RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual

Order No. AA-JS13A-TC

RSX-11M-PLUS Version 4.0
Micro/RSX Version 4.0

Digital Equipment Corporation Maynard, Massachusetts

First Printing, May 1979
Revised, January 1982
Updated, April 1982
Revised, April 1983
Revised, July 1985
Revised, September 1987

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

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

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

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

The following are trademarks of Digital Equipment Corporation:

DEC	EduSystem	UNIBUS
DEC/CMS	IAS	VAX
DEC/MMS	MASSBUS	VAXcluster
DECnet	MicroPDP–11	VMS
DECsystem–10	Micro/RSX	VT
DECSYSTEM–20	PDP	
DECUS	PDT	
DECwriter	RSTS	digital
DIBOL	RSX	

ZK3074

HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS

USA & PUERTO RICO*	CANADA	INTERNATIONAL
Digital Equipment Corporation P.O. Box CS2008 Nashua, New Hampshire 03061	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario K2K 2A6 Attn: Direct Order Desk	Digital Equipment Corporation PSG Business Manager c/o Digital's local subsidiary or approved distributor

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575).

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminster, Massachusetts 01473.

This document was prepared using an in-house documentation production system. All page composition and make-up was performed by TEX, the typesetting system developed by Donald E. Knuth at Stanford University. TEX is a trademark of the American Mathematical Society.

Contents

Preface vii

Summary of Technical Changes xiii

Chapter 1 Preparing for and Running CDA

1.1 Making a Crash Dump Driver Part of Your System 1-1
1.1.1 Larger Systems 1-1
1.1.2 Smaller Systems 1-2
1.1.2.1 Selecting or Changing a Crash Dump Device 1-3
1.1.2.2 Displaying the Current Crash Dump Device 1-3
1.1.2.3 Unloading a Crash Dump Driver 1-4
1.2 Obtaining a Crash Dump 1-4
1.2.1 Larger Systems 1-4
1.2.1.1 Processor Trap and Executive Debugging Tool (XDT) in System 1-4
1.2.1.2 Processor Trap and XDT not in System 1-4
1.2.1.3 HALT or Infinite Loop 1-5
1.2.1.4 Crash Dump Driver Procedure 1-5
1.2.2 Smaller Systems 1-5
1.2.2.1 A System Crash with a Driver Loaded and XDT Unloaded 1-6
1.2.2.2 A System Crash With a Driver Loaded and XDT Loaded 1-6
1.2.2.3 A System Crash With Only XDT Loaded 1-6
1.3 Running CDA 1-6
1.3.1 CDA Installed and Command Line Interpreter (CLI) is MCR 1-7
1.3.2 CDA Installed and CLI is DCL 1-7
1.3.3 CDA Not Installed and CDA.TSK in System Directory 1-7
1.3.4 CDA Not Installed and CDA.TSK in Directory for Current User Identification Code (UIC) 1-7
1.3.5 DCL Command to Run CDA 1-7

Chapter 2 Command Lines

2.1 CDA Command Lines 2-1
2.1.1 CDA Command Line Switches 2-3
2.1.1.1 Analysis Switches 2-3
2.1.1.2 Function Switches 2-7
2.1.2 CDA Command Line Examples 2-10
2.2 The DCL Command ANALYZE/CRASH_DUMP 2-11
2.2.1 ANALYZE/CRASH_DUMP Command Qualifiers 2-12
2.2.1.1 Command Qualifiers 2-12
2.2.1.2 Crash-input File Qualifiers 2-15
2.2.2 ANALYZE/CRASH_DUMP Command Examples 2-20

Chapter 3 Analysis Listings

3.1 System Information 3-1
3.1.1 Volatile Registers 3-2
3.1.2 Kernel Stack 3-4
3.1.3 Low Core Memory Dump 3-4
3.1.4 System Common 3-5
3.1.5 System Common Alphabetized Dump 3-10
3.1.6 Pool Statistics 3-12
3.1.7 Assign Table 3-14
3.1.8 Group Global Event Flags 3-14
3.1.9 Error Log Packets 3-14
3.2 Optional Information 3-15
3.2.1 Active Tasks 3-15
3.2.2 Task Headers 3-21
3.2.3 Command Line Interpreter Parser Block (CPB) 3-24
3.2.4 Partition Information 3-24
3.2.5 Common Block Directory 3-29
3.2.6 Device Information 3-30
3.2.7 System Task Directory 3-36
3.2.8 Pool Dump 3-36
3.2.9 Task Dump 3-39
3.2.10 Clock Queue 3-41
3.2.11 Controller Information 3-42
3.2.12 Kernel Data Space 3-44
3.2.13 Kernel Instruction Space 3-44
3.2.14 Cache Dump 3-44

Chapter 4 Interpreting a Crash Dump Listing

4.1 Helpful Concepts 4-1
4.1.1 Determining What Was Mapped 4-1
4.1.2 Interpreting The Kernel Stack 4-2

Appendix A CDA Messages

Appendix B Definitions of Values in Low Core Memory Dump and System Common Alphabetized Dump

B.1 Low Core Memory Dump B-1
B.2 System Common Alphabetized Dump B-12

Appendix C System Data Structures and Symbolic Definitions

Appendix D Common Error Code Definitions

Index

Examples

3–1 Volatile Registers 3-3
3–2 Kernel Stack 3-4
3–3 Low Core Memory 3-5
3–4 System Common 3-7
3–5 System Common Alphabetized Dump 3-10
3–6 Pool Statistics 3-13
3–7 Logical Assignment Table 3-14
3–8 Group Global Event Flags 3-14
3–9 Error Log Packets 3-15
3–10a Active Tasks with Receive Queue, AST Queue, and Receive-by-Reference Queue Listings 3-19
3–10b Active Tasks with OCB Listing 3-21
3–11 Task Headers (Truncated) 3-23
3–12 CLI Parser Blocks 3-24
3–13 Partition Information 3-25
3–14 Partition Control Blocks and Attachment Descriptors 3-28

3–14 Common Block Directory 3-30
3–16 Device Information and I/O Packet (Truncated) 3-35
3–17 System Task Directory (Truncated) 3-36
3–18 Pool Dump (Truncated) 3-37
3–19 Task Dump (Truncated) 3-40
3–20 Clock Queue 3-42
3–21 Controller Information 3-43
3–22 Kernel Data Space 3-44
3–23 Kernel Instruction Space 3-44
3–24 Cache Information 3-45
4–1 Kernel Page Address Registers 4-2

Tables

1–1 Crash Dump Devices for Non-Pregenerated RSX–11M–PLUS Systems 1-2
1–2 Crash Dump Devices for Micro/RSX and Pregenerated RSX–11M–PLUS Systems 1-3
2–1 File Default Values 2-3
2–2 Summary of CDA Analysis Switches 2-3
2–3 Summary of CDA Function Switches 2-8
2–4 Summary of ANALYZE/CRASH_DUMP Command Qualifiers 2-13
2–5 Summary of ANALYZE/CRASH_DUMP Crash-input Qualifiers 2-15
C–1 Summary of System Data Structure Macros C-1

Preface

Manual Objectives

This manual describes the operation of the Crash Dump Analyzer (CDA). It does not attempt to describe the operation of the RSX–11M–PLUS Executive or the significance of the individual data structures. The *RSX–11M–PLUS and Micro/RSX Executive Reference Manual* and the *RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver* describe these data structures.

