HEWLETT PACKARD

RTE Operating System

Driver Writing Manual

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014

MANUAL PART NO. 92200-93005
Printed in U.S.A. December 1983
E1283

Printing History

The Printing History below identifies the Edition of this Manual and any Updates that are included. Periodically, Update packages are distributed which contain replacement pages to be merged into the manual, including an updated copy of this Printing History page. Also, the update may contain write-in instructions.

Each reprinting of this manual will incorporate all past Updates, however, no new information will be added. Thus, the reprinted copy will be identical in content to prior printings of the same edition with its user-inserted update information. New editions of this manual will contain new information, as well as all Updates.

To determine what software manual edition and update is compatible with your current software revision code, refer to the appropriate Software Numbering Catalog, Software Product Catalog, or Diagnostic Configurator Manual.

Fifth Edition	Feb 1980	
Update 1	Jul 1981	Manual enhancement
Reprint	Jul 1981	Update 1 incorporated
Sixth Edition	Dec 1983	RTE-6/VM compatibility

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Copyright © 1981, 1983 by HEWLETT-PACKARD COMPANY

DOCUMENTATION MAP

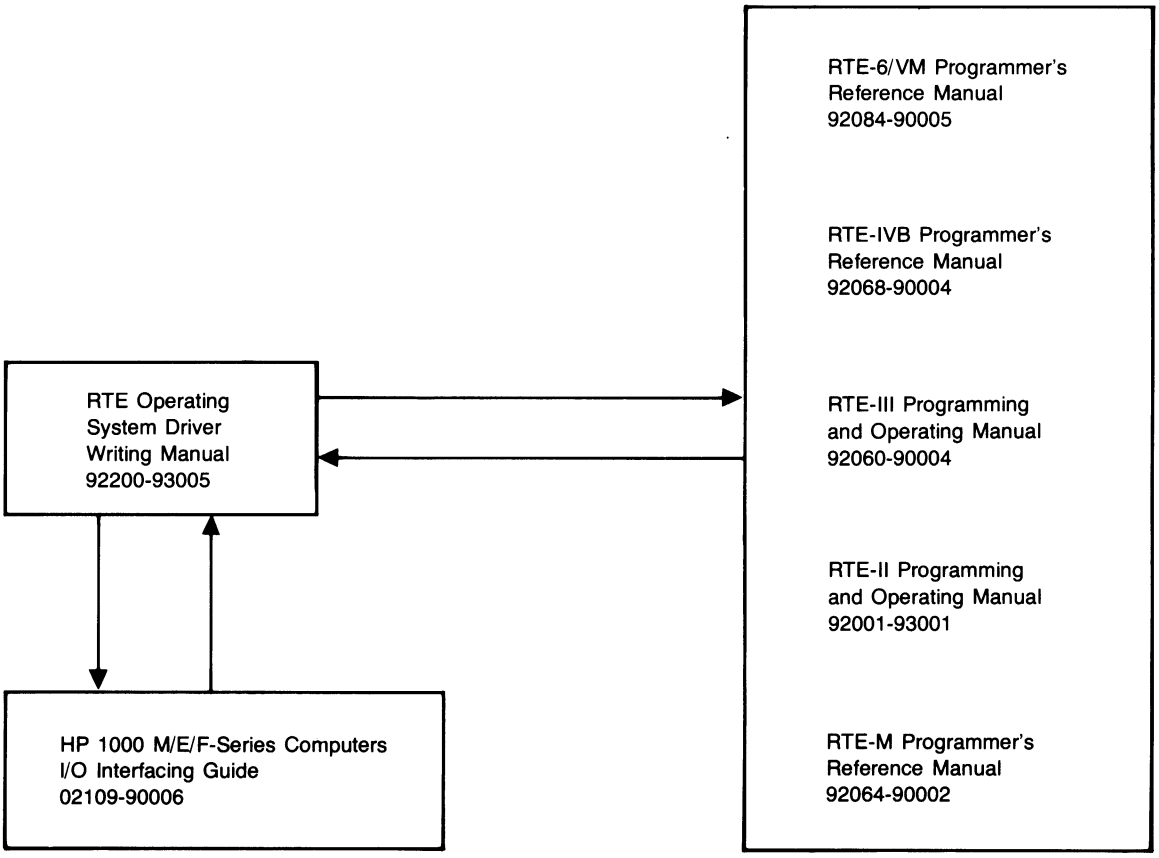


Table of Contents

Chapter 1 Introduction

Purpose 1-1
Scope 1-1
Supporting Documentation 1-2

Chapter 2 RTE Input/Output Structure

Introduction 2-1
Software I/O Structure 2-2
  Input/Output Device Drivers 2-2
  System I/O Processor 2-2
  Base Page Communications Area 2-3
  Equipment Table 2-5
  Logical Unit Numbers 2-9
  Device Reference Table 2-9
  Computer Interrupt Mechanism 2-12
  Interrupt Table 2-13
  Driver Mapping Table 2-16
General Operation of RTE I/O 2-18
  I/O Initiation 2-21

Chapter 3 Writing Standard RTE Drivers

Introduction 3-1
General Driver Structure and Operation 3-1
Driver Naming Requirements 3-2
Initiation Section 3-3
  Functions of the Initiation Section 3-4
Continuation/Completion Section 3-9
Device Clear on Program Abort 3-13
I/O Controller Time-Out 3-14
  Driver Processing of Time-Out 3-15
  System Processing of Time-Out 3-16

DCPC Processing 3-17
RTE Control of DCPC Assignment 3-17
DCPC Assignment by RTE 3-18
Preferred Method 3-19
Alternate Methods 3-19
Alternate Method I: Initiation Request 3-19
Alternate Method II: Continuation Request 3-20
Determining The DCPC Assignment Method 3-21
Returning DCPC Channels to RTE 3-23
Handling the DCPC Interrupt 3-24
Intermixed DCPC and Non-DCPC Operations 3-26
Driver Automatic "Up" 3-26
Power-Fail Processing 3-26
Power Down Sequence 3-27
Power Up Sequence 3-27
Restart I/O Sequence 3-28
Program Scheduling by Drivers 3-29
Determination of Operating System Environment 3-32
Subroutines for Special Mapping Functions 3-34
Mapping in RTE-III and RTE-M/III 3-35
Mapping in RTE-IV and RTE-6/VM 3-37
Obtaining the Subchannel 3-39
Operating System Trap Cell Instructions 3-39
Sample Standard RTE Driver 3-40

Chapter 4 Writing Privileged RTE Drivers

Introduction 4-1
General Privileged Driver Structure and Operation 4-3
Initiation Section 4-5
Privileged Section 4-7
Completion Section 4-11
Privileged Driver Design Considerations 4-11
Communication with User Programs 4-12
Discussion of Sample DMS Privileged Driver 4-13
Initiation Section 4-13
Privileged Section 4-14
Completion Section 4-15
Time-Out Values for Privileged Drivers 4-15
Subroutines for Special Mapping Functions 4-15
Mapping in RTE-III and RTE-M/III 4-17
Mapping in RTE-IV and RTE-6/VM 4-19
Sample Privileged Drivers 4-22

List of Illustrations

Title	Page
Equipment Table Entry Format	2-6
Expansion of CONWD Word (EQT Entry Word 6)	2-8
Device Reference Table Entry Format	2-10
Device Reference Table	2-11
Interrupt Table	2-15
Driver Mapping Table	2-17
Unbuffered I/O Read Request	2-19
EQT Words Set by RTE	3-5
EQT Words the Driver Can Modify	3-6
I/O Driver Initiation Section	3-7
I/O Driver Continuation/Completion Section	3-11
DCPC Channel Assignment Words	3-18
Determining DCPC Assignment	3-22
Standard RTE-Driver Example	3-40
DMS Privileged RTE Driver Example	4-22
Non-DMS Privileged RTE Driver Example	4-30

List of Tables

Title	Page
Base Page Communications Area -- I/O Operations	2-4
$OPSY Word Format	3-33

Chapter 1
Introduction

Purpose

The RTE Operating System Driver Writing Manual is a reference for those users who wish to develop their own device drivers. A device driver provides the software interface between a peripheral device and the RTE operating system. Many drivers for HP peripherals have already been written and are available from HP. Users who wish to interface peripherals that are not supported by HP will require specialized drivers. The information in this manual will aid the user in the development of such routines.

Note that it is not the purpose of the manual to describe the various HP-supplied drivers in any detail. Each of these is described in a separate manual specific to the driver.

Scope

The manual first provides the reader with a general description of the input/output (I/O) characteristics of the RTE family of operating systems. The techniques and requirements for developing device drivers are then presented in subsequent sections.

Since all of the RTE operating systems have the same general I/O structure, the manual can be used to develop general purpose drivers for use in any of these RTE systems. There are some areas where differences between operating systems may affect driver structure and operation; these areas are clearly pointed out in the text with notations such as "RTE-IV only" or "RTE-III only." Phrases such as "RTE-III only" should be interpreted as referring to both RTE-III (disc-based system) and RTE-M/III (memory-based equivalent of RTE-III).

Supporting Documentation

To use this manual effectively, the reader should be thoroughly familiar with HP Assembly Language and with the Programming and Operating Manual for the RTE system in which the driver is to be used. Refer to the Documentation Map at the front of this manual for information on these and other available manuals. For specific information on an HP supplied driver, refer to the appropriate driver manual.

Chapter 2
RTE Input/Output Structure

Introduction

In RTE, centralized control and logical referencing of input and output (I/O) operations effect simple device-independent programming. By means of several user-defined I/O tables, I/O drivers, and program EXEC calls, the programmer is relieved of most I/O problems. To understand the software I/O characteristics of RTE, the user should be familiar with two hardware related terms used in this manual:

I/O Controller — A combination of I/O card, cable, and (for some devices) controller box used to control one or more I/O devices on a computer I/O select code.

I/O Device — A physical unit (or portion of a unit) identified in the RTE operating system by means of an Equipment Table entry and a subchannel assignment.

Each I/O device is interfaced to the computer through an I/O controller. This controller is associated with one or more of the computer I/O select codes. Interrupts from controllers on specific select codes are directed to specific memory locations in the computer for system processing.

It is also important to note the difference between a synchronous device and a non-synchronous device. An interrupt from a synchronous device controller must be processed within a specified time period, or the data will be lost. Examples of synchronous devices are moving-head disc drives and nine-track magnetic tape drives. Non-synchronous devices have no such requirement, and interrupts from these device controllers can be serviced whenever the computer is able to do so. Examples of non-synchronous devices include paper tape punches and readers.

Software I/O Structure

The RTE I/O structure is made up of two general types of software (the system I/O processor and the various device drivers) and a number of I/O tables and a communications area (the Equipment Table, the Device Reference Table, the Interrupt Table, the Driver Mapping Table (RTE-IV and RTE-6/VM only), and the Base Page Communications Area). These tables and areas are used for communication between the system and the drivers, and for control of the many I/O operations that can be in progress simultaneously. Each component of the I/O structure is discussed individually in this subsection. A summary of the overall I/O process is given in the next subsection.

Input/Output Device Drivers

Input/Output device driver provide the software interface between peripheral I/O devices and the operating system. Drivers are responsible for the initiation, continuation, and completion of all data transfers between an I/O device and the computer. Drivers communicate with the system directly via parameter passing, and indirectly through the various tables and communications areas (particularly the Equipment Table and the Base Page Communications Area) that are discussed later in this subsection. There are two types of drivers; standard and privileged. Standard drivers are simpler and can be used for most asynchronous devices and some high speed and synchronous devices (if DCPC transfers are used); these drivers are discussed in Chapter 3. Privileged drivers are more complex and are generally used for high speed and synchronous devices that require driver interaction on each data word transferred (i.e.,DCPC transfers cannot be used); these drivers are discussed in Chapter 4.

System I/O Processor

The system I/O processor provides the software interface between user programs that perform I/O and the drivers that actually handle the I/O operations. The system I/O processor checks user I/O calls for validity, suspends programs while their I/O is in progress (if necessary), calls drivers to initiate the I/O data transfers, directs controller interrupts to the appropriate drivers, and restarts programs suspended for I/O. The mechanism for communication between the system I/O processor and user programs is the EXEC call and its associated parameter. Communication between the system I/O processor and drivers is handled directly via parameters and indirectly through the various I/O tables discussed in this section.

Two general areas within the system I/O processor are discussed in this manual; IOC and CIC. The Input/Output Control module (IOC) is entered when a user program makes an I/O request. IOC is responsible for initiating the I/O transfer by calling the appropriate driver. The Central Interrupt Control module (CIC) is entered when a device controller interrupt is detected. CIC is responsible for calling the correct driver to handle the interrupt.

Base Page Communications Area

A block of storage in base page contains the system's communications area and is used by RTE to define request parameters, I/O tables, scheduling lists, operating parameters, memory bounds, etc. The RTE Assembler and Macroassembler allow absolute references to addresses less than octal 2000 so that user programs can read information from the base page. Programs cannot alter the base page, however, because of the memory protect feature of RTE. Table 2-1 illustrates the portion of the Base Page Communications Area that pertains to I/O operations. The meaning and use of the various words illustrated in the table will become clear in subsequent sections of this manual. (For a complete description of the Base Page Communications Area, refer to the appropriate RTE System Programming and Operating Manual.)

Table 2-1. Base Page Communications Area -- I/O Operations

OCTAL LOCATION	CONTENTS	DESCRIPTION
. . .		
01650	EQTA	Address of Equipment Table (EQT)
01651	EQT#	Number of EQT entries
01652	DRT	Address of Device Reference Word 1 Table
01653	LUMAX	Number of logical units (in Device Reference Table)
01654	INTBA	Address of Interrupt Table
01655	INTLG	Number of Interrupt Table entries
01656	TAT	Address of Track Assignment Table (disc-based systems only)
01657	KEYWD	Address of keyword block
01660 01661 01662 01663 01664 01665 01666 01667 01670 01671 01672	EQT1 EQT2 EQT3 EQT4 EQT5 EQT6 EQT7 EQT8 EQT9 EQT10 EQT11	Addresses of first 11 words of current EQT entry (see location of 01771 for last 4 words)
01673	CHAN	Current DCPC Select Code (6 or 7)
. . .		
01717	XEQT	ID segment address of current program
. . .		
01737	DUMMY	I/O channel of privileged interrupt card (0 if none)
. . .		
01770	MPTFL	Memory Protect On/Off (0/1) flag.
01771 01772 01773 01774	EQT12 EQT13 EQT14 EQT15	Addresses of last 4 words of current EQT entry

Equipment Table

The Equipment Table (EQT) is used to maintain a list of all the I/O equipment in the system. This table consists of a number of EQT entries, with one EQT entry for each I/O controller defined in the system at generation time. The EQT entry contains all of the information required by the system and the associated driver to operate the equipment, including: the I/O select code in which the controller is interfaced to the computer, the driver type, and the various requirements and specifications of the controller or driver (e.g., DCPC, buffering, time-out, power-fail, etc.) To distinguish between multiple I/O devices connected to a single controller, the system also inserts the subchannel number of the device being referenced into the EQT entry before calling the driver.

The format of each EQT entry is illustrated in Figure 2-1. Some information in the EQT entry is static; other parts are dynamic. information marked <A> is fixed at generation time (or, for the I/O select code number in RTE-IV and RTE-6/VM , at reconfiguration time) and never changes during on-line operation of the system. Words marked <B> are also fixed at generation time (or, for RTE-IV and RTE-6/VM, at reconfiguration time) but can be changed on-line via operator commands. Information marked <C> is modified or set up for the driver prior to each I/O initialization; it informs the driver of the nature of the request. Words marked <D> are not used by the system and are therefore available to the driver for use as temporary storage for the duration of each I/O request.

EQT words 9 and 10 are available for use as temporary storage unless optional parameters were specified in the EXEC call. For the case of an EXEC call with the control word's Z bit clear, EQT words 9 and 10 contain the values of the optional parameters. If the Z bit was set, however, EQT word 9 contains the address of the optional buffer while word 10 contains the length of the buffer.

RTE Input/Output Structure

WORD	CONTENTS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	I/O REQUEST LIST POINTER <C>														
2	R	DRIVER "INITIATION" SECTION ADDRESS <A>														
3	<F> A	DRIVER "CONTINUATION/COMPLETION" SECTION ADDRESS <A>														
4	<A> D	<B> B	<E> P	<E> S	<C> T	LOWER 5 BITS OF SUBCHANNEL #<C>					I/O SELECT CODE # <A>					
5	AV <F>	EQUIPMENT TYPE CODE <A>						STATUS <E>								
6	CONWD (CURRENT I/O REQUEST WORD) <C>													<C> U		
7	REQUEST BUFFER ADDRESS <C>															
8	REQUEST BUFFER LENGTH <C>															
9	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>															
10	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>															
11	TEMPORARY STORAGE FOR DRIVER <D>															
12	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION SIZE, IF ANY <A>															
13	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION STARTING ADDRESS, IF ANY <A>															
14	DEVICE TIME-OUT RESET VALUE <B>															
15	DEVICE TIME-OUT CLOCK <C>															

WHERE THE LETTERS IN BRACKETS (<>) INDICATE THE NATURE OF EACH DATA ITEM, AS FOLLOWS:

<A> = FIXED AT GENERATION TIME (OR, FOR I/O SELECT CODE # IN RTE-IV, AT RECONFIGURATION TIME); NEVER CHANGES

<B> = FIXED AT GENERATION TIME; CAN BE CHANGED ON-LINE.

<C> = SET UP OR MODIFIED AT EACH I/O INITIALIZATION.

<D> = AVAILABLE FOR USE AS TEMPORARY STORAGE BY DRIVER.

<E> = CAN BE SET BY DRIVER.

<F> = MAINTAINED BY SYSTEM.

AND WHERE:

R = (RESERVED FOR SYSTEM USE)

I/O REQUEST LIST POINTER = POINTER TO LIST OF REQUESTS QUEUED UP ON THIS EQT ENTRY. FIRST ENTRY IN LIST IS CURRENT REQUEST IN PROGRESS; ZERO IF NO REQUESTS.

A = 1 IF DCPC WAS ALLOCATED DYNAMICALLY AT REQUEST OF DRIVER CONTINUATION (RTE-IVB ONLY)

D = 1 IF DCPC REQUIRED

B = 1 IF AUTOMATIC OUTPUT BUFFERING USED

P = 1 IF DRIVER IS TO PROCESS POWER FAIL

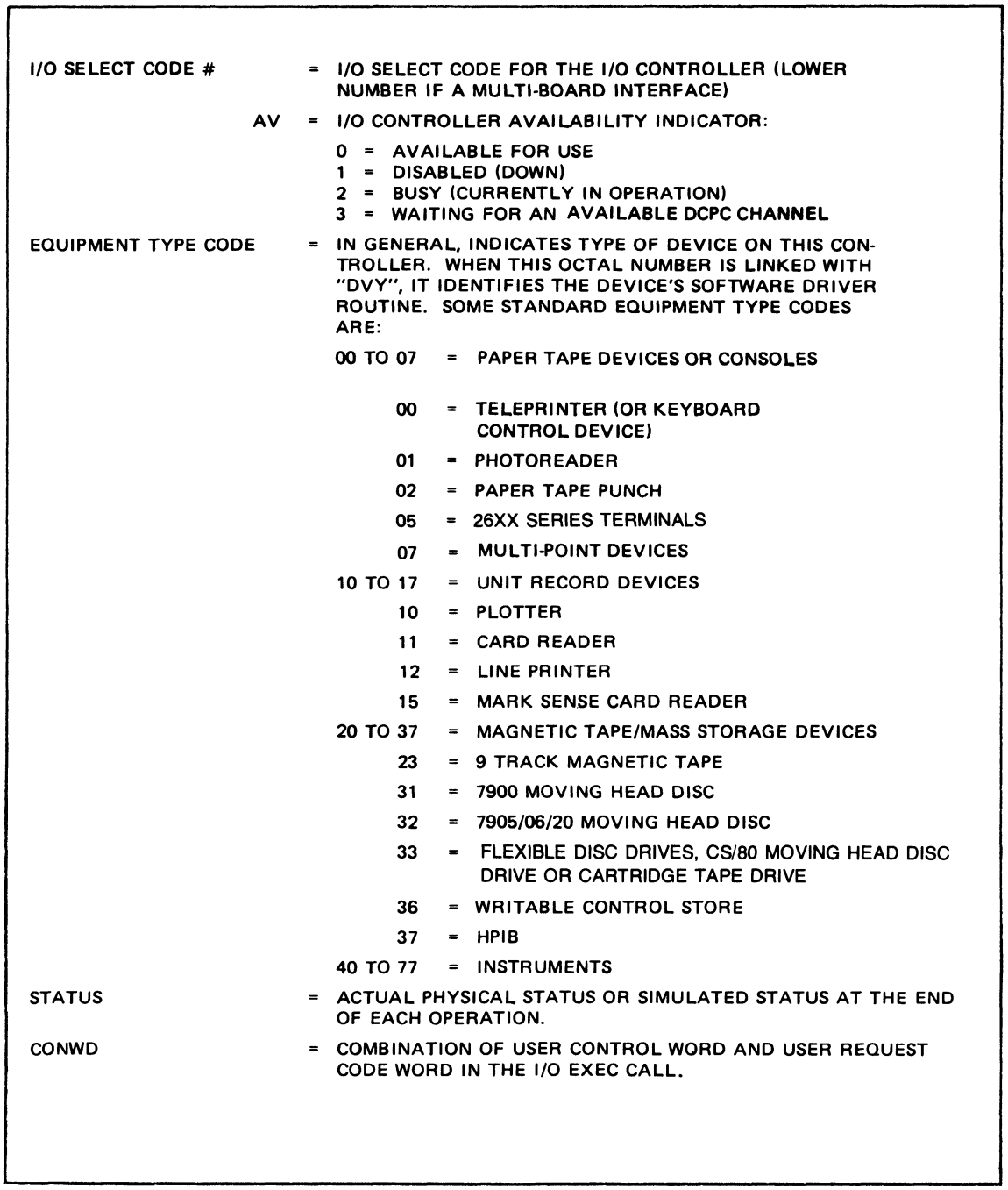
S = 1 IF DRIVER IS TO PROCESS TIME-OUT

T = 1 IF DEVICE TIMED OUT (SYSTEM SETS TO ZERO BEFORE EACH I/O REQUEST)

U = BIT 6 OF SUBCHANNEL # (RTE-6/VM ONLY)

SUBCHANNEL # = LAST SUBCHANNEL ADDRESSED

Figure 2-1. Equipment Table Entry Format (Sheet 1 of 2)



8300-96

Figure 2-1. Equipment Table Entry Format (Sheet 2 of 2)

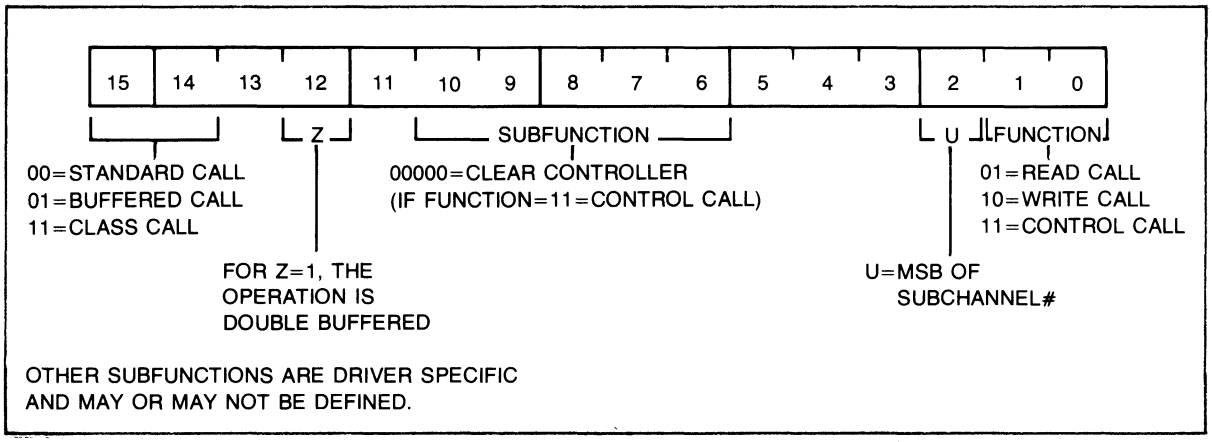


Figure 2-2. Expansion of CONWD Word (EQT Entry Word 6)

If the number of words marked <D> does not provide sufficient temporary storage for the driver, additional space can be allocated at generation time by specifying that an EQT entry extension is needed for a particular EQT entry. This space can only be used to extend the referenced EQT entry and therefore should only be allocated for drivers that need the additional space. When an EQT entry extension is specified, EQT entry words 12 and 13 are used to identify the location and length of the extension (since the extension does not immediately follow the EQT entry) and therefore should not be modified by the driver. Otherwise, these words are available as temporary storage.

For programming convenience, the addresses of the words in the current EQT entry (except for words in the extension, if an extension exists) are placed in the Base Page Communications Area by the system before calling the driver to initiate or continue an I/O operation. In RTE-6/VM, if the first word is set up correctly it is assumed that the others are correct and are not reset. A driver should use these addresses instead of computing them from the EQT entry number and the start of the Equipment Table. In this way, the driver can remain independent of the actual organization of the Equipment Table in memory.

All Equipment Table entries are located sequentially in memory beginning with EQT entry number 1. The address of the first entry and the total number of entries in the table can be found in the Base Page Communications Area.

Logical Unit Numbers

Logical unit numbers (LUs) provide the RTE user with the capability of logically addressing the physical devices defined by the Equipment Table. LU numbers are maintained by the Device Reference Table (see below), and their definition can be changed on-line by the LU operator request. This scheme allows the programmer to reference changeable logical units instead of fixed physical units.

The functions of Logical Units 0 through 6 are predefined in the RTE system as follows:

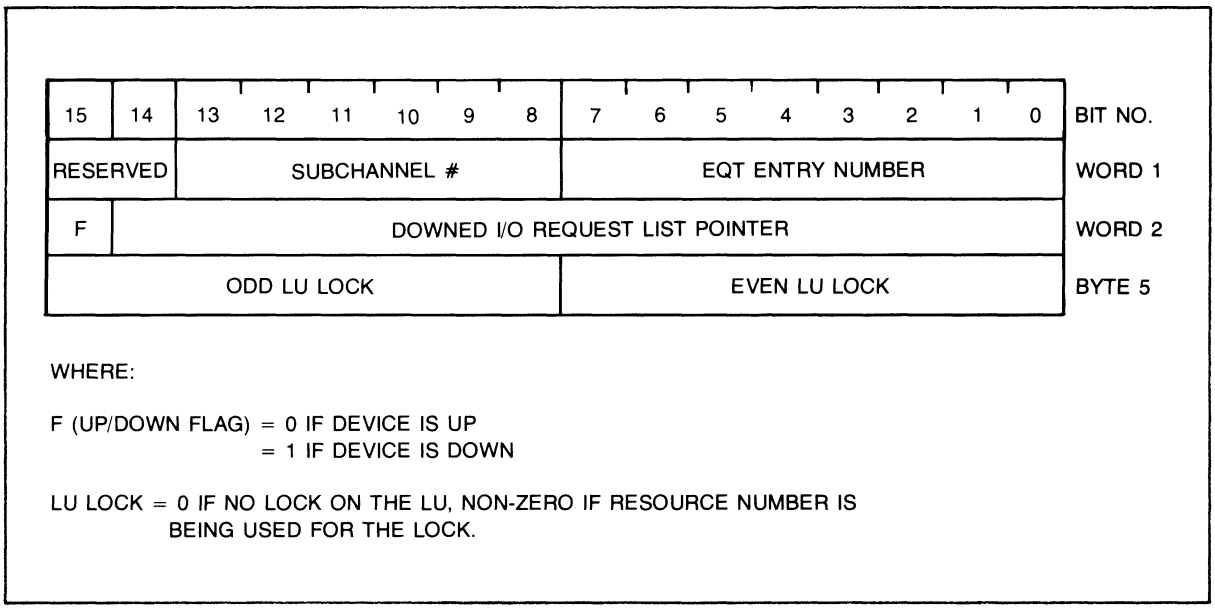
0 - "bit bucket" (null device, no entry in Device Reference Table)
1 - system console
2 - reserved for system (system disc subchannel in disc-based systems)
3 - reserved for system (auxiliary disc subchannel in disc-based systems)
4 - standard output device
5 - standard input device
6 - standard list device

Logical Unit 8 is recommended for the magnetic tape device, if one is present in the system. Peripheral discs must be assigned logical units greater than 6. Additional logical units may be assigned for any functions desired.

Device Reference Table

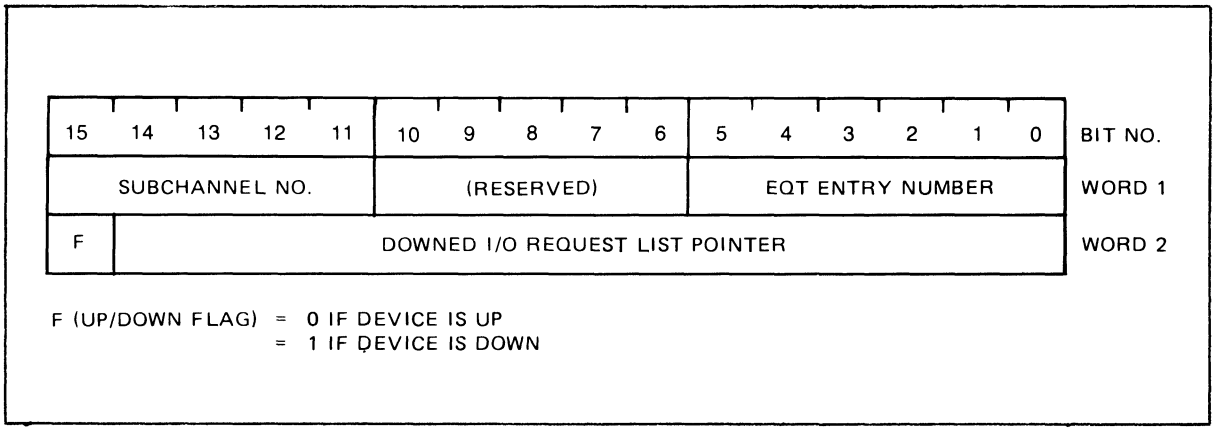
The Device Reference Table (DRT) is part of the mechanism by which unit numbers for I/O are implemented. RTE users request I/O by specifying a logical unit number. The DRT is used to translate this logical unit number into a physical device, as specified by an EQT entry number and subchannel. The DRT is also used to queue requests for I/O on a device when it is unavailable (down). (The DRT is not used to queue requests when the device is up. The request list for available (i.e., up) devices originates from word 1 of the EQT entry as illustrated in Figure 2-1.)

Each DRT entry is two words long (two and one-half words for RTE-6/VM). There is one entry for each logical unit number defined at generation time, beginning with logical unit 1. The format of each entry is illustrated in Figure 2-3. The word of the entry contains several items, including: 1) the EQT entry number of the controller assigned to the logical unit, and 2) the subchannel number of the specific device on that controller to be referenced. The second word of each entry contains the status of the logical unit: up (available) or down (unavailable). If the device is down, word two also contains a pointer to the list of requests waiting to access the LU. The extra half word in RTE-6/VM is used to indicate an LU lock (zero for no lock, non-zero for a resource number lock).



8300-95

Figure 2-3A. Device Reference Table Entry Format (for RTE-6/VM)



8300-97

Figure 2-3B. Device Reference Table Entry Format (except for RTE-6/VM)

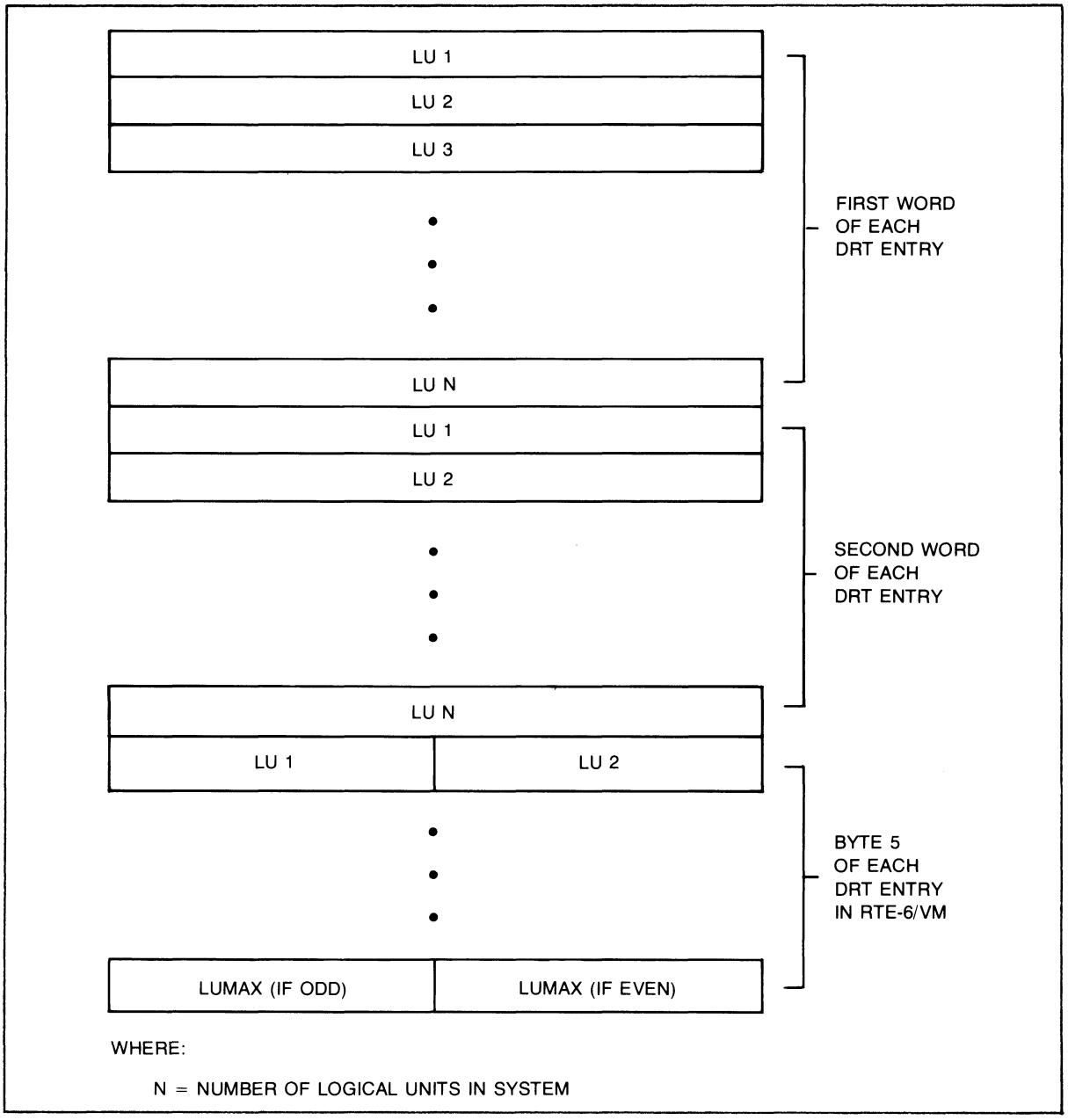


Figure 2-4. Device Reference Table

The DRT table is actually split into two separate parts (three parts for RTE-6/VM). The first part contains word 1 of each DRT entry, the second part contains word 2 of each DRT entry, and the third part (in RTE-6/VM) contains byte 5 of each DRT entry. This format is illustrated in Figure 2-4. The starting address and length of part one of the table can be found in the Base Page Communications Area. Part two is located in memory immediately following part one and has the same length as part one. Part three is located in memory immediately following part two.

