARTIOS.LB
```

RTOS:

```
RLDR/C $LPT/L MAINPROG (SUBPROG1...SUBPROGn) ↑
BRTOS.RB DTABLE BRTIOS.LB ↑
BRTOS.LB
```

END OF CHAPTER

APPENDIX A

Error Messages

This appendix supplies all the error messages in octal for the RTIOS system.

RTIOS ERRORS

Origin	Assembly Language	FORTRAN	Mnemonic	Meaning
RTIOS	0	3	ERFNO=0	Device not sysgened or incomplete transfer. (also see IDEF errors)
DCONS	3	6	ERICD(=3)	Bad command for this device
DI0BS	0	3	ERFNO(=0)	Bad command for this device
D4066	21	24	ERUFT(=21)	Device already in use.
D4066	3	6	ERICD(=3)	Bad command.
D4067	3	6	ERICD(=3)	Bad command.
D4068	3	6	ERICD(=3)	Bad command.
D4120	21	24	ERUFT(=21)	Device already in use.
D4120	3	6	ERICD(=3)	Bad command.
D4180	3	6	ERICD(=3)	Bad command.
D4183	3	6	ERICD(=3)	Bad command.
D5062	0	3	ERFNO(=0)	Not sysgened.
J5602	3	6	ERICD(=3)	Bad command.
DINPI	3	6	ERICD(=3)	Bad command.
DINPI	21	24	ERUFT(=21)	Device in use.
DINPN	3	6	ERICD(=3)	Bad command.
DEXTS	3	6	ERICD(=3)	Bad command.
DEXTS	21	24	ERUFT(=21)	Device in use.
WRAIS	3	6	ERICD(=3)	Bad command.
WRAIS	21	24	ERUFT(=21)	Device in use.

RTIOS ERRORS (continued)

Origin	Assembly Language	FORTRAN	Mnemonic	Meaning
AWRAI	3	6	ERICD(=3)	Bad command.
AWRAI	21	24	ERUFT(=21)	Device in use.
SCOPE F4PWR		(see IDEF and .TASK +3 FORTRAN accessible only)		Illegal Scope mode.
.IDEF	36	41	ERDNM	Illegal device code (>77_8). Device code 77_8 is reserved for the power monitor/auto restart option.
	45	50	ERIBS	Interrupt device code in use or 10 user devices already outstanding.
	65	70	ERDCH	Insufficient room in data channel map.
	74	77	ERMPR	Address outside address space (mapped systems only).
.TASK	42	45	ERNOT	No TCBs.
	61	64	ERTID	A task with the requested I.D. (except 0) already exists.

END OF APPENDIX

APPENDIX B

Special Products Considerations

Because of the nature of 5602 special products digital I/O hardware, special procedures must be used if that device is sysgened into the system. These measures are designed to insure that all interrupts from sub-devices on the 5602 are masked off before a program which uses RTIOS returns to normal RDOS operations. Failure to follow these procedures will cause the system to panic if an interrupt is received from a sub-device on the 5602 after program termination:

(1) If a program using the 5602 terminates normally, it must make a call to CTLA2 or XRTIO after the last RTIOS call, as part of the termination procedures.

(2) If the user program executes a .RTN, .ERTN, or a .BREAK, either CTLA2 or XRTIO must be called prior to the .RTN, .ERTN, or .BREAK.

(3) For abnormal terminations, use of console break (Control/C), foreground interrupt (Control/F), or of console interrupt (Control/A) in the foreground, will not permit the console interrupt service routine to be executed, and may result in a system panic.

(4) For programs running in the background:

(A). If the user is servicing console interrupts (Control/A) by setting USTIT in the user status table, then from within the program servicing the interrupts, he should call either CTLA2 (from FORTRAN program) or XRTIO (from assembly language program).

(B). Otherwise, if the user is not setting USTIT, he will have no cleanup (other than that required for RTIOS) after interrupts. To provide RTIOS cleanup, the program should initially call CTLA1 (from FORTRAN program) or CTRLA (from assembly language program), before making any RTIOS calls.

(5) For programs running in the foreground, since console interrupt (Control/A) and console break (Control/C) are disallowed, the user may arrange interrupts by:

(A) Monitoring a console switch or other input device

(B) Communication from a background program

(C) Communication via OPCOM.

Special Products Considerations (Continued)

(5) (Continued)

Console interrupts and breaks may be disabled by a .SYSTM .ODIS. In any case, the foreground program must call CTLA2 (from FORTRAN program) or XRTIO (from assembly language program), before terminating.

END OF APPENDIX

DataGeneral

SOFTWARE DOCUMENTATION
REMARKS FORM

Document Title	Document No.	Tape No.

SPECIFIC COMMENTS: List specific comments. Reference page numbers when applicable. Label each comment as an addition, deletion, change or error if applicable.

GENERAL COMMENTS: Also, suggestions for improvement of the Publication.

FROM:

Name	Title	Date

Company Name

Address (No. & Street)	City	State	Zip Code

Form No. 10-24-004 Rev. 7-75

FOLD DOWN FIRST FOLD DOWN

FIRST
CLASS
PERMIT
No. 26
Southboro
Mass. 01772

BUSINESS REPLY MAIL

No Postage Necessary If Mailed In The United States

Postage will be paid by:

Data General Corporation

Southboro, Massachusetts 01772

ATTENTION: Software Documentation

FOLD UP SECOND FOLD UP

STAPLE