Intended Audience

This manual is intended for system managers who are responsible for interpreting system failures and for system operators who run CDA to generate dumps. Understanding CDA output requires a working knowledge of assembly language programming and of the Executive data structures.

Structure of This Document

Chapter 1 explains the function of the Crash Dump Analyzer. It tells how to make a crash dump driver part of your system, how to get a crash dump, and how to run CDA to get the listing.

Chapter 2 describes the two ways that you can use CDA: by entering the CDA command line or by entering the DCL command ANALYZE/CRASH_DUMP. The chapter begins by describing the CDA command line format, including command line specifications and switches. Two summary tables provide quick reference on switch operation. The chapter concludes with a description of the DCL command ANALYZE/CRASH_DUMP.

Chapter 3 consists of examples and descriptions of CDA output listings.

Chapter 4 contains helpful hints for interpreting CDA output listings.

Appendix A contains a short description of each CDA error message.

Appendix B defines values displayed in the Low Core Memory Dump and in the System Common Alphabetized Dump.

Appendix C lists system macros that supply symbolic offsets for system data structures for RSX–11M–PLUS.

Appendix D lists error code definitions for the Bugcheck facility.

Associated Documents

Refer to the *RSX-11M-PLUS Information Directory and Master Index* for a brief description of each manual in the RSX-11M-PLUS documentation set.

Conventions Used in This Document

The following conventions are used in this manual:

Convention	Meaning
>	A right angle bracket is the default prompt for the Monitor Console Routine (MCR), which is one of the command interfaces used on RSX-11M-PLUS systems. All systems include MCR.
$	A dollar sign followed by a space is the default prompt of the DIGITAL Command Language (DCL), which is one of the command interfaces used on RSX-11M-PLUS and Micro/RSX systems. Many systems include DCL.
MCR>	This is the explicit prompt of the Monitor Console Routine (MCR).
DCL>	This is the explicit prompt of the DIGITAL Command Language (DCL).
xxx>	Three characters followed by a right angle bracket indicate the explicit prompt for a task, utility, or program on the system.
UPPERCASE	Uppercase letters in a command line indicate letters that must be entered as they are shown. For example, utility switches must always be entered as they are shown in format specifications.
command abbreviations	Where short forms of commands are allowed, the shortest form acceptable is represented by uppercase letters. The following example shows the minimum abbreviation allowed for the DCL command DIRECTORY: `$ DIR`
lowercase	Any command in lowercase must be substituted for. Usually the lowercase word identifies the kind of substitution expected, such as a filespec, which indicates that you should fill in a file specification. For example: `filename.filetype;version` This command indicates the values that comprise a file specification; values are substituted for each of these variables as appropriate.

Convention	Meaning
/keyword, /qualifier, or /switch	A command element preceded by a slash (/) is an MCR keyword; a DCL qualifier; or a task, utility, or program switch. Keywords, qualifiers, and switches alter the action of the command they follow.
parameter	Required command fields are generally called parameters. The most common parameters are file specifications.
[option]	Square brackets indicate optional entries in a command line or a file specification. If the brackets include syntactical elements, such as periods (.) or slashes (/), those elements are required for the field. If the field appears in lowercase, you are to substitute a valid command element if you include the field. Note that when an option is entered, the brackets are not included in the command line.
[, . . .]	Square brackets around a comma and an ellipsis mark indicate that you can use a series of optional elements separated by commas. For example, (argument[, . . .) means that you can specify a series of optional arguments by enclosing the arguments in parentheses and by separating them with commas.
{ }	Braces indicate a choice of required options. You are to choose from one of the options listed.
:argument	Some parameters and qualifiers can be altered by the inclusion of arguments preceded by a colon. An argument can be either numerical (COPIES:3) or alphabetical (NAME:QIX). In DCL, the equal sign (=) can be substituted for the colon to introduce arguments. COPIES=3 and COPIES:3 are the same.
( )	Parentheses are used to enclose more than one argument in a command line. `SET PROT = (S:RWED,O:RWED)` Commas are used as separators for command line parameters and to indicate positional entries on a command line. Positional entries are those elements that must be in a certain place in the command line. Although you might omit elements that come before the desired element, the commas that separate them must still be included.

Convention	Meaning
[g,m] [directory]	The convention [g,m] signifies a User Identification Code (UIC). The g is a group number and the m is a member number. The UIC identifies a user and is used mainly for controlling access to files and privileged system functions. This may also signify a User File Directory (UFD), commonly called a directory. A directory is the location of files. Other notations for directories are: [ggg,mmm], [gggmmm], [ufd], [name], and [directory]. The convention [directory] signifies a directory. Most directories have 1- to 9-character names, but some are in the same [g,m] form as the UIC. Where a UIC, UFD, or directory is required, only one set of brackets is shown (for example, [g,m]). Where the UIC, UFD, or directory is optional, two sets of brackets are shown (for example, [[g,m]]).
filespec	A full file specification includes device, directory, file name, file type, and version number, as shown in the following example: `DL2:[46,63]INDIRECT.TXT;3` Full file specifications are rarely needed. If you do not provide a version number, the highest numbered version is used. If you do not provide a directory, the default directory is used. Some system functions default to particular file types. Many commands accept a wildcard character (*) in place of the file name, file type, or version number. Some commands accept a filespec with a DECnet node name.
.	A period in a file specification separates the file name and file type. When the file type is not specified, the period may be omitted from the file specification.
;	A semicolon in a file specification separates the file type from the file version. If the version is not specified, the semicolon may be omitted from the file specification.
@	The at sign invokes an indirect command file. The at sign immediately precedes the file specification for the indirect command file, as follows: `@filename[.filetype;version]`

Convention	Meaning
. . .	A horizontal ellipsis indicates the following: • Additional, optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis shows where elements of command input or statements in an example or figure have been omitted because they are irrelevant to the point being discussed.
boldface	Boldface type indicates that the term is being defined; for example, command parameters and glossary entries.
italics	Italic type indicates emphasis, a book title, or that a new term is being used in the text.
KEYNAME	This typeface denotes one of the keys on the terminal keyboard. For example, the RETURN key.
"print" and "type"	As these words are used in the text, the system prints and the user types.
black ink	In examples, what the system prints or displays is printed in black.
red ink	In interactive examples, what the user types is printed in red. System responses appear in black.
[xxx]	A symbol with a 1- to 3-character abbreviation, such as [x] or [RET], indicates that you press a key on the terminal. For example, [RET] indicates the RETURN key, [LF] indicates the LINE FEED key, and [DEL] indicates the DELETE key.
[CTRL/a]	The symbol [CTRL/a] means that you are to press the key marked CTRL while pressing another key. Thus, [CTRL/Z] indicates that you are to press the CTRL key and the Z key together in this fashion. [CTRL/Z] is echoed on some terminals as ^Z. However, not all control characters echo.