Computer Interrupt Mechanism

When a device controller interrupts RTE, the computer transfers control to one of a group of memory locations on base page known as the interrupt trap cell. The I/O select code of the interrupting controller determines the location of the transfer. For example, interrupts from select code 12 cause a transfer to memory location 12. Interrupts from select code 13 cause a transfer to memory location 13, and so on. Select code numbers range from 4 to 77 (octal). Thus, the group of memory locations from 4 to 77 (octal) comprises the entire set of interrupt trap cells.

Transferring control to an interrupt trap cell causes the instruction located there to be executed. For all devices operating under the control of CIC, this instruction is a JSB LINK,I, where LINK is a base page link containing the address of the entry point to CIC. This instruction is initially set up by the RTE generator, and is reset into the trap cell each time the system is rebooted. In RTE-6/VM, the JSB LINK, I on an E-or F-Series CPU is replaced by four new instructions that handle these interrupts. The new instructions are described under Operating System Trap Cell instructions in the next chapter. The fact that the JSB instruction references an indirect address causes the computer to hold off any further interrupts for one instruction after the JSB. This gives CIC a chance to issue a CLF 0 instruction (which disables the interrupt system entirely) to prevent further interrupts from occurring while the current one is being processed.

Since CIC is entered at the same location for all device controller interrupts under its control, a method is needed by which the select code of the interrupting device controller can be determined. CIC obtains the interrupting select code number by accessing the contents of the computer's Central Interrupt Register via an LIA 4 instruction. CIC can then use this information to index into the Interrupt Table (see next subsection) to determine how to process the interrupt.

The interrupt trap cells are not limited to containing a JSB LINK,I instruction (where LINK contains the address of CIC). Other instructions can be placed in a trap cell by the generator or by a system routine. However, the trap cell should not contain any instruction other than a HALT instruction, an OS trap cell instruction (RTE-6/VM only) or a JSB indirect to an interrupt processing routine (such as CIC or a user-written routine) that saves the state of the machine on entry and restores it to its original state on exit. This includes saving and restoring the registers, state of the memory protect fence, etc.

Specifically, I/O instructions and NOP instructions must not be put into trap cells because they do not provide any way to restore the system to its original state, Microcode macro's (i.e., jumps to microcoded routines) may be used if a microcoded driver is used to process the interrupts, or if you are using the OS trap cell instructions (RTE-6/VM only).

Note that if a JSB instruction is placed in the interrupt trap cell, it must reference an indirect address. The indirect address keeps the interrupt system suppressed for one instruction after the JSB, as explained above. This allows the interrupt processing routine to issue a CLF 0 instruction to prevent further interrupts from occurring while the state of the machine is being saved. (Note that the generator automatically provides a base page link for all JSB instructions it places in the interrupt trap cells. A JSB indirect instruction is created whenever an "ENT," "PRG," or "EQT" entry is specified during generation.)

Systems without the power-fail/automatic restart feature have a HALT 4 instruction inserted by the generator into the power-fail interrupt trap cell (memory location 4). As a result, the computer will halt when a power-fail interrupt occurs. An example of a JSB to a user-written interrupt processing routine is discussed later in the "Writing Privileged Drivers" section of this manual.

Interrupt Table

The Interrupt Table directs CIC's actions when an interrupt occurs on any I/O select code that contains a JSB LINK,I instruction (where LINK contains the address of CIC). CIC can call a driver, schedule a specified program, or handle the interrupt itself.

There is one Interrupt Table entry for each I/O select code from 6 up to the highest select code defined in the system at generation. (Systems with I/O reconfiguration ability at boot-up (e.g., RTE-IV and RTE-6/VM) always include Interrupt Table entries for all select codes, even if some select codes were not defined in the initial generation.) Each Interrupt Table entry is one word long and can have three possible values: zero, positive, or negative.

1. If the entry is zero, the select code is undefined in the Interrupt Table. Any interrupts on this select code are illegal and cause the following message to be printed:

   ILL INT xx

   where xx is the octal I/O select code number. RTE then clears the interrupt flag on the select code and returns to the suspended process at the point of interruption. (Note that an Interrupt Table entry can also be zero if interrupts on the associated select code are handled by a special routine instead of by CIC and a driver. Refer to the "Writing Privileged Drivers" section later in this manual for more information on this subject.)

2. If the contents are positive, the entry contains the address of the EQT entry associated with the controller on the select code.

3. If the contents are negative, the entry contains the negative of the address of the ID segment of the program to be scheduled whenever an interrupt occurs on the select code. If such a program is not dormant when an interrupt occurs on the select code, the following message is output to the system console:

   SC03 INT xxxxx

   where xxxxx is the program name. RTE then clears the interrupt flag on the select code and control is returned to the suspended process at the point of interruption. This implies the program should be quick if scheduled often, and preferably memory-resident.

All Interrupt Table entries are located sequentially in memory beginning with the entries for I/O select codes 6 and 7 (DCPC). This format is illustrated in Figure 2-5. There are no entries for I/O select codes 4 and 5 because the system is able to process interrupts from these select codes (power-fail interrupts, memory protect violations, etc.) without the need for an Interrupt Table entry. The address of the first word of the table and the number of entries in the table can be found in the Base Page Communications Area.

NOTE

The reader should not confuse the interrupt trap cell area of the computer, which is located on base page, with the Interrupt Table of RTE, which is located elsewhere. The interrupt trap cells are those memory locations (4 to 77 octal) to which control is transferred when an interrupt occurs. The Interrupt Table, on the other hand, is merely a convenient way for RTE to record what action CIC should take when an interrupt occurs on a select code under CIC's control.

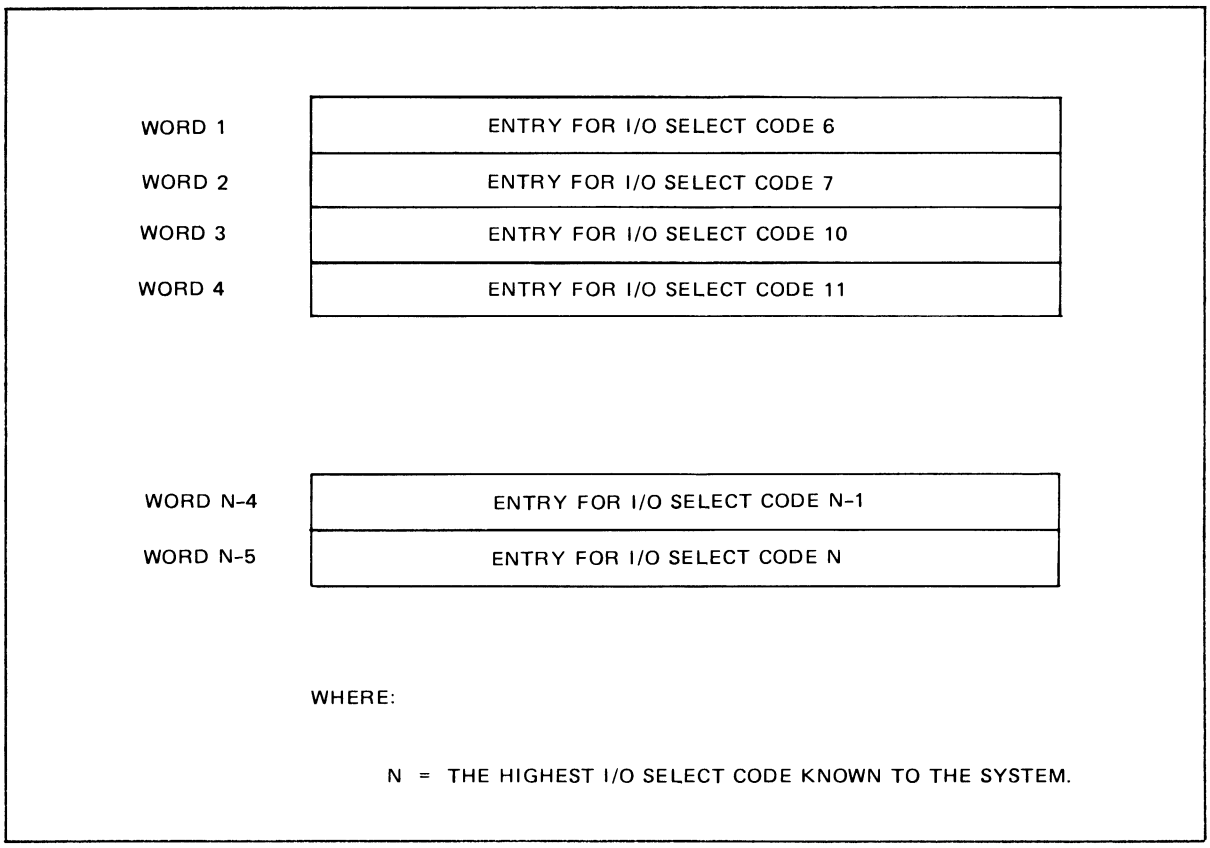


Figure 2-5. Interrupt Table

Driver Mapping Table (RTE-IV and RTE-6/VM Only)

In the RTE-IV and RTE-6/VM Operating Systems, drivers can be placed in one of two areas: in the System Driver Area (SDA) or in one of the driver partitions. Most standard drivers are placed in driver partitions. The SDA is primarily used for privileged drivers, drivers that do their own mapping, and very large drivers.

The Driver Mapping Table (DMT) is used to record where a driver resides in physical memory and other static and dynamic information about the driver and the location of the I/O request buffer.

There is one DMT entry associated with each EQT entry defined at generation time. Each entry is two words long, as illustrated in Figure 2-6. Word 1 is set up at generation time and its contents are never changed. It indicates whether the drive resides in the System Driver Area (SDA) or in a driver partition. If it is in the SDA, it also indicates whether or not the driver is doing its own memory mapping. (See the "Subroutines for Special Mapping Functions" subsection in Chapter 3.) If the driver is in a partition, word 1 also indicates the starting physical memory page number of the driver partition in which it is located.

Word 2 of the DMT entry is dynamic in nature and is set up at each I/O initialization of the associated EQT entry. This word indicates whether the I/O request buffer is located within a disc-resident program, memory-resident program, or system area. If a disc-resident program is making the request and the I/O request buffer is located within the program (i.e., an unbuffered request), word 2 also indicates the physical memory page number of the disc-resident program's base page. This information is used to save time on setting up the proper map when processing interrupts handled by the driver.

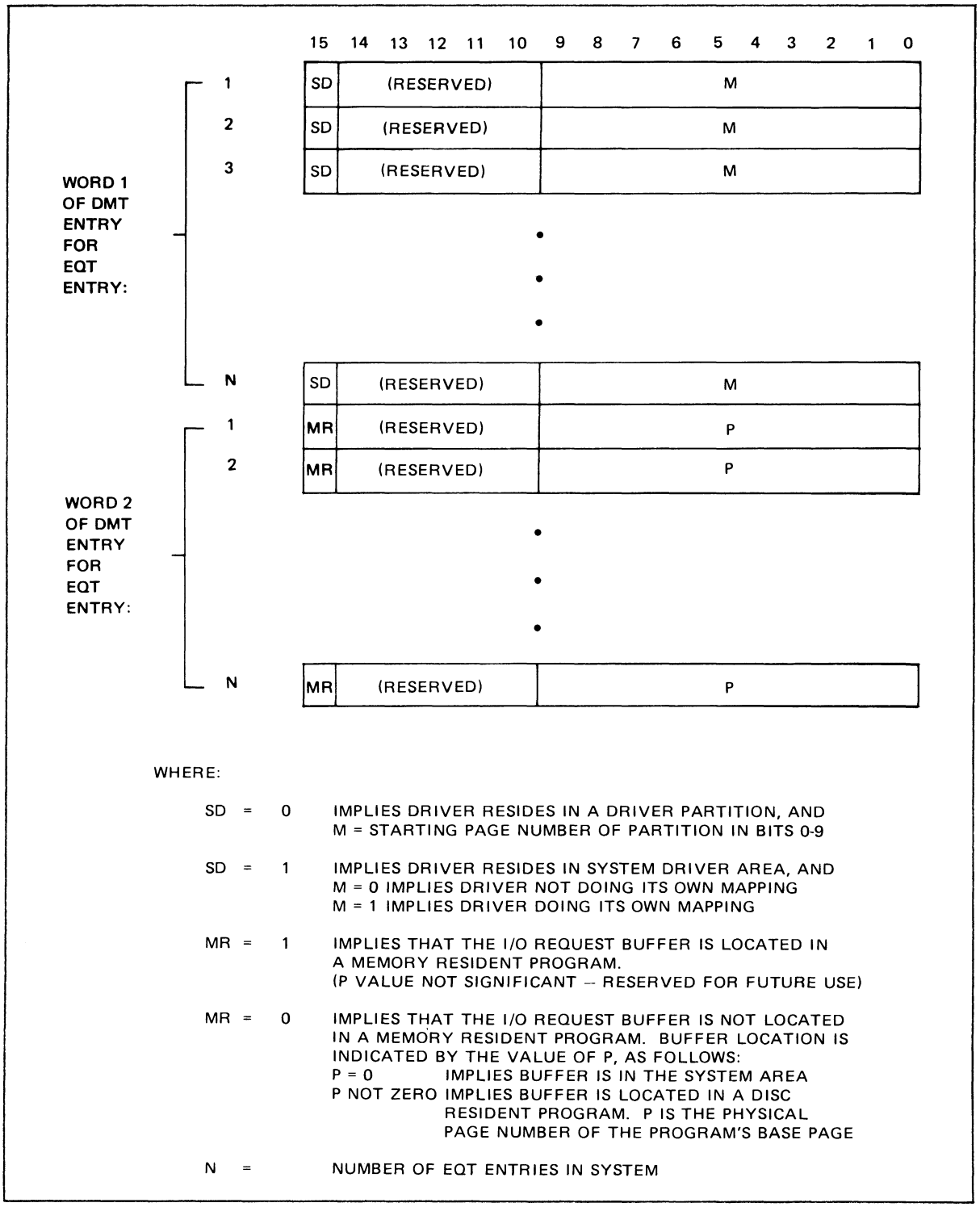
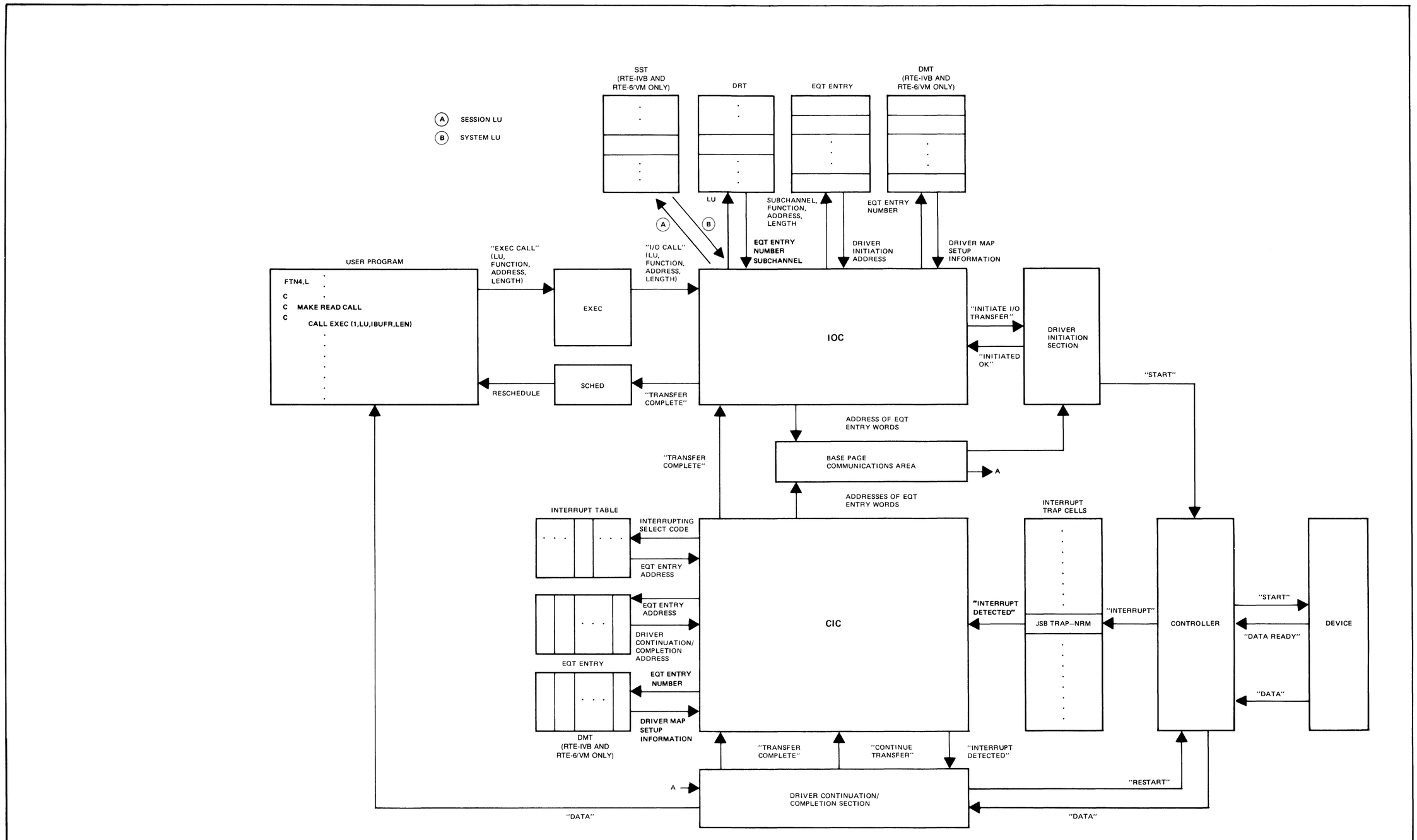


Figure 2-6. Driver Mapping Table

General Operation of RTE I/O

Input/Output transfers in RTE can be conveniently broken into three parts for discussion: initiation, continuation, and completion. A user program is involved only in the initiation and completion phases; the system I/O processor and the device drivers are involved in all three phases. The following is a simplified discussion of each phase. As an aid to understanding this explanation, the general flow of events for an unbuffered I/O READ request is illustrated in Figure 2-7.



7700-140A

Figure 2-7. Unbuffered I/O READ Request

I/O Initiation

A user program makes an EXEC call to initiate I/O transfers. Parameters passed along with this call specify the logical unit, control information, buffer location, buffer length, and type of request (READ, WRITE, or CONTROL) to be made. The user request is channelled to the IOC (Input/Output Control) module of the system by the RTE request processor. The request is checked for legality and rejected if any errors are found. If there are no errors, the logical unit number supplied is used to index into the DRT (Device Reference Table) to determine which I/O controller (EQT entry number) and device (subchannel) are actually being referenced. The I/O request is then linked into the request list for the referenced controller.

If the device controller is available (i.e., no prior requests are pending), the parameters of the request are put into the associated EQT entry, the addresses of the EQT entry words are set into the Base Page Communications Area for convenience, the proper map (System or User) is enabled (performed in systems with Dynamic Mapping only), and the "initiation" section of the driver is called. This section initializes the device controller, starts the data transfer or control function, and returns to IOC.

IOC then returns to the system's dispatching module to begin execution of the highest priority scheduled program. If the operation was successfully initiated by the driver, the data transfer is now under way.

I/O Continuation

When the device controller finishes transferring a data word, or block of words, it interrupts the computer. This causes a transfer to one of the interrupt locations in the computer's memory, and the instruction located there is executed. For most I/O devices, this instruction is a JSB LINK,I (where LINK contains the address of the entry point to CIC). Execution of this instruction causes control to be transferred to CIC, the Central Interrupt Control module of the system. CIC obtains the number of the interrupting select code from the computer's Central Interrupt Register and uses it to index into the Interrupt Table. In RTE-6/VM, most of this is handled by a trap cell instruction.

For those I/O processes operating under the control of CIC and a driver, the Interrupt Table tells CIC which EQT entry is associated with the interrupting select code. CIC looks at the EQT entry, determines which driver is responsible for handling the interrupt, enables the correct map (System or User) in systems with Dynamic Mapping, and calls the driver's "continuation/completion" section to process the interrupt. The driver either accepts the data from the device (read operation) or sends more data to the device (write operation) and restarts the device. Return is then made to CIC with a code indicating that more interrupts are expected. This process (interrupt, CIC, driver, CIC) is repeated once for each word or block of words transferred until the entire transfer is complete.

I/O Completion

Eventually the driver will determine that the required amount of data has been transferred and that the I/O process is now complete. The driver then returns to CIC with a special code indicating that the I/O operation is complete and can be terminated; no more interrupts are expected.

CIC, in turn, transfers control back to IOC to terminate the IOC process. IOC causes the program that made the initial I/O request to be placed back into the scheduled list and checks to see if there are any other I/O requests pending for this controller. If at least one request is pending, the initiation section of the driver is again called to begin the next operation. IOC then returns control to the system's dispatching module to begin execution of the highest priority scheduled program.

Chapter 3
Writing Standard RTE Drivers

Introduction

This section describes in detail the structure, operation, and design of standard RTE drivers. Standard drivers are fairly simple in structure and can generally be used to control most asynchronous devices. They can also be used to control synchronous and high-speed devices if these devices are driven under DCPC control. DCPC processing is also described in this section.

An alternate method for controlling synchronous and high-speed devices is to employ the more complex privileged driver. Chpter 4 of this manual describes the differences in the design of privileged drivers versus standard drivers. Thus, if the user wishes to design a privileged driver, the material in this section should be read and understood before continuing with the privileged driver discussion in Chapter 4.

Note that the operation of RTE requires the synchronous and high-speed devices be driven either by a standard driver utilizing DCPC transfers, or by a privileged driver. This is necessary to ensure that interrupts from such devices are serviced within the required response time. The reader should keep this requirement in mind when deciding upon the type of driver to be written.

General Driver Structure and Operation

An I/O driver, operating under control of the Input/Output Control (IOC) and Central Interrupt Control (CIC) modules of RTE, is responsible for all data transfers between an I/O device controller and the computer. Each driver is written in two functional sections: an initiation section and a continuation/completion section. The section is responsible for starting up the device and initiating the first data transfer. The continuation/completion section is responsible for processing each interrupt generated by the device under its control. This involves accepting data from the device (read operation), sending more data to the device (write operation), and then restarting the device to continue the transfer. Eventually, the continuation/completion section determines that a sufficient amount of data has been transferred and terminates the I/O operation.

A standard RTE driver operates with the interrupt system disabled (or effectively disabled, if the system contains a Privileged Interrupt card. Refer to the "Writing Privileged Drivers" section of this manual.) This means that once a driver is entered to process an interrupt, no other interrupts (except privileged interrupts) can be serviced until the driver completes its operation and returns to CIC. (CIC turns the interrupt system back on to allow other interrupts to occur.) Drivers should therefore be coded as efficiently as possible to minimize the amount of time that the interrupt system is disabled and the processing of other interrupts is delayed.

Driver Naming Requirements

To facilitate the identification of driver programs and entry points, the following naming scheme has been devised. This scheme must be incorporated into the design of all RTE drivers so that the RTE system generator programs can identify the drivers and relocate them in the proper memory area of the operating system.

a. Driver names must be five characters in length, beginning with the characters "DV" and ending with a two-digit octal number (known as the equipment type code of the device).

b. The initiation and continuation/completion sections must have entry points whose names are four characters in length, beginning with the character "I" or "C", respectively, and ending with the same two-digit octal number used in the driver name.

Thus, if "nn" is the octal equipment type code, Ixnn and Cxnn are the entry point names of the initiation and continuation/completion sections, respectively. DVynn is the driver name.

The user is allowed some flexibility in the choice of the "x" (in Ixnn and Cxnn) and "y" (in DVynn) characters referred to above. This flexibility allows several drivers with the same octal equipment type code to have unique names and entry points. The rules for the choice of "x" and "y" are:

If "y" is "R" then "x" = "."

If "y" is not "R" then "x" = "y"

Using the above rules, a driver named DVR66 has entry points I.66 and C.66. A driver named DVA66 has entry points IA66 and CA66.

The octal equipment type code (nn) can be any octal number between 00 and 77. A table of "standard" type codes is given in Figure 2-1. Care should be taken to choose the type code and/or "x" and "y" characters so that new driver names and entry points do not conflict with those of any standard HP drivers or other user written drivers present in the system.

Initiation Section

The IOC module of RTE calls the driver initiation section when an I/O transfer is initiated. (Note that the initiation section may also be entered when a DCPC channel is assigned by IOC in response to a dynamic DCPC request by the driver. Refer to the "DCPC Processing" subsection of this manual.) Prior to actually entering the driver initiation section, IOC sets up all information needed by the driver to process the call in the associated EQT entry and in the Base Page Communications Area, as follows:

a. Locations EQT1 through EQT15 in the Base Page Communications Area are set to contain the addresses of each word of the EQT entry associated with the call. Base page word EQT1 is set to contain the address of EQT entry word 1, base page word EQT2 is set to contain the address of EQT entry word 2, and so on. If the driver uses DCPC (that is, if bit 15 of EQT entry word 4 is set), IOC also assigns a DCPC channel to the driver and stores the DCPC channel number in base page word CHAN.

b. Words 6 through 10 of the EQT entry pointed to by the Base Page Communications Area are set to contain the request parameters from the user's EXEC call (request code, subfunction, buffer address, buffer length, and optional parameters, if present). Note that EQT entry word 6 (CONWD) contains the CONWD from the user's EXEC call, modified to contain the request code in bits 0 and 1 in place of the logical unit. The subchannel being referenced by the call is placed into bits 6 through 10 of EQT entry word 4. For RTE-6/VM only, the subchannel number occupies six bits, the sixth (most significant) bit being bit 2 of word 6. Refer to the EQT diagram in Figure 2-1.

c. CIC also sets up and enables the correct map (System or User) needed by the driver to process the call. (This step is performed in systems with Dynamic Mapping only.)

After performing these tasks, IOC enters the driver directly via a jump subroutine to the initiation entry point Ixnn (JSB Ixnn). Upon entry, the A-Register contains the I/O select code of the controller being referenced in the call. (This same information is present in bits 0 through 5 of EQT entry word 4.) Later, when the driver has completed (or rejected, if necessary) the initialization procedure, it must return to IOC via a jump indirect through the Ixnn entry point (JMP Ixnn,I).

Once entered, the driver is free to use EQT entry words 6 through 13 in any way, but words 1 through 4 and 14 must not be altered, except for the D, P and S bits of EQT word 4. If an EQT entry extension was specified at generation, the space in the extention is also available to the driver. In this case words 12 and 13 (which define the extention) must not be modified. The driver can also update the status field in word 5, if appropriate, but this must be done without altering the rest of word 5. Finally, EQT entry word 15 may be modified, if desired, to set a time-out value for the device. (Refer to the "I/O Controller Time-Out" subsection in this manual.)

Figure 3-1 shows the EQT words set by RTE, while Figure 3-2 shows the EQT words the driver can modify.

Functions of the Initiation Section

As part of the general I/O structure of RTE, the initiation section of a standard driver performs the functions illustrated in Figure 3-3. A more detailed description of the initiation section functions is given below.

a. Checks for power-fail/automatic restart entry by examining bit bit 15 of EQT entry word 5, which is set to 1 only on this type of entry. If bit 15 is set, the appropriate power-fail/automatic restart processing should be done. This check need only be made by drivers that are designed to process power-fail interrupts (as described in the "Power-Fail Processing" subsection of this manual).

b. Rejects the request by following the procedure described in step "g" if:

    1. A status check of the device or controller indicates that it is inoperable, or

    2. The request code or other parameters are illegal.

c. Configures all I/O instructions in the driver to reference the specific I/O select code (and DCPC channel, if used) of the device controller.

d. Initializes DCPC, if used. (Refer to the "DCPC Processing" subsection of this manual.)

e. Initializes software flags and activates the device controller. All variable information pertinent to the transmission must be saved in the EQT entry associated with the controller because the driver may be called for another controller before the first operation is complete.

f. Optionally sets the device controller time-out clock (EQT entry word 15) to modify the time-out value inserted there by the system. (Refer to the "I/O Controller Time-Out" subsection of this manual.)

WORD	CONTENTS																
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	I/O REQUEST LIST POINTER <C>															
2	R	DRIVER "INITIATION" SECTION ADDRESS <A>															
3	<F> A	DRIVER "CONTINUATION/COMPLETION" SECTION ADDRESS <A>															
4	<A> D	<B> B	<E> P	<E> S	<C> T	LOWER 5 BITS OF SUBCHANNEL #<C>					I/O SELECT CODE # <A>						
5	AV <F>	EQUIPMENT TYPE CODE <A>						STATUS <E>									
6	CONWD (CURRENT I/O REQUEST WORD) <C>													<C> U			
7	REQUEST BUFFER ADDRESS <C>																
8	REQUEST BUFFER LENGTH <C>																
9	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>																
10	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>																
11	TEMPORARY STORAGE FOR DRIVER <D>																
12	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION SIZE, IF ANY <A>																
13	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION STARTING ADDRESS, IF ANY <A>																
14	DEVICE TIME-OUT RESET VALUE <B>																
15	DEVICE TIME-OUT CLOCK <C>																

Figure 3-1. EQT Words Set by RTE (shaded portions)

WORD	CONTENTS															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	R	I/O REQUEST LIST POINTER <C>														
2	R	DRIVER "INITIATION" SECTION ADDRESS <A>														
3	<F> A	DRIVER "CONTINUATION/COMPLETION" SECTION ADDRESS <A>														
4	<A> D	<B> B	<E> P	<E> S	<C> T	LOWER 5 BITS OF SUBCHANNEL #<C>					I/O SELECT CODE # <A>					
5	AV <F>	EQUIPMENT TYPE CODE <A>						STATUS <E>								
6	CONWD (CURRENT I/O REQUEST WORD) <C>													<C> U		
7	REQUEST BUFFER ADDRESS <C>															
8	REQUEST BUFFER LENGTH <C>															
9	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>															
10	TEMPORARY STORAGE <D> OR OPTIONAL PARAMETER <C>															
11	TEMPORARY STORAGE FOR DRIVER <D>															
12	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION SIZE, IF ANY <A>															
13	TEMPORARY STORAGE FOR DRIVER <D> OR EQT EXTENSION STARTING ADDRESS, IF ANY <A>															
14	DEVICE TIME-OUT RESET VALUE <B>															
15	DEVICE TIME-OUT CLOCK <C>															

Figure 3-2. EQT words the Driver Can Modify (shaded portions)

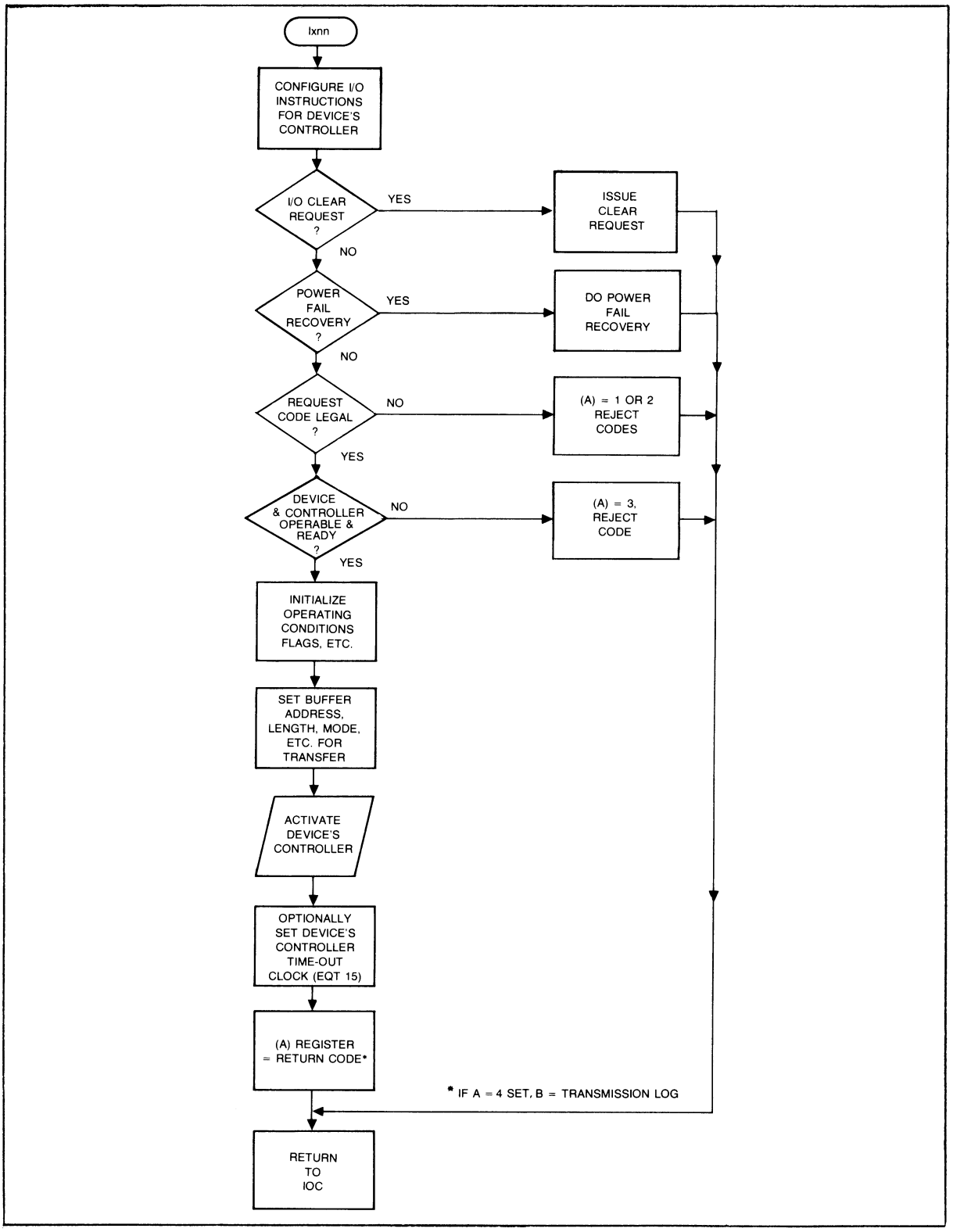


Figure 3-3. I/O Driver Initiation Section

g. Returns to IOC (via JMP Ixnn,I) with the A-Register set to indicate initiation or rejection (and the cause of the rejection) as follows:

A-Reg Status

0 The operation was initiated successfully.

1 Operation rejected: read or write illegal for device (IO07, program aborted).

2 Operation rejected: control request illegal or undefined.

3 Operation rejected: equipment malfunction or not ready (IONR message issued).

4 The operation was immediately completed. This means that the driver was able to completely satisfy the request without the need of a subsequent interrupt and that the program making the I/O call can be rescheduled immediately. The B-Register should be set to the positive number of words or characters (depending upon which the user specified) transferred. This value is known as the transmission log.