Summary of Technical Changes

The following sections list features, commands, qualifiers, error messages, and restrictions that are new to the Crash Dump Analyzer Utility or have been modified for the RSX–11M–PLUS and Micro/RSX Version 4.0 operating systems. These new or modified features are documented in this revision of the *RSX–11M–PLUS and Micro/RSX Crash Dump Analyzer Reference Manual.*

In addition, major changes to the organization of the manual are included at the end of this summary.

New or Modified Features

The Crash Dump Analyzer now displays information on cache memory at the time of a system failure.

New or Modified Qualifiers

The Crash Dump Analyzer has the following new or modified qualifiers:

- The /CACHE qualifier to a CDA command or the DCL command ANALYZE/CRASH_DUMP.
- The :CSR qualifier to the MCR command SET /CRASH_DEVICE.

(The :CSR qualifier is not new to Version 4.0 of RSX–11M–PLUS or Micro/RSX but is documented here for the first time.)

/CACHE

The /CACHE qualifier produces a listing of information about cache memory at the time of a system failure.

:CSR

The :CSR qualifier lets you specify a control and status register for the crash dump device, if it is not yet installed on your system.

New or Modified Error Messages

The CDA produces the following new error messages:

- Cache region is not in memory.
- Cache region was not found.
- Loop found in cache extent descriptors at nnnnn.
- Loop found in cache statistics blocks at nnnnn.
- Loop found in linked list in pool at nnnnn.
- No devices have caching associated with this region.
- Requested partition is not a cache region.

Changes to the Document

Chapter 1 of the *RSX-11M-PLUS and Micro/RSX Crash Dump Analyzer Reference Manual* has been reorganized to reflect the steps needed before you can request crash dumps: loading a crash dump driver, obtaining a crash dump after the system fails, and running the CDA program to obtain listings of memory at the time of the system failure.

Chapter 1
Preparing for and Running CDA

The Crash Dump Analyzer (CDA) produces readable listings of the contents of memory after a system crash, or failure. These listings help you track down the cause of the system failure so you can fix the problem. The CDA is a nonprivileged task that any user can run.

After the system fails, you must pass control to a special program called a *crash dump driver.* Section 1.1 tells you how to make such a driver a part of your system. Section 1.2 tells how to transfer control to the crash dump driver after a system crash.

The crash dump driver writes (dumps) the contents of memory onto a disk or tape, which you specify, called the crash dump device. Once memory has been dumped, you reboot the system and run CDA to format and print the listings. Section 1.3 tells how to run CDA.

As with all other utilities, you can enter CDA command lines directly from the terminal or from an indirect command file. However, CDA indirect command files must not contain a reference to another command file.

1.1 Making a Crash Dump Driver Part of Your System

How you make a crash dump driver part of your system depends on the size of the system. A larger system is the non-pregenerated RSX-11M-PLUS system. Smaller systems include Micro/RSX and pregenerated RSX-11M-PLUS systems.

1.1.1 Larger Systems

For a non-pregenerated RSX-11M-PLUS system, you select one of several available crash dump drivers during system generation. See the *RSX-11M-PLUS System Generation and Installation Guide* for instructions.

Briefly, you select the crash notification device and the crash dump device. The system then builds a crash dump driver into the Executive. Thereafter, when the system fails, the crash dump driver displays a message on the crash notification device and writes the contents of memory onto the specified crash dump device. If you decide to change the crash dump or crash notification devices, you must perform another system generation.

Since the crash dump driver overwrites the contents of the crash dump device, you should not specify the system device as the crash dump device. Also, the following fixed disks cannot serve as crash dump devices: RA80, RA81, RD51, RD52, RD53, RD54, and RCF25. Table 1–1 lists the devices that can be specified for crash dump devices on non-pregenerated RSX–11M–PLUS Systems.

Table 1–1: Crash Dump Devices for Non-Pregenerated RSX–11M–PLUS Systems

Device	Mnemonic
RP04/RP05/RP06 disk packs	DB:
DECtape II (TU58)	DD:
RK05/J/F disk cartridge	DK:
RL01/RL02 disks	DL:
RK06/RK07 disk cartridges	DM:
RM02/RM03/RM05 disk packs	DR:
DECtape (TU56)	DT:
RC25 removable disk pack	DU:
RA60 disk packs	DU:
RX50/RX33 diskette	DU:
RX02 diskette	DY:
TU45/TU16/TE16/TU77 magnetic tapes	MM:
TS11/TU80/TSV05/TK25 magnetic tapes	MS:
TS03/TU10/TE10 magnetic tapes	MT:
TK50 magnetic tapes	MU:

1.1.2 Smaller Systems

The smaller systems — Micro/RSX operating systems with the Advanced Programmer's Kit and pregenerated RSX–11M–PLUS operating systems — support loadable crash dump drivers. You can load a driver for a crash dump device while the system is operating. This saves space, as you can unload the driver when you do not need crash dump support. Table 1–2 lists the crash dump devices you can use with the smaller systems.

Table 1–2: Crash Dump Devices for Micro/RSX and Pregenerated RSX–11M–PLUS Systems

Device	Mnemonic
RL01/RL02 disks	DL:
RD51/RD52/RD53/RD54 disks	DU:
RX50/RX33 diskettes	DU:
TSV05/TK25 magnetic tape	MS:
TK50 magnetic tapes	MU:

1.1.2.1 Selecting or Changing a Crash Dump Device

You select (or change) a crash dump driver with the following DCL command:

```
SET SYSTEM /CRASH_DEVICE:ddn:
```

If the device is not currently on the system, you use the REGISTER qualifier to specify the control and status register (CSR) of the desired device:

```
SET SYSTEM /CRASH_DEVICE:ddn/REGISTER:csraddr
```

When the crash driver is successfully loaded, you receive the following message:

```
SET -- Crash device ddn: has been successfully loaded
```

If the device you specified is not in the current system, the following error message is displayed:

```
SET -- Device not in system
```

On RSX–11M–PLUS pregenerated systems, you can also use the following MCR command:

```
SET /CRASH_DEVICE:ddnn[:CSRaddr]
```

or

```
SET /CRASHDEV:ddnn[:CSRaddr]
```

1.1.2.2 Displaying the Current Crash Dump Device

To display the current crash dump device unit, use the following DCL command:

```
SHOW SYSTEM /CRASH_DEVICE
```

The system then displays:

```
CRASHDEV=ddn:
```

1.1.2.3 Unloading a Crash Dump Driver

Use the following DCL command to unload a crash dump driver when you no longer want crash dump support:

```
SET SYSTEM /NOCRASH_DEVICE
```

The following MCR command unloads a crash dump driver:

```
SET /NOCRASH_DEVICE
```

In response, the system displays:

```
SET -- Crash device ddn: is being unloaded
SET -- WARNING, Crash dump support is inactive
```

When there is no crash dump driver resident in memory, the Bugcheck facility services system crashes. See Appendix D for a list of error code definitions used by Bugcheck.

1.2 Obtaining a Crash Dump

To obtain a crash dump, you must transfer control to the crash dump driver after a system crash. Again, the process for doing this depends on the type of system.

1.2.1 Larger Systems

For non-pregenerated RSX–11M–PLUS systems, transferring processor control to the crash dump driver depends on how the failure occurred and whether you built the Executive Debugging Tool (XDT) into your system during system generation. System crashes can result from any of the following causes:

- The processor encounters a program condition that causes it to trap to location 40 or to XDT.
- An infinite loop condition occurs.
- The processor encounters an unintentional HALT instruction in kernel mode (000000).

1.2.1.1 Processor Trap and Executive Debugging Tool (XDT) in System

When a program condition causes a processor trap and XDT is included in your system, control transfers automatically to XDT. You can then type X at the console terminal, as follows:

```
XDT>X
```

This causes XDT to transfer control to the crash dump driver. Refer to the *RSX–11M–PLUS and Micro/RSX XDT Reference Manual* for a description of XDT.

1.2.1.2 Processor Trap and XDT not in System

If your system does not include XDT, a processor trap causes control to be transferred directly to the crash dump driver.

1.2.1.3 HALT or Infinite Loop

When a system failure is the result of a HALT instruction or an infinite loop condition, you must restart the processor manually at location 40.

1.2.1.4 Crash Dump Driver Procedure

Regardless of how control is transferred, once the processor enters the crash dump routine, the routine prints the following informational message on the crash notification device:

```
CRASH-CONT WITH SCRATCH MEDIA ON ddnn
```

After displaying the message, the crash dump routine halts the processor so you can put the crash dump device on line. When the device is on line, restart the processor. (For many processors, you restart by pressing the CONTINUE switch on the processor console. See the *PDP-11 Processor Handbook* for your processor for specific instructions on how to restart the processor.) The crash dump routine then dumps memory on the crash dump device and halts the processor when the dump finishes. The volume in the crash dump device now contains a binary representation of the contents of memory at the time of the crash. These contents are then input to CDA. You can then reboot the system and run CDA to analyze the dump.

If you specified an illegal device as the crash dump device during system generation (a device not listed in Table 1-2), the crash dump driver displays the following message on the crash notification device:

```
CRASH -- ILLEGAL CRASH DEVICE
```

After displaying the message, the crash dump driver halts. The illegal crash device error occurs if you specified a fixed-media device as the crash dump device. If you have a removable media device on the same controller, you can switch the physical unit number plugs on the devices to assign the removable media device to the crash device. Then press the Continue key on the operator console (or other appropriate action to restart your processor) and the crash dump driver will attempt the dump again.

1.2.2 Smaller Systems

When a Micro/RSX or pregenerated RSX-11M-PLUS operating system crashes, the reaction of the system depends on the type of crash support that is loaded when the crash occurs:

- A crash dump driver is loaded but XDT is not
- Both a crash dump driver and XDT are loaded
- XDT is loaded but a crash dump driver is not

1.2.2.1 A System Crash with a Driver Loaded and XDT Unloaded

If a system fails when a crash dump driver is loaded but XDT is not loaded, the crash dump routine notifies you of the failure with the following message:

```
CRASH -- CONT WITH SCRATCH MEDIA ON ddn
```

After displaying this message, the crash dump driver halts the hardware processor so that you can make sure there is a scratch media in the crash device. When you have the crash device ready, restart the processor. (For Micro/RSX systems, press the P key followed by a carriage return. For other processors, perform the action defined in the *PDP-11 Processor Handbook* to restart.

In response to your command to proceed, the crash dump driver dumps memory to the designated crash dump device. When the dump is completed, the processor is again halted. During the memory dump, the processor Run light is on; when the dump is completed, the processor Run light goes off.

At this point, the medium in the crash dump device contains a binary representation of the contents of memory at the time the system crash occurred. This memory dump is the input to CDA.

1.2.2.2 A System Crash With a Driver Loaded and XDT Loaded

If a system fails when a crash driver and XDT are loaded, control is transferred to XDT. After you use XDT to debug the system, if you want to obtain a crash dump, press the X key followed by a carriage return. The following message is then displayed:

```
CRASH -- CONT WITH SCRATCH MEDIA ON ddn
```

Now you can follow the procedure in Section 1.2.2.1 to obtain the crash dump.

1.2.2.3 A System Crash With Only XDT Loaded

If a system fails when XDT is loaded but a crash driver is not loaded, control is transferred to XDT. However, when you enter the X command, the following is displayed:

```
SYSTEM FAULT DETECTED AT PC=xxxxxx FACILITY=xxxxxx ERROR CODE=xxxxxx
CRASH -- CRASH DRIVER NOT LOADED
```

There is no recovery from this situation. You may want to reboot the system and duplicate the problem with a crash driver installed.

1.3 Running CDA

After the crash dump driver has written the contents of memory to the crash dump device, you reboot the system and run CDA to obtain its formatted listing of memory.

There are several ways to run CDA. You can run it as either an installed or an uninstalled task, and you can run it from either the DCL or MCR command line interpreter (CLI). This section describes the alternative ways of running CDA.