5 A DCPC channel is required but none was assigned by IOC. This can only occur when the "DCPC channel required" bit is not set in the EQT entry, and the driver decides that it needs a DCPC channel to process this specific call. (Refer to the "DCPC Processing" subsection of this manual.)

6 A DCPC channel was assigned by IOC but the driver did not use it and wants to give it up. After IOC deallocates the channel, processing continues as for a successful initiation return (A = 00. (RTE-IV and RTE-6/VM only.)

7 to 99 IOxx message issued: the program making the I/O request is aborted (unless the no-abort bit was set in the call), and an I/O error message is printed on the system console. (Note that this return can be used for unbuffered user I/O requests only. This therefore excludes the use of return codes 6 through 99 on any Class, buffered or system I/O requests.) The error message has the following format:

```
IOxx yyyyy
NNNNN ABORTED
```

where:

xx = The return code from the driver (decimal 06 to decimal 99),

yyyyy = The address of the aborted I/O request in program NNNNN, and

NNNNN = The name of the program that made the I/O request.

This type of return can be used by drivers to generate their own I/O error messages at the system console. Note that certain codes are reserved for system use, as follows:

Return Code	Reserved for
7 - 59	HP system modules and system drivers.
60 - 99	User written drivers

Continuation/Completion Section

The CIC module of RTE calls the continuation/completion section of a driver whenever an interrupt is recognized on an I/O controller associated with the driver. Before calling the driver to process the interrupt, CIC issues a clear flag instruction (CLF) to the interrupting select code, sets the addresses of the associated EQT entry into the Base Page Communications Area, and sets the interrupt source code (I/O select code of interrupting controller) into the A-Register. (The interrupting I/O select code address is also available in EQT entry word 4.)

CIC also sets up and enables the correct map (System or User) needed by the driver to process the call. (This step is performed in systems with Dynamic Mapping only.) The driver is then entered with the correct map enabled by executing a jump subroutine to the continuation/completion section of the driver at entry point Cxnn (JSB Cxnn). This call has the following format:

Location	Action
	(Set A-Register equal to interrupt source code)
P	JSB Cxnn
P+1	Completion return from Cxnn
P+2	Continuation return from Cxnn
P+3	Get/give up DCPC continuation return from Cxnn (RTE-IV and RTE-6/VM only).

The return points from Cxnn to CIC indicate whether:

1. The transfer has been completed.

2. The transfer is continuing (i.e., further interrupts are expected from the device controller).

3. The driver is requesting the allocation or deallocation of a DCPC channel.

The continuation/completion section of the driver is flowcharted in Figure 3-4 and performs the following functions. Note that steps "a" through "e" are always executed whenever the driver is entered at Cxnn. Then, depending on whether the I/O operation is now complete or is still continuing, the driver executes either step "f" (completion return) or step "g" (continuation return), respectively.

a. Checks whether bits 14-0 of EQT entry word 1 (the controller I/O request list pointer) equal zero. If so, a spurious interrupt has occurred (i.e., no I/O operation was in progress on the controller). The driver can handle the spurious interrupt in one of two ways:

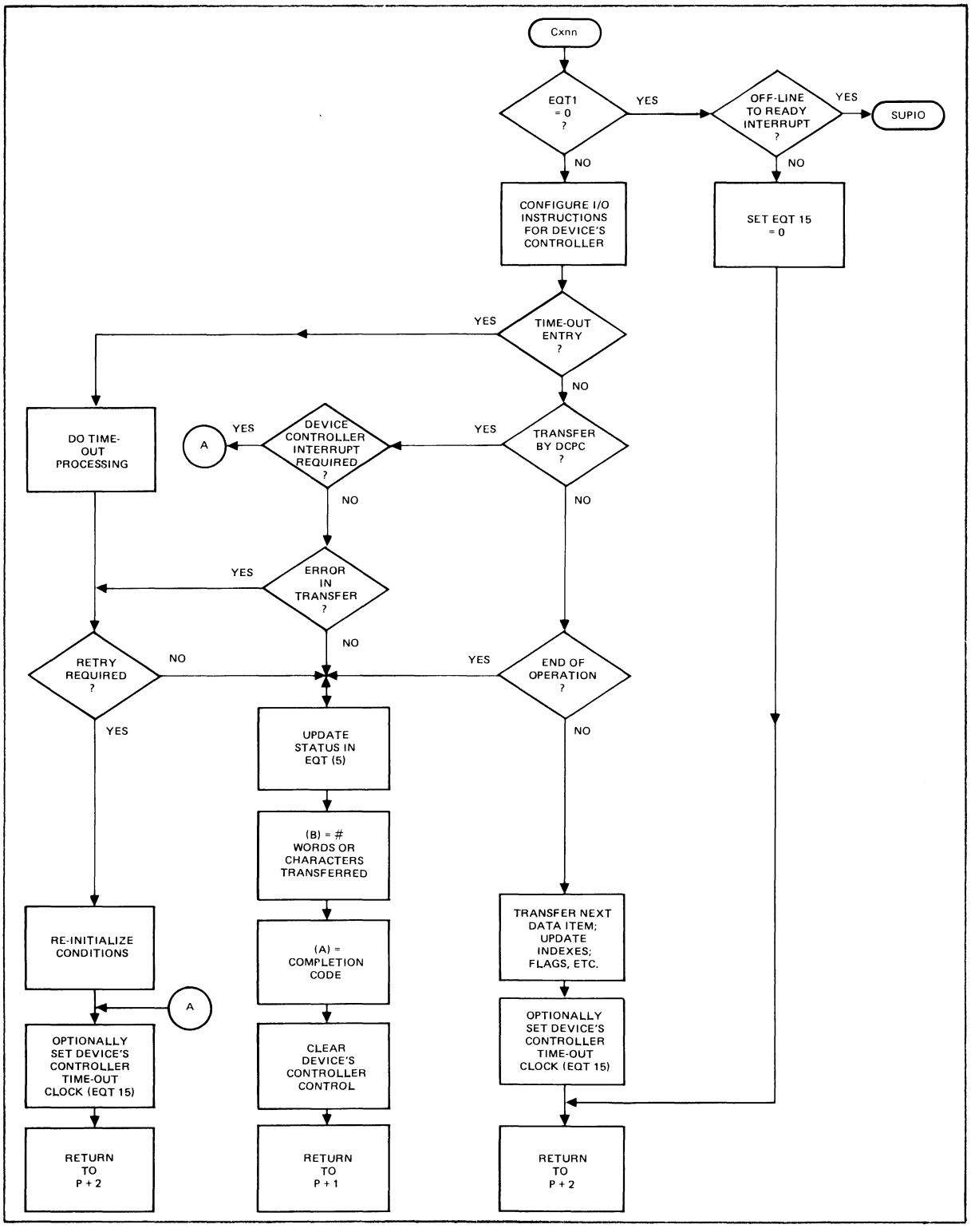
    1. Ignore the interrupt by setting EQT entry word 15 (the time-out clock) to zero to prevent time-out and making a continuation return as described in step "g" below.

    2. Ignore the interrupt and have RTE display a message by making a completion return as described in step "f" below. No special A-Register return value is needed. Upon return from the driver, RTE looks at the I/O request list pointer and, if the pointer is zero, displays the following message on the system console (where the "sc" indicates the interrupting select code):

        ILL INT sc

b. If the interrupt is valid (i.e., bits 14-0 of word 1 are non-zero), the driver configures all I/O instructions in the continuation/completion section to reference the interrupting select code.

c. Checks to see if a time-out has occurred on the device by checking bit 11 of EQT entry word 4. If this bit is set, the device has timed out, and any required time-out processing should be done. Note that this check need only be made by drivers that are designed to process time-out interrupts (as described in the "I/O Controller Device Time-Out" subsection of this manual). Drivers not processing time-out are not entered on device time-out.



7700-127

Figure 3-4. I/O Driver Continuation/Completion Section

d. If both the DCPC and the device controller interrupts are expected, but only the device controller interrupt is significant, the DCPC interrupt can be ignored by making a continuation return to CIC as described in step "g" below. (Refer to the "DCPC Processing" subsection of this manual for a method to suppress the DCPC interrupt entirely.)

e. Performs the input or output of the next data item if the device is driven under program control. One of three possible actions is then taken:

   1. If the transfer is not complete, the driver follows the procedure described in step "g" below to make a continuation return.

   2. If the transfer is complete, the driver follows the procedure described in step "f" below to make a completion return.

   3. If the driver detects a transmission error, it can reinitiate the transfer and attempt a retransmission. A counter for the number of retry attempts can be kept in the EQT entry. After initiating each retry, the driver makes a continuation return to CIC as described in step "g" below.

f. (Completion Return.) At the end of a successful transfer or after completing the retry procedure, the driver performs the following steps before returning to CIC:

   1. Sets the actual or simulated device controller status into bits 7 through 0 of EQT entry word 5 without altering the rest of word 5.

   2. Sets the number of words or characters (depending on which the user requested) transmitted into the B-Register. This value is known as the transmission log.

   3. Clears the device controller (and DCPC if used).

   4. Sets the A-Register to indicate successful completion and the reason as follows:

      If A = 0 The operation was successfully completed.

      If A = 1,2,3,4 The operation was not completed, where:
      1 = device or controller malfunction or not ready (IONR issued)
      2 = end-of-tape or end-of-information (IOET issued)
      3 = transmission parity error (IOPE issued)
      4 = device time-out (IOTO issued)

   5. Return to CIC at P+1 (JMP Cxnn,I).

g. (Continuation return.) When the driver wishes to continue the transfer (i.e., additional interrupts are expected), the driver performs the following steps before returning to CIC at P+2:

   1. Sets the device controller (and DCPC if used) for the next transfer or retry.

   2. Optionally sets the device time-out clock (EQT entry word 15) to modify the value inserted there by the system. (Refer to the "I/O Controller Time-Out" subsection of this manual.)

   3. Returns to CIC at P+2 or (for RTE-IV and RTE-6/VM) P+3 as follows:

```
ISZ Cxnn          or          ISZ Cxnn
JMP Cxnn,I                    ISZ Cxnn
                              JMP Cxnn,I
```

      for the P+3 return, the content of the A-Register determines the action taken by the system. The A-Register should be set as follows:

      A = 5 Request the allocation of a DCPC channel. After assigning a DCPC channel, the system will reenter the driver at Ixnn.

      A = 6 Request the deallocation of a DCPC channel. After releasing the DCPC channel, the system continues processing as if this were a P+2 continuation return.

Device Clear on Program Abort

If an I/O suspended program is aborted while waiting for a controller, the system attempts to clear the controller by issuing a clear control request to the driver (i.e., request code 3, subfunction code 00, as indicated in EQT entry word 6). All drivers written for use in RTE must be prepared to handle this request even if no other control requests are supported for the controller.

If the controller can be cleared without interrupt (i.e., immediately), the driver should perform the clear operation and return to IOC with the A-Register equal to 2 (control request illegal) or 4 (immediate completion). Either return is sufficient in this case; they are both treated equivalently by the system.

If an interrupt is required, the driver should return with the A-Register equal to 0. In this case, the system forces a one second time-out for the controller. When the device controller interrupts, the driver should complete the clear operation and make a successful completion return (A-Register = 0) to CIC at P+1. However, if the interrupt does not occur within the one second time-out, the system itself issues a clear control command (CLC) to the controller's select code(s). Note that in this case the driver is not entered to process the time-out even if it had previously set the "driver processes time-out" bit in EQT entry word 4. (Refer to the "I/O Controller Time-Out" subsection in this manual.)

I/O Controller Time-Out

Each I/O controller can have a time-out clock to prevent indefinite I/O suspension. Indefinite I/O suspension can occur when a program initiates I/O, and the device controller fails to return a flag indicating that the transfer is complete. This can occur as the result of either a hardware malfunction or improper program encoding. Without the controller time-out, the program making the I/O call would remain in I/O suspension indefinitely, awaiting the completion indication from the device controller.

EQT entry words 14 and 15 function as an I/O controller's time-out clock. EQT entry word 15 is the actual working clock. Prior to each call to the driver, word 15 is set to a value "m," where "m" is a negative number of ten millisecond time intervals. Thereafter, this counter is incremented by one at each ten millisecond tick of the system's real-time clock. If the controller does not interrupt within the required time interval (i.e., before the counter in EQT entry word 15 goes to 0), it is considered as having "timed out."

The EQT entry word 15 clock for each controller can be individually set by either of the following two methods:

a. The system always inserts the contents of EQT entry word 14 (a negative number) into EQT entry word 15 before the initiation or continuation/completion section of a driver is entered, unless it is running (EQT word 15 is not zero). Word 14 can be preset to "m" by entering a time-out value during the EQT entry phase of generation, or it can be set or modified on-line with the TO operator request

b. When the driver initiates I/O and expects to be entered due to a subsequent interrupt, the driver itself can set the value "m" into EQT entry word 15 just before it exits. The value "m" can either be coded permanently into the driver or can be passed to the driver as an I/O request parameter.

NOTE

> The system always inserts the contents of EQT entry word 14 into EQT entry word 15 before entering a driver, with the following exceptions: 1) if an initiation call is being made and word 15 is already non-zero, it is not reset, and 2) if a continuation call is being made and word 14 is zero, word 15 is not reset. In any case, a time-out value inserted by the driver directly into word 15 overrides any value previously set by the system.

A time-out value of zero is equivalent to not using the time-out feature for a particular controller. If a time-out parameter is not entered, its value remains zero and time-out is disabled for the controller.

Time-out is enabled for a controller only while the controller is processing an I/O request. The working time-out clock (EQT entry word 15) is set as described above. If the controller does not time out, the clock is then cleared by the system after one of these driver returns:

1. An Initiation return indicating a rejected initiation request (A-Register not equal to zero).

2. A Completion return.

Driver Processing of Time-Out

When a controller times out, a driver has the option of performing its own time-out processing, or of letting the system handle it entirely. A driver that processes its own time-outs indicates this to the system by setting bit 12 of EQT entry word 4. Since the system never clears this bit, it needs to be set only once. When bit 12 is set, the following action takes places upon controller time-out:

a. Bit 1 in EQT entry word 4 is set by the system.

b. The driver is entered at Cxnn with the A-Register set to the I/O select code of the controller that timed out. (The same information is available in EQT entry word 4.)

c. The driver recognizes that the entry is for time-out by examining bit 11 of EQT entry word 4. When bit 11 is set, a time-out has occurred, and the driver should perform whatever processing is necessary. This can involve completing the operation in progress or restarting the device and continuing the operation. If the latter option is taken, the driver should clear bit 11 prior to exiting in case it is entered again prior to completion of the operation. This enables the driver to distinguish between a normal continuation entry (bit 11 = 0) and another time-out entry (bit 11 = 1). Note that IOC only clears bit 11 prior to entering the driver at Ixnn on an initiation call.

d. The driver may decide to continue (i.e., restart the device) or complete i.e., terminate) the operation, as follows:

   1. If the driver decides to complete the operation, it sets the A-Register equal to 4 (to indicate that a time-out has occurred), sets the B-Register equal to the transmission log, and returns to CIC at P+1. Ths causes CIC to set the LU down and to print the following message:

      I/O TO L #x E #y S #z

      where:

      #x is the LU number being set down,

      #y is the number of the EQT entry associated with this LU, and

      #z is the channel associated with this LU.

      It is possible to complete the operation without having a message printed. To do this, the driver simply makes a normal completion return (A-Register = 0, B-Register = transmission log) to CIC at P+1.

      In either case (A = 4 or A = 0), CIC reschedules the calling program and passes it the transmission log returned by the driver.

   2. If the driver decides to continue the operation, it makes a normal continuation return to CIC at P+2.

System Processing of Time-Out

When a time-out occurs and bit 12 of EQT entry word 4 is not set, the system handles the interrupt itself in the following way:

a. The program that made the initial I/O request is rescheduled and a zero transmission log is returned to it.

b. The LU is set down and the following message is printed:

   I/O TO L #x E #y S #z

where:

#x is the LU being set down,

#y is the number of the EQT entry associated with this LU, and

#z is the subchannel associated with this LU.

c. A clear control instruction is issued to the controller's select code(s) through the EQT entry number located in the Interrupt Table.

Note that the driver is never entered for time-out processing when bit 12 of EQT entry word 4 is zero. This means that only those drivers that set bit 12 to indicate that they are to process time-out need to check for a time-out entry.

Since the system issues a CLC instruction to the controller's select code(s), each controller interface card requires an entry in the Interrupt Table during generation. Otherwise, the system would not be able to issue the CLC instruction when a controller timed out.

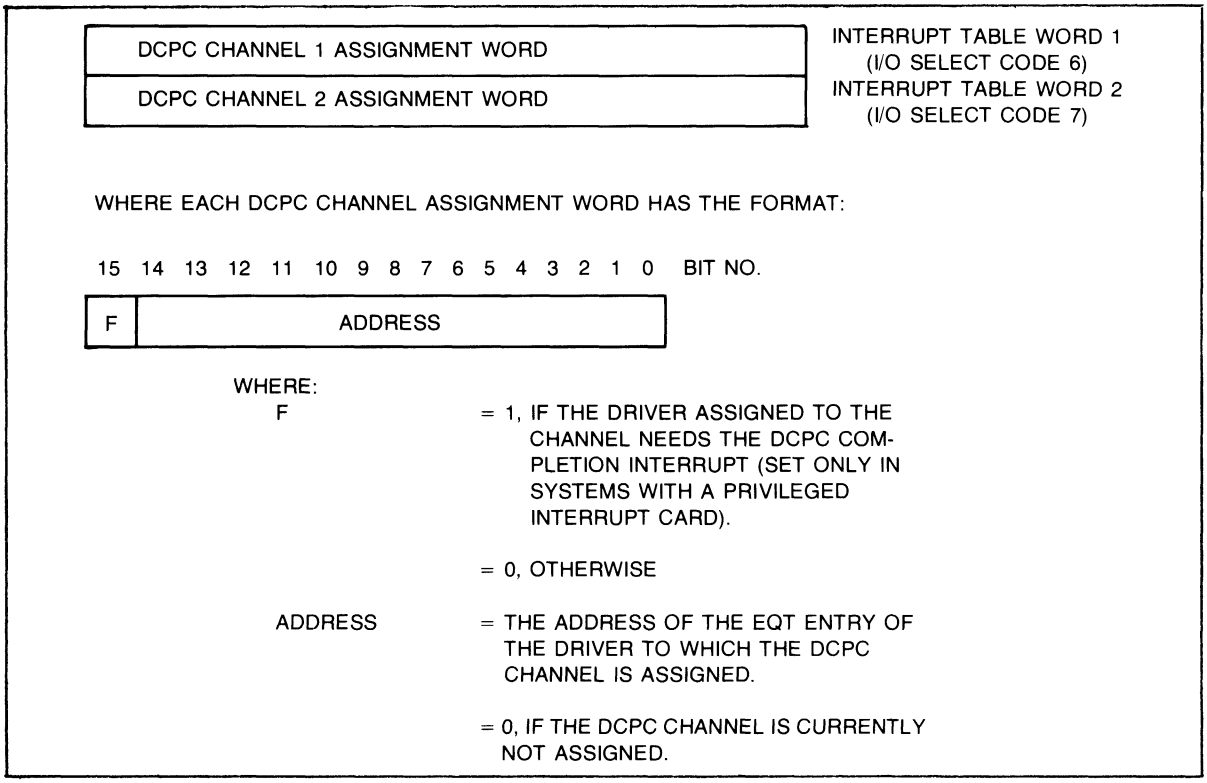
DCPC Processing

The Dual Channel Port Controller (DCPC) feature of the HP 1000 M/E/F series of computers can be used to transfer blocks of data between I/O devices and the computer at high data transfer rates. The DCPC transfers are initiated in software, but the actual word-by-word transfer is handled under hardware control. Words are transferred to or from the computer via a "cycle stealing" technique that operates concurrently with the normal execution of programs. This design eliminates the overhead associated with driver processing of individual interrupts and allows synchronous and high-speed devices to be controlled by standard RTE drivers.

This subsection discusses some of the aspects of DCPC transfers in the RTE operating systems. It is assumed that the reader is already familiar with the general techniques of DCPC programming as described in the appropriate computer reference manual.

RTE Control of DCPC Assignment

RTE controls the allocation of the two DCPC channels available via the first two words of the Interrupt Table. Interrupt Table entry word 1 records the current assignment of DCPC channel 1, and word 2 records the current assignment of DCPC channel 2. This arrangement is illustrated in Figure 3-5. This figure also illustrates the format of each individual DCPC Assignment Word. (Note that DCPC channels 1 and 2 generate interrupts on I/O select codes 6 and 7, respectively, and hence are often referred to as DCPC channels 6 and 7.)



8300-93

Figure 3-5. DCPC Channel Assignment Words

Each DCPC Channel Assignment Word in the Interrupt Table can be in one of two states; assigned or unassigned. If the entire word is zero, the respective DCPC channel is unassigned, and is therefore available for use. A non-zero word implies that the DCPC channel is currently assigned. Bits 0 through 14 contain the address of the EQT entry (which in turn points to the driver) to which the DCPC channel has been allocated. Once a DCPC channel is allocated, a driver can set bit 15 in the appropriate DCPC Channel Assignment Word as a flag to the operating system. Use of this flag is fully explained later in this subsection.

DCPC Assignment by RTE

Before a driver can initiate a DCPC transfer, it must be assigned by RTE the exclusive use of a DCPC channel. This prevents simultaneous access to the channel by several drivers. A driver can be assigned a DCPC channel in the following two ways:

Preferred Method

If the driver's EQT entry had a "D" specified at generation time, the "DCPC channel required" bit is permanently set in the EQT entry (EQT entry word 4, bit 15). In this case, the system always assigns a DCPC channel to the driver at each I/O initiation. The assigned DCPC channel number can be found (at initiation only) in the Base Page Communications Area word CHAN. It should then be saved in one of the temporary storage words of the EQT entry since it is not available in CHAN on later entries to the driver for the same I/O request.

Alternate Methods

If a driver needs a DCPC channel only for a certain subset of the functions that it performs, it can dynamically ask the system to assign it a DCPC channel as required. In this case, the DCPC option is not selected for the driver's EQT entry at generation time, and hence the "DCPC channel required" bit is not set in the EQT entry.

A driver can dynamically request a DCPC channel from either its Initiation or (RTE-IV and RTE-6/VM) Continuation/Completion sections. Each method is described below.

Alternate Method I - Initiation Request

The driver will be entered at Ixnn without a DCPC channel assigned by IOC. The driver must analyze the request and determine if a DCPC channel is required. If so, the driver requests a DCPC channel from IOC by returning via a jump indirect through Ixnn (JMP Ixnn,I) with the A-Register equal to 5. IOC then assigns a DCPC channel and recalls the driver.

However, IOC does not differentiate between the initial call to the driver and the recall with the DCPC channel assigned. The EQT entry is set up identically in both cases, and the driver is entered at Ixnn. Furthermore, it is possible that the driver may never be recalled with the DCPC channel assigned for a particular I/O request. For example, this can occur if the program making the request is aborted before IOC has a chance to assign the DCPC channel. If the program is aborted, the driver will not be entered again until another program requests I/O for a device under the driver's control.

Thus, a driver can never know from the calling parameters or from its past history whether it is being called for the first time for an I/O request (i.e., no DCPC channel is assigned) or whether it is being recalled with a DCPC channel now assigned. The only way the driver can distinguish between these two cases is to access the two DCPC Channel Assignment Words (in the Interrupt Table) to determine whether a DCPC channel is currenty assigned to the driver.

If the value in either DCPC Channel Assignment Word is equal to the address of the EQT entry currently being serviced by the driver, that DCPC channel is currently assigned to the driver. The driver can then assume that it has been recalled with a DCPC channel assigned, and can initiate the transfer on that DCPC channel. If neither value matches, no DCPC channel is assigned to the driver, and it must return to IOC to request that one be assigned. (Note that in this case the driver cannot use Base Page Communications Area word CHAN as an indication of whether or not it has been assigned a DCPC channel. This is because CHAN indicates only which channel was last assigned to any driver; not to whom it was assigned.)

The following code illustrates the DCPC assignment check technique. In reviewing this and subsequent examples, remember that the Base Page Communications Area word INTBA contains the address of the Interrupt Table, and that base page word EQT1 contains the address of the EQT entry currently being serviced by the driver.

```
CHDCP EQU *          Execute this code if DCPC required
      DLD INTBA,I    Access DCPC Channel Assignment Words
      CPA EQT1       Is DCPC channel 1 assigned to this driver?
      JMP CH1        Yes, configure and initiate transfer on channel 1
      CPB EQT1       Is DCPC channel 2 assigned assigned to this driver?
      JMP CH2        Yes, configure and initiate transfer on channel 2
      LDA =B5        No. A DCPC channel is not assigned. Set
      JMP Isnn,I     A = 5 to request one from IOC, and return
```

Note that if a driver obtains a DCPC channel in this way, a special procedure must also be followed to return the DCPC channel to RTE. The return procedure is discussed in the "Returning DCPC Channels to RTE" subsection.

Alternate Method II - Continuation Request

Alternately, in RTE-IV and RTE-6/VM, the driver can request a DCPC channel from IOC by returning via a jump indirect through Cxnn with the A-Register equal to 5 as follows:

```
(Set A-Register to 5
ISZ Cxnn
ISZ Cxnn
JMP Cxnn,I
```

IOC then assigns a DCPC channel and recalls the driver, entering at Ixnn. In this case, IOC does not reset the EQT before entering at Ixnn.

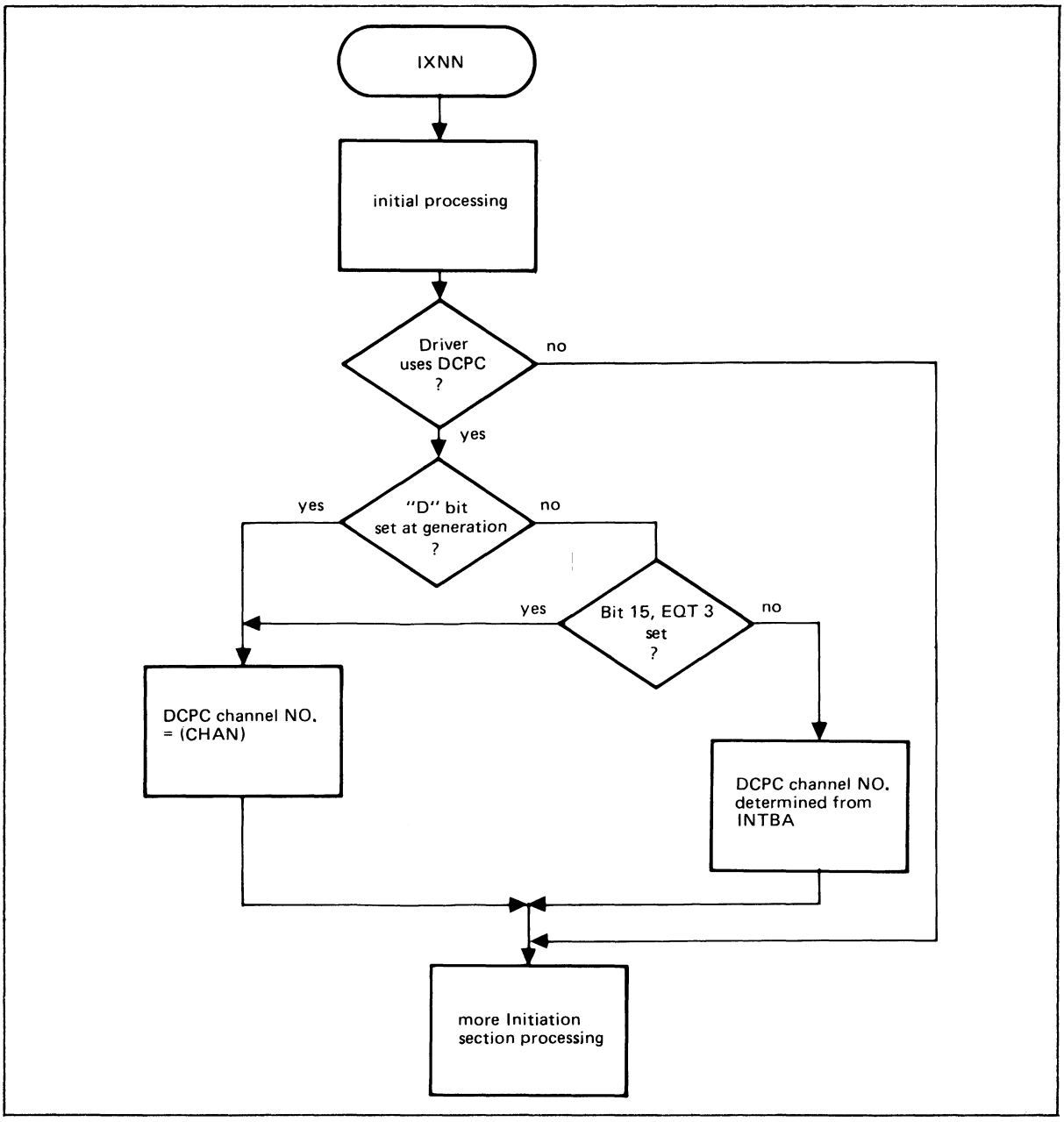
The Initiation section of the driver can determine if this method is being used to acquire a DCPC channel by examining bit 15 of word 3 in the EQT. If this bit is set, then the Initiation section was entered as a result of a DCPC request from the driver's Continuation section. The DCPC channel assigned is available in Base Page Communications Area word CHAN.

Note that a DCPC channel obtained this way must be returned to RTE by a special procedure. The return procedure is discussed in the "Returning DCPC Channels to RTE" subsection.

Determining the DCPC Assignment Method

These alternate methods of obtaining a DCPC channel are more complex than the preferred method and should only be used by drivers that: 1) process a mixture of DCPC and non-DCPC operations, and 2) cannot afford to tie up a DCPC channel during the non-DCPC operations.

It should also be noted that the Initiation section of the driver may have to use different methods to determine if a DCPC channel was assigned by IOC. Figure 3-6 summarizes these methods.



8300-226

Figure 3-6. Determining DCPC Assignment

Regardless of the method used to obtain a DCPC channel, RTE records the assignment by putting the address of the EQT entry being assigned the DCPC channel into the appropriate DCPC Channel Assignment Word in the Interrupt Table.

If a DCPC channel is not available, the requesting EQT is set into a "waiting for DCPC" state. As soon as another driver releases a DCPC channel, the lowest-numbered EQT waiting for DCPC is assigned the DCPC channel, and its I/O request is initiated.

Returning DCPC Channels to RTE

As soon as a driver completes a DCPC transfer and the associated I/O request, the DCPC channel must be returned to the available pool so that it can be used by other drivers. This occurs in two different ways, depending on how the DCPC channel was assigned:

1. If the DCPC channel was assigned automatically (Preferred Method discussed above), the DCPC channel is returned automatically by RTE when the driver makes its completion return to CIC. No special driver processing is required.

2. If the DCPC channel was assigned as the result of a specific request by the driver (Alternate Method discussed above), the driver must explicitly inform RTE of this fact when the I/O request is completed. This is done by setting the sign bit in the A-Register on the completion return to CIC. This bit may be set at all times - even when the driver has not been assigned a DCPC channel. However, some extra system overhead is created if the sign bit is set when not required. Note that the sign bit is set in addition to the normal completion code, as illustrated below:

```
LDA COMCD      Set A = completion code determined earlier
IOR =B100000   Set sign bit to indicate dynamic DCPC assignment
JMB Cxnn,I     Return to CIC
```

In either of the above cases, RTE implements the return of the DCPC channel by clearing the appropriate DCPC Channel Assignment Word in the Interrupt Table. DCPC Channel 1 Assignment Word (Interrupt Table word 1) is cleared if DCPC channel 1 was assigned to the driver; DCPC Channel 2 Assignment Word (Interrupt Table word 2) is cleared if DCPC channel 2 was assigned to the driver. No action is taken if a DCPC channel was not assigned to the driver.