1.3.1 CDA Installed and Command Line Interpreter (CLI) is MCR

If CDA is an installed task on your system, you can enter the CDA command line at the CLI prompt. After CDA processes your command, the CLI prompt returns. In the following example, MCR is the CLI:

```
>CDA CRASH_DUMP.LST,COPY.CDA=[1,54]/STB,DR5: RET
>
```

1.3.2 CDA Installed and CLI is DCL

If CDA is installed and you want to enter commands directly to it, invoke the command level of the CDA utility by typing CDA and and pressing RETURN. When you are finished using CDA, you exit from CDA by pressing CTRL/Z, which returns control to the CLI. In the following example, DCL is the CLI:

```
$ CDA RET
CDA>command line RET
CDA>command line RET
CDA>^Z
$
```

1.3.3 CDA Not Installed and CDA.TSK in System Directory

If CDA is an uninstalled task, the system has to find and install the CDA task image file before it can run CDA. Therefore, the command you use depends upon the location of the CDA task image file (CDA.TSK). If CDA.TSK is in the system directory, type the following:

```
RUN $CDA RET
CDA>command line RET
CDA>
```

1.3.4 CDA Not Installed and CDA.TSK in Directory for Current User Identification Code (UIC)

If CDA.TSK is present in the directory that corresponds to the current UIC on the default system device (the current directory for the terminal from which the command is entered), you can run CDA by entering the following command:

```
RUN CDA RET
CDA>command line RET
```

1.3.5 DCL Command to Run CDA

Finally, you can run CDA by using the DCL command ANALYZE/CRASH_DUMP:

```
$ ANALYZE/CRASH_DUMP RET
```

If your CLI is MCR, but your terminal also supports DCL, you can enter the ANALYZE /CRASH_DUMP command by typing DCL and a space before the command. For example:

```
>DCL ANALYZE/CRASH_DUMP RET
```

Chapter 2 shows you how to use CDA command lines and the ANALYZE/CRASH_DUMP command.

Chapter 2
Command Lines

CDA commands control how the Crash Dump Analyzer processes a memory dump and how it formats the output listings that it generates. You can use CDA command lines to enter commands directly to the CDA utility or, if your terminal supports the DIGITAL Command Language (DCL), you can use the DCL command ANALYZE/CRASH_DUMP to run CDA. This chapter describes CDA command lines and the ANALYZE/CRASH_DUMP command by showing the format of the command lines, the command specifications and qualifiers, and examples of how the commands work.

2.1 CDA Command Lines

This section shows the CDA command line format, lists and describes command line switches, and provides some examples of CDA command lines.

The CDA command line has the following format:

```
CDA>[listfile/sw],[binaryfile/sw]=[symbolfile/STB],crash-input[/sw]
```

The CDA command line specifies the input to CDA and the output from CDA. The specifications to the left of the equal sign in the command line are output specifications, and those on the right side of the equal sign are input specifications.

You must include at least one output specification and one input specification in the command line. For output from CDA, you can specify a list file only, a binary file only, or both a list file and a binary file. For input to CDA, you must specify the crash-input, but the symbol file specification is optional.

Output file specifications are position dependent. Position dependent means that when you include both output specifications, you must place them in the positions shown in the command line. If you omit the list file, you must place a comma before the binary file specification.

Input file specifications are position independent and can appear in any order.

The remainder of this section describes CDA command line specifications.

Output Specifications:

listfile

The output specification of the formatted CDA analysis listings. You can use either a device or a file as the list file specification. If you specify a file, CDA creates the file and writes the output listings to the file. By default, CDA then spools the file to the system line printer queue, unless you specify otherwise. If you specify a device for the list file, CDA displays or prints its output listings on that specific device. For example, if you specify your terminal (TTnn: or TI:) as the list file, CDA displays the output listings on your terminal. Chapter 3 describes the analysis listings that CDA generates.

binaryfile

The file specification for the optional binary file. This file is a copy of the binary data that the crash dump routine wrote on the crash dump device. It allows you to selectively create a historical record of crash dumps. If you create this file during an initial analysis, you can use it for input to CDA at a later time. Since the crash dump routine overwrites the information on the crash dump volume with each successive dump, this feature allows you to use a single volume for all crash dumps.

If the crash dump device on your system is a secondary storage or sequential device, you can reduce CDA analysis time by copying the crash input to a binary file on another device. Then you can use the binary file as input to CDA for analysis.

Input Specifications:

symbolfile/STB

The file specification of the symbol table file for the crashed system. The /STB switch is an integral part of this file specification because CDA uses the data in the symbol table file to format the binary memory dump into readable formats. If you omit this file specification and switch, CDA uses the default symbol table file, which is the file named RSX11M.STB in the directory that corresponds to the current UIC.

crash-input

The source of the binary input to CDA. This specification can be either a device name (the crash dump device) or a binary file that was created during a previous CDA analysis. However, if the crash-input specification is a binary file, you cannot also include a binary file output specification in the command line.

Switches:

/sw

An optional CDA switch. The list file, binary file, and crash-input file specifications can include optional switches that modify CDA action. Each specification in the command line has its own switches. Section 2.1.1 describes the CDA switches and lists the specification to which they apply.

File specifications in the CDA command line can appear in complete Files-11 format, with device name, directory, file name, file type, and version number. When you omit any of these elements, CDA uses the defaults shown in Table 2-1. However, not all of the elements in file specifications have defaults.

Table 2-1: File Default Values

	Default Value			
File	**Device**	**Directory**	**File Name**	**File Type**
List file	SY:	Current	None	.LST
Binary file	SY:	Current	None	.CDA
Symbol file/STB	SY:	Current	RSX11M	.STB
Crash-input	SY:	Current	None	.CDA

See Section 2.1.2 for examples of CDA command lines, which include examples that show how CDA uses default file types.

2.1.1 CDA Command Line Switches

Two kinds of command line switches — analysis switches and function switches — allow you to control CDA operation.

Analysis switches determine which analysis routines CDA applies to the crash input. Thus, you can select the types of data that you want CDA to output. For example, analysis switches can list information about all of the devices in the system, or they can list information about active devices only.

Function switches provide a number of options for controlling CDA output. For example, function switches can terminate an analysis after CDA encounters a specified number of errors, or they can limit the number of pages of output listings.

Both types of switches are file specific. That is, each switch applies to a particular file and may not be used without that file or with any other file.

2.1.1.1 Analysis Switches

Table 2-2 summarizes the analysis switches and gives brief descriptions of their effects. Some of the switches in Table 2-2 have synonyms or alternate mnemonics. These are shown under each switch. Expanded descriptions of each switch follow the table.

Table 2-2: Summary of CDA Analysis Switches

Switch	Function	Applies to File
/ACT /ATL	List the contents of the Task Control Block (TCB) for each active task	Crash-input
/ADV	Lists information for all devices in the system	Crash-input

Table 2–2 (Cont.): Summary of CDA Analysis Switches

Switch	Function	Applies to File
/ALL	Lists the output of all analysis routines	Crash-input
/CACHE[:[region]]	Lists information on cache region	Crash-input
/CLI /CPB	List the contents of the CLI Parser Blocks in the system	Crash-input
/CLQ	Lists the contents of the clock queue	Crash-input
/CTL	Lists information for each device controller	Crash-input
/DEV /DCB /SCB /UCB	List information for all active devices in the system	Crash-input
/DUMP:a:b:[c] /DMP:a:b:[c]	List the contents of physical memory between address a and address b; (c is an optional virtual starting address)	Crash-input
/HDR	Lists the contents of the task headers for each task resident in memory	Crash-input
/KDS:a:b	Lists the contents of the kernel data space from virtual address a to virtual address b (RSX–11M–PLUS systems only)	Crash-input
/KIS:a:b	Lists the contents of kernel instruction space from virtual address a to virtual address b (RSX–11M–PLUS systems only)	Crash-input
/PCB /PAR	List the contents of each Partition Control Block	Crash-input
/POOL	Lists the contents of the system's pool	Crash-input
/SECPOL[:a:b]	Lists the contents of system secondary pool (RSX–11M–PLUS systems only)	Crash-input
/-SYS	Suppresses listing of the system information	Crash-input
/TASK:name:a:b /TAS:name:a:b /TSK:name:a:b	List the contents of task "name" between virtual address a and virtual address b; list the contents of task data space (if task includes data space) on RSX–11M–PLUS	Crash-input
/TCB /TAL /STD	List the contents of the TCB for every task in the System Task Directory	Crash-input
/TDS:name:a:b	Lists the contents of task data space (RSX–11M–PLUS only)	Crash-input

Table 2–2 (Cont.): Summary of CDA Analysis Switches

Switch	Function	Applies to File
/TIS:name:a:b	Lists the contents of task instruction space (RSX–11M–PLUS systems only)	Crash-input

/ACT or /ATL (Task Control Blocks for Active Tasks)

File: Crash-input

Effect: CDA lists the contents of the Task Control Block (TCB) for each active task.

/ADV (All Devices)

File: Crash-input

Effect: CDA lists the contents of the control blocks for all devices in the system. To list active devices, use the /DEV switch.

/ALL (All Analysis Routines)

File: Crash-input

Effect: CDA applies all of its analysis routines (except those associated with memory and task dumps) to the specified crash-input. The output from these routines is listed in the following order:

1. System information
2. Active tasks information
3. Task headers information
4. Partition information
5. Common Block Directory entries
6. Device information
7. Clock queue contents
8. Device controller information
9. Pool contents
10. Cache region listing

/CACHE[:[region]] (Cache Memory)

File: Crash-Input

Effect: CDA lists information about the contents of cache memory at the time of the dump. The region is the name of the cache region specified in the SET CACHE command or the MOUNT command with the CACHE qualifier.

/CLI or /CPB (Command Line Interpreter Parser Blocks)

File: Crash-input

Effect: CDA lists the contents of all Command Line Interpreter Parser Blocks (CPBs) in the system.

/CLQ (Clock Queue)

File: Crash-input

Effect: CDA lists the contents of the clock queue.

/CTL (Device Controllers)

File: Crash-input

Effect: CDA lists the contents of the controller table and Controller Request Block (KRB) for each device controller in the system.

/DEV, /DCB, /SCB, or /UCB (Devices in System)

File: Crash-input

Effect: CDA scans the system device tables and lists the contents of the control blocks for each active device in the system. To list all devices, use the /ADV switch.

/DUMP:a:b:[c] or /DMP (Physical Memory)

File: Crash-input

Effect: If only a and b are specified, CDA dumps the contents of physical addresses a through b, inclusive, and labels them with their physical addresses. If a, b, and c are specified, CDA dumps the contents of physical addresses a through b but labels them with dummy virtual addresses, starting at the address specified by c.

CDA allows you to specify a virtual starting address because RSX–11M–PLUS and Micro/RSX systems use physical memory in terms of virtual addresses. If you dump physical memory labeled with the corresponding virtual addresses, you do not have to translate physical addresses to virtual addresses as you read the dump.

/HDR (Headers for Memory-Resident Tasks)

File: Crash-input

Effect: CDA lists the contents of the task headers for each task resident in memory.

/KDS:a:b (Kernel Data Space)

File: Crash-input

Effect: CDA lists the contents of kernel data space between the virtual addresses a and b, inclusive.

/KIS:a:b (Kernel Instruction Space)

File: Crash-input

Effect: CDA lists the contents of kernel instruction space between the virtual addresses a and b, inclusive.

/PCB or /PAR (Partition Control Blocks)

File: Crash-input

Effect: CDA outputs a map that lists all the occupants of memory and the contents of each Partition Control Block (PCB).

/POOL (System Pool)

File: Crash-input

Effect: CDA lists the system pool in octal, Radix–50, and ASCII.

/SECPOL[:a:b] (Secondary Pool)

File: Crash-input

Effect: Lists the contents of the secondary pool on RSX–11M–PLUS systems.

/STD, /TCB, or /TAL (System Task Directory)

File: Crash-input

Effect: CDA lists the contents of all of the Task Control Blocks in the System Task Directory (STD) at the time of the crash.

/-SYS (System Information)

File: Crash-input

Effect: CDA suppresses the system information listing.

/TASK:name:a:b, /TAS, or /TSK (Task Virtual Address Space)

File: Crash-input

Effect: CDA lists the virtual address space from the 16-bit virtual address a through b for the task specified by "name." If you do not specify addresses, CDA lists the task's entire virtual address space.

/TDS:name[:a:b] (Task Data Space)

File: Crash-input

Effect: CDA lists the contents of the task data space between the virtual addresses a and b, inclusive. If you do not specify addresses, CDA lists the entire task data space.

/TIS:name:a:b (Task Instruction Space)

File: Crash-input

Effect: CDA lists the contents of the task instruction space between the virtual addresses a and b, inclusive. If you do not specify addresses, CDA lists the entire task instruction space.

2.1.1.2 Function Switches

Table 2–3 summarizes the function switches and gives brief descriptions of their effects. Expanded descriptions of each switch follow the table.

Table 2–3: Summary of CDA Function Switches

Switch	Function	Applies to File	Default[1]
/BL:n	Identifies the starting block number of the crash-input device; the value of n must be less than 65535	Crash-input	n=1
/DENS:n :HIGH :LOW	Sets density of crash input tape to 800 or 1600 bits per inch (bpi)	Crash-input	n=800
/EXIT:n	Terminates analysis after encountering n analysis errors	List file	–
/LIMIT:n	Limits output listing to n pages	List file	n=300.
/LINES:n	Limits page length to n lines	List file	n=60.
/MEMSIZ:n	Saves nK words of memory from crash in a binary file	Binary file	n=124.
/KMR	Forces the assignment of kernel address register values for the crashed system	Crash-input	/-KMR
/-SP	Does not print analysis output listing	List file	/SP
/STB	Identifies the file specification that contains the Executive symbol table	Symbol file	–

[1] n can be expressed as an octal or decimal number. A decimal point (.) following the number denotes decimal.