In RTE-IV and RTE-6/VM, a driver may also return a DCPC channel assigned to it through an Initiation or a Continuation return. In each case, the A-Register having value 6 selects this deallocate DCPC option. IOC will deallocate the DCPC channel, if one was indeed allocated, and then proceed as for a normal Initiation or Continuation return. In addition, if a DCPC channel is deallocated from a Continuation return, IOC will check to see if another EQT is waiting for a DCPC channel. If so, IOC assigns the DCPC channel to that EQT and initiates its I/O request, thus increasing DCPC utilization.

Handling the DCPC Interrupt

An end-of-operation interrupt is generated by the DCPC hardware when a DCPC transfer is complete. Depending upon the nature of the device under control, the associated driver may or may not wish to recognize the DCPC interrupt.

If the driver does not require or use the DCPC completion interrupt, it can be disabled by issuing a clear control instruction (CLC) to the DCPC select code (6 or 7) after initializing the DCPC transfer. No further special processing is necessary.

If the driver uses the DCPC completion interrupt, some special processing must be included in the driver to ensure that the completion interrupt occurs only at the correct time in systems using a Privileged Interrupt card. (These systems are described more fully in Chapter 4.)

The following potential problem exists: In systems using a Privileged Interrupt card, the interrupt system is always ON, even when a driver is executing. It is therefore possible that a driver using DCPC could start the DCPC transfer and be interrupted by its own DCPC completion interrupt before it had a chance to complete the initiation procedure and return to IOC.

To eliminate this problem, a scheme has been designed to hold off the DCPC completion interrupt until the standard driver using DCPC completes the initiation procedure and returns to IOC. This scheme requires the cooperation of the standard driver utilizing DCPC and of both RTE and any privileged drivers present in the system, as follows: After disabling the interrupt system and initializing the DCPC transfer, the standard driver clears control on the DCPC select code (6 or 7) to inhibit the completion interrupt while the standard driver completes the initiation procedure. The standard driver also sets a flag (F in the interrupt table, Figure 3-5) to inform RTE that the standard driver actually needs the interrupt, and that RTE should reenable the interrupt later, after the driver returns to IOC.

The flag is also used by privileged drivers. A privileged driver disables the DCPC completion interrupts upon entry so that the privileged driver will not be interrupted while processing the privileged interrupt. A privileged driver will reenable a DCPC completion interrupt before exiting only if it is needed by a standard driver (as indicated by the flag being set).

Bit 15 of each DCPC Channel Assignment word in the Interrupt Table is used as the flag for the respective DCPC channel. If this flag is set, RTE and the privileged drivers will enable the DCPC interrupt on the correct DCPC channel at the appropriate time. No action is taken if the flag is not set.

The section of code listed below illustrates the special processing required when a standard driver uses the DCPC completion interrupt. Note that although this processing must be included in all drivers using DCPC, it need only be executed when the driver is operating in a privileged system. The type of system in which a driver is operating can be determined by examining base page word DUMMY. If DUMMY is 0, the system is non-privileged (i.e., no Privileged Interrupt card is present); otherwise the system is privileged (i.e., a Privileged Interrupt card is present).

```
        CLF 0           Disable the interrupt system

        STC DCPC,C      Initiate transfer on DCPC channel

        CLA             Bypass section below if
        CPA DUMMY       DUMMY = 0 (non-privilged system)
        JMP X           and special processing not needed.

        CLC DCPC        Clear DCPC control to inhibit DCPC
        LDB INTBA       interrupt.  Set B = address of the appropriate
        LDA CHAN        DCPC Channel Assignment word in the
        CPA =D7         Interrupt Table.

        LDA B,I         Set bit 15 of DCPC channel assignment entry
        IOR =B100000    equal to 1 as flag to system to turn  DCPC
        STA B,I         interrupts back on later.  Reenable the
        STF 0           interrupt system.

X       EQU *           Continue processing.
```

Intermixed DCPC and Non-DCPC Operations

Occasionally a driver may have a special requirement to intermix a series of non-DCPC operations with DCPC operations during the same I/O request. If it is necessary or desirable to retain assignment of the DCPC channel throughout the non-DCPC operation, the following special assignment word should be cleared prior to beginning the non-DCPC operations. This prevents the system from reenabling the DCPC completion interrupt when it is not desired. Note that this processing need only be done if the flag was previously set under the conditions discussed in the preceeding paragraphs.

Driver Automatic "Up"

A driver has the capability of automatically returning its EQT entry and all associated LUs to the "up" state through a JMP instruction. For example, if a driver makes a not ready, parity error, end-of-information, or time-out return to the system, the system sets the associated LU and EQT entry into the "down" state. If the driver subsequently detects an interrupt (or time-out) entry that signals that the controller is now ready, it may return the EQT entry and associated LUs to the "up" state as follows:

```
    JMP $UPIO
```

The device controller's EQT entry and all associated LUs are reset to the up (available) state by $UPIO. If an I/O request is pending, $UPIO restarts the request by entering the driver at the initiation entry point Ixnn. If there are no requests pending, $UPIO goes to the dispatcher to start the next program.

Power Fail Processing

When an RTE system is generated, the user has the option of including DVP43, the power-fail/automatic restart driver, and AUTOR, the automatic restart program. If DVP43 is not included, the system executes a HALT 4 when power is restored to the computer.

If the power-fail/automatic restart modules are included in the generation, they enable the system to recover automatically from a power failure. Power-fail/automatic restart processing can be divided into three parts:

a. The power down sequence.

b. The power up sequence.

c. The sequence required to restart any I/O transfers that were in progress when the power-fail occurred. (A driver has the option of restarting its own I/O, or of letting the system restart it from the beginning of the request. These two alternatives are discussed below.)

Power Down Sequence

When a power-fail occurs, a power-fail interrupt is generated and DVP43 is entered to process the interrupt. In the brief period of time available before the system becomes completely inoperable, DVP43 performs the following steps to save the state of the machine:

a. Stops all DCPC transfers.

b. Saves all user accessible registers (A,B,E,O...).

c. Saves the status of the memory protect fence. Also saves the status of the Dynamic Mapping System (DMS) in systems which include the DMS feature.

d. Saves all map registers (System Map, User Map, and the two DCPC maps). This step is performed in systems with Dynamic Mapping only.

DVP43 then executes a HALT 4 instruction before power fails completely.

Power Up Sequence

When power is restored to the computer, an interrupt is generated an DVP43 is reentered to process the interrupt. DVP43 performs the following steps to restart the system:

a. Sets a software flag to prevent resaving the state of the machine if a subsequent power failure occurs before the system is completely restored.

b. Reenables the power-fail hardware.

c. Restores the state of the memory protect fence. Also restores all map registers and the status of the Dynamic Mapping System in systems which include the DMS feature.

d. Saves the time of the power-fail.

e. Finds the power-fail EQT entry (i.e., the EQT entry associated with DVP43) and sets up a very short time out on this EQT entry by setting EQT entry word 15 (the time-out clock) to -1. This causes DVP43 to be reentered after one tick of the real-time clock. DVP43 can then begin to restart any I/O transfers that were in progress at the time of the power failure.

f. Restarts the real-time clock.

g. Restores all user-accessible registers.

h. Clears the software flag that was set in step "a", so that the state of the machine will be saved as usual on any subsequent power failures.

i. Returns to the suspended process (i.e., the process that was in operation when the power-fail occurred) at the point of interruption.

Restart I/O Sequence

As soon as the power-fail EQT entry times out, DVP43 is entered again since it previously set the "driver processes time-out" bit. DVP43 now attempts to restart any I/O transfers that were in progress at the time of the power fail by performing the following steps:

a. Makes the following checks for each I/O controller:

   1. Checks bits 14 and 15 of EQT entry word 5. The value of bits 14 and 15 indicate whether the I/O controller was "down" or "busy" at the time of the power failure.

   2. Checks bit 13 of EQT entry word 4 to see if the driver associated with the EQT entry is prepared to process a power-fail/automatic restart entry. Drivers that are prepared to process power fail/automatic restart entries will have previously set bit 13 to one. Otherwise this bit is zero. (Note that the system never clears bit 13, so a driver only needs to set it once.)

   3. Checks to see if any EQT entries are currently waiting for a DCPC channel.

b. Depending upon the above information, one of the following three actions is taken for each controller or device in the system:

   Case 1. Controller (EQT entry) busy and "driver processes time-out" bit set:

   If the controller was reading or writing data when the power fail occurred and the driver is designed to handle power-fail, the driver has the responsibility to recover from the power-fail in the best possible manner. The system simply sets bit 15 of EQT entry word 5 to 1 indicate that a power-fail has occurred, and enters the driver at the initiation entry point Ixnn.

Case 2. Controller waiting for a DCPC channel.

If the controller was waiting for a DCPC channel when the power failure occurred, no action is taken. The I/O transfer will be initiated as usual when a DCPC channel is released by another driver.

Case 3. All other EQT entries.

For all EQT entries not falling under Case 1 or Case 2 above, DVP43 makes a call to $UPIO to up the EQT entry and all associated LUs. (See the "Driver Automatic Up" subsection of this manual.) $UPIO restarts any I/O requests that were in progress (i.e., EQT entry was busy) or pending (i.e., EQT entry or LU was down) at the time of the power failure. This is done by resetting the parameters of the original call into the EQT entry and reentering the driver at the initiation point Ixnn.

After the above action is taken for each I/O controller in the system, an HP-supplied program call AUTOR auto-restart is scheduled. AUTOR sends the time of power failure to all user consoles on the system (thereby reenabling all terminals). Note that the request sent to user consoles (drivers 00 and 05) by AUTOR specifies number of characters to be output, not number of words.

AUTOR is written in FORTRAN, with the source tape supplied so that it can be easily modified for site-specific applications.

Program Scheduling by Drivers

Occasionally some I/O applications may require that a driver schedule a program to perform a certain task. The system list processor, $LIST, has several calls available that provide drivers with this capability. These calls are illustrated below. All of these calls cause a program to be scheduled. They differ only in the format of the calling sequence, and in the type of information that each call may specify is to be stored in the ID segment of the program to scheduled.

Method 1. (Used to put five parameters in the ID segment and then schedule the program.)

```
          EXT $LIST
             .
             .
             .
          JSB $LIST
          OCT 701
          DEF RTN      Return point. (Must be immediately after the call)
          DEF PNAME    Address of three word array containing program name
          DEF P1       Addresses of up to five optional parameters to be
          DEF P2       placed in program's ID segment.
          DEF P3
          DEF P4
          DEF P5
RTN          .         Return point. Must be located immediately after
                       call.
             .         (See below for error status return in A &
                       B-Registers)
             .
             .
PNAME     ASC 3,XXXXX Name of program to be scheduled
P1        OCT A       Up to five optional parameters to be placed in
P2        OCT B       program's ID segment (temporary storage area)
P3        OCT C       prior to scheduling it.
P4        OCT D
P5        OCT E
```

This call causes the system to place whatever number (1-5) of optional parameters are supplied into the temporary storage area of the ID segment of the program whose ASCII name is contained in the variable PNAME. The system then schedules the program to run at its own priority.

CAUTION

You must pass at least one parameter (P1). The driver will fail if it calls $LIST with no parameters.

Method 2. (Same as Method 1, except that the ID segment address rather than the program name is supplied. Note that in some RTE systems a driver may not be able to search the ID Segment Table for a program's ID segment address. It is therefore recommended that drivers scheduling programs do so by specifying the program's name (function code 701 call to $LIST), rather than the ID segment address.)

```
          EXT $LIST
             .
             .
             .
          JSB $LIST
          OCT 001
          DEF RTN       Return point. (Must be immediately after call.)
          OCT IDADR     ID segment address of program to be scheduled.
          DEF P1        Address of up to five optional parameters to be
          DEF P2        placed in program's ID segment.
          DEF P3
          DEF P4
          DEF P5
RTN          .          Return point.  Must be located immediately after
                        call.
             .          (See below for error status return in A &
                        B-Registers)
             .
             .
P1        OCT A         Up to five optional parameters to placed in
P2        OCT B         program's ID segment (temporary storage area)
P3        OCT C         prior to scheduling it.
P4        OCT D
P5        OCT E
```

This call causes the system to place whatever number (1-5) of optional parameters are supplied into the temporary storage area of the ID segment specified by IDADR. The system then schedules the program associated with the ID segment to run at its own priority.

Method 3. (Not for use in RTE-6/VM; otherwise, used to put a value into the "B-Register at suspension" word in the ID segment and then schedule the program. This call can be used to set the B-Register to point to a scheduling parameter storage area. The scheduled program can then recover the parameters via a call to subroutine RMPAR. The driver should make sure that the parameters are placed in a memory area that is mapped with the user program.)

```
          EXT $LIST
             .
             .
             .
          JSB $LIST
          OCT 601
          OCT IDADR     ID segment address of program to be scheduled.
          OCT BVAL      Value to be put into "B-Register at suspension"
                        word
```

```
RTN     .       Return is always made to here by $LIST
        .       (See below for error status return in A &
                B-Registers)
        .
```

This call causes the system to place the value BVAL into the "B-Register at suspension" word of the ID segment specified by IDADR. If this value is an address that points to a set of scheduling parameters, the program can recover the parameters by making a call to subroutine RMPAR. The system then schedules the program associated with the ID segment to run at its own priority.

Error Conditions:

When $LIST returns from any of the program schedule calls described above, the A and B-Registers indicate whether or not the program was successfully scheduled, as follows:

If A = 0	The program was successfully scheduled. (The B-Register contains the ID segment address of the scheduled program.)
If A is non-zero	The program could not be scheduled. The B-Registerindicates the reason, as follows:
	B = 3 Illegal status (program not dormant)
	B = 5 No such program.

Determination of Operating System Environment

There are times when it may be necessary for a driver to know the operating system within which it is executing. The system entry point $OPSY provides this and other information in the form of one-word table, as illustrated in Table 3-1.

Table 3-1. $OPSY Word Format

System	$OPSY Value	Bit 15 1=RTE	Bit 3 Type	Bit 2 0=Memory Based 1=Disc Based	Bit 1 0=No DMS 1=DMS	Bit 0 0= 64 Word Disc 1=128 Word Disc
RTE-M/I	-7	1	1	0	0	1
RTE-M/II	-15	1	0	0	0	1
RTE-M/III	-5	1	1	0	1	1
RTE-II	-3	1	1	1	0	1
RTE-III	-1	1	1	1	1	1
RTE-IV	-9	1	0	1	1	1
RTE-IVE	-13	1	0	0	1	1
RTE-6/VM	-17	1	1	1	1	1

$OPSY can be referenced simply by loading it into a register and testing the appropriate bits. This technique is illustrated below:

```
EXT $OPSY
   .
   .
   .
LDA $OPSY      Access $OPSY information
AND MASK       Isolate appropriate bits
   .
   .           Take appropriate action
   .
```

In Dynamic Mapping System ($OPSY bit 1 = 1), it may also be necessary to determine whether the System Map or User Map is currently enabled. This can be done in a driver by accessing the status of the Dynamic Mapping System via an RSA instruction, and looking at bit 12. Bit 12 is 0 if the System Map is enabled, and 1 if the User Map is enabled. The following code illustrates this procedure.

```
RSA            Access Dynamic Mapping System Status
ALF            Position Bit 12 into Bit 0
SLA            System Map enabled?
JMP USER       No, User Map Enabled.
JMP SYSTM      Yes, System Map Enabled.
```

Subroutines for Special Mapping Functions (DMS Systems Only)

By using the Dynamic Mapping System (DMS) feature of the HP 1000 M/E/F series of computers, RTE provides the capability for addressing memory configurations larger than 32K words. This is accomplished by translating memory addresses through one of four "memory maps." A memory map consists of a set of hardware registers. These registers provide the interface between memory addresses used by programs (logical memory addresses) and actual memory addresses (physical memory addresses). There are four distinct maps: the System Map, the User Map, and the two DCPC maps. The DCPC Maps are loaded (i.e., set up) by the system as necessary to describe the logical memory configuration required by the currently executing program or DCPC transfer.

Prior to entering a driver to process an I/O request, RTE loads and enables the correct map (System or User) needed to describe the buffer of the request. The driver can access the buffer through the request buffer address word in the EQT without having to consider which map contains the buffer.

Certain drivers may, however, need to access a buffer in addition to the primary request buffer. If a driver is entered with the System Map enabled (e.g., when the primary request buffer is in System Available Memory or System Common), it must access the second buffer through the User Map if the second buffer is located in the program making the I/O request. In addition, the driver must load the User Map since the current contents of the map may or may not describe the desired program.

Subroutine $XDMP can be called by standard drivers to reload the User Map to describe the program containing the second buffer. The driver uses $XDMP as follows:

1. Save the contents of the current User Map.

2. Load the A-Register with the ID Segment address of the program containing the second buffer.

3. Call $XDMP (JSB $XDMP) to load the User Map for the program specified in the A-Register.

4. Check for an error return. $XDMP returns with the A-Register set to zero if the specified program was not in memory and the User Map could not be set up.

5. If no error was detected, access the second buffer through the newly loaded User Map.

6. After all accesses to the second buffer have been made, restore the User Map to its original contents and continue with normal operation.

Notice that the use of $XDMP requires that the driver know the ID Segment address of the program making the I/O request. Since the driver can not be sure that this ID Segment address is available on base page or in the EQT, the program making the I/O request should cooperate with the driver and pass its ID Segment address. This could be done by using one of the optional parameters and letting the driver retrieve the address through the appropriate optional parameter word in the EQT or by including the address in one of the words of the primary request buffer.

The recommended procedure for using $XDMP and accessing the second buffer is somewhat different in RTE-III, RTE-IV and RTE-6/VM. Each of these procedures is described in a separate section below.

> Subroutine $XDMP can be called by standard RTE drivers only. Privileged drivers wishing to perform the same function must use subroutine $PVMP, which is described in Chapter 4 of this manual.

Mapping in RTE-III and RTE-M/III

Any standard driver operating in RTE-III may use subroutine $XDMP to perform the memory map switching discussed above. The driver first saves the current state of the User Map registers and the Dynamic Mapping System, and then calls $XDMP to reload the User Map registers to describe the user program. The driver can then enable the User Map and access the memory described by it. After all accesses have been made, the driver reenables the System Map and restores the original state of the User Map registers before continuing with its normal processing under the System Map.

Note that subroutine $XDMP need only be used to reload the User Map registers when the driver is entered with the System Map enabled. The calling sequence is as follows:

```
EXT $XDMP
   .
   .

RSA          Get Dynamic Mapping System (DMS) status
ALF          Position bit 12 into bit 0
SLA          Is System Map Enabled?
JMP USER     No, so do not need to execute code below
   .
   .         Normal driver processing under System Map
   .
RSA          Get Dynamic Mapping System (DMS) status
```

```
        RAL,RAL         Position current status in upper bits
        STA DMSST       Save status for later.

        LDA MAPAD       Set A = address of User Map storage area
        IOR SIGN        Set sign bit indicating STORE Map in memory
        USA             Save current User Map in memory for later.

        LDA IDADR       Get ID address of program that contains buffer
        JSB $XDMP       Call $XDMP to set up User Map for this program
        SZA,RSS         Check for error return
        JMP ERROR       Error exists, go handle it

        UJP CONT        No errors.  Enable new User Map and continue

CONT      .
          .             Process buffer under new User Map
          .

        SJP NEXT        Reenable System Map
NEXT    LDA MAPAD       Access address of User Map storage area
        USA             Restore original contents of User Map
        JRS DMSST NXT   Restore original DMS status (i.e., System Map)

NXT       .
          .             Proceed with normal processing under System Map
          .
MAPAD   DEF MAP         Address of User Map storage area
MAP     BSS 32          User Map storage area
SIGN    OCT 100000
IDADR   BSS 1           Storage for ID segment address
DMSST   BSS 1           Temporary storage for DMS status
```

Remember that any driver using $XDMP must save the original contents of the User Map before calling $XDMP. It must also restore the original User Map before continuing with its normal operation under the System Map. Also remember that if the driver accesses the memory described by the User Map by enabling the map, it must save and restore the DMS status in addition to the original contents of the User Map. The example above illustrates this procedure.

NOTE

> The driver could also access the buffer in the user program through a series pf cross-map loads and stores without actually enabling the User Map. This is in fact the recommended procedure for using $XDMP in an RTE-IV or RTE-6/VM system, and use of it by a driver would allow the same driver to be used in either type of system.

Mapping in RTE-IV and RTE-6/VM

Drivers in RTE-IV and RTE-6/VM may be located in one of the Driver partitions or in the System Driver Area of memory. If a driver resides in a driver partition, RTE automatically includes the correct driver partition as it loads and enables the correct map (System or User) prior to entering the driver. If one of these drivers is entered under the System Map and needs to access a second buffer in the program which made the I/O request, the driver can use the $XDMP subroutine as described in the previous section.

The other location for a driver in RTE-IV and RTE-6/VM is the System Driver Area (SDA) of memory. Drivers are placed in SDA by specifying either "M" or "S" in the EQT entry definition phase during system generation. The "S" option specifies that the driver will be entered under the appropriate map (System or User) depending on the location of the I/O request buffer. The "M" option specifies that the driver will always be entered under System Map regardless of the location of the I/O buffer.

The "S" and "M" options also differ in the types of checks performed by RTE prior to entering the driver. If the "S" option is selected and the I/O buffer is in the user program, RTE will check to see if SDA is included in the program's logical address space (map). If not, RTE aborts the program with an IO11 error, "Type 4 program made an unbuffered I/O request to a driver that did not do its own mapping." If the "M" option is selected, RTE will not check to see if the I/O request buffer and SDA are in the same logical address space; rather, RTE enters the driver under the System Map, assuming that the drive will perform any needed mapping. This is why the "M" option is sometimes called the "driver does its own mapping" option.

Thus, a driver in SDA operating under the System Map may need to access a buffer in a User map for two reasons. The first reason is to access a second buffer located in the user program. The second is when an "M" option driver must process an I/O request with the request buffer, and possibly a second buffer, located in the user program. In either case, the driver can use the $XDMP subroutine to reload the User Map to describe the appropriate program.

The procedure for using $XDMP is as follow: The driver first saves the current contents of the User Map registers, and then calls $XDMP to reload the User Map to describe the desired program. The driver can then use a series of cross-map loads and stores to access the buffer described by the User Map. Note that the User Map should not be enabled since drivers in SDA are not necessarily included in all user maps. After all accesses have been made, the driver restores the original state of the User Map registers before continuing with its normal processing.

The following example shows how an SDA driver operating under the System Map might use the $XDMP subroutine to access a buffer in a user program.

```
        EXT $XDMP
           .
           .          Normal SDA driver processing under System Map
           .
        LDA MAPAD     Set A = address of User Map storage area
        IOR SIGN      Set sign bit indicating STORE Map in memory
        USA           Save current User Map in Memory for later

        LDA IDADR     Get ID Segment address of program that contains
                      buffer
        JSB $XDMP     Call $XDMP to set up User Map for this program
        SZA,RSS       Check for error return.  If A-Reg = 0,
                      specified program was not found in memory and
                      user map is not set up.
        JMP ERROR     Error exists, go handle it
           .
           .          No errors.  Access buffer via series of
           .          cross-map loads and stores, since System
           .          Map is still enabled
           .
        LDA MAPAD     Access address of User Map storage area
        USA           Restore original contents of User Map
           .
           .          Proceed with normal processing under System Map
           .

MAPAD   DEF MAP       Address of User Map storage area
MAP     BSS 32        User map storage area
SIGN    OCT 100000
IDADR   BSS 1         Storage for ID segment address
```

Remember that the driver using this routine must save the current contents of the User Map registers before calling $XDMP, and must restore the registers to their original value after all accesses have been made to the buffer. The example above illustrates this procedure.

NOTE

Before a standard driver in SDA is entered to process an I/O request, the system automatically reloads the User Map registers to describe the requesting program if the I/O buffer is located in the program. This is done regardless of whether the driver is entered under the System Map or the User Map. Thus, if the driver is entered under the System Map, and if the buffer that the driver wishes to access is located within the calling program, some processing time can be saved by not calling $XDMP to reload the User Map registers. A call to $XDMP is not needed in this case since the system has already reloaded the User Map registers with the correct information. The driver must still access the buffer via cross-map loads and stores however.

Obtaining the Subchannel

There are two ways to obtain the subchannel number. The first is to pull it out of the EQT. The second, for RTE-6/VM only, is to use the system subroutine $SUBC. $SUBC will return the subchannel number in the low bits of the A-Register.

```
EXT $SUBC
    .
    .
    .
JSB $SUBC
subchannel # returned in the A-Register
```

Operating System Trap Cell Instructions

In RTE-6/VM, for E- and F-Series computers only, there are four instructions available for handling interrupts, excluding power-fail. These instructions replace the JSB LINK,I that is in the trap cell in RTE-M, -II, -III and IV systems.

The four interrupt-handling instructions perform some of the operations handled by the central interrupt processor, thereby speeding up the interrupt processing and reducing overhead.

The four instructions speed up interrupt processing in two ways. First, the driver does not need to execute code to determine what type of interrupt occurred (memory protect, DCPC, TBG or normal device). Second, the operations are performed in microcode, allowing faster exection of the functions.

Operating System trap cell instructions do not have to be generated into the system. On the E- and F-Series computers, the system places them in the trap cells for you.

Sample Standard RTE Driver

The sample driver illustrated in Figure 3-7 demonstrates some of the principles involved in writing a standard I/O driver for the RTE operating system. Note that this driver is for tutorial purposes only and is not one of the drivers supplied with the system.

```
               ASMB,Q
                     NAM DVR70   **STANDARD RTE DRIVER EXAMPLE **
*
*
                     ENT I.70,C.70
*
*
* DRIVER 70 OPERATES UNDER THE CONTROL OF THE I/O CONTROL (IOC)
* AND THE CENTRAL INTERRUPT CONTROL (CIC) MODULES OF RTE.
* THIS DRIVER IS RESPONSIBLE FOR CONTROLLING OUTPUT
* TRANSMISSION TO A 16 BIT EXTERNAL DEVICE.
* I.70 IS THE ENTRY POINT FOR THE *INITIATION* SECTION
* AND C.70 IS THE ENTRY POINT FOR THE *CONTINUATION/COMPLETION*
* SECTION.
*
* NOTE THAT THIS DRIVER DOES NOT PROCESS TIME-OUTS OR
* POWER FAIL.  THESE PROCEDURES ARE LEFT ENTIRELY UP TO
* THE SYSTEM.
*
* REMEMBER THAT RTE SETS THE ADDRESSES OF EACH WORD OF
* THE 15 WORD EQT ENTRY FOR THE DEVICE BEING SERVICED INTO
* THE BASE PAGE COMMUNICATIONS AREA ON EACH ENTRY TO THE
* DRIVER.
* THIS DRIVER REFERENCES THESE ADDRESSES THROUGH VARIABLES
* EQT1 THROUGH EQT15.
*
* **********************
* * INITIATION SECTION *
* **********************
*
* THE INITIATION SECTION IS CALLED FROM I/O CONTROL (IOC) TO
* INITIALIZE A DEVICE AND INITIATE AN OUTPUT OPERATION
```

Figure 3-7. Standard RTE-Driver Example

```
*
* THE CALLING SEQUENCE FOR THE INITIATION SECTION IS:
*
*              (SET A = SELECT CODE OF I/O DEVICE)
*         P    JSB I.70
*         P+1  (RETURN POINT)
*
*         ON RETURN, A REGISTER INDICATES STATUS, AS FOLLOWS:
*
*               A = 0,   OPERATION SUCCESSFULY INITIATED
*               A NOT 0, OPERATION REJECTED FOR THE FOLLOWING
*                        REASON:
*
*                        A = 1 = ILLEGAL READ REQUEST
*                        A = 2 = ILLEGAL CONTROL REQUEST
*
* (NOTE, HOWEVER, THAT A "CLEAR" CONTROL REQUEST FROM THE
*  SYSTEM WILL BE PROCESSED BY THE DRIVER, AS REQUIRED.)
*
* ************************************
* * CONTINUATION/COMPLETION SECTION *
* ************************************
*
* THE CONTINUATION/COMPLETION SECTION IS CALLED BY CENTRAL
* INTERRUPT CONTROL (CIC) TO CONTINUE OR COMPLETE AN OPERATION WHEN
* AN INTERRUPT IS DETECTED ON THE DEVICE
*
* THE CALLING SEQUENCE FOR THE COMPLETION SECTION IS:
*
*              (SET A = SELECT CODE OF I/O DEVICE)
*         P    JSB C.70
*         P+1  COMPLETION RETURN
*         P+2  CONTINUATION RETURN
*
*         ON RETURN, A & B REGISTERS INDICATE STATUS, AS FOLLOWS:
*
*               ON A COMPLETION RETURN:
*
*               A = 0, SUCCESSFUL COMPLETION, WITH
*                      B = NUMBER OF WORDS TRANSMITTED
*
*               A = 2, TRANSMISSION ERROR DETECTED
*
*               ON A CONTINUATION RETURN, THE REGISTERS ARE
*               MEANINGLESS
*
```

Figure 3-7. Standard RTE Driver Example (Continued)

```
* RECORD FORMAT:
*
*     THIS DRIVER PROVIDES A 16 BIT BINARY WORD
*     TRANSFER ONLY.
*
*     ***********************
*     * INITIATION SECTION *
*     ***********************
  00000 000000  I.70  NOP           ENTRY FROM IOC
*
  00001 000100R       JSB SETIO     CONFIGURE I/O INSTRUCTNS FOR DEVICE
  00002 001665        LDA EQT6,I    GET CONTROL WORD OF REQUEST, AND
  00003 000117R       AND =B3       ISOLATE THE REQUEST TYPE
*
  00004 000115R       CPA =B1       IF REQUEST IS FOR INPUT
  00005 000000R       JMP I.70,I    THEN REJECT (A=1=ILLEGAL READ)
  00006 000116R       CPA =B2       IF REQUEST IS FOR OUTPUT
  00007 000017R       JMP D.X1      THEN GO PROCESS WRITE REQUEST
*
* CONTROL REQUEST CHECK IF IT IS A "CLEAR" CONTROL REQUEST
* IF SO, ASSUME IT WAS ISSUED BY SYSTEM, CLEAR DEVICE, AND RETURN
*
  00010 001665        LDA EQT6,I    ACCESS CONTROL WORD
  00011 000124R       AND =B3700    ISOLATE SUBFUNCTION
  00012 002002        SZA           "CLEAR" REQUEST?
  00013 000015R       JMP REJCT     NO, SO REJECT REQUEST AS ILLEGAL
*
  00014 106700  I.0   CLC SC        YES, CLEAR DEVICE AND RETURN
*
* REQUEST ERROR - CAUSE REJECT RETURN TO IOC
*
  00015 000116R REJCT LDA =B2       SET A=2=ILLEGAL CONTROL REQUEST
  00016 000000R       JMP I.70,I    AND RETURN
*                   (A=2=ILLEGAL CONTROL REQUEST)
*
*WRITE REQUEST PROCESSING
*
  00017 001666  D.X1  LDA EQT7,I    GET REQUEST BUFFER ADDRESS
  00020 001670        STA EQT9,I    AND SET IT AS CURRENT ADDRESS
  00021 001667        LDA EQT8,I    GET REQUEST BUFFER LENGTH
  00022 003004        CMA,INA       MAKE NEGATIVE AND
  00023 001671        STA EQT10,I   SAVE AS REMAINING BUFFER LENGTH
  00024 002002        SZA           IS BUFFER LENGTH = 0?
  00025 000031R       JMP D.X3      NO, PROCESS AS USUAL
  00026 000120R       LDA =B4       YES, MAKE IMMEDIATE COMPLETN RETURN
  00027 006400        CLB           SET TRANSMISSION LOG = 0 INTO B
  00030 000000R       JMP I.70,I    AND RETURN (A=4=IMMED. COMPLETION)
```

Figure 3-7. Standard RTE Driver Example (Continued)

```
*
* CALL THE CONTINUATION/COMPLETION SECTION TO WRITE FIRST WORD
*
  00031 000114R D.X3  LDA P2         ADJUST RETURN ADDRESS SO WILL
  00032 000036R       STA C.70       RETURN HERE (INITIATION SECTION)
  00033 000047R       JMP D.X2       GO TO COMPLETION SECTION
*
  00034 002400  IEXIT CLA            NOW RETURN TO IOC WITH
  00035 000000R       JMP I.70,I     OPERATION INITIATED (A = 0 = OK)
*
*     ************************************
*     * CONTINUATION/COMPLETION SECTION *
*     ************************************
*
  00036 000000  C.70  NOP            CONTINUATION/COMPLETION ENTRY POINT
*
  00037 000100R       JSB SETIO      CONFIGURE I/O INSTRUCTIONS
*
  00040 001660        LDA EQT1,I     CHECK FOR SPURIOUS INTERRUPT
  00041 000126R       AND =B77777    ISOLATE I/O REQST LIST PTR (15 BITS)
  00042 002002        SZA            IS REQST IN PROGRESS?
  00043 000047R       JMP D.X2       YES, GO PROCESS REQUEST
*
  00044 001774        STA EQT15,I    NO SPURIOUS INTRUPT-ZERO TO CLOCK
  00045 000036R       ISZ C.70       ADJUST RETURN TO P+2 (CONTINUATION)
  00046 000036R       JMP C.70,I     MAKE CONTINUATION RETURN TO CIC
*
  00047 002400  D.X2  CLA            IF CURRENT BUFFER LENGTH  = 0,
  00050 001671        CPA EQT10,I    THEN GO TO STATUS
  00051 000063R       JMP I.3        SECTION. (I.E., TRANSFER DONE NOW)
*
  00052 001670        LDB EQT9,I     GET CURRENT BUFFER ADDRESS
*
  00053 001670        ISZ EQT9,I     ADD 1 FOR NEXT WORD
  00054 000001        LDA B,I        GET WORD TO BE WRITTEN TO DEVICE
  00055 001671        ISZ EQT10,I    INCREMENT WORD COUNT ALSO
  00056 000000        NOP            IGNORE P+1 SKIP IF LAST WORD
*
  00057 102600  I.1   OTA SC         OUTPUT WORD TO INTERFACE
  00060 103700  I.2   STC SC,C       TURN DEVICE ON
*
  00061 000036R       ISZ C.70       ADJUST RETURN TO P+2 (CONTINUATION)
  00062 000036R       JMP C.70,I     MAKE CONTINUATION RETURN
*
* STATUS AND COMPLETION SECTION
*
```

Figure 3-7. Standard RTE Driver Example (Continued)

```
  00063 102500  I.3   LIA SC         GET STATUS WORD FROM DEVICE
  00064 000121R       AND =B77       STRIP OFF UNUSED BITS
  00065 000001        STA B          SAVE IN B TEMPORARILY
  00066 001664        LDA EQT5,I     REMOVE PREVIOUS STATUS
  00067 000127R       AND =B177400   BITS IN EQT WORD 5
  00070 000001        IOR B          OR IN NEW BITS
  00071 001664        STA EQT5,I     AND RESET INTO EQT WORD 5
*
  00072 002400        CLA            SET A = 0 = OK RETURN CODE
  00073 000120R       CPB =B4        ERROR STATUS BIT ON?
  00074 000116R       LDA =B2        YES, SET A = 2 = ERROR RETURN
*
  00075 001667        LDB EQT8,I     SET B = TRANSMISSION LOG
*
  00076 106700  I.4   CLC SC         CLEAR DEVICE CONTROLLER
*
  00077 000036R       JMP C.70,I     MAKE COMPLETION RETURN TO CIC
*
*
*     **********************
*     * SUBROUTINE SETIO *
*     **********************
*
* SUBROUTINE <SETIO> CONFIGURES ALL I/O INSTRUCTIONS IN DRIVER
*
  00100 000000  SETIO NOP            ENTRY POINT
*
  00101 000113R       IOR LIA        COMBINE LIA WITH I/O
  00102 000063R       STA I.3        SELECT CODE AND SET IN CODE
*
  00103 000122R       ADA =B100      CONSTRUCT OTA INSTRUCTION
  00104 000057R       STA I.1
*
  00105 000123R       ADA =B1100     CONSTRUCT STC,C INSTRUCTION
  00106 000060R       STA I.2
*
  00107 000125R       XOR =B5000     CONSTRUCT CLC INSTRUCTION
  00110 000014R       STA I.0
  00111 000076R       STA I.4
*
  00112 000100R       JMP SETIO,I    RETURN
*
*
*     *************
*     * DATA AREA *
*     *************
*
```

Figure 3-7. Standard RTE Driver Example (Continued)

```
* CONSTANT AND STORAGE AREA
*
        000000  A      EQU 0          A-REGISTER
        000001  B      EQU 1          B-REGISTER
*
        000000  SC     EQU 0          DUMMY I/O SELECT CODE NUMBER
  00113 102500  LIA    LIA 0          CODE FOR LIA INSTRUCTION
  00114 000033R P2     DEF IEXIT-1    RETURN POINT IN INITIATION SECTION
*
* ** BASE PAGE COMMUNICATIONS AREA DEFINITIONS **
*
        001650  .      EQU 1650B
*
        001660  EQT1   EQU .+8
        001661  EQT2   EQU .+9
        001662  EQT3   EQU .+10
        001663  EQT4   EQU .+11
        001664  EQT5   EQU .+12
        001665  EQT6   EQU .+13
        001666  EQT7   EQU .+14
        001667  EQT8   EQU .+15
        001670  EQT9   EQU .+16
        001671  EQT10  EQU .+17
        001672  EQT11  EQU .+18
        001771  EQT12  EQU .+81
        001772  EQT13  EQU .+82
        001773  EQT14  EQU .+83
        001774  EQT15  EQU .+84
*
*
  00115 000001
  00116 000002
  00117 000003
  00120 000004
  00121 000077
  00122 000100
  00123 001100
  00124 003700
  00125 005000
  00126 077777
  00127 177400
                       END
```

Figure 3-7. Standard RTE-Driver Example

Chapter 4
Writing Privileged RTE Drivers

Introduction

Peripheral devices that are synchronous in nature, or that generate interrupts at very high rates, need special attention in an RTE system. Such devices cannot be controlled by standard RTE drivers on a word-by-word transfer basis, because this method cannot guarantee that the interrupts generated by such device controllers will be processed within the required response time. There are two reasons why the response time may be exceeded:

1. An interrupt is not recognized immediately by RTE if the interrupt system is disabled at the time the interrupt occurs. For example, this happens if the interrupt occurs while a standard driver is processing a previous interrupt, or while RTE itself is executing.