/BL:n (Identify Starting Block Number)

File: Crash-input

Effect: CDA reads the dump from the input device beginning at block n. If the crash dump device is not a disk or a DECtape, CDA ignores this switch.

Default: n = 1

/DENS:n (Set Tape Density)

:HIGH
:LOW

File: Crash-input

Effect: CDA reads the crash input tape at the density specified: 800 or 1600 bpi. You can also use LOW to indicate 800 bpi or HIGH to indicate 1600 bpi.

Default: n=800

/EXIT:n (Exit After n Errors)

File: List file

Effect: CDA maintains an error count. As it encounters inconsistencies in the system data structures, it increments this count. If you specify the /EXIT:n switch, CDA terminates analysis after n errors. If you specify the /EXIT switch but do not specify n, CDA exits after one error.

Default: CDA runs to completion.

/LIMIT:n (Limit Output Listing)

File: List file

Effect: The /LIMIT:n switch limits the number of pages of analysis output. When CDA has generated n pages, it terminates the analysis and prints a message on the user terminal indicating that it has done so.

Default: n = 300.

/LINES:n (Print n Lines per Page)

File: List file

Effect: This switch lets you specify the number of lines you want CDA to print per page. After n lines are printed, a new page is ejected.

Default: n=60.

/MEMSIZ:n (Establish Size of Binary Output File)

File: Binary file

Effect: This switch causes CDA to create a binary output file 4n blocks long and to transfer nK words to it from the crash-input file. The value of n must be greater than 16.

This switch is particularly useful when transferring binary crash dumps from disk or DECtape. Since disks and DECtapes have no physical end-of-files (EOFs), it is necessary to specify the size of the actual memory dump.

When the crash input resides on magnetic tape, the binary output file is filled with zeroes if the EOF is read before nKb words are transferred.

Default: n = 124.

/KMR (Assign Kernel Mapping Register Values)

File: Crash-input

Effect: On mapped systems, when CDA reads incorrect Page Address Register (PAR) values from the crash stack, it aborts the analysis and prints an error message on the terminal. If this happens, you can use the /KMR switch to retry the analysis. When you specify /KMR, CDA uses standard mapping values to convert kernel virtual addresses to physical memory addresses.

Default: CDA uses existing Page Address Registers.

/-SP (Do Not Spool)

File: List file

Effect: CDA does not spool the analysis output listing to the system line printer queue. Instead, it creates an output list file on the device indicated in the output file specification. If you do not specify a device in the output file specification when you use the /-SP qualifier, CDA creates the output list file on SY0:.

Default: /SP

/STB (File Specified Contains the Executive Symbol Table)

File: Symbol file (RSX11M.STB)

Effect: The /STB switch identifies a file containing the Executive symbol table. This file must correspond to the failed system. CDA opens the symbol file and extracts the necessary symbol values. If it fails to find any required symbol values, CDA aborts the analysis and returns an error message.

Default: [current UIC]RSX11M.STB

2.1.2 CDA Command Line Examples

The following examples illustrate CDA command lines. Assume that the user in these examples is logged in under UIC [301,356], that the crash dump device is DR5:, and that CDA is running as an installed task. Also, note how CDA uses default file types.

Example 1

```
>CDA RET
CDA>DUMP,DUMP=RSX11M.STB/STB,DR5: RET
```

This command line creates:

- A list file, DUMP.LST, in directory [301,356], which is printed automatically
- A binary file, DUMP.CDA, in directory [301,356]

CDA reads the binary crash dump input from the crash dump device (DR5:), makes a binary copy of the crash dump input named DUMP.CDA, analyzes the crash dump input according to the information in the Executive symbol table file named RSX11M.STB in directory [301,356], and writes a formatted output listing to a file named DUMP.LST. CDA then spools DUMP.LST to the system line printer queue.

Example 2

```
>CDA RET
CDA>,DUMP=[1,54]/STB,DR5: RET
```

This command line creates a binary file named DUMP.CDA in directory [301,356].

CDA reads the binary crash dump input from DR5: and analyzes it according to the information in the Executive symbol table file, which is named RSX11M.STB in directory [1,54].

Example 3

```
>CDA LP:=[1,54]/STB,DUMP RET
>
```

This command line creates an output listing on device LP:.

CDA reads the binary input from a previously created binary file named DUMP.CDA and analyzes it in accordance with the information contained in the Executive symbol table file named RSX11M.STB in directory [1,54]. The CDA output listings are then printed on LP:.

This command line is also an example of a CDA command that is issued from the CLI prompt. Thus, the CLI prompt returns after the command is issued.

Example 4

```
>CDA TI:=DUMP RET
>
```

This command line creates an output listing that is displayed on the terminal from which the command was issued.

CDA reads the binary input from a previously created binary file named DUMP.CDA and analyzes it according to the information in the default symbol table file, (the file named RSX11M.STB in the directory that currently corresponds to UIC [301,356]). The CDA output listings are then displayed on TI.

2.2 The DCL Command ANALYZE/CRASH_DUMP

If your terminal supports the DIGITAL Command Language (DCL) command line interpreter, you can run the CDA utility by using the DCL command ANALYZE/CRASH_DUMP as an alternative to the CDA command line. This section describes the ANALYZE/CRASH_DUMP command line format and qualifiers. The section concludes with some examples of ANALYZE /CRASH_DUMP command lines.

The ANALYZE/CRASH_DUMP command line has the following format:

```
ANALYZE/CRASH_DUMP[/qualifiers] crash-input[/qualifiers]
```

You use the ANALYZE/CRASH_DUMP command to specify CDA input and output. The command qualifiers that you place immediately after the command name specify the CDA output files and, optionally, the symbol table file that CDA uses to process the crash dump input. The crash-input specification is mandatory because it directs CDA to the source of the binary crash dump input.

Output Specifications:

You must specify at least one of the following command qualifiers as an output specification in the command line:

- /LIST: Specifies the output list file

  /BINARY: Specifies a binary copy of the crash-input file

- /SYMBOLS: Specifies the symbol definition file

You can specify /LIST: only, /BINARY: only, or /LIST: and /BINARY: together. Optionally, you can specify /SYMBOLS: with any combination of the /LIST: and /BINARY: qualifiers. However, if you do specify /SYMBOLS, you must include at least one of the other command qualifiers (the symbol definition file is not an output file; rather, it is used by CDA to generate an output file). Section 2.2.1.1. provides complete descriptions of the functions of each of the command qualifiers.

If you omit the crash-input specification from the command line, CDA prompts you for it, as shown in the following example:

```
$ ANALYZE/CRASH_DUMP/LIST:LP: RET
Crash input? DR5: RET
```

If you enter the command name only, CDA prompts you for input and output, as shown in the following example:

```
$ ANALYZE/CRASH_DUMP RET
Crash output? /LIST:SY:[301,356]CRASH.LST:/BINARY:COPY.CDA RET
Crash input? DUMP.CDA RET
```

Note that, if you enter an output file in this way, you must include the /LIST: or /BINARY: qualifiers as part of the output file specification.

Input Specification:

crash-input

Specifies the location of the binary input to the ANALYZE/CRASH_DUMP command. The crash-input specification can be the name of the crash dump device, or it can be a binary file that was created during a previous crash dump analysis.

When you enter an ANALYZE/CRASH_DUMP command line, you can include command qualifiers, qualifiers for the crash-input parameter, or both. Section 2.2.1 describes qualifiers.

2.2.1 ANALYZE/CRASH_DUMP Command Qualifiers

You can control the way CDA processes the crash input and how it formats the output listings by using command qualifiers in the command line. You can select the information that you want in the CDA output listings by using qualifiers for the crash-input specification. Section 2.2.1.1 describes command qualifiers. Section 2.2.1.2 describes the qualifiers that you can use when you specify the crash input.

2.2.1.1 Command Qualifiers

You can use command qualifiers with the ANALYZE/CRASH_DUMP command to control how CDA processes the binary crash-input and how it formats the output analysis listings. You place command qualifiers immediately after the command name in the command line. Table 2–4 summarizes the command qualifiers and gives brief descriptions of their effects. Expanded descriptions of each qualifier follow the table.

Table 2–4: Summary of ANALYZE/CRASH_DUMP Command Qualifiers

Command Qualifier	Function	Applies to File
/LIST:listfile[/qualifiers]	Specifies the output list file or device	List file
listfile qualifiers:		
/ERROR_LIMIT	Specifies an error limit at which CDA analysis terminates	List file
/PAGE_COUNT:n	Specifies the number of output pages	List file
/PAGE_LENGTH:n	Specifies the number of output lines per page	List file
/[NO]PRINTER	Specifies whether the output should be printed on the system line printer	List file
/BINARY:binaryfile[/qual]	Specifies an optional copy of the binary input file	Crash-input
binaryfile qualifier:		
/MEMORY_SIZE:n	Copies nK words of memory from a crashed system	Crash-input
/SYMBOLS:symbolfile	Specifies the symbol definition file	Crash-input

Command Qualifier Descriptions:

/LIST:listfile[/qualifiers]
/ERROR_LIMIT[:n]
/PAGE_COUNT:n
/PAGE_LENGTH:n
/[NO]PRINTER

File: List file

Effect: Specifies the optional formatted CDA output list file. This list file consists of the analysis report listings that are described in Chapter 3. You can also specify a device for the list file, in which case CDA displays or prints its output listings on the specified device. You can control the list file output by using the following file qualifiers.

List File Qualifiers:

/ERROR_LIMIT[:n]

Effect: CDA maintains an error count. As it encounters inconsistencies in the system data structures, it increments the error count. CDA terminates the crash dump analysis when it finds the number of errors that you specify with this qualifier. If you use the /ERROR_LIMIT qualifier without specifying a number, the crash dump analysis terminates after one error.

Default: CDA runs the analysis until it is completed.

/PAGE_COUNT:n

Effect: This qualifier limits the number of pages of analysis output. When CDA has generated n pages, it terminates the analysis and prints a message on the terminal indicating that the analysis has terminated.

Default: Analysis terminates after 300 pages.

/PAGE_LENGTH:n

Effect: This qualifier lets you specify the number of lines that you want CDA to print per output page. After the specified number of lines are printed, CDA breaks to a new page.

Default: CDA prints 60 lines per page.

/[NO]PRINTER

Effect: This qualifier prevents the printing of the analysis output on the system line printer. Instead, CDA creates the output list file on the device in the list file specification. If a device is not specified in the list file specification, CDA creates the output file on the default user disk (SY0:).

Default: CDA prints all output on the system line printer.

/BINARY:binaryfile[/qualifier]
/MEMORY_SIZE:n

File: Crash-input

Effect: Specifies that an optional binary file should be created. This file is a copy of the binary data that the crash dump routine wrote on the crash dump device. If you create the file during an initial analysis, you can use it as input to the ANALYZE/CRASH_DUMP command at a later time. Also, because the crash dump routine overwrites the contents of the crash dump volume with each crash dump, this qualifier allows you to save the results of crash dumps. You can then reuse the same volume for successive crash dumps while maintaining a record of previous crash dumps.

Binary File Qualifier:

/MEMORY_SIZE:n

Effect: Specifies memory size for the binary copy of the crash dump input file. You specify n, where n is the number of K words. CDA then creates a binary file 4n words long and transfers nK words to it from the crash-input file. The value of n must be greater than 16(decimal).

Default: n=124

/SYMBOLS:symbolfile

File: Symbol definition file

Effect: Specifies the symbol definition file for the failed system, which contains the Executive symbol table. The symbol file must correspond to the failed system. CDA opens the file and extracts the necessary symbol values. If it fails to find any required symbol values, CDA aborts the analysis and returns an error message. If you omit this file specification, CDA uses the default file, which is the file named RSX11M.STB in the directory that corresponds to the current UIC.

2.2.1.2 Crash-input File Qualifiers

You can select the analysis listings that you want CDA to output by using qualifiers for the crash-input file specification in the ANALYZE/CRASH_DUMP command line. Table 2–5 summarizes the crash-input qualifiers and gives brief descriptions of their effects. Expanded descriptions of each qualifier follow the table.

Table 2–5: Summary of ANALYZE/CRASH_DUMP Crash-input Qualifiers

Qualifier or Argument	Function	Applies to File
/ACTIVE:(arg[, . . .])	Lists data on active tasks and/or devices	Crash-input
/ACTIVE arguments:		
DEVICES	Lists data about active devices	Crash-input
TASKS	Lists contents of the Task Control Blocks for active tasks	Crash-input
/ALL	Lists all available crash dump data	Crash-input
/BLOCK:n	Specifies the block number where crash dump begins on the crash dump device	Crash-input
/CACHE[:[region]]	Lists information about cache memory at the time of the dump	Crash-input
/CLOCK_QUEUE	Lists the contents of the clock queue	Crash-input
/CONTROLLERS	Lists device controller data	Crash-input
/DATA_STRUCTURES:(arg[, . . .])	Specifies which data structures are to be formatted and listed	Crash-input
/DATA_STRUCTURES arguments:		
COMMAND_PARSER	Lists contents of CLI Parser Blocks	Crash-input
DEVICE STATUS UNIT	List contents of the control blocks for active devices	