2. Once an interrupt is recognized, the system overhead required to direct the interrupt to the appropriate driver for processing may be too long.

One way to guarantee a fast interrupt response time is to utilize DCPC transfers for synchronous and high speed devices. The special DCPC hardware allows the transfer to occur simultaneously with other RTE operations, thereby eliminating the above problems.

However, DCPC transfers do not allow the driver to perform any processing that might be required on each data word as it is transferred. For example, it might be necessary to check a parity bit on each word as it is received from the device. Thus, a special interrupt processing method is needed for any high speed or synchronous device that requires driver. interaction on each data word transferred. This interrupt processing method must have the following properties:

a. The ability to recognize interrupts immediately, regardless of what other RTE operation is in progress.

b. A means to eliminate the system overhead associated with processing an interrupt.

c. Driver interaction on each data word transferred.

Privileged interrupt processing was specifically designed to meet these criteria. This method requires that a special I/O card, known as the Privileged Interrupt card, be present in the system. The Privileged Interrupt card is inserted in the computer such that it physically separates the I/O cards into two groups. All devices whose I/O cards are in lower-numbered (i.e.,higher priority) select codes are known as privileged devices; these are the high speed and synchronous devices that require driver interaction on each word transferred. The I/O cards of all other devices in the system are placed in higher-numbered (i.e.,lower priority) select codes and are known as non-privileged devices.

Systems with Privileged Interrupt cards are referred to as privileged systems, and a special type of RTE driver (known as a privileged driver) is required for each privileged controller present in the system. Standard RTE drivers are used for the remaining non-privileged devices.

The Privileged Interrupt card can be any standard I/O card that contains the normal control and flag flip-flop circuitry. Because of the position of the Privileged Interrupt card in the I/O priority chain, setting of the control flip-flop on the card holds off all interrupts from the non-privileged device controllers, while at the same time allowing the privileged device controllers to interrupt.

When a Privileged Interrupt card is present in the system and a non-privileged interrupt occurs (or when the system is requested to perform some function via an EXEC call or operator request), RTE performs the following functions before entering the standard driver (or system routine) to process the interrupt:

a. Disables the interrupt system and saves the state of the machine.

b. Sets the control flip-flop on the Privileged Interrupt card to hold off any further non-privileged interrupts.

c. Disables the DCPC completion interrupts. These interrupts are not held off by the Privileged Interrupt card since the DCPC completion interrupts occur on the highest priority select codes (6 and 7).

d. Reenables the interrupt system and enters the driver to process the interrupt.

The above means that a privileged system processes standard (i.e., non-privileged) interrupts and requests for system functions with the interrupt system in a hold-off state, rather than with the interrupt system disabled (as it does in non-privileged systems). The privileged interrupts are always enabled and can interrupt any process taking place and be serviced almost immediately.

When servicing of the non-privileged interrupt is completed, RTE clears the control flip-flop on the Privileged Interrupt card and reenables the DCPC completion interrupts if they are needed by the standard driver using DCPC. This returns the system to a state where any interrupt (privileged or non-privileged) can occur and be recognized almost immediately.

To eliminate the system overhead associated with processing interrupts, the contents of the interrupt trap cells for the privileged device controllers are changed from a JSB LINK.I instruction (where LINK contains the address of the entry point to CIC), to a JSB $JPNN,I instruction (where $JPNN contains the address of the entry point of the privileged interrupt routine). When a privileged interrupt occurs, the privileged routine is entered directly, rather than from CIC. This eliminates the system overhead. The tradeoff is that the privileged driver must be somewhat more complex than a standard driver, since it must perform some of the housekeeping duties normally handled by CIC.

RTE records a system as privileged by storing at generation time (or, for RTE-IV, at reconfiguration time) the I/O select code address of the Privileged Interrupt card in Base Page Communications Area word DUMMY. Systems without a Privileged Interrupt card have a a zero in base page word DUMMY.

In general, privileged drivers are very similar to standard drivers; thus most of the material presented in Chapter 3 for standard drivers also applies to privileged drivers. Since only the differences are pointed out in this section, the reader should be familiar with the material presented in Chapter 3 before continuing with the privileged driver considerations described below.

General Privileged Driver Structure and Operation

Privileged drivers are responsible for the initiation, continuation, and completion of all I/O requests for privileged devices. Since privileged drivers operate independently of RTE, there are several additional requirements and restrictions that must be followed to ensure the integrity of the operating system and the proper operation of the driver. These restrictions and requirements are described in subsequent subsections.

Privileged drivers are generally designed in three sections: 1) an initiation section, 2) a privileged section, and 3) a completion section. The drive must have a name in the form DVynn, and the initiation and completion entry points must have names in the forms Ixnn and Cxnn respectively. The rules for the choice of "x," "y" and "nn" are the same as those given previously for standard drivers. There are no special rules for the entry point name of the privileged section. For consistency with Ixnn and Cxnn, it is suggested that a name such as Pxnn be chosen where "x" and "nn" agree with the characters chosen for Ixnn and Cxnn.

IOC calls the initiation section of a privileged driver when an I/O request for a privileged device is made. This call has the same format as the call to the initiation section of a standard driver, described in the previous chapter.

The privileged section of a privileged driver is somewhat similar to the continuation section of a standard driver. The privileged section is entered on each interrupt from the privileged device controller and is responsible for reading and writing the next data word and restarting the device. Since the privileged section is entered directly from the trap cell on interrupt (rather than from CIC), it must save and restore the state of the computer on entry and exit. (These tasks are performed by CIC for standard drivers.)

The completion section of a privileged driver has an entry point named Cxnn, and is responsible for returning to RTE when the I/O transfer is complete.

The overall operation of a privileged I/O request from initiation to completion is summarized below:

a. The privileged driver is called by a standard EXEC I/O call.

b. If the request is being made by a user program and the call is not buffered, the calling program is placed into I/O suspension.

c. The interrupt trap cell for the privileged device controller is changed by the privileged driver from a JSB LINK,I instruction (where LINK contains the address of the entry point to CIC) to a JSB $JPBB,I instruction (where $JPNN contains the address of the privileged section entry point, Pxnn).

d. Each time the device controller interrupts, the system overhead is circumvented because the privileged section Pxnn is entered directly.

e. After each interrupt, if another data transfer is still required to satisfy the buffer length, the device controller is restarted and the privileged section is exited.

f. When the entire data buffer has been filled, the driver needs a way to inform RTE that the transfer is complete. This is accomplished by allowing the driver to time out, which causes IOC to reenter the driver at Cxnn.

g. Cxnn returns the transmission log (via the B-Register) and a successful completion indication (via the A-Register) to IOC.

h. IOC then reschedules the program that made the I/O request.

Initiation Section

The initiation section of a privileged driver performs the functions listed below. The list is similar to the one given earlier for standard drivers with the exception that no DCPC processing can be done by privileged drivers. (See the "Privileged Driver Design Considerations" subsection of this manual.)

a. Checks for power fail/automatic restart entry by examining bit 15 of EQT entry word 5, which is set to 1 only on this type of entry. If bit 15 is set, the appropriate power-fail/automatic restart processing should be done. This check need only be made by drivers that are designed to process power-fail interrupts (as described in the "Power-Fail Processing" subsection of this manual).

b. Configures all I/O intructions in the driver to reference the specific I/O select code of the device controller. This step is done only on the first entry to the driver since there is one privileged driver for each privileged device controller in the system. (See the "Privileged Driver Design Considerations" subsection of this manual.)

c. Clears bit 12 in EQT entry word 4 if time-outs are to be handled by the system. This bit will be reset to 1 by the privileged section when it sets up to complete the call.

d. Rejects the request and follows the procedure described in step "f" if:

   1. A status check of the device or controller indicates that it is inoperable, or

   2. The request code or other parameters are illegal.

e. Initializes software flags and activates the device controller. All variable information pertinent to the transmission can be saved in the EQT entry associated with the controller, providing that the driver saves the addresses of the EQT entry internally in the driver itself at initiation. These addresses are not available on base page on subsequent entries to the driver. (See the "Privileged Driver Design Considerations" subsection of this manual.)

f. Returns to IOC (via JMP Ixnn,I) with the A-Register set to indicate initiation or rejection (and the cause of the rejection) as follows:

A-Reg Status

0 The operation was initiated successfully.

1 Operation rejected: read or write illegal for device

2 Operation rejected: control request illegal or undefined

3 Operation rejected: equipment malfunction or not ready

4 The operation was immediately completed. This means that the driver was able to completely satisfy the request without the need of a subsequent interrupt and that the program making the I/O call can be rescheduled immediately. The B-Register should be set to the number of words or characters (depending upon which the user specified) transferred. This value is known as the transmission log.

5 This return must NOT be used by privileged drivers.

6 to 99 The program making the I/O request is aborted (unless the no-abort bit was set in the call) and an I/O error message is printed on the system console. (Note that this return can be used for unbuffered I/O requests only. This therefore excludes the use of return codes 6 through 99 on any Class, buffered or system I/O request.) The error message has the following format:

```
IOxx yyyyy
NNNNN ABORTED
```

where:

xx = The return code from the driver (decimal 06 to decimal 99),

yyyyy = The address of the aborted I/O request in program NNNNN, and

NNNNN = The name of the program that made the I/O request.

This type of return can be used by drivers to generate their own I/O error messages at the system console. Note that certain codes are reserved for system use, as follows:

Return Code	Reserved for
6 - 59	HP system modules and drivers
60 - 99	User written drivers

Before returning to IOC, the initiation section modifies the trap cell for the privileged device controller to contain a JSB $JPNN,I instruction, where $JPNN contains the address of the privileged section entry point Pxnn. This causes all interrupts from the privileged device controller to be directed immediately to the privileged section of the driver. The driver needs to perform this step only once since the contents of the trap cell are not modified by any other program or system routine.

In setting up the trap cell, the driver must be sure that it is operating with the System Map enabled so that it has access to the trap cells. This can be ensured by having the calling program reference a buffer in SYSTEM COMMON when making the I/O request to the driver. (See the "Communication with User Programs" subsection later in this manual.)

An alternative method of setting up the trap cell is to point the trap cell directly at the privileged section entry point when the system is generated. This is done by entering sc,ENT,Pxnn (where "sc" is the select code of the privileged controller) during the Interrupt Table definition phase of the generation. When the generator detects an entry of this form it places a JSB $JPNN,I instruction (where $JPNN contains the address of Pxnn) into the appropriate interrupt trap cell. The generator also places a zero in the corresponding Interrupt Table entry to indicate that interrupts on the select code are not handled by RTE. This method requires that the privileged section entry point, Pxnn, be declared as an entry point in the privileged driver (via the ENT pseudo-instruction).

Privileged Section

When a privileged interrupt occurs, the operation currently in progress is suspended, and the privileged section of the driver is entered directly via the JSB $JPNN,I instruction in the trap cell. In addition to the normal tasks associated with continuing the data transfer, the privileged section is required to perform several housekeeping functions that are normally performed by CIC. This includes saving and restoring the state of the computer on entry and exit and disabling the DCPC completion interrupts so that the driver's operation is not interrupted.

The privileged section of a privileged driver performs the following functions:

a. Executes the following tasks done by CIC for standard drivers:

   1. Disables the entire interrupt system with a CLF 0 instruction so that the driver is not interrupted while performing the housekeeping functions.

   2. Disables the DCPD completion interrupts by issuing a CLC 6 instruction and a CLC 7 instruction. The DCPC completion interrupts are associated with I/O select codes 6 and 7, and therefore precede the Privileged Interrupt card in the I/O priority chain. Interrupts from these device controllers are not held off by the Privileged Interrupt card and must be disabled to prevent an interrupt from occurring while the privileged driver is executing.

   3. Saves the current contents of all program accessible registers (A,B,E,O, and if present, X, and Y) in a local buffer. These registers must be restored to their original contents before exiting the driver.

   4. Saves the previous state of the memory protect fence. When an interrupt occurs, the memory protect fence (if ON) is automatically turned off. The driver can determine the previous state of the memory protect fence (which is the state to which it should be restored after processing the interrupt) by examining Base Page Communications Area word MPTFL. If MPTFL equals zero, memory protect was ON when the privileged interrupt occurred and the privileged section must turn the fence back on before exiting. If MPTFL is non-zero, memory protect was OFF, and the privileged section must not restore the memory protect fence.

   5. Sets base page word MPTFL to 1 to indicate that the memory protect fence is now OFF.

   6. Saves the status of the Dynamic Mapping System so that it can be restored to its original state before returning to the point of interruption. This is done by executing an SSM instruction. (Applicable to systems with Dynamic Mapping only.)

   7. Reenables the interrupt system by executing an STF 0 instruction. This allows a higher priority privileged controller (if one exists) to interrupt the driver. All non-privileged interrupts are held off because the flag is still set on the card that caused the privileged interrupt.

b. Checks whether bits 0 - 14 of EQT word 1 (the controller I/O request list pointer) equal zero. If so, a spurious interrupt has occurred (i.e., no I/O operation was in progress at the time of the interrupt). The driver ignores the interrupt as follows:

1. Disables the interrupt system via a CLF 0 instruction so that the driver is not interrupted while clearing the controller.

2. Clears the control and flag flip-flops on the controller (usually via a CLC DEVIC,C instruction).

3. Proceeds to step "e-3" below to restore the computer to its original state before exiting.

c. Performs the input or output of the next data item. One of the following three actions is then taken:

1. If the transfer is not complete, the driver follows the procedure described in step "e" below to return to the suspended process at the point of interruption.

2. If the transfer is complete, the driver follows the procedure described in step "d" below to set up for a completion return to IOC.

3. If the driver detects a transmission error, it may reinitiate the transfer and attempt a retransmission. A counter for the number of retry attempts can be kept in the EQT entry. After initiating each retry, the driver follows the procedure described in step "e" below to return to the suspended process at the point of interruption.

d. Once the transfer is complete, the driver needs a way to indicate this fact to RTE so that the program that made the I/O request can be rescheduled. This is accomplished by letting the driver time-out. To do this, the driver performs the following steps:

1. Disables the interrupt system with a CLF 0 instruction so that no interruptions occur while the time-out is being set up and the computer is being restored to its original state.

2. Turns off the device controller to prevent further interrupts (usually with a CLC instruction).

3. Sets the time-out clock (EQT entry word 15) to -1 to cause a time-out of the privileged device controller at the next tick of the real-time clock. This will cause the completion section of the driver to be entered from IOC so that a normal completion return can be made to RTE.

4. Sets the "driver processes time-out" bit in EQT entry word 4 to one so that the driver will be reentered when the time-out occurs.

5. Follows the procedure described in step "e-3" below to restore the computer to its original state before exiting.

e. Before returning to the point of interruption, the privileged section performs the following steps to restore the computer to its original state upon entry:

1. Disables the interrupt system so that no interruptions occur while the computer is being restored to its original state.

2. Encodes the device controller to initiate the next data transfer, usually via a STC DEVIC,C instruction. Note that the device controller must not be encoded until the interrupt system is disabled and the driver is about to return to the point of interruption. If the device controller were encoded earlier, the driver might be reentered at Pxnn by the next interrupt before completely servicing the previous interrupt. Clearing of the flag on the privileged device controller's I/O card will also allow any lower priority interrupts to be recognized by the system when the interrupt system is reenabled.

3. Checks to see if either of the DCPC completion interrupts needs to be reenabled, a follows:

   * If the memory protect fence was initially OFF, the driver must not reenable the DCPC completion interrupts. If the memory protect fence was initially OFF, the privileged driver interrupted the operation of the system or another privileged driver. These routine operate with the DCPC completion interrupts disabled and assume that the completion interrupts will remain disabled if they are interrupted.

   * If the memory protect fence was initially ON, the DCPC completion interrupts are turned back on only if the standard driver currently using the DCPC channel needs the interrupt. Standard drivers that need the DCPC completion interrupt set bit 15 of the appropriate DCPC Assignment Control Word (in the Interrupt Table) to 1 as a flag. (See the "DCPC Processing" subsection earlier in this manual. If bit 15 is set, a privileged driver must reenable the appropriate DCPC completion interrupt by issuing a STC 6 or STC 7 instruction. If bit 15 is not set, the privileged driver must not reenable the interrupt.

4. Restores all saved registers to their original values.

5. Restores base page word MPTFL to its original value. (This word is used to indicate the current status (ON/OFF) of the memory protect fence).

6. Turns the interrupt system back on via a STF 0 instruction to allow other interrupts to occur.

7. Turns the memory protect fence back on via a STC 5 instruction if the fence was ON initially.

8. Performs one of the following actions depending on whether or not the system includes the Dynamic Mapping feature:

   * Restores the Dynamic Mapping System to its original value at interrupt and returns to the suspended process at the point of interruption by executing a jump and restore status (JRS) instruction indirect through the entry point Pxnn. (Performed only in systems with the Dynamic Mapping feature.)

   * Returns to the suspended process at the point of interruption via a jump indirect through the entry point Pxnn. (Performed only in systems without the Dynamic Mapping feature.)

   NOTE

   If the memory protect fence was turned on in step 7, execution of the JRS instruction (in step 8) to restore DMS status can only be performed if the System Map is currently enabled. An attempt to execute it with the User Map enabled will result in a DMS violation. Thus, if the driver switches to operation under the User Map for any reason, the System Map must be reenabled before executing the JRS instruction. The explanation of map switching given in the "Subroutines for Special Mapping Functions" subsection of this manual illustrates this procedure.

Completion Section

When the time-out set up by the privileged section occurs, IOC enters the continuation section of the privileged driver at entry point Cxnn. The continuation section sets the A-Register equal to the appropriate completion code and the B-Register equal to the transmission log. It then returns to IOC via a jump indirect through the entry point Cxnn (JMP Cxnn,I). The return point (P+1) and allowable completion codes (0-4) are the same as those described earlier for the completion section of a standard driver.

Privileged Driver Design Considerations

Privileged drivers operate independently of RTE. In fact, the operation of the RTE operating system itself may be suspended while a privileged interrupt is being serviced. As a result, the writer of a privileged driver must adhere to the following design requirements:

a. Privileged drivers must not use any of the features or request calls of RTE. Calling a system process might involve entering RTE while it is processing another request. This cannot be allowed because RTE is not reentrant.

b. Privileged drivers cannot use either DCPC channel because it is very difficult to coordinate the use of DCPC with the operating system and other drivers that may be using DCPC.

c. If a privileged driver wishes to use the EQT entry for temporary storage, the initiation section must read the EQT entry addresses from the Base Page Communications Area and save them internally in the driver. These addresses are not available in base page on subsequent interrupts since the privileged driver is entered directly from the trap cell instead of from CIC. (CIC is the module that places these addresses into the base page before calling a standard RTE driver to process an interrupt.)

d. Since privileged drivers are required to keep information relating to the I/O request internally (see "c" above), a separate privileged driver is required for each privileged device controller present in the system. For each additional controller of the same type, an additional copy of the privileged driver must be generated into the system. Each copy of the driver must have unique names for DVynn, Ixnn, Pxnn, and Cxnn.

Communication with User Programs (DMS Systems Only)

Privileged drivers are automatically entered with the System Map enabled when a privileged interrupt occurs. If the I/O request buffer for the privileged call is located in a user program, the driver will have to switch maps before it can access the buffer. Any privileged driver in a DMS system should therefore be designed for user communication through SYSTEM COMMON or the Subsystem Global Area (SSGA) to avoid the overhead of map switching. These areas can be specified at generation to be included in both the System Map and User Map, and hence can be accessed directly by both user programs and privileged drivers without any map switching.

Otherwise, if the I/O request buffer is located in a user program, some map switching will have to be done before the privileged driver can access the buffer. This map switching procedure is described in detail in the "Subroutines for Special Mapping Functions" subsections of this manual.

Discussion of Sample DMS Privileged Driver

The following discussion describes Figure 4-1, an example of a privileged driver written specifically for use in a system with the Dynamic Mapping feature. (Figure 4-2 shows a similar driver written specifically for a system that does not include the Dynamic Mapping feature.) For the purposes of the discussion, this driver has been given the generalized name of DVYNN.

The device controller transfers one word of data each time it interrupts, and the data is stored in a buffer passed to the driver via the call parameters. Note that the design of the DMS privileged driver assumes that the I/O request buffer is located in SYSTEM COMMON for two reasons: 1) it ensures that the driver's initiation section is entered with the System Map enabled. This is necessary for the proper operation of the trap cell modification technique used in that section; 2), it allows the driver to place data values directly in the I/O request buffer without any map switching.

Note that the driver does not process power-fail interrupts nor does it process any time-outs, except for the time-out it creates as a means to complete the I/O request and return to IOC.

Initiation Section

Refer to the partial listing of the sample privileged driver in Figure 4-1 (or 4-2). A standard I/O call to input from the device causes the calling program to be I/O suspended and the driver to be entered at Ixnn.

Since this driver can control just one device controller, there is no need to configure the I/O instructions more than once. Therefore, the driver is configured the first time it is entered, and the switch at subsequent entry to the driver. The initiation section also saves the addresses of those EQT entry words that will be used by the privileged section since these addresses will not be available in base page on subsequent interrupts.

The modification of the trap cell is also performed just once (as part of the configuring routine) and is not modified again on any later entries into the initiation section. The trap cell is altered so that the device controller interrupts are channelled to the privileged section of the driver (Pxnn) instead of to CIC. The JSB $JPNN,I instruction (where $JPNN contains the address of Pxnn) is established by coding a JSB instruction on the base page (see listing).

The request code is checked for validity. All write request and control requests (except a "clear" control request from the system) are rejected. For read requests, a counter is established for the number of readings to be taken, and the buffer address for the storage of the data is saved. The "driver processes time-out" bit in EQT entry word 4 is cleared so that any unexpected time-outs are handled by the system. This bit is later reset to 1 by the privileged section when it sets up a time-out as a means of returning to IOC at the end of the I/O request. Finally, the initiation section sets up and encodes the device controller to begin a read operation and returns to IOC.

Privileged Section

When the device controller interrupts, the privileged section (Pxnn) is entered directly as a result of the controller's trap cell modification.

Because entry is made directly into Pxnn, Pxnn must do the housekeeping that is normally done by CIC when a standard interrupt occurs. Thus, before Pxnn can turn the interrupt system back on to allow higher priority privileged interrupts to be recognized, Pxnn must ensure that the DCPC channels cannot interrupt, save the user-programmable registers, save the old memory protect status, and set its new status. For systems with Dynamic Mapping, Pxnn must also save the Dynamic Mapping System status at the time of interrupt.

Pxnn then loads and stores the data in the next unfilled buffer word. If there is yet another data point to be taken, Pxnn sets up the device controller for the next reading, disables the interrupt system, encodes the device controller, restores memory protect status and its flag, restores the user programmable registers, turns the interrupt system back on, and exits. For systems with Dynamic Mapping, Pxnn must also restore the Dynamic Mapping System status to its original value. All of this basically resets the system to its state before Pxnn was entered.

When the privileged section wants to perform system operations such as scheduling a program, and still continue with the privileged section, then Pxnn should set EQT word 15 to -1 so a time-out will occur at the next tick of the real-time clock. The continuation section can perform the required system task and then do a continuation exit.

When the last reading is taken, Pxnn disables the interrupt system, turns off the device controller, and sets up the privileged controller's EQT so that a time-out will occur at the next tick of the real-time clock. Pxnn then resets the system to its original state and returns to the suspended process at the point of interruption.

Completion Section

The status of the device controller and the driver is now unchanged until the Time Base Generator (TBG) interrupts. The TBG causes a time-out of the privileged controller (because a -1 was set into EQT entry word 15), which in turn causes IOC to pass control to the completion section at Cxnn. The completion section simply sets the A- and B-Registers to the status and transmission log, respectively, and returns to IOC. IOC then reschedules the calling program and initiates any remaining requests for the controller as if it were a standard (non-privileged) controller.

Time-Out Values for Privileged Drivers

If the user wishes to specify a time-out value for the privileged controller (to prevent indefinite suspension in the event that the controller malfunctions), the time-out value must be long enough to cover the entire period from I/O initiation to completion. This is different from the time-out value for a standard driver, which is normally only long enough to cover the expected time between interrupts from the standard device controller.

Each time IOC or CIC enters a standard driver to initiate or continue an I/O request, it resets the time-out clock (EQT entry word 15) to the value specified at generation. However, since privileged drivers are not entered from CIC on interrupt, the time-out value is inserted into the privileged controller's time-out clock only at initiation. If this value is not long enough to cover the entire I/O transfer period, a time-out will occur while the data transfer is still in progress, and the transfer will be prematurely terminated. This can be prevented by specifying a suitably long time-out value, or by specifying a time-out value of zero (which disables the time-out feature entirely).

The time-out value set by the user to prevent indefinite suspension should not be confused with the time-out set up by the privileged driver to complete the call and return to IOC. In the latter case, the driver overrides the user-specified time-out by inserting its own value (-1) directly into the time-out clock before returning.

Subroutines for Special Mapping Functions (DMS Systems Only)

The mapping considerations for the initiation and continuation/completion sections of privileged drivers are the same as those for standard RTE drivers (see Chapter 3 of this manual). Prior to entering either of these sections, RTE will load and enable the correct map (System or User) needed to describe the I/O request buffer.

The privileged section of a privileged driver requires special mapping considerations however. The System Map will always be enabled when the privileged section is entered since this section is entered directly from the interrupt trap cell. If the I/O buffer is not accessible under the System Map, the privileged section must reload the User Map to describe the appropriate program before accessing the buffer. Therefore, if the privileged driver only processes I/O requests from buffers in System Common or System Available Memory, the privileged section can access the I/O buffer directly without making special mapping considerations.

Subroutine $PVMP can be used by privileged drivers to perform any needed map switching. The driver uses $PVMP as follows:

1. Save the contents of the current User Map.

2. Load the A-Register with the ID Segment address of the program containing the buffer.

3. Call $PVMP (JSB $PVMP) to load the User Map for the program specified in the A-Register.

4. Check for an error return. $PVMP returns with the A-Register set to zero if the specified program was not in memory and the User Map could not be set up.

5. If no error was detected, access the buffer through the newly loaded User Map.

6. After all accesses to the buffer have been made, restore the User Map to its original contents and continue with normal operation.

Notice that the use of $PVMP requires that the driver know the ID Segment address of the program making the I/O request. Since the driver can not be sure that this ID Segment address is available on base page or in the EQT, the program making the I/O request should cooperate with the driver and pass its ID Segment Address. This could be done by using one of the optional parameters and letting the driver retrieve the address through the appropriate optional parameter word in the EQT.

The recommended procedure for using $PVMP is somewhat different in RTE-III and RTE-IV. Each of these procedures is described in a separate section below.

NOTE

> Subroutine $PVMP can be called by privileged RTE drivers only. Standard drivers wishing to perform the same function must use subroutine $XDMP, which is described in Chapter 3 of this manual.

Mapping in RTE-III and RTE-M/III

Before entering the initiation or continuation section of a privileged driver, IOC enables the correct map needed to process the call, as follows:

a. When the I/O request buffer is located in SYSTEM COMMON or System Available Memory, IOC enables the System Map before entering the driver.

b. When the I/O request buffer is located in the calling program, IOC enables the User Map before entering the driver.

However, the System Map is always enabled upon entry to the privileged section, since the privileged section is entered directly from the interrupt trap cell.

Since the buffer, in general, may be located in either the User Map (prgram not swappable) or the System Map, the driver needs to identify which map is used. If necessary, the driver can then use $PVMP to access the buffer while in the privileged section.

In the following example, the initiation section of the privileged driver checks the status of the Dynamic Mapping System and sets a flag to indicate whether the System or User Map is enabled. The initiator also retrieves the ID Segment address of the program making the I/O request. Note that the driver is assuming that the program making the I/O request has cooperated with the driver and placed its ID Segment address in the first optional parameter.

The privileged section of the driver then saves the current state of the User Map registers and Dynamic Mapping System. After the driver has determined that the User Map is needed, it calls $PVMP to reload the User Map registers to describe the calling program. The driver then switches to operation under this map and accesses the memory described by it. After all accesses have been made, the driver reenables the System Map and restores the original state of the User Map registers before continuing with its normal processing under the System Map.

```
        EXT $PVMP

IXNN    NOP             Initiation Section entry point
                         System Map or User Map enabled depending on
                        location of I/O request buffer.
          .
          .
          .
        CLA
        STA IDADR       Clear IDADR signifying System Map used
        RSA             Access Dynamic Mapping System Status
```

```
        ALF             Position bit 12 into bit 0
        SLA,RSS         System Map enabled?
        JMP PROCD       Yes, System map enabled
                        No, user map enabled

        LDA EQT9,I      Access address of program making request
        STA IDADR       Save for use of Privileged Section later

PROCD   NOP
          .
          .             Continue Initiation Section processing
          .

        JMP IXNN,I      Return to IOC
PXNN    NOP             Privileged Section entry point (System Map
                        Enabled)
          .
          .             Normal driver processing under System Map
          .
        RSA             Get Dynamic Mapping System (DMS) status
        RAL,RAL         Position current status in upper bits
        STA DMSST       Save status for later

        LDA MAPAD       Set A = address of User Map storage area
        IOR SIGN        Set sign bit indicating Store Map in Memory
        USA             Save current User Map in memory for later

        LDA IDADR       Access IDADR to determine if User or System
                        Map used
        SZA,RSS         System Map used?
        JMP CONT        Yes, System Map used
                        No, User Map used
*
        JSB $PVMP       Call $PVMP to set up User Map for this program
        SZA,RSS         Check for error return
        JMP ERROR       Error exists, go handle it

        UJP CONT        No, errors.  Enable new User Map and continue.
CONT      .
          .             Process buffer under new User Map.
          .
        LDA IDADR       Determine if System or User Map is used
        SZA,RSS         System Map used?
        JMP NXT         Yes, System Map used
*                       No, User Map used

        SJP NEXT        Reenable System Map
NEXT    LDA MAPAD       Access address of User Map storage area
        USA             Restore original contents of User Map

        JRS DMSSTNXT    Restore original DMS status (i.e., System Map)
```

```
NXT     .           Enable Interrupt System.
        .
        .           Proceed with normal processing under System Map
        .
MAPAD DEF MAP       Address of User Map storage area
MAP   BSS 32        User Map storage area
SIGN  OCT 100000
IDADR BSS 1         ID segment address saved by initiation section
DMSST BSS 1         Temporary storage for DMS status
```

Remember that any driver using this routine must save the original contents of the User Map registers before calling $PVMP. It must also restore the original User Map after all accesses to the buffer have been made. Also remember that if the driver accesses the memory described by the User Map by enabling the map, it must save and restore the DMS status in addition to the original contents of the User Map. The example above illustrates this procedure.

NOTE

> The privileged driver could also access the buffer in the user program through a series of cross-map loads and stores without actually enabling the User Map. This in fact is the recommended procedure for using $PVMP in an RTE-IV system, and use of it by a privileged driver would allow the same privileged driver to be used in either type of system.

Mapping in RTE-IV and RTE-6/VM

Privileged drivers in RTE-IV must reside in the System Driver Area (SDA) of memory since the privileged section is always entered under the System Map directly from the trap cell. The driver can be placed in SDA by selecting either the "M" or "S" option during the EQT entry definition phase of system generation (see Chapter 3 of this manual for a discussion of the "M" and "S" options). The initiation and continuation/completion sections of the driver may be entered under either the System Map or the User Map depending on the location of the I/O buffer and the option selected at generation. The privileged section, however, will always be entered under the System Map.

If the I/O buffer being processed by the driver is located in System Common on System Available Memory, the privileged driver will always have access to the buffer under the System Map. If, however, the I/O buffer is located in the user program, the driver must consider two possible situations in which it is operating under the System Map while the buffer is in the User Map.

1. The initiation and continuation/completion sections of the driver have to access the buffer through the User Map if the "M" option was selected.

2. The privileged section must access the buffer through the User Map regardless of the option selected.

A privileged driver that needs to access a buffer in a user program while operating under the System Map can use the $PVMP subroutine to reload the User Map to describe the desired program.

In the following example of an "S" option privileged driver, the initiation section of the driver checks the DMS status and sets a flag to indicate whether the System Map or User Map is enabled. If the User Map is enabled, the initiator also retrieves the ID Segment address of the program making the I/O request. Note that the driver is assuming that the program has cooperated and placed its ID Segment address in the first optional parameter.

The privileged section of the driver then saves the current contents of the User Map. If the User Map is needed to access the buffer, the privileged section then calls $PVMP to reload the User Map registers to describe the calling program. The driver then uses a series of cross-map loads and stores to access the buffer described by the User Map. Note that the User Map should not be enabled since drivers in SDA are not necessarily included in all user maps. After all accesses have been made, the driver restores the original state of the User Map registers before continuing with its normal processing under the System Map.

```
        EXT $PVMP

IXNN    NOP             Initiation Section entry point
                        (System Map or User Map enabled depending on
                        location of I/O request buffer.)
          .
          .
          .
        CLA
        STA IDADR       Clear IDADR signifying System Map used
        RSA             Access Dynamic Mapping System Status

        ALF             Position bit 12 into bit 0
        SLA,RSS         System Map enabled?
        JMP PROCD       Yes, System map enabled
*                       No, user map enabled

        LDA EQT9,I      Access address of program making request
        STA IDADR       Save for use of Privileged Section later

PROCD   NOP
          .
```

```
            .           Continue Initiation Section processing
            .
            .
        JMP IXNN,I      Return to IOC
PXNN    NOP             Privileged Section entry point (System Map
                        Enabled)
            .
            .           Normal driver processing under System Map.
            .
        LDA MAPAD       Set A = address of User Map storage area
        IOR SIGN        Set sign bit indicating Store Map in memory
        USA             Save current User Map in memory for later

        LDA IDADR       Access IDADR to determine if User or System
                        Map used
        SZA,RSS         System Map used?
        JMP SYACC       Yes, System Map used - access buffer directly
*                       No, User Map used
        JSB $PVMP       Call $PVMP to set up User Map for this program
        SZA,RSS         Check for error return
        JMP ERROR       Error exists, go handle it

            .
            .           No errors.  System Map is still enabled.
            .           Access buffer via a series of cross-map
            .           loads and stores since SDA drivers are not
            .           included in all User maps.
            .
        LDA MAPAD       Access address of User Map storage area
        USA             Restore original contents of User Map

            .
            .           Proceed with normal processing under System Map
            .

MAPAD DEF MAP           Address of User Map storage area
MAP   BSS 32            User Map storage area
SIGN  OCT 100000
IDADR BSS 1             ID segment address saved by initiation section
```

Remember that any driver using this routine must save the original contents of the User Map registers before calling $PVMP and must restore the registers to their original value after all acceses to the buffer have been made. The example above illustrates this procedure.

Sample Privileged Drivers

The sample drivers illustrated in Figures 4-1 and 4-2 demonstrate some of the principles involved in writing a privileged I/O driver for use in an RTE system with Dynamic Mapping (Figure 4-1) and without Dynamic Mapping (Figure 4-2). Note that the drivers are examples only and are not drivers supplied with the system.

```
               ASMB,Q
*
                     NAM DVYNN       **DMS PRIVILEGED DRIVER EXAMPLE**
                     SUP
*
                     ENT IXNN,CXNN
*
*************************************************************
* SAMPLE RTE PRIVILEGED DRIVER DVYNN - FOR DMS SYSTEMS *
*************************************************************
*
* HANDLES USER PROGRAM REQUESTS RO READ FROM A PRIVILEGED
* CONTROLLER
*
* USER PROGRAM CALLING SEQUENCE:
*
*   JSB EXEC       CALL EXEC
*   DEF *+5        RETURN POINT
*   DEF RCODE      REQUEST CODE (MUST BE READ REQUEST)
*   DEF CONWD      CONTROL WORD
*   DEF BUFFR      ADDRESS OF BUFFER (MUST BE IN SYSTEM COMMON)
*   DEF LENTH      LENGTH OF BUFFER
*
* CAUTION:
*
* THIS DRIVER WILL NOT WORK WITH MORE THAN ONE PRIVILEGED
* CONTROLLER.  IF MORE THAN ONE PRIVILEGED CONTROLLER
* EXISTS IN A SYSTEM, DVYNN MUST BE
* RE-ASSEMBLED WITH ALL NAMES CONTAINING "NN" CHANGED SO
* THAT EACH COPY OF THE DRIVER HAS UNIQUE ENTRY POINTS.
* THEN ONE DRIVER PER CONTROLLER MUST BE PUT
* INTO THE SYSTEM AT GENERATION TIME.
*
```

Figure 4-1. DMS Privileged RTE Driver Example

```
* NOTE:
*
* 1.) THE DESIGN OF THIS DRIVER ASSUMES THAT THE I/O
*     BUFFER BEING PROCESSED IS LOCATED IN SYSTEM COMMON.
*     CONSEQUENTLY, THE DRIVER IS ENTERED WITH THE
*     SYSTEM MAP ENABLED.  THIS IS NECESSARY FOR THE
*     CORRECT OPERATION OF THE TRAP CELL MODIFICATION
*     TECHNIQUE ILLUSTRATED BELOW.  IN ADDITION, THE
*     BUFFER IN SYSTEM COMMON ALLOWS THE DRIVER TO PUT THE
*     DATA VALUES DIRECTLY INTO THE BUFFER, WITHOUT
*     THE NEED FOR MAP SWITCHING
*
* 2.) THIS DRIVER DOES NOT PROCESS POWER-FAIL INTERRUPTS.
*
* 3.) THIS DRIVER DOES NOT PROCESS ANY TIME-OUTS EXCEPT
*     FOR THE TIME-OUT THAT IT CREATES AS A MEANS TO
*     COMPLETE THE I/O REQUEST AND RETURN TO IOC
*
*   **********************
*   * INITIATION SECTION *
*   **********************
*
  00000 000000  IXNN  NOP           INITIATION SECTION ENTRY POINT
  00001 000201R       STA SCODE     SAVE SELECT CODE OF CONTROLLER
*
  00002 000204R       LDB FIRST     ACCESS FIRST TIME THROUGH FLAG
  00003 006002        SZB           IS THIS THE FIRST TIME THRU?
  00004 000020R       JMP INIT      NO, SO SKIP CONFIGURATION CODE
*
* CONFIGURE I/O INSTRUCTIONS
*
  00005 000220R       IOR LIA       CREATE LIA INSTRUCTION
*
* MODIFY TRAP CELL
*
  00006 000221R       LDA $JSB      SET TRAP CELL TO
  00007 000201R       STA SCODE,I   JSB $JPNN,I ($JPNN=ADDR OF PXNN)
*
* SAVE EQT ADDRESSES
*
  00010 001774        LDA EQT15     SAVE EQT15
  00011 000216R       STA EQ15
  00012 001663        LDA EQT4      EQT4
  00013 000215R       STA EQ4
  00014 001660        LDA EQT1      AND EQT1
  00015 000214R       STA EQ1       ADDRESSES
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
*
  00016 002404         CLA,INA        SET FLAG TO PREVENT CONFIGURING ON
  00017 000204R        STA FIRST      SUBSEQUENT INITIATIONS
*
* CLEAR THE "DRIVER PROCESSES TIME-OUT" BIT TO ALLOW
* NORMAL TIME-OUT OPERATION
*
  00020 001663  INIT   LDA EQT4,I     ACCESS EQT WORD 4
  00021 000227R        AND =B167777   CLEAR BIT 12
  00022 001663         STA EQT4,I     AND RESET EQT WORD 4
*
* CHECK THE REQUEST CODE
*
  00023 001665         LDA EQT6,I     ACCESS REQUEST CODE
  00024 000224R        AND =B3        ISOLATE REQUEST TYPE
  00025 000222R        CPA =B1        READ REQUEST?
  00026 000041R        JMP PROC       YES, GO PROCESS READ REQUEST
*
  00027 000224R        CPA =B3        CONTROL REQUEST?
  00030 000033R        JMP CNTRL      YES, GO PROCESS CONTROL REQUEST
*
  00031 002404         CLA,INA        NO, REJECT AS ILLEGAL WRITE REQUEST
  00032 000000R        JMP IXNN,I
*
* CONTROL REQUEST. CHECK IF IT IS A "CLEAR" CONTROL REQUEST
* IF SO, ASSUME IT WAS ISSUED BY SYSTEM, CLEAR DEVICE, AND RETURN.
*
  00033 001665  CNTRL  LDA EQT6,I     ACCESS CONTROL WORD
  00034 000225R        AND =B3700     ISOLATE SUBFUNCTION
  00035 002002         SZA            "CLEAR" REQUEST?
  00036 000037R        JMP REJCT      NO, REJECT AS ILLEGAL CONTRL REQUEST
*        .
*        .           EXECUTE CODE TO CLEAR CONTROLLER
*        .
  00037 000223R REJCT  LDA =B2        REJECT AS ILLEGAL CONTROL REQUEST
  00040 000000R        JMP IXNN,I
*
* SET UP FOR THE DATA TRANSFER
*
  00041 001667  PROC   LDA EQT8,I     ACCESS $ OF CONVERSIONS REQUIRED
  00042 003004         CMA,INA        NEGATE FOR CONVERSION COUNTER
  00043 000202R        STA CVCTR      AND SAVE
  00044 002021         SSA,RSS        REJECT IF
  00045 000037R        JMP REJCT      NUMBER <0
  00046 001666         LDA EQT7,I     SAVE DATA BUFFER ADDRESS
  00047 000203R        STA DAPTR      FOR PXNN
*
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
* INITIATE A READ AND RETURN
*
  00050 000054R        JSB READ      START A READ
  00051 103700  I.1    STC SC,C      ENCODE DEVICE
  00052 002400         CLA
  00053 000000R        JMP IXNN,I    RETURN TO IOC
*
* SUBROUTINE TO INITIATE A READ
*
  00054 000000  READ   NOP           ROUTINE CONTAINING
*         .            CONFIGURED I/O
*         .            INSTRUCTION TO
*         .            SET UP THE DEVICE
*         .            TO INITIATE ONE READING
  00055 000054R          JMP READ,I
*
*   **********************
*   * PRIVILEGED SECTION *
*   **********************
*
* SAVE STATE OF COMPUTER AT INTERRUPT
*
  00056 000000  PXNN   NOP           PRIVILEGED SECTION ENTRY POINT
*
  00057 103100         CLF 0         TURN OFF INTERRUPT SYSTEM
*
  00060 106706         CLC 6         TURN OFF DCPC COMPLETION INTERRUPTS
  00061 106707         CLC 7
*
  00062 000205R        STA ASV       SAVE REGISTERS
  00063 000206R        STB BSV
  00064 001520         ERA,ALS
  00065 102201         SOC
  00066 002004         INA
  00067 000207R        STA EOSV
  00070 105743         STX XSV       SAVE X REGISTER
  00072 105753         STY YSV       SAVE Y REGISTER
  00074 105714         SSM DMSTS     SAVE DYNAMIC MAPPING SYSTEM STATUS
*
  00076 001770         LDA MPTFL     SAVE OLD MEMORY PROTECT FLAG
  00077 000213R        STA MPFSV
  00100 002404         CLA,INA       SET MEMORY PROTECT FLAG TO OFF
  00101 001770         STA MPTFL     SINCE MEMORY PROTECT IS NOW OFF
*
  00102 102100         STF 0         TURN INTERRUPT SYSTEM BACK ON
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
*
* CHECK FOR SPURIOUS INTERRUPT
*
  00103 000214R        LDA EQ1,I     ACCESS REQUEST LIST POINTER WORD
  00104 000226R        AND =B77777   ISOLATE REQUEST LIST POINTER
  00105 002002         SZA           IS A REQUEST IN PROGRESS?
  00106 000112R        JMP PREAD     YES, GO PROCESS INTERRUPT
*
  00107 103100         CLF 0         NO, TURN OFF INTERRUPT SYSTEM
  00110 107700  I.2    CLC SC C      RESET CONTROLLER, AND
  00111 000122R        JMP EXIT      IGNORE SPURIOUS INTRUPT BY RETURNING
*
* PROCESS READ REQUEST
*
        000112R PREAD EQU *
*         .          LOAD IN DATA FROM DEVICE
*         .          VIA CONFIGURED I/O INSTRUCTIONS
*         .
*
  00112 000203R        STA DAPTR,I   STORE WORD IN DATA BUFFER
  00113 000202R        ISZ CVCTR     IS THIS THE LAST CONVERSION?
  00114 002001         RSS           NO
  00115 000165R        JMP DONE      YES, GO SET UP TO TERMINATE CALL
*
  00116 000203R        ISZ DAPTR     NO, SET UP FOR NEXT CONVERSION
  00117 000054R        JSB READ      INITIATE IT
*
* RESTORE MACHINE TO ORIGINAL STATE ON INTERRUPT
*
  00120 103100         CLF 0         TURN OFF INTRUPT SYSTEM TEMPORARILY
*
  00121 103700  I.3    STC SC,C      ENCODE DEVICE
*
  00122 000213R EXIT   LDA MPFSV     ACCESS PREVIOUS STATE OF MEM PROTECT
  00123 002002         SZA           WAS MEMORY PROTECT ON?
  00124 000135R        JMP EXIT1     NO, DO NOT TURN ON DCPC INTERRUPTS
*
  00125 001654         LDB INTBA     YES, TURN DCPC COMPLETION INTERRUPTS
  00126 000001         LDA B,I       BACK ON IF THEY WERE ON INITIALLY.
  00127 002020         SSA           ON/OFF STATUS IS INDICATED BY BIT 15
  00130 102706         STC 6         OF EACH DCPC ASSIGNMENT WORD IN THE
  00131 006004         INB           INTERRUPT TABLE
  00132 000001         LDA B,I
  00133 002020         SSA
  00134 102707         STC 7
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
*
  00135 000207R EXIT1 LDA EOSV      RESTORE E AND O REGISTERS
  00136 103101        CLO
  00137 000036        SLA,ELA
  00140 102101        STF 1
  00141 000206R       LDB BSV       RESTORE B-REGISTER
  00142 105745        LDX XSV       RESTORE X REGISTER
  00144 105755        LDY YSV       RESTORE Y REGISTER
*
  00146 000213R       LDA MPFSV     RESTORE MEMORY PROTECT FLAG
  00147 001770        STA MPTFL     IN BASE PAGE
  00150 002002        SZA           WAS MEMORY PROTECT ON AT INTERRUPT?
  00151 000160R       JMP EXIT2     NO
*
  00152 000205R       LDA ASV       YES, RESTORE A-REGISTER
  00153 102100        STF 0         TURN ON INTERRUPT SYSTEM
  00154 102705        STC 5         SET MEMORY PROTECT ON
  00155 105715        JRS DMSTS PXNN,I     RESTORE DMS STATUS AND RETURN
*
*                          (NOTE: EXECUTION OF A "JRS"
*                          INSTRUCTION AFTER TURNING THE
*                          MEMORY PROTECT FENCE ON IS
*                          ALLOWED ONLY IF SYSTEM MAP
*                          IS CURRENTLY ENABLED.  THIS
*                          DRIVER HAS BEEN DESIGNED SUCH
*                          THAT THIS IS ALWAYS THE CASE.
*
  00160 000205R EXIT2 LDA ASV       NO, RESTORE A-REGISTER
  00161 102100        STF 0         TURN ON INTERRUPTS
  00162 105715        JRS DMSTS PXNN,I     RESTORE DMS STATUS AND RETURN
*
* THIS CODE SETS UP THE TIME OUT TO COMPLETE THE CALL
*
  00165 103100  DONE  CLF 0         TURN OFF THE INTERRUPT SYSTEM
  00166 106700  I.4   CLC SC        TURN OFF PRIVILEGED DEVICE
  00167 003400        CCA           SET TIME OUT FOR
  00170 000216R       STA EQ15,I    ONE TICK AND SET
  00171 000215R       LDA EQ4,I     BIT12 IN EQT4 SO
  00172 000217R       IOR BIT12     RTIOC WILL
  00173 000215R       STA EQ4,I     CALL CXNN ON TIME-OUT
  00174 000122R       JMP EXIT      GO TO EXIT ROUTINE
*
*   **********************
*   * COMPLETION SECTION *
*   **********************
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
*
  00175 000000  CXNN  NOP           COMPLETION SECTION ENTRY POINT
*
  00176 002400        CLA           SET A = 0 = NORMAL RETURN
  00177 001667        LDB EQT8,I    SET B = TRANSMISSION LOG
  00200 000175R       JMP CXNN,I    RETURN TO IOC
*
* CONSTANT AND STORAGE AREA
*
        000000  A     EQU 0
        000001  B     EQU 1
        000000  SC    EQU 0         DUMMY I/O SELECT CODE NUMBER
*
  00201         SCODE BSS 1
  00202         CVCTR BSS 1
  00203         DAPTR BSS 1
  00204         FIRST BSS 1
  00205         ASV   BSS 1
  00206         BSV   BSS 1
  00207         EOSV  BSS 1
  00210         XSV   BSS 1
  00211         YSV   BSS 1
  00212         DMSTS BSS 1
  00213         MPFSV BSS 1
  00214         EQ1   BSS 1
  00215         EQ4   BSS 1
  00216         EQ15  BSS 1
  00217 010000  BIT12 OCT 10000
  00220 102500  LIA   LIA 0
*
* BASE PAGE COMMUNICATIONS AREA DEFINITION
*
        001650  .     EQU 1650B
        001654  INTBA EQU .+4
        001660  EQT1  EQU .+8
        001663  EQT4  EQU .+11
        001665  EQT6  EQU .+13
        001666  EQT7  EQU .+14
        001667  EQT8  EQU .+15
        001774  EQT15 EQU .+84
        001770  MPTFL EQU .+80
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
*
* CODE TO SET UP JSB $JPNN,I INSTRUCTION ON BASE PAGE
*
  00000                 ORB           RESET LOCATION COUNTER TO BASE PAGE
  00000 000056R $JPNN   DEF PXNN      PRIV. SECTION ENTRY POINT ADDR
*
  00221                 ORR           RESET LOC COUNTER TO RELOCATABLE
  00221 000000B $JSB    JSB $JPNN,I   JSB INSTR. TO PRIV SECTION, INDIRECT
*
                        END
```

Figure 4-1. DMS Privileged RTE Driver Example (Continued)

```
               ASMB,Q
*
                     NAM DVYNN  **NON-DMS PRIVILEGED DRIVER EXAMPLE **
*
                     ENT IXNN,CXNN
*
*****************************************************************
* SAMPLE RTE PRIVILEGED DRIVER DVYNN - FOR NON-DMS SYSTEMS *
*****************************************************************
*
* HANDLES USER PROGRAM REQUESTS TO READ FROM A PRIVILEGED
* CONTROLLER
*
* USER PROGRAM CALLING SEQUENCE:
*
*   JSB EXEC       CALL EXEC
*   DEF *+5        RETURN POINT
*   DEF RCODE      REQUEST CODE (MUST BE READ REQUEST)
*   DEF CONWD      CONTROL WORD
*   DEF BUFFR      ADDRESS OF BUFFER (MUST BE IN COMMON)
*   DEF LENTH      LENGTH OF BUFFER
*
* CAUTION:
*
* THIS DRIVER WILL NOT WORK WITH MORE THAN ONE PRIVILEGED
* CONTROLLER.  IF MORE THAN ONE PRIVILEGED CONTROLLER
* EXISTS IN A SYSTEM, DVYNN MUST BE
* RE-ASSEMBLED WITH ALL NAMES CONTAINING "NN" CHANGED SO
* THAT EACH COPY OF THE DRIVER HAS UNIQUE ENTRY POINTS.
* THEN ONE DRIVER PER CONTROLLER MUST BE PUT
* INTO THE SYSTEM AT GENERATION TIME.
*
* NOTE:
*
* 1.) THIS DRIVER DOES NOT PROCESS POWER-FAIL INTERRUPTS.
*
* 2.) THIS DRIVER DOES NOT PROCESS ANY TIME-OUTS EXCEPT
*     FOR THE TIME-OUT THAT IT CREATES AS A MEANS TO
*     COMPLETE THE I/O REQUEST AND RETURN TO IOC
*
*   **********************
*   * INITIATION SECTION *
*   **********************
*
  00000 000000  IXNN  NOP          INITIATION SECTION ENTRY POINT
  00001 000163R       STA SCODE    SAVE SELECT CODE OF CONTROLLER
```

Figure 4-2. Non-DMS Privileged RTE Driver Example

```
*
  00002 000166R       LDB FIRST     ACCESS FIRST TIME THROUGH FLAG
  00003 006002        SZB           IS THIS THE FIRST TIME THRU?
  00004 000020R       JMP INIT      NO, SO SKIP CONFIGURATION CODE
*
* CONFIGURE I/O INSTRUCTIONS
*
  00005 000177R       IOR LIA       CREATE LIA INSTRUCTION
*
* MODIFY TRAP CELL
*
  00006 000200R       LDA $JSB      SET TRAP CELL TO
  00007 000163R       STA SCODE,I   JSB $JPNN,I ($JPNN=ADDR OF PXNN)
*
* SAVE EQT ADDRESSES
*
  00010 001774        LDA EQT15     SAVE EQT15
  00011 000175R       STA EQ15
  00012 001663        LDA EQT4      EQT4
  00013 000174R       STA EQ4
  00014 000173R       LDA EQ1       AND EQT1
  00015 000173R       STA EQ1       ADDRESSES
*
  00016 002404        CLA,INA       SET FLAG TO PREVENT CONFIGURATING ON
  00017 000166R       STA FIRST     SUBSEQUENT INITIATIONS
*
* CLEAR THE "DRIVER PROCESSES TIME-OUT" BIT TO ALLOW
* NORMAL TIME-OUT OPERATION
*
  00020 001663  INIT  LDA EQT4,I    ACCESS EQT WORD 4
  00021 000206R       AND =B167777  CLEAR BIT 12
  00022 001663        STA EQT4,I    AND RESET EQT WORD 4
*
* CHECK THE REQUEST CODE
*
  00023 001665        LDA EQT6,I    ACCESS REQUEST CODE
  00024 000203R       AND =B3       ISOLATE REQUEST TYPE
  00025 000201R       CPA =B1       READ REQUEST?
  00026 000041R       JMP PROC      YES, GO PROCESS READ REQUEST
*
  00027 000203R       CPA =B3       CONTROL REQUEST?
  00030 000033R       JMP CNTRL     YES, GO PROCESS CONTROL REQUEST
*
  00031 002404        CLA,INA       NO, REJECT AS ILLEGAL WRITE REQUEST
  00032 000000R       JMP IXNN,I
*
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

```
* CONTROL REQUEST. CHECK IF IT IS A "CLEAR" CONTROL REQUEST
* IF SO, ASSUME IT WAS ISSUED BY SYSTEM, CLEAR DEVICE, AND RETURN.
*
  00033 001665  CNTRL LDA EQT6,I    ACCESS CONTROL WORD
  00034 000204R       AND =B3700    ISOLATE SUBFUNCTION
  00035 002002        SZA           "CLEAR" REQUEST?
  00036 000037R       JMP REJCT     NO, REJECT AS ILLEGAL CNTRL REQUEST
*       .
*       .          EXECUTE CODE TO CLEAR CONTROLLER
*       .
*       .
  00037 000202R REJCT LDA =B2       REJECT AS ILLEGAL CONTROL REQUEST
  00040 000000R       JMP IXNN,I
*
* SET UP FOR THE DATA TRANSFER
*
  00041 001667  PROC  LDA EQT8,I    ACCESS $ OF CONVERSIONS REQUIRED
  00042 003004        CMA,INA       NEGATE FOR CONVERSION COUNTER
  00043 000164R       STA CVCTR     AND SAVE
  00044 002021        SSA,RSS       REJECT IF
  00045 000037R       JMP REJCT     NUMBER <0
  00046 001666        LDA EQT7,I    SAVE DATA BUFFER ADDRESS
  00047 000165R       STA DAPTR     FOR PXNN
*
* INITIATE A READ AND RETURN
*
  00050 000054R       JSB READ      START A READ
  00051 103700  I.1   STC SC,C      ENCODE DEVICE
  00052 002400        CLA
  00053 000000R       JMP IXNN,I    RETURN TO IOC
*
* SUBROUTINE TO INITIATE A READ
*
  00054 000000  READ  NOP           ROUTINE CONTAINING
*       .          CONFIGURED I/O
*       .          INSTRUCTIONS TO
*       .          SET UP THE DEVICE
*       .          TO INITIATE ONE READING
  00055 000054R       JMP READ,I
*
*   **********************
*   * PRIVILEGED SECTION *
*   **********************
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

```
*
* SAVE STATE OF COMPUTER AT INTERRUPT
*
  00056 000000  PXNN  NOP           PRIVILEGED SECTION ENTRY POINT
*
  00057 103100        CLF 0         TURN OFF INTERRUPT SYSTEM
*
  00060 106706        CLC 6         TURN OFF DCPC COMPLETION INTERRUPTS
  00061 106707        CLC 7
*
  00062 000167R       STA ASV       SAVE REGISTERS
  00063 000170R       STB BSV
  00064 001520        ERA,ALS
  00065 102201        SOC
  00066 002004        INA
  00067 000171R       STA EOSY
*
  00070 001770        LDA MPTFL     SAVE OLD MEMORY PROTECT FLAG
  00071 000172R       STA MPFSV
  00072 002404        CLA,INA       SET MEMORY PROTECT FLAG TO OFF,
  00073 001770        STA MPTFL     SINCE MEMORY PROTECT IS NOW OFF
*
  00074 102100        STF 0         TURN INTERRUPT SYSTEM BACK ON
*
* CHECK FOR SPURIOUS INTERRUPT
*
  00075 000173R       LDA EQ1,I     ACCESS REQUEST LIST POINTER WORD
  00076 000205R       AND =B77777   ISOLATE REQUEST LIST POINTER
  00077 002002        SZA           IS A REQUEST IN PROGRESS?
  00100 000104R       JMP PREAD     YES, GO PROCESS INTERRUPT
*
  00101 103100        CLF 0         NO, TURN OFF INTERRUPT SYSTEM
  00102 107700  I.2   CLC SC,C      RESET CONTROLLER, AND
  00103 000114R       JMP EXIT      IGNORE SPURIOUS INTRUPT BY RETURNING
*
* PROCESS READ REQUEST
*
        000104R PREAD EQU *
*         .         LOAD IN DATA FROM DEVICE
*         .         VIA CONFIGURED I/O INSTRUCTIONS
*         .
  00104 000165R       STA DAPTR,I   STORE WORD IN DATA BUFFER
  00105 000164R       ISZ CVCTR     IS THIS THE LAST CONVERSION?
  00106 002001        RSS           NO
  00107 000147R       JMP DONE      YES, GO SET UP TO TERMINATE CALL
*
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

```
  00110 000165R        ISZ DAPTR     NO, SET UP FOR NEXT CONVERSION
  00111 000054R        JSB READ      INITIATE IT
*
* RESTORE MACHINE TO ORIGINAL STATE ON INTERRUPT
*
  00112 103100         CLF 0         TURN OFF INTRUPT SYSTEM TEMPORARILY
*
  00113 103700  I.3    STC SC,C      ENCODE DEVICE
*
  00114 000172R EXIT   LDA MPFSV     ACCESS PREVIOUS STATE OF MEM PROTECT
  00115 002002         SZA           WAS MEMORY PROTECT ON?
  00116 000127R        JMP EXIT1     NO, DO NOT TURN ON DCPC INTERRUPTS
*
  00117 001654         LDB INTBA     YES, TURN DCPC COMPLETION INTERRUPTS
  00120 000001         LDA B,I       BACK ON IF THEY WERE ON INITIALLY.
  00121 002020         SSA           ON/OFF STATUS IS INDICATED BY BIT 15
  00122 102706         STC 6         OF EACH DCPC ASSIGNMENT WORD IN THE
  00123 006004         INB           INTERRUPT TABLE
  00124 000001         LDA B,I
  00125 002020         SSA
  00126 102707         STC 7
*
  00127 000171R EXIT1  LDA EOSY      RESTORE E AND O-REGISTERS
  00130 103101         CLO
  00131 000036         SLA,ELA
  00132 102101         STF 1
  00133 000170R        LDB BSV       RESTORE B-REGISTER
*
  00134 000172R        LDA MPFSV     RESTORE MEMORY PROTECT FLAG
  00135 001770         STA MPTFL     IN BASE PAGE
  00136 002002         SZA           WAS MEMORY PROTECT ON A1 INTERRUPT?
  00137 000144R        JMP EXIT2     NO
*
  00140 000167R        LDA ASV       YES, RESTORE A-REGISTER
  00141 102100         STF 0         TURN ON INTERRUPT SYSTEM
  00142 102705         STC 5         SET MEMORY PROTECT ON
  00143 000056R        JMP PXNN,I    RETURN TO POINT OF INTERRUPTION
*
  00144 000167R EXIT2  LDA ASV       NO, RESTORE A-REGISTER
  00145 102100         STF 0         TURN ON INTERRUPT SYSTEM
  00146 000056R        JMP PXNN,I    RETURN TO POINT OF INTERRUPTION
*
* THIS CODE SETS UP THE TIME OUT TO COMPLETE THE CALL
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

```
*
  00147 103100  DONE  CLF 0          TURN OFF THE INTERRUPT SYSTEM
  00150 106700        CLC SC         TURN OFF PRIVILEGED DEVICE
  00151 003400        CCA            SET TIME OUT FOR
  00152 000175R       STA EQ15,I     ONE TICK AND SET
  00153 000174R       LDA EQ4,I      BIT12 IN EQT4 SO
  00154 000176R       IOR BIT12      RTIOC WILL
  00155 000174R       STA EQ4,I      CALL CXNN ON TIME-OUT
  00156 000114R       JMP EXIT       GO TO EXIT ROUTINE
*
*   **********************
*   * COMPLETION SECTION *
*   **********************
*
  00157 000000  CXNN  NOP            COMPLETION SECTION ENTRY POINT
*
  00160 002400        CLA            NO, SET A = 0 = NORMAL POINT
  00161 001667        LDB EQT8,I     SET B = TRANSMISSION LOG
  00162 000157R       JMP CXNN,I     MAKE COMPLETION RETURN (P+1) TO IOC
*
* CONSTANT AND STORAGE AREA
*
        000000  A     EQU 0
        000001  B     EQU 1
        000000  SC    EQU 0          DUMMY I/O SELECT CODE NUMBER
*
  00163         SCODE BSS 1
  00164         CVCTR BSS 1
  00165         DAPTR BSS 1
  00166         FIRST BSS 1
  00167         ASV   BSS 1
  00170         BSV   BSS 1
  00171         EOSY  BSS 1
  00172         MPFSV BSS 1
  00173         EQ1   BSS 1
  00174         EQ4   BSS 1
  00175         EQ15  BSS 1
  00176 010000  BIT12 OCT 10000
  00177 102500  LIA   LIA 0
*
* BASE PAGE COMMUNICATIONS AREA DEFINITION
*
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

```
        001650  .      EQU 1650B
        001654  INTBA  EQU .+4
        001660  EQT1   EQU .+8
        001663  EQT4   EQU .+11
        001665  EQT6   EQU .+13
        001666  EQT7   EQU .+14
        001667  EQT8   EQU .+15
        001774  EQT15  EQU .+84
        001770  MPTFL  EQU .+80
*
* CODE TO SET UP JSB $JPNN,I INSTRUCTION ON BASE PAGE
*
  00000                ORB           RESET LOCATION COUNTER TO BASE PAGE
  00000 000056R $JPNN  DEF PXNN      PRIV. SECTION ENTRY POINT ADDR
*
  00200                ORR           RESET LOC COUNTER TO RELOCATABLE
  00200 000000B $JSB   JSB $JPNN,I   JSB INSTR. TO PRIV SECTION, INDIRECT
*
  00201 000001
  00202 000002
  00203 000003
  00204 003700
  00205 077777
  00206 167777
                       END
```

Figure 4-2. Non-DMS Privileged RTE Driver Example (Continued)

Index

$JPNN, 4-3
$LIST: program scheduling, 3-30
$OPSY: operating system environment, 3-32
$SUBC, 3-39
$XDMP, 3-34

A

AUTOR :, 3-29

B

base page communications area, 2-3

C

CHAN: base page word, 3-3
CIC: Central Interrupt Control module, 2-2
Communication with User Programs, 4-12
Completion Section, 4-14
Continuation/Completion Section, 3-9
controller, 2-1

D

DCPC: Dual Channel Port Controller, 3-17
device, 2-1
Device Clear on Program Abort, 3-13
DMS: Dynamic Mapping System, 3-33
DMT: Driver Mapping Table, 2-16
driver, 2-2
   Mapping Table, 2-16
   naming requirements, 3-2
   privileged, 4-2
   Processing of Time-Out, 3-15
   standard, 2-22
driver structure and operation
   general, 3-1
   privileged, 4-3
DRT: Device Reference Table, 2-9

E

Environment, 3-32
EQT: Equipment Table, 2-5
   driver initiation, 3-2
EXEC call, 2-2

I

I/O
   Completion, 2-22
   Continuation, 2-21
   Initiation, 2-21
   processor, 2-2
Initiation Section, 4-13
initiation section standard driver, 3-2
interrupt mechanism, 2-12
IOC: Input/Output Control module, 2-2

L

LINK: base page link, 2-12
LU: Logical Unit Numbers, 2-8

M

Mapping
   in RTE-III and RTE-M/III, 3-35, 4-16
   in RTE-IV and RTE-6/VM, 3-36, 4-19
   Table, 2-16

P

power-fail, 3-4
   Processing, 3-26
privileged
   driver, 4-2
   Driver Design Considerations, 4-11
   Interrupt card, 4-2
   Section, 4-14
Program Scheduling by Drivers, 3-29

S

SDA: System Driver Area, 2-16
select code, 2-12
standard driver, 2-22, 3-1
Subroutines for Special Mapping Functions, 3-33, 4-15
synchronous device, 2-1
System
   Map, 3-33
   Processing of Time-Out, 3-16

T

Time-Out Values for Privileged drivers, 4-15
trap cell, 2-12

U

User Map, 3-33

READER COMMENT SHEET

RTE Operating System
Driver Writing Manual

92200-93005 **December 1983**

Update No._____
(If Applicable)

We welcome your evaluation of this manual. Your comments and suggestions help us improve our publications. Please use additional pages if necessary. If you have technical questions or if you want to order publications, make your inquiries to a representative of your local HP office. Thank you for your comments! No postage necessary if mailed in the U.S.A.

FROM:

Name _____

Company _____

Address _____

_____

Phone No. _____ **Ext.** _____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 141 CUPERTINO, CA.

— POSTAGE WILL BE PAID BY —

Hewlett-Packard Company
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014
ATTN: Technical Publications

FOLD FOLD

SALES & SUPPORT OFFICES

Arranged alphabetically by country

Product Line Sales/Support Key

Key Product Line
A Analytical
CM Components
C Computer Systems Sales only
CH Computer Systems Hardware Sales and Services
CS Computer Systems Software Sales and Services
E Electronic Instruments & Measurement Systems
M Medical Products
MP Medical Products Primary SRO
MS Medical Products Secondary SRO
P Personal Computation Products
*** Sales only for specific product line**
**** Support only for specific product line**

IMPORTANT: These symbols designate general product line capability. They do not insure sales or support availability for all products within a line, at all locations. Contact your local sales office for information regarding locations where HP support is available for specific products.

HP distributors are printed in italics.

HEADQUARTERS OFFICES

If there is no sales office listed for your area, contact one of these headquarters offices.

NORTH/CENTRAL AFRICA

Hewlett-Packard S.A.
7, Rue du Bois-du-Lan
CH-1217 **MEYRIN** 2, Switzerland
Tel: (022) 83 12 12
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

ASIA

Hewlett-Packard Asia Ltd.
6th Floor, Sun Hung Kai Centre
30 Harbour Rd.
G.P.O. Box 795
HONG KONG
Tel: 5-832 3211
After Jan. 1, 1984
47th Floor, China Resources Bldg.
26 Harbour Rd., Wanchai
HONG KONG
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG

CANADA

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
Telex: 610-492-4246

EASTERN EUROPE

Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O.Box 72
A-1222 **VIENNA,** Austria
Tel: (222) 2365110
Telex: 1 3 4425 HEPA A

NORTHERN EUROPE

Hewlett-Packard S.A.
Uilenstede 475
P.O.Box 999
NL-1180 AZ **AMSTELVEEN**
The Netherlands
Tel: 20 437771

SOUTH EAST EUROPE

Hewlett-Packard S.A.
7, Rue du Bois-du-Lan
CH-1217 **MEYRIN** 2, Switzerland
Tel: (022) 83 12 12
Telex: 27835 hpse
Cable: HEWPACKSA Geneve

OTHER EUROPE

Hewlett-Packard S.A.
P.O. Box
150, Rte du Nant-D'Avril
CH-1217 **MEYRIN** 2, Switzerland
Tel: (022) 83 8111
Telex: 22486 hpsa
Cable: HEWPACKSA Geneve

MEDITERRANEAN AND MIDDLE EAST

Hewlett-Packard S.A.
Mediterranean and Middle East Operations
Atrina Centre
32 Kifissias Ave.
Paradissos-Amarousion, **ATHENS**
Greece
Tel: 682 88 11
Telex: 21-6588 HPAT GR
Cable: HEWPACKSA Athens

EASTERN USA

Hewlett-Packard Co.
4 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 258-2000

MIDWESTERN USA

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800

SOUTHERN USA

Hewlett-Packard Co.
2000 South Park Place
P.O. Box 105005
ATLANTA, GA 30348
Tel: (404) 955-1500

WESTERN USA

Hewlett-Packard Co.
3939 Lankershim Blvd.
P.O. Box 3919
LOS ANGELES, CA 91604
Tel: (213) 506-3700

OTHER INTERNATIONAL AREAS

Hewlett-Packard Co.
Intercontinental Headquarters
3495 Deer Creek Road
PALO ALTO, CA 94304
Tel: (415) 857-1501
Telex: 034-8300
Cable: HEWPACK

ANGOLA

Telectra
Empresa Técnica de Equipamentos
R. Barbosa Rodrigues, 41-I DT.
Caixa Postal 6487
LUANDA
Tel: 35515,35516
E,P

ARGENTINA

Hewlett-Packard Argentina S.A.
Avenida Santa Fe 2035
Martinez 1640 **BUENOS AIRES**
Tel: 798-5735, 792-1293
Telex: 17595 BIONAR
Cable: HEWPACKARG
A,E,CH,CS,P

Biotron S.A.C.I.M. e I.
Av Paseo Colon 221, Piso 9
*1399 **BUENOS AIRES***
Tel: 30-4846, 30-1851
Telex: 17595 BIONAR
M

AUSTRALIA

Adelaide, South Australia Office
Hewlett-Packard Australia Ltd.
153 Greenhill Road
PARKSIDE, S.A. 5063
Tel: 272-5911
Telex: 82536
Cable: HEWPARD Adelaide
A*,CH,CM,,E,MS,P

Brisbane, Queensland Office
Hewlett-Packard Australia Ltd.
10 Payne Road
THE GAP, Queensland 4061
Tel: 30-4133
Telex: 42133
Cable: HEWPARD Brisbane
A,CH,CM,E,M,P

Canberra, Australia Capital Territory Office
Hewlett-Packard Australia Ltd.
121 Wollongong Street
FYSHWICK, A.C.T. 2609
Tel: 80 4244
Telex: 62650
Cable: HEWPARD Canberra
CH,CM,E,P

Melbourne, Victoria Office
Hewlett-Packard Australia Ltd.
31-41 Joseph Street
BLACKBURN, Victoria 3130
Tel: 895-2895
Telex: 31-024
Cable: HEWPARD Melbourne
A,CH,CM,CS,E,MS,P

Perth, Western Australia Office
Hewlett-Packard Australia Ltd.
261 Stirling Highway
CLAREMONT, W.A. 6010
Tel: 383-2188
Telex: 93859
Cable: HEWPARD Perth
A,CH,CM,E,MS,P

Sydney, New South Wales Office
Hewlett-Packard Australia Ltd.
17-23 Talavera Road
P.O. Box 308
NORTH RYDE, N.S.W. 2113
Tel: 887-1611
Telex: 21561
Cable: HEWPARD Sydney
A,CH,CM,CS,E,MS,P

AUSTRIA

Hewlett-Packard Ges.m.b.h.
Grottenhofstrasse 94
A-8052 **GRAZ**
Tel: (0316) 291 5 66
Telex: 32375
CH,E

Hewlett-Packard Ges.m.b.h.
Lieblgasse 1
P.O. Box 72
A-1222 **VIENNA**
Tel: (0222) 23 65 11-0
Telex: 134425 HEPA A
A,CH,CM,CS,E,MS,P

BAHRAIN

Green Salon
P.O. Box 557
Manama
BAHRAIN
Tel: 255503-255950
Telex: 84419
P

Wael Pharmacy
P.O. Box 648
BAHRAIN
Tel: 256123
Telex: 8550 WAEL BN
E,C,M

BELGIUM

Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

BRAZIL

Hewlett-Packard do Brasil I.e.C. Ltda.
Alameda Rio Negro, 750
Alphaville
06400 **BARUERI SP**
Tel: (011) 421.1311
Telex: (011) 33872 HPBR-BR
Cable: HEWPACK Sao Paulo
A,CH,CM,CS,E,M,P

Hewlett-Packard do Brasil I.e.C. Ltda.
Avenida Epitacio Pessoa, 4664
22471 **RIO DE JANEIRO**-RJ
Tel: (02l) 286.0237
Telex: 021-21905 HPBR-BR
Cable: HEWPACK Rio de Janeiro
A,CH,CM,E,MS,P*

ANAMED I.C.E.I. Ltda.
Rua Bage, 103
*04012 **SAO PAULO***
Tel: (011) 570-5726
Telex: 021-21905 HPBR-BR
M

SALES & SUPPORT OFFICES

Arranged alphabetically by country

CANADA

Alberta
Hewlett-Packard (Canada) Ltd.
3030 3rd Avenue N.E.
CALGARY, Alberta T2A 6T7
Tel: (403) 235-3100
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
11120A-178th Street
EDMONTON, Alberta T5S 1P2
Tel: (403) 486-6666
A,CH,CM,CS,E,MS,P

British Columbia
Hewlett-Packard (Canada) Ltd.
10691 Shellbridge Way
RICHMOND,
British Columbia V6X 2W7
Tel: (604) 270-2277
Telex: 610-922-5059
A,CH,CM,CS,E*,MS,P*

Manitoba
Hewlett-Packard (Canada) Ltd.
380-550 Century Street
WINNIPEG, Manitoba R3H 0Y1
Tel: (204) 786-6701
A,CH,CM,E,MS,P*

Nova Scotia
Hewlett-Packard (Canada) Ltd.
P.O. Box 931
900 Windmill Road
DARTMOUTH, Nova Scotia B2Y 3Z6
Tel: (902) 469-7820
CH,CM,CS,E*,MS,P*

Ontario
Hewlett-Packard (Canada) Ltd.
3325 N. Service Rd., Unit 6
BURLINGTON, Ontario P3A 2A3
Tel: (416) 335-8644
CS,M*

Hewlett-Packard (Canada) Ltd.
552 Newbold Street
LONDON, Ontario N6E 2S5
Tel: (519) 686-9181
A,CH,CM,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
6877 Goreway Drive
MISSISSAUGA, Ontario L4V 1M8
Tel: (416) 678-9430
A,CH,CM,CS,E,MP,P

Hewlett-Packard (Canada) Ltd.
2670 Queensview Dr.
OTTAWA, Ontario K2B 8K1
Tel: (613) 820-6483
A,CH,CM,CS,E*,MS,P*

Hewlett-Packard (Canada) Ltd.
220 Yorkland Blvd., Unit #11
WILLOWDALE, Ontario M2J 1R5
Tel: (416) 499-9333
CH

Quebec
Hewlett-Packard (Canada) Ltd.
17500 South Service Road
Trans-Canada Highway
KIRKLAND, Quebec H9J 2M5
Tel: (514) 697-4232
A,CH,CM,CS,E,MP,P*

Hewlett-Packard (Canada) Ltd.
Les Galeries du Vallon
2323 Du Versont Nord
STE. FOY, Quebec G1N 4C2
Tel: (418) 687-4570
CH

CHILE

Jorge Calcagni y Cia. Ltda.
Av. Italia 634 Santiago
Casilla 16475
***SANTIAGO** 9*
Tel: 222-0222
Telex: Public Booth 440001
A,CM,E,M

Olympia (Chile) Ltda.
Av. Rodrigo de Araya 1045
Casilla 256-V
***SANTIAGO** 21*
Tel: (02) 22 55 044
Telex: 240-565 OLYMP CL
Cable: Olympiachile Santiagochile
CH,CS,P

CHINA, People's Republic of

China Hewlett-Packard Rep. Office
P.O. Box 418
1A Lane 2, Luchang St.
Beiwei Rd., Xuanwu District
BEIJING
Tel: 33-1947, 33-7426
Telex: 22601 CTSHP CN
Cable: 1920
A,CH,CM,CS,E,P

COLOMBIA

Instrumentación
H. A. Langebaek & Kier S.A.
Carrera 4A No. 52A-26
Apartado Aereo 6287
***BOGOTA** 1, D.E.*
Tel: 212-1466
Telex: 44400 INST CO
Cable: AARIS Bogota
CM,E,M

Casa Humboldt Ltda.
Carrera 14, No. 98-60
Apartado Aereo 51283
***BOGOTA** 1, D.E.*
Tel: 256-1686
Telex: 45403 CCAL CO.
A

COSTA RICA

Cientifica Costarricense S.A.
Avenida 2, Calle 5
San Pedro de Montes de Oca
Apartado 10159
SAN JOSÉ
Tel: 24-38-20, 24-08-19
Telex: 2367 GALGUR CR
CM,E,M

CYPRUS

Telerexa Ltd.
P.O. Box 4809
14C Stassinos Avenue
NICOSIA
Tel: 62698
Telex: 2894 LEVIDO CY
E,M,P

DENMARK

Hewlett-Packard A/S
Datavej 52
DK-3460 **BIRKEROD**
Tel: (02) 81-66-40
Telex: 37409 hpas dk
A,CH,CM,CS,E,MS,P

Hewlett-Packard A/S
Rolighedsvej 32
DK-8240 **RISSKOV**, Aarhus
Tel: (06) 17-60-00
Telex: 37409 hpas dk
CH,E

DOMINICAN REPUBLIC

Microprog S.A.
Juan Tomás Mejía y Cotes No. 60
Arroyo Hondo
SANTO DOMINGO
Tel: 565-6268
Telex: 4510 ARENTA DR (RCA) P

ECUADOR

CYEDE Cia. Ltda.
Avenida Eloy Alfaro 1749
Casilla 6423 CCI
QUITO
Tel: 450-975, 243-052
Telex: 2548 CYEDE ED
CM,E,P

Hospitalar S.A.
Robles 625
Casilla 3590
QUITO
Tel: 545-250, 545-122
Telex: 2485 HOSPTL ED
Cable: HOSPITALAR-Quito
M

EGYPT

International Engineering Associates
24 Hussein Hegazi Street
Kasr-el-Aini
CAIRO
Tel: 23829, 21641
Telex: IEA UN 93830
CH,CS,E,M

EGYPOR
P.O.Box 2558
42 El Zahraa Street
***CAIRO**, Egypt*
Tel: 65 00 21
Telex: 93 337
P

EL SALVADOR

IPESA de El Salvador S.A.
29 Avenida Norte 1216
SAN SALVADOR
Tel: 26-6858, 26-6868
Telex: 20539 IPESASAL
A,CH,CM,CS,E,P

FINLAND

Hewlett-Packard Oy
Revontulentie 7
PL 24
SF-02101 **ESPOO** 10
Tel: (90) 4550211
Telex: 121563 hewpa sf
CH,CM,CS,P

Hewlett-Packard Oy
(Olarinluoma 7)
PL 24
02101 **ESPOO** 10
Tel: (90) 4521022
A,E,MS

Hewlett-Packard Oy
Aatoksenkatv 10-C
SF-40720-72 **JYVASKYLA**
Tel: (941) 216318
CH

Hewlett-Packard Oy
Kainvuntie 1-C
SF-90140-14 **OULU**
Tel: (981) 338785
CH

FRANCE

Hewlett-Packard France
Z.I. Mercure B
Rue Berthelot
F-13763 Les Milles Cedex
AIX-EN-PROVENCE
Tel: 16 (42) 59-41-02
Telex: 410770F
A,CH,E,MS,P*

Hewlett-Packard France
64, rue Marchand Saillant
F-61000 **ALENCON**
Tel: 16 (33) 29 04 42

Hewlett-Packard France
Boite Postale 503
F-25026 **BESANCON**
28 rue de la Republique
F-25000 **BESANCON**
Tel: 16 (81) 83-16-22
CH,M

Hewlett-Packard France
13, Place Napoleon III
F-29000 **BREST**
Tel: 16 (98) 03-38-35

Hewlett-Packard France
Chemin des Mouilles
Boite Postale 162
F-69130 **ECULLY** Cedex (Lyon)
Tel: 16 (78) 833-81-25
Telex: 310617F
A,CH,CS,E,MP

Hewlett-Packard France
Tour Lorraine
Boulevard de France
F-91035 **EVRY** Cedex
Tel: 16 6 077-96-60
Telex: 692315F
E

Hewlett-Packard France
Parc d'Activite du Bois Briard
Ave. du Lac
F-91040 **EVRY** Cedex
Tel: 16 6 077-8383
Telex: 692315F
E

Hewlett-Packard France
5, avenue Raymond Chanas
F-38320 **EYBENS** (Grenoble)
Tel: 16 (76) 25-81-41
Telex: 980124 HP GRENOB EYBE
CH

Hewlett-Packard France
Centre d'Affaire Paris-Nord
Bâtiment Ampère 5 étage
Rue de la Commune de Paris
Boite Postale 300
F-93153 **LE BLANC MESNIL**
Tel: 16 (1) 865-44-52
Telex: 211032F
CH,CS,E,MS

Hewlett-Packard France
Parc d'Activités Cadera
Quartier Jean Mermoz
Avenue du Président JF Kennedy
F-33700 **MERIGNAC** (Bordeaux)
Tel: 16 (56) 34-00-84
Telex: 550105F
CH,E,MS

Hewlett-Packard France
Immueble "Les 3 B"
Nouveau Chemin de la Garde
ZAC de Bois Briand
F-44085 **NANTES** Cedex
Tel: 16 (40) 50-32-22
CH**

FRANCE (Cont'd)

Hewlett-Packard France
125, rue du Faubourg Bannier
F-45000 **ORLEANS**
Tel: 16 (38) 68 01 63

Hewlett-Packard France
Zone Industrielle de Courtaboeuf
Avenue des Tropiques
F-91947 Les Ulis Cedex **ORSAY**
Tel: (6) 907-78-25
Telex: 600048F
A,CH,CM,CS,E,MP,P

Hewlett-Packard France
Paris Porte-Maillot
15, Avenue de L'Amiral Bruix
F-75782 **PARIS** CEDEX 16
Tel: 16 (1) 502-12-20
Telex: 613663F
CH,MS,P

Hewlett-Packard France
124, Boulevard Tourasse
F-64000 **PAU**
Tel: 16 (59) 80 38 02

Hewlett-Packard France
2 Allée de la Bourgonnette
F-35100 **RENNES**
Tel: 16 (99) 51-42-44
Telex: 740912F
CH,CM,E,MS,P*

Hewlett-Packard France
98 Avenue de Bretagne
F-76100 **ROUEN**
Tel: 16 (35) 63-57-66
CH**,CS

Hewlett-Packard France
4 Rue Thomas Mann
Boite Postale 56
F-67033 **STRASBOURG** Cedex
Tel: 16 (88) 28-56-46
Telex: 890141F
CH,E,MS,P*

Hewlett-Packard France
Le Péripole
20, Chemin du Pigeonnier de la Cépière
F-31083 **TOULOUSE** Cedex
Tel: 16 (61) 40-11-12
Telex: 531639F
A,CH,CS,E,P*

Hewlett-Packard France
9, rue Baudin
F-26000 **VALENCE**
Tel: 16 (75) 42 76 16

Hewlett-Packard France
Carolor
ZAC de Bois Briand
F-57640 **VIGY** (Metz)
Tel: 16 (8) 771 20 22
CH

Hewlett-Packard France
Immeuble Péricentre
F-59658 **VILLENEUVE D'ASCQ** Cedex
Tel: 16 (20) 91-41-25
Telex: 160124F
CH,E,MS,P*

GERMAN FEDERAL REPUBLIC

Hewlett-Packard GmbH
Geschäftsstelle
Keithstrasse 2-4
D-1000 **BERLIN** 30
Tel: (030) 24-90-86
Telex: 018 3405 hpbln d
A,CH,E,M,P

Hewlett-Packard GmbH
Geschäftsstelle
Herrenberger Strasse 130
D-7030 **BÖBLINGEN**
Tel: (7031) 14-0
Telex:
A,CH,CM,CS,E,MP,P

Hewlett-Packard GmbH
Geschäftsstelle
Emanuel-Leutze-Strasse 1
D-4000 **DUSSELDORF**
Tel: (0211) 5971-1
Telex: 085/86 533 hpdd d
A,CH,CS,E,MS,P

Hewlett-Packard GmbH
Geschäftsstelle
Schleefstr. 28a
D-4600 **DORTMUND**-Aplerbeck
Tel: (0231) 45001

Hewlett-Packard GmbH
Vertriebszentrale Frankfurt
Berner Strasse 117
Postfach 560 140
D-6000 **FRANKFURT** 56
Tel: (0611) 50-04-1
Telex: 04 13249 hpffm d
A,CH,CM,CS,E,MP,P

Hewlett-Packard GmbH
Geschäftsstelle
Aussenstelle Bad Homburg
Louisenstrasse 115
D-6380 **BAD HOMBURG**
Tel: (06172) 109-0

Hewlett-Packard GmbH
Geschäftsstelle
Kapstadtring 5
D-2000 **HAMBURG** 60
Tel: (040) 63804-1
Telex: 021 63 032 hphh d
A,CH,CS,E,MS,P

Hewlett-Packard GmbH
Geschäftsstelle
Heidering 37-39
D-3000 **HANNOVER** 61
Tel: (0511) 5706-0
Telex: 092 3259
A,CH,CM,E,MS,P

Hewlett-Packard GmbH
Geschäftsstelle
Rosslauer Weg 2-4
D-6800 **MANNHEIM**
Tel: (0621) 70050
Telex: 0462105
A,C,E

Hewlett-Packard GmbH
Geschäftsstelle
Messerschmittstrasse 7
D-7910 **NEU ULM**
Tel: 0731-70241
Telex: 0712816 HP ULM-D
A,C,E*

Hewlett-Packard GmbH
Geschäftsstelle
Ehhericherstr. 13
D-8500 **NÜRNBERG** 10
Tel: (0911) 5205-0
Telex: 0623 860
CH,CM,E,MS,P

Hewlett-Packard GmbH
Geschäftsstelle
Eschenstrasse 5
D-8028 **TAUFKIRCHEN**
Tel: (089) 6117-1
Telex: 0524985
A,CH,CM,E,MS,P

GREAT BRITAIN

See United Kingdom

GREECE

Kostas Karaynnis S.A.
8 Omirou Street
***ATHENS** 133*
Tel: 32 30 303, 32 37 371
Telex: 215962 RKAR GR
A,CH,CM,CS,E,M,P

PLAISIO S.A.
G. Gerardos
24 Stournara Street
ATHENS
Tel: 36-11-160
Telex: 221871
P

GUATEMALA

IPESA
Avenida Reforma 3-48, Zona 9
GUATEMALA CITY
Tel: 316627, 314786
Telex: 4192 TELTRO GU
A,CH,CM,CS,E,M,P

HONG KONG

Hewlett-Packard Hong Kong, Ltd.
G.P.O. Box 795
5th Floor, Sun Hung Kai Centre
30 Harbour Road
HONG KONG
Tel: 5-8323211
Telex: 66678 HEWPA HX
Cable: HEWPACK HONG KONG
E,CH,CS,P

CET Ltd.
1402 Tung Wah Mansion
199-203 Hennessy Rd.
*Wanchia, **HONG KONG***
Tel: 5-729376
Telex: 85148 CET HX
CM

Schmidt & Co. (Hong Kong) Ltd.
Wing On Centre, 28th Floor
Connaught Road, C.
HONG KONG
Tel: 5-455644
Telex: 74766 SCHMX HX
A,M

ICELAND

Elding Trading Company Inc.
Hafnarnvoli-Tryggvagotu
P.O. Box 895
IS-REYKJAVIK
Tel: 1-58-20, 1-63-03
M

INDIA

Computer products are sold through Blue Star Ltd. All computer repairs and maintenance service is done through Computer Maintenance Corp.

Blue Star Ltd.
Sabri Complex II Floor
24 Residency Rd.
***BANGALORE** 560 025*
Tel: 55660
Telex: 0845-430
Cable: BLUESTAR
A,CH,CM,CS*,E*

Blue Star Ltd.
Band Box House
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-3101
Telex: 011-3751
Cable: BLUESTAR
A,M

Blue Star Ltd.
Sahas
414/2 Vir Savarkar Marg
Prabhadevi
***BOMBAY** 400 025*
Tel: 422-6155
Telex: 011-4093
Cable: FROSTBLUE
A,CH,CM,CS*,E,M*

Blue Star Ltd.
Kalyan, 19 Vishwas Colony
*Alkapuri, **BORODA**, 390 005*
Tel: 65235
Cable: BLUE STAR
A

Blue Star Ltd.
7 Hare Street
***CALCUTTA** 700 001*
Tel: 12-01-31
Telex: 021-7655
Cable: BLUESTAR
A,M

Blue Star Ltd.
133 Kodambakkam High Road
***MADRAS** 600 034*
Tel: 82057
Telex: 041-379
Cable: BLUESTAR
A,M

Blue Star Ltd.
Bhandari House, 7th/8th Floors
91 Nehru Place
***NEW DELHI** 110 024*
Tel: 682547
Telex: 031-2463
Cable: BLUESTAR
A,CH,CM,CS*,E,M*

Blue Star Ltd.
15/16:C Wellesley Rd.
***PUNE** 411 011*
Tel: 22775
Cable: BLUE STAR
A

Blue Star Ltd.
2-2-47/1108 Bolarum Rd.
***SECUNDERABAD** 500 003*
Tel: 72057
Telex: 0155-459
Cable: BLUEFROST
A,E

Blue Star Ltd.
T.C. 7/603 Poornima
Maruthankuzhi
***TRIVANDRUM** 695 013*
Tel: 65799
Telex: 0884-259
Cable: BLUESTAR
E

Computer Maintenance Corporation Ltd.
115, Sarojini Devi Road
***SECUNDERABAD** 500 003*
Tel: 310-184, 345-774
Telex: 031-2960
*CH***

SALES & SUPPORT OFFICES

Arranged alphabetically by country

INDONESIA

BERCA Indonesia P.T.
P.O.Box 496/Jkt.
Jl. Abdul Muis 62
JAKARTA
Tel: 21-373009
Telex: 46748 BERSAL IA
Cable: BERSAL JAKARTA
P

BERCA Indonesia P.T.
P.O.Box 2497/Jkt
Antara Bldg., 17th Floor
Jl. Medan Merdeka Selatan 17
JAKARTA-PUSAT
Tel: 21-344-181
Telex: BERSAL IA
A,CS,E,M

BERCA Indonesia P.T.
P.O. Box 174/SBY.
Jl. Kutei No. 11
SURABAYA
Tel: 68172
Telex: 31146 BERSAL SB
Cable: BERSAL-SURABAYA
A,E,M,P*

IRAQ

Hewlett-Packard Trading S.A.
Service Operation
Al Mansoor City 9B/3/7
BAGHDAD
Tel: 551-49-73
Telex: 212-455 HEPAIRAQ IK
CH,CS

IRELAND

Hewlett-Packard Ireland Ltd.
82/83 Lower Leeson Street
DUBLIN 2
Tel: 0001 608800
Telex: 30439
A,CH,CM,CS,E,M,P

Cardiac Services Ltd.
Kilmore Road
Artane
***DUBLIN** 5*
Tel: (01) 351820
Telex: 30439
M

ISRAEL

Eldan Electronic Instrument Ltd.
P.O.Box 1270
***JERUSALEM** 91000*
16, Ohaliav St.
***JERUSALEM** 94467*
Tel: 533 221, 553 242
Telex: 25231 AB/PAKRD IL
A

Electronics Engineering Division
Motorola Israel Ltd.
16 Kremenetski Street
P.O. Box 25016
TEL-AVIV 67899
Tel: 3 88 388
Telex: 33569 Motil IL
Cable: BASTEL Tel-Aviv
CH,CM,CS,E,M,P

ITALY

Hewlett-Packard Italiana S.p.A
Traversa 99C
Via Giulio Petroni, 19
I-70124 **BARI**
Tel: (080) 41-07-44
M

Hewlett-Packard Italiana S.p.A.
Via Martin Luther King, 38/III
I-40132 **BOLOGNA**
Tel: (051) 402394
Telex: 511630
CH,E,MS

Hewlett-Packard Italiana S.p.A.
Via Principe Nicola 43G/C
I-95126 **CATANIA**
Tel: (095) 37-10-87
Telex: 970291
C,P

Hewlett-Packard Italiana S.p.A.
Via G. Di Vittorio 9
I-20063 **CERNUSCO SUL NAVIGLIO**
(Milano)
Tel: (02) 923691
Telex: 334632
A,CH,CM,CS,E,MP,P

Hewlett-Packard Italiana S.p.A.
Via C. Colombo 49
I-20090 **TREZZANO SUL NAVIGLIO**
(Milano)
Tel: (02) 4459041
Telex: 322116
C,M

Hewlett-Packard Italiana S.p.A.
Via Nuova San Rocco a
Capodimonte, 62/A
I-80131 **NAPOLI**
Tel: (081) 7413544
Telex: 710698
A,CH,E

Hewlett-Packard Italiana S.p.A.
Viale G. Modugno 33
I-16156 **GENOVA PEGLI**
Tel: (010) 68-37-07
Telex: 215238
E,C

Hewlett-Packard Italiana S.p.A.
Via Pelizzo 15
I-35128 **PADOVA**
Tel: (049) 664888
Telex: 430315
A,CH,E,MS

Hewlett-Packard Italiana S.p.A.
Viale C. Pavese 340
I-00144 **ROMA EUR**
Tel: (06) 54831
Telex: 610514
A,CH,CM,CS,E,MS,P*

Hewlett-Packard Italiana S.p.A.
Via di Casellina 57/C
I-50018 **SCANDICCI-FIRENZE**
Tel: (055) 753863

Hewlett-Packard Italiana S.p.A.
Corso Svizzera, 185
I-10144 **TORINO**
Tel: (011) 74 4044
Telex: 221079
CH,E

JAPAN

Yokogawa-Hewlett-Packard Ltd.
152-1, Onna
ATSUGI, Kanagawa, 243
Tel: (0462) 28-0451
CM,C*,E

Yokogawa-Helwett-Packard Ltd.
Meiji-Seimei Bldg. 6F
3-1 Hon Chiba-Cho
CHIBA, 280
Tel: 472 25 7701
E,CH,CS

Yokogawa-Hewlett-Packard Ltd.
Yasuda-Seimei Hiroshima Bldg.
6-11, Hon-dori, Naka-ku
HIROSHIMA, 730
Tel: 82-241-0611

Yokogawa-Hewlett-Packard Ltd.
Towa Building
2-3, Kaigan-dori, 2 Chome Chuo-ku
KOBE, 650
Tel: (078) 392-4791
C,E

Yokogawa-Hewlett-Packard Ltd.
Kumagaya Asahi 82 Bldg
3-4 Tsukuba
KUMAGAYA, Saitama 360
Tel: (0485) 24-6563
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Asahi Shinbun Daiichi Seimei Bldg.
4-7, Hanabata-cho
KUMAMOTO,860
Tel: (0963) 54-7311
CH,E

Yokogawa-Hewlett-Packard Ltd.
Shin-Kyoto Center Bldg.
614, Higashi-Shiokoji-cho
Karasuma-Nishiiru
Shiokoji-dori, Shimogyo-ku
KYOTO, 600
Tel: 075-343-0921
CH,E

Yokogawa-Hewlett-Packard Ltd.
Mito Mitsui Bldg
4-73, Sanno-maru, 1 Chome
MITO, Ibaraki 310
Tel: (0292) 25-7470
CH,CM,E

Yokogawa-Hewlett-Packard Ltd.
Sumitomo Seimei 14-9 Bldg.
Meieki-Minami, 2 Chome
Nakamura-ku
NAGOYA, 450
Tel: (052) 571-5171
CH,CM,CS,E,MS

Yokogawa-Hewlett-Packard Ltd.
Chuo Bldg.,
4-20 Nishinakajima, 5 Chome
Yodogawa-ku
OSAKA, 532
Tel: (06) 304-6021
Telex: YHPOSA 523-3624
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
27-15, Yabe, 1 Chome
SAGAMIHARA Kanagawa, 229
Tel: 0427 59-1311

Yokogawa-Hewlett-Packard Ltd.
Daiichi Seimei Bldg.
7-1, Nishi Shinjuku, 2 Chome
Shinjuku-ku,**TOKYO** 160
Tel: 03-348-4611
CH,E

Yokogawa-Hewlett-Packard Ltd.
29-21 Takaido-Higashi, 3 Chome
Suginami-ku **TOKYO** 168
Tel: (03) 331-611
Telex: 232-2024 YHPTOK
A,CH,CM,CS,E,MP,P*

Yokogawa-Hewlett-Packard Ltd.
Daiichi Asano Building
2-8, Odori, 5 Chome
UTSUNOMIYA, Tochigi 320
Tel: (0286) 25-7155
CH,CS,E

Yokogawa-Hewlett-Packard Ltd.
Yasuda Seimei Nishiguchi Bldg.
30-4 Tsuruya-cho, 3 Chome
YOKOHAMA 221
Tel: (045) 312-1252
CH,CM,E

JORDAN

Mouasher Cousins Company
P.O. Box 1387
AMMAN
Tel: 24907, 39907
Telex: 21456 SABCO JO
CH,E,M,P

KENYA

ADCOM Ltd., Inc., Kenya
P.O.Box 30070
NAIROBI
Tel: 331955
Telex: 22639
E,M

KOREA

Samsung Electronics HP Division
12 Fl. Kinam Bldg.
San 75-31, Yeoksam-Dong
Kangnam-Ku
Yeongdong P.O. Box 72
SEOUL
Tel: 555-7555, 555-5447
Telex: K27364 SAMSAN
A,CH,CM,CS,E,M,P

KUWAIT

Al-Khaldiya Trading & Contracting
P.O. Box 830 Safat
KUWAIT
Tel: 42-4910, 41-1726
Telex: 22481 Areeg kt
CH,E,M

Photo & Cine Equipment
P.O. Box 270 Safat
KUWAIT
Tel: 42-2846, 42-3801
Telex: 22247 Matin kt
P

LEBANON

G.M. Dolmadjian
Achrafieh
P.O. Box 165.167
BEIRUT
Tel: 290293
*MP***

Computer Information Systems
P.O. Box 11-6274
BEIRUT
Tel: 89 40 73
Telex: 22259
C

LUXEMBOURG

Hewlett-Packard Belgium S.A./N.V.
Blvd de la Woluwe, 100
Woluwedal
B-1200 **BRUSSELS**
Tel: (02) 762-32-00
Telex: 23-494 paloben bru
A,CH,CM,CS,E,MP,P

MALAYSIA

Hewlett-Packard Sales (Malaysia)
Sdn. Bhd.
1st Floor, Bangunan British
American
Jalan Semantan, Damansara Heights
KUALA LUMPUR 23-03
Tel: 943022
Telex: MA31011
A,CH,E,M,P*

SALES & SUPPORT OFFICES

Arranged alphabetically by country

MAYLAYSIA (Cont'd)

Protel Engineering
P.O.Box 1917
Lot 6624, Section 64
23/4 Pending Road
*Kuching, **SARAWAK***
Tel: 36299
Telex: MA 70904 PROMAL
Cable: PROTELENG
A,E,M

MALTA

Philip Toledo Ltd.
Notabile Rd.
MRIEHEL
Tel: 447 47, 455 66
Telex: Media MW 649
E,P

MEXICO

Hewlett-Packard Mexicana, S.A. de C.V.
Av. Periferico Sur No. 6501
Tepepan, Xochimilco
16020 **MEXICO D.F.**
Tel: 6-76-46-00
Telex: 17-74-507 HEWPACK MEX
A,CH,CS,E,MS,P

Hewlett-Packard Mexicana, S.A. de C.V.
Ave. Colonia del Valle 409
Col. del Valle
Municipio de Garza Garcia
MONTERREY, Nuevo Leon
Tel: 78 42 41
Telex: 038 410
CH

ECISA
José Vasconcelos No. 218
Col. Condesa Deleg. Cuauhtémoc
***MEXICO** D.F. 06140*
Tel: 553-1206
Telex: 17-72755 ECE ME
M

MOROCCO

Dolbeau
81 rue Karatchi
CASABLANCA
Tel: 3041-82, 3068-38
Telex: 23051, 22822
E

Gerep
2 rue d'Agadir
Boite Postale 156
CASABLANCA
Tel: 272093, 272095
Telex: 23 739
P

NETHERLANDS

Hewlett-Packard Nederland B.V.
Van Heuven Goedhartlaan 121
NL 1181KK **AMSTELVEEN**
P.O. Box 667
NL1180 AR **AMSTELVEEN**
Tel: (020) 47-20-21
Telex: 13 216 HEPA NL
A,CH,CM,CS,E,MP,P

Hewlett-Packard Nederland B.V.
Bongerd 2
NL 2906VK **CAPELLE A/D IJSSEL**
P.O. Box 41
NL 2900AA **CAPELLE A/D IJSSEL**
Tel: (10) 51-64-44
Telex: 21261 HEPAC NL
A,CH,CS,E

Hewlett-Packard Nederland B.V.
Pastoor Petersstraat 134-136
NL 5612 LV **EINDHOVEN**
P.O. Box 2342
NL 5600 CH **EINDHOVEN**
Tel: (040) 326911
Telex: 51484 hepae nl
A,CH**,E,M

NEW ZEALAND

Hewlett-Packard (N.Z.) Ltd.
5 Owens Road
P.O. Box 26-189
Epsom, **AUCKLAND**
Tel: 687-159
Cable: HEWPACK Auckland
CH,CM,E,P*

Hewlett-Packard (N.Z.) Ltd.
4-12 Cruickshank Street
Kilbirnie, **WELLINGTON 3**
P.O. Box 9443
Courtenay Place, **WELLINGTON 3**
Tel: 877-199
Cable: HEWPACK Wellington
CH,CM,E,P

Northrop Instruments & Systems Ltd.
369 Khyber Pass Road
P.O. Box 8602
AUCKLAND
Tel: 794-091
Telex: 60605
A,M

Northrop Instruments & Systems Ltd.
110 Mandeville St.
P.O. Box 8388
CHRISTCHURCH
Tel: 486-928
Telex: 4203
A,M

Northrop Instruments & Systems Ltd.
Sturdee House
85-87 Ghuznee Street
P.O. Box 2406
WELLINGTON
Tel: 850-091
Telex: NZ 3380
A,M

NORTHERN IRELAND

See United Kingdom

NORWAY

Hewlett-Packard Norge A/S
Folke Bernadottes vei 50
P.O. Box 3558
N-5033 **FYLLINGSDALEN** (Bergen)
Tel: 0047/5/16 55 40
Telex: 16621 hpnas n
CH,CS,E,MS

Hewlett-Packard Norge A/S
Österndalen 16-18
P.O. Box 34
N-1345 **ÖSTERÅS**
Tel: 0047/2/17 11 80
Telex: 16621 hpnas n
A,CH,CM,CS,E,M,P

OMAN

Khimjil Ramdas
P.O. Box 19
MUSCAT
Tel: 722225, 745601
Telex: 3289 BROKER MB MUSCAT
P

Suhail & Saud Bahwan
P.O.Box 169
MUSCAT
Tel: 734 201-3
Telex: 3274 BAHWAN MB

PAKISTAN

Mushko & Company Ltd.
1-B, Street 43
Sector F-8/1
ISLAMABAD
Tel: 51071
Cable: FEMUS Rawalpindi
A,E,M

Mushko & Company Ltd.
Oosman Chambers
Abdullah Haroon Road
***KARACHI** 0302*
Tel: 524131, 524132
Telex: 2894 MUSKO PK
Cable: COOPERATOR Karachi
*A,E,M,P**

PANAMA

Electrónico Balboa, S.A.
Calle Samuel Lewis, Ed. Alfa
Apartado 4929
PANAMA 5
Tel: 63-6613, 63-6748
Telex: 3483 ELECTRON PG
A,CM,E,M,P

PERU

Cía Electro Médica S.A.
Los Flamencos 145, San Isidro
Casilla 1030
***LIMA** 1*
Tel: 41-4325, 41-3703
Telex: Pub. Booth 25306
CM,E,M,P

PHILIPPINES

The Online Advanced Systems Corporation
Rico House, Amorsolo Cor. Herrera Street
Legaspi Village, Makati
P.O. Box 1510
*Metro **MANILA***
Tel: 85-35-81, 85-34-91, 85-32-21
Telex: 3274 ONLINE
A,CH,CS,E,M

Electronic Specialists and Proponents Inc.
690-B Epifanio de los Santos Avenue
*Cubao, **QUEZON CITY***
P.O. Box 2649 Manila
Tel: 98-96-81, 98-96-82, 98-96-83
Telex: 40018, 42000 ITT GLOBE MACKAY BOOTH
P

PORTUGAL

Mundinter
Intercambio Mundial de Comércio S.A.R.L.
P.O. Box 2761
Av. Antonio Augusto de Aguiar 138
P-LISBON
Tel: (19) 53-21-31, 53-21-37
Telex: 16691 munter p
M

Soquimica
Av. da Liberdade, 220-2
*1298 **LISBOA** Codex*
Tel: 56 21 81/2/3
Telex: 13316 SABASA
P

Telectra-Empresa Técnica de Equipmentos Eléctricos S.A.R.L.
Rua Rodrigo da Fonseca 103
P.O. Box 2531
***P-LISBON** 1*
Tel: (19) 68-60-72
Telex: 12598
CH,CS,E,P

PUERTO RICO

Hewlett-Packard Puerto Rico
Ave. Muñoz Rivera #101
Esq. Calle Ochoa
HATO REY, Puerto Rico 00918
Tel: (809) 754-7800

Hewlett-Packard Puerto Rico
Calle 272 Edificio 203
Urb. Country Club
RIO PIEDRAS, Puerto Rico
P.O. Box 4407
CAROLINA, Puerto Rico 00628
Tel: (809) 762-7255
A,CH,CS

QATAR

Computearbia
P.O. Box 2750
DOHA
Tel: 883555
Telex: 4806 CHPARB
P

Eastern Technical Services
P.O.Box 4747
DOHA
Tel: 329 993
Telex: 4156 EASTEC DH

Nasser Trading & Contracting
P.O.Box 1563
DOHA
Tel: 22170, 23539
Telex: 4439 NASSER DH
M

SAUDI ARABIA

Modern Electronic Establishment
Hewlett-Packard Division
P.O. Box 22015
Thuobah
AL-KHOBAR
Tel: 895-1760, 895-1764
Telex: 671 106 HPMEEK SJ
Cable: ELECTA AL-KHOBAR
CH,CS,E,M

Modern Electronic Establishment
Hewlett-Packard Division
P.O. Box 1228
Redec Plaza, 6th Floor
JEDDAH
Tel: 644 38 48
Telex: 4027 12 FARNAS SJ
Cable: ELECTA JEDDAH
CH,CS,E,M

Modern Electronic Establishment
Hewlett-Packard Division
P.O.Box 22015
RIYADH
Tel: 491-97 15, 491-63 87
Telex: 202049 MEERYD SJ
CH,CS,E,M

Abdul Ghani El Ajou
P.O. Box 78
RIYADH
Tel: 40 41 717
Telex: 200 932 EL AJOU
P

SCOTLAND

See United Kingdom

SINGAPORE

Hewlett-Packard Singapore (Sales)
Pte. Ltd.
#08-00 Inchcape House
450-2 Alexandra Road
P.O. Box 58 Alexandra Rd. Post Office
SINGAPORE, 9115
Tel: 631788
Telex: HPSGSO RS 34209
Cable: HEWPACK, Singapore
A,CH,CS,E,MS,P

SALES & SUPPORT OFFICES

Arranged alphabetically by country

SINGAPORE (Cont'd)

Dynamar International Ltd.
Unit 05-11 Block 6
Kolam Ayer Industrial Estate
***SINGAPORE** 1334*
Tel: 747-6188
Telex: RS 26283
CM

SOUTH AFRICA

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 120
Howard Place **CAPE PROVINCE** 7450
Pine Park Center, Forest Drive,
Pinelands
CAPE PROVINCE 7405
Tel: 53-7954
Telex: 57-20006
A,CH,CM,E,MS,P

Hewlett-Packard So Africa (Pty.) Ltd.
P.O. Box 37099
92 Overport Drive
DURBAN 4067
Tel: 28-4178, 28-4179, 28-4110
Telex: 6-22954
CH,CM

Hewlett-Packard So Africa (Pty.) Ltd.
6 Linton Arcade
511 Cape Road
Linton Grange
PORT ELIZABETH 6000
Tel: 041-302148
CH

Hewlett-Packard So Africa (Pty.) Ltd.
P.O.Box 33345
Glenstantia 0010 **TRANSVAAL**
1st Floor East
Constantia Park Ridge Shopping
Centre
Constantia Park
PRETORIA
Tel: 982043
Telex: 32163
CH,E

Hewlett-Packard So Africa (Pty.) Ltd.
Private Bag Wendywood
SANDTON 2144
Tel: 802-5111, 802-5125
Telex: 4-20877
Cable: HEWPACK Johannesburg
A,CH,CM,CS,E,MS,P

SPAIN

Hewlett-Packard Española S.A.
Calle Entenza, 321
E-**BARCELONA** 29
Tel: 322.24.51, 321.73.54
Telex: 52603 hpbee
A,CH,CS,E,MS,P

Hewlett-Packard Española S.A.
Calle San Vicente S/No
Edificio Albia II
E-**BILBAO** 1
Tel: 423.83.06
A,CH,E,MS

Hewlett-Packard Española S.A.
Crta. de la Coruña, Km. 16, 400
Las Rozas
E-**MADRID**
Tel: (1) 637.00.11
CH,CS,M

Hewlett-Packard Española S.A.
Avda. S. Francisco Javier, S/no
Planta 10. Edificio Sevilla 2,
E-**SEVILLA** 5
Tel: 64.44.54
Telex: 72933
A,CS,MS,P

Hewlett-Packard Española S.A.
Calle Ramon Gordillo, 1 (Entlo.3)
E-**VALENCIA** 10
Tel: 361-1354
CH,P

SWEDEN

Hewlett-Packard Sverige AB
Sunnanvagen 14K
S-22226 **LUND**
Tel: (046) 13-69-79
Telex: (854) 17886 (via Spånga
office)
CH

Hewlett-Packard Sverige AB
Östra Tullgatan 3
S-21128 **MALMÖ**
Tel: (040) 70270
Telex: (854) 17886 (via Spånga
office)

Hewlett-Packard Sverige AB
Våstra Vintergatan 9
S-70344 **ÖREBRO**
Tel: (19) 10-48-80
Telex: (854) 17886 (via Spånga
office)
CH

Hewlett-Packard Sverige AB
Skalholtsgatan 9, Kista
Box 19
S-16393 **SPÅNGA**
Tel: (08) 750-2000
Telex: (854) 17886
Telefax: (08) 7527781
A,CH,CM,CS,E,MS,P

Hewlett-Packard Sverige AB
Frötallisgatan 30
S-42132 **VÄSTRA-FRÖLUNDA**
Tel: (031) 49-09-50
Telex: (854) 17886 (via Spånga
office)
CH,E,P

SWITZERLAND

Hewlett-Packard (Schweiz) AG
Clarastrasse 12
CH-4058 **BASEL**
Tel: (61) 33-59-20
A

Hewlett-Packard (Schweiz) AG
7, rue du Bois-du-Lan
Case Postale 365
CH-1217 **MEYRIN** 2
Tel: (0041) 22-83-11-11
Telex:27333 HPAG CH
CH,CM,CS

Hewlett-Packard (Schweiz) AG
Allmend 2
CH-8967 **WIDEN**
Tel: (0041) 57 31 21 11
Telex: 53933 hpag ch
Cable: HPAG CH
A,CH,CM,CS,E,MS,P

SYRIA

General Electronic Inc.
Nuri Basha Ahnaf Ebn Kays Street
P.O. Box 5781
DAMASCUS
Tel: 33-24-87
Telex: 411 215
Cable: ELECTROBOR DAMASCUS
E

Middle East Electronics
P.O.Box 2308
Abu Rumnaneh
DAMASCUS
Tel: 33 4 5 92
Telex: 411 304
M

TAIWAN

Hewlett-Packard Far East Ltd.
Kaohsiung Office
2/F 68-2, Chung Cheng 3rd Road
KAOHSIUNG
Tel: (07) 241-2318
CH,CS,E

Hewlett-Packard Far East Ltd.
Taiwan Branch
8th Floor
337 Fu Hsing North Road
TAIPEI
Tel: (02) 712-0404
Telex: 24439 HEWPACK
Cable:HEWPACK Taipei
A,CH,CM,CS,E,M,P

Ing Lih Trading Co.
3rd Floor, 7 Jen-Ai Road, Sec. 2
TAIPEI 100
Tel: (02) 3948191
Cable: INGLIH TAIPEI
A

THAILAND

Unimesa
30 Patpong Ave., Suriwong
***BANGKOK** 5*
Tel: 235-5727
Telex: 84439 Simonco TH
Cable: UNIMESA Bangkok
A,CH,CS,E,M

Bangkok Business Equipment Ltd.
5/5-6 Dejo Road
BANGKOK
Tel: 234-8670, 234-8671
Telex: 87669-BEQUIPT TH
Cable: BUSIQUIPT Bangkok
P

TRINIDAD & TOBAGO

Caribbean Telecoms Ltd.
50/A Jerningham Avenue
P.O. Box 732
PORT-OF-SPAIN
Tel: 62-44213, 62-44214
Telex: 235,272 HUGCO WG
CM,E,M,P

TUNISIA

Tunisie Electronique
31 Avenue de la Liberte
TUNIS
Tel: 280-144
E,P

Corema
1 ter. Av. de Carthage
TUNIS
Tel: 253-821
Telex: 12319 CABAM TN
M

TURKEY

Teknim Company Ltd.
Iran Caddesi No. 7
*Kavaklidere, **ANKARA***
Tel: 275800
Telex: 42155 TKNM TR
E

E.M.A.
Medina Eldem Sokak No.41/6
Yuksel Caddesi
ANKARA
Tel: 175 622
Telex: 42 591
M

UNITED ARAB EMIRATES

Emitac Ltd.
P.O. Box 2711
ABU DHABI
Tel: 82 04 19-20
Cable: EMITAC ABUDHABI

Emitac Ltd.
P.O. Box 1641
SHARJAH
Tel: 591 181
Telex: 68136 Emitac Sh
CH,CS,E,M,P

UNITED KINGDOM

GREAT BRITAIN

Hewlett-Packard Ltd.
Trafalgar House
Navigation Road
ALTRINCHAM
Cheshire WA14 1NU
Tel: 061 928 6422
Telex: 668068
A,CH,CS,E,M,MS,P

Hewlett-Packard Ltd.
Elstree House, Elstree Way
BOREHAMWOOD, Herts WD6 1SG
Tel: 01 207 5000
Telex: 8952716
E,CH,CS,P

Hewlett-Packard Ltd.
Oakfield House, Oakfield Grove
Clifton **BRISTOL,** Avon BS8 2BN
Tel: 0272 736806
Telex: 444302
CH,CS,E,P

Hewlett-Packard Ltd.
Bridewell House
Bridewell Place
LONDON EC4V 6BS
Tel: 01 583 6565
Telex: 298163
CH,CS,P

Hewlett-Packard Ltd.
Fourier House
257-263 High Street
LONDON COLNEY
Herts. AL2 1HA, St. Albans
Tel: 0727 24400
Telex: 1-8952716
CH,CS

Hewlett-Packard Ltd.
Pontefract Road
NORMANTON, West Yorkshire WF6 1RN
Tel: 0924 895566
Telex: 557355
CH,CS,P

Hewlett-Packard Ltd.
The Quadrangle
106-118 Station Road
REDHILL, Surrey RH1 1PS
Tel: 0737 68655
Telex: 947234
CH,CS,E,P

GREAT BRITAIN (Cont'd)
Hewlett-Packard Ltd.
Avon House
435 Stratford Road
Shirley, **SOLIHULL**, West Midlands
B90 4BL
Tel: 021 745 8800
Telex: 339105
CH,CS,E,P

Hewlett-Packard Ltd.
West End House
41 High Street, West End
SOUTHAMPTON
Hampshire S03 3DQ
Tel: 04218 6767
Telex: 477138
CH,CS,P

Hewlett-Packard Ltd.
Eskdale Rd.
Winnersh, **WOKINGHAM**
Berkshire RG11 5DZ
Tel: 0734 696622
Telex: 848884
E

Hewlett-Packard Ltd.
King Street Lane
Winnersh, **WOKINGHAM**
Berkshire RG11 5AR
Tel: 0734 784774
Telex: 847178
A,CH,CS,E,M,MP,P

Hewlett-Packard Ltd.
Nine Mile Ride
Easthampstead, **WOKINGHAM**
Berkshire, 3RG11 3LL
Tel: 0344 773100
Telex: 848805
CH,CS,E,P

IRELAND

NORTHERN IRELAND
Hewlett-Packard Ltd.
Cardiac Services Building
95A Finaghy Road South
BELFAST BT10 0BY
Tel: 0232 625-566
Telex: 747626
CH,CS

SCOTLAND
Hewlett-Packard Ltd.
SOUTH QUEENSFERRY
West Lothian, EH30 9TG
Tel: 031 331 1188
Telex: 72682
CH,CM,CS,E,M,P

UNITED STATES

Alabama
Hewlett-Packard Co.
700 Century Park South, Suite 128
BIRMINGHAM, AL 35226
Tel: (205) 822-6802
A,CH,M

Hewlett-Packard Co.
420 Wynn Drive
HUNTSVILLE, AL 35805
P.O. Box 7700
HUNTSVILLE, AL 35807
Tel: (205) 830-2000
CH,CM,CS,E,M*

Arizona
Hewlett-Packard Co.
8080 Pointe Parkway West
PHOENIX, AZ 85044
Tel: (602) 273-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2424 East Aragon Road
TUCSON, AZ 85706
Tel: (602) 889-4631
CH,E,MS**

California
Hewlett-Packard Co.
99 South Hill Dr.
BRISBANE, CA 94005
Tel: (415) 330-2500
CH,CS

Hewlett-Packard Co.
P.O. Box 7830 (93747)
5060 E. Clinton Avenue, Suite 102
FRESNO, CA 93727
Tel: (209) 252-9652
CH,CS,MS

Hewlett-Packard Co.
P.O. Box 4230
1430 East Orangethorpe
FULLERTON, CA 92631
Tel: (714) 870-1000
CH,CM,CS,E,MP

Hewlett-Packard Co.
320 S. Kellogg, Suite B
GOLETA, CA 93117
Tel: (805) 967-3405
CH

Hewlett-Packard Co.
5400 W. Rosecrans Boulevard
LAWNDALE, CA 90260
P.O. Box 92105
LOS ANGELES, CA 90009
Tel: (213) 970-7500
Telex: 910-325-6608
CH,CM,CS,MP

Hewlett-Packard Co.
3155 Porter Oaks Drive
PALO ALTO, CA 94304
Tel: (415) 857-8000
CH,CS,E

Hewlett-Packard Co.
4244 So. Market Court, Suite A
P.O. Box 15976
SACRAMENTO, CA 95852
Tel: (916) 929-7222
A*,CH,CS,E,MS

Hewlett-Packard Co.
9606 Aero Drive
P.O. Box 23333
SAN DIEGO, CA 92139
Tel: (619) 279-3200
CH,CM,CS,E,MP

Hewlett-Packard Co.
2305 Camino Ramon "C"
SAN RAMON, CA 94583
Tel: (415) 838-5900
CH,CS

Hewlett-Packard Co.
3005 Scott Boulevard
SANTA CLARA, CA 95050
Tel: (408) 988-7000
Telex: 910-338-0586
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
5703 Corsa Avenue
WESTLAKE VILLAGE, CA 91362
Tel: (213) 706-6800
E*,CH*,CS*

Colorado
Hewlett-Packard Co.
24 Inverness Place, East
ENGLEWOOD, CO 80112
Tel: (303) 649-5000
A,CH,CM,CS,E,MS

Connecticut
Hewlett-Packard Co.
47 Barnes Industrial Road South
P.O. Box 5007
WALLINGFORD, CT 06492
Tel: (203) 265-7801
A,CH,CM,CS,E,MS

Florida
Hewlett-Packard Co.
2901 N.W. 62nd Street
P.O. Box 24210
FORT LAUDERDALE, FL 33307
Tel: (305) 973-2600
CH,CS,E,MP

Hewlett-Packard Co.
6177 Lake Ellenor Drive
P.O. Box 13910
ORLANDO, FL 32859
Tel: (305) 859-2900
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
5750B N. Hoover Blvd., Suite 123
P.O. Box 15200
TAMPA, FL 33614
Tel: (813) 884-3282
A*,CH,CM,CS,E*,M*

Georgia
Hewlett-Packard Co.
2000 South Park Place
P.O. Box 105005
ATLANTA, GA 30348
Tel: (404) 955-1500
Telex: 810-766-4890
A,CH,CM,CS,E,MP

Hawaii
Hewlett-Packard Co.
Kawaiahao Plaza, Suite 190
567 South King Street
HONOLULU, HI 96813
Tel: (808) 526-1555
A,CH,E,MS

Illinois
Hewlett-Packard Co.
304 Eldorado Road
P.O. Box 1607
BLOOMINGTON, IL 61701
Tel: (309) 662-9411
CH,MS**

Hewlett-Packard Co.
1100 31st Street, Suite 100
DOWNERS GROVE, IL 60515
Tel: (312) 960-5760
CH,CS

Hewlett-Packard Co.
5201 Tollview Drive
ROLLING MEADOWS, IL 60008
Tel: (312) 255-9800
Telex: 910-687-1066
A,CH,CM,CS,E,MP

Indiana
Hewlett-Packard Co.
7301 No. Shadeland Avenue
P.O. Box 50807
INDIANAPOLIS, IN 46250
Tel: (317) 842-1000
A,CH,CM,CS,E,MS

Iowa
Hewlett-Packard Co.
1776 22nd Street, Suite 1
WEST DES MOINES, IA 50265
Tel: (515) 224-1435
CH,MS**

Kansas
Hewlett-Packard Co.
7804 East Funston Road, #203
WICHITA, KS 67207
Tel: (316) 684-8491
CH

Kentucky
Hewlett-Packard Co.
10300 Linn Station Road, #100
LOUISVILLE, KY 40223
Tel: (502) 426-0100
A,CH,CS,MS

Louisiana
Hewlett-Packard Co.
160 James Drive East
ST. ROSE, LA 70087
P.O. Box 1449
KENNER, LA 70063
Tel: (504) 467-4100
A,CH,CS,E,MS

Maryland
Hewlett-Packard Co.
3701 Koppers Street
BALTIMORE, MD 21227
Tel: (301) 644-5800
Telex: 710-862-1943
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
2 Choke Cherry Road
ROCKVILLE, MD 20850
Tel: (301) 948-6370
A,CH,CM,CS,E,MP

Massachusetts
Hewlett-Packard Co.
1775 Minuteman Road
ANDOVER, MA 01810
Tel: (617) 682-1500
A,C,CH,CS,CM,E,MP,P*

Hewlett-Packard Co.
32 Hartwell Avenue
LEXINGTON, MA 02173
Tel: (617) 861-8960
CH,CS,E

Michigan
Hewlett-Packard Co.
4326 Cascade Road S.E.
GRAND RAPIDS, MI 49506
Tel: (616) 957-1970
CH,CS,MS

Hewlett-Packard Co.
1771 W. Big Beaver Road
TROY, MI 48084
Tel: (313) 643-6474
CH,CS

Minnesota
Hewlett-Packard Co.
2025 W. Larpenteur Ave.
ST. PAUL, MN 55113
Tel: (612) 644-1100
A,CH,CM,CS,E,MP

Missouri
Hewlett-Packard Co.
11131 Colorado Avenue
KANSAS CITY, MO 64137
Tel: (816) 763-8000
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
13001 Hollenberg Drive
BRIDGETON, MO 63044
Tel: (314) 344-5100
A,CH,CS,E,MP

SALES & SUPPORT OFFICES

Arranged alphabetically by country

UNITED STATES (Cont'd)

Nebraska
Hewlett-Packard
10824 Old Mill Rd., Suite 3
OMAHA, NE 68154
Tel: (402) 334-1813
CM,MS

New Jersey
Hewlett-Packard Co.
120 W. Century Road
PARAMUS, NJ 07652
Tel: (201) 265-5000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
60 New England Av. West
PISCATAWAY, NJ 08854
Tel: (201) 981-1199
A,CH,CM,CS,E

New Mexico
Hewlett-Packard Co.
11300 Lomas Blvd.,N.E.
P.O. Box 11634
ALBUQUERQUE, NM 87112
Tel: (505) 292-1330
CH,CS,E,MS

New York
Hewlett-Packard Co.
5 Computer Drive South
ALBANY, NY 12205
Tel: (518) 458-1550
A,CH,E,MS

Hewlett-Packard Co.
9600 Main Street
P.O. Box AC
CLARENCE, NY 14031
Tel: (716) 759-8621
CH

Hewlett-Packard Co.
200 Cross Keys Office Park
FAIRPORT, NY 14450
Tel: (716) 223-9950
CH,CM,CS,E,MS

Hewlett-Packard Co.
7641 Henry Clay Blvd.
LIVERPOOL, NY 13088
Tel: (315) 451-1820
A,CH,CM,E,MS

Hewlett-Packard Co.
No. 1 Pennsylvania Plaza
55th Floor
34th Street & 8th Avenue
MANHATTAN NY 10119
Tel: (212) 971-0800
CH,CS,E*,M*

Hewlett-Packard Co.
250 Westchester Avenue
WHITE PLAINS, NY 10604
Tel: (914) 684-6100
CM,CH,CS,E

Hewlett-Packard Co.
3 Crossways Park West
WOODBURY, NY 11797
Tel: (516) 921-0300
A,CH,CM,CS,E,MS

North Carolina
Hewlett-Packard Co.
5605 Roanne Way
P.O. Box 26500
GREENSBORO, NC 27420
Tel: (919) 852-1800
A,CH,CM,CS,E,MS

Ohio
Hewlett-Packard Co.
9920 Carver Road
CINCINNATI, OH 45242
Tel: (513) 891-9870
CH,CS,MS

Hewlett-Packard Co.
16500 Sprague Road
CLEVELAND, OH 44130
Tel: (216) 243-7300
A,CH,CM,CS,E,MS

Hewlett-Packard Co.
962 Crupper Ave.
COLUMBUS, OH 43229
Tel: (614) 436-1041
Eff: Nov. 25, 1983
675 Brooksedge Blvd.
WESTERVILLE, OH 43081
CH,CM,CS,E*

Hewlett-Packard Co.
330 Progress Rd.
DAYTON, OH 45449
Tel: (513) 859-8202
A,CH,CM,E*,MS

Oklahoma
Hewlett-Packard Co.
304 N. Meridian, Suite A
P.O. Box 75609
OKLAHOMA CITY, OK 73147
Tel: (405) 946-9499
A*,CH,E*,MS

Hewlett-Packard Co.
3840 S. 103rd E. Avenue, #100
P.O. Box 35747
TULSA, OK 74153
Tel: (918) 665-3300
A**,CH,CS,M*

Oregon
Hewlett-Packard Co.
9255 S. W. Pioneer Court
P.O. Box 328
WILSONVILLE, OR 97070
Tel: (503) 682-8000
A,CH,CS,E*,MS

Pennsylvania
Hewlett-Packard Co.
111 Zeta Drive
PITTSBURGH, PA 15238
Tel: (412) 782-0400
A,CH,CS,E,MP

Hewlett-Packard Co.
2750 Monroe Boulevard
P.O. Box 713
VALLEY FORGE, PA 19482
Tel: (215) 666-9000
A,CH,CM,E,M

South Carolina
Hewlett-Packard Co.
Brookside Park, Suite 122
1 Harbison Way
P.O. Box 21708
COLUMBIA, SC 29221
Tel: (803) 732-0400
CH,E,MS

Hewlett-Packard Co.
Koger Executive Center
Chesterfield Bldg., Suite 124
GREENVILLE, SC 29615
Tel: (803) 297-4120

Tennessee
Hewlett-Packard Co.
224 Peters Road, Suite 102
P.O. Box 22490
KNOXVILLE, TN 37922
Tel: (615) 691-2371
A*,CH,MS

Hewlett-Packard Co.
3070 Directors Row
MEMPHIS, TN 38131
Tel: (901) 346-8370
A,CH,MS

Texas
Hewlett-Packard Co.
4171 North Mesa
Suite C-110
EL PASO, TX 79902
Tel: (915) 533-3555
CH,E*,MS**

Hewlett-Packard Co.
10535 Harwin Drive
P.O. Box 42816
HOUSTON, TX 77042
Tel: (713) 776-6400
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
930 E. Campbell Rd.
P.O. Box 1270
RICHARDSON, TX 75080
Tel: (214) 231-6101
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
1020 Central Parkway South
P.O. Box 32993
SAN ANTONIO, TX 78216
Tel: (512) 494-9336
CH,CS,E,MS

Utah
Hewlett-Packard Co.
3530 W. 2100 South
SALT LAKE CITY, UT 84119
Tel: (801) 974-1700
A,CH,CS,E,MS

Virginia
Hewlett-Packard Co.
4305 Cox Road
GLEN ALLEN, VA 23060
P.O. Box 9669
RICHMOND, VA 23228
Tel: (804) 747-7750
A,CH,CS,E,MS

Washington
Hewlett-Packard Co.
15815 S.E. 37th Street
BELLEVUE, WA 98006
Tel: (206) 643-4000
A,CH,CM,CS,E,MP

Hewlett-Packard Co.
Suite A
708 North Argonne Road
SPOKANE, WA 99212
Tel: (509) 922-7000
CH,CS

West Virginia
Hewlett-Packard Co.
4604 MacCorkle Ave.
P.O. Box 4297
CHARLESTON, WV 25304
Tel: (304) 925-0492
A,MS

Wisconsin
Hewlett-Packard Co.
150 S. Sunny Slope Road
BROOKFIELD, WI 53005
Tel: (414) 784-8800
A,CH,CS,E*,MP

URUGUAY

Pablo Ferrando S.A.C. e I.
Avenida Italia 2877
Casilla de Correo 370
MONTEVIDEO
Tel: 80-2586
Telex: Public Booth 901
A,CM,E,M

VENEZUELA

Hewlett-Packard de Venezuela C.A.
3RA Transversal Los Ruices Norte
Edificio Segre 1, 2 & 3
Apartado 50933
CARACAS 1071
Tel: 239-4133
Telex: 251046 HEWPACK
A,CH,CS,E,MS,P

Hewlett-Packard de Venezuela C.A.
Calle-72-Entre 3H y 3Y, No. 3H-40
Edificio Ada-Evelyn, Local B
Apartado 2646
4001, **MARACAIBO**, Estado Zulia
Tel: (061) 80.304
C,E*

Hewlett-Packard de Venezuela C.A.
Calle Vargas Rondon
Edificio Seguros Carabobo, Piso 10
VALENCIA
Tel: (041) 51 385
CH,CS,P

Bioelectronica Medica C.A.
Calle Buen Pastor
Edif. Cota Mil-Piso 2 y Semi Sotano 1
Boleita Norte
Apartado 50710 ***CARACAS*** *1050A*
Tel: 239 84 41
Telex: 26518

ZIMBABWE

Field Technical Sales
45 Kelvin Road, North
P.B. 3458
SALISBURY
Tel: 705 231
Telex: 4-122 RH
C,E,M,P

July 1983 5952-6900

Indicates main office

HP distributors are printed in italics.

MANUAL PART NO. 92200-93005
Printed in U.S.A. December 1983
E1283

HEWLETT-PACKARD COMPANY
Data Systems Division
11000 Wolfe Road
Cupertino, California 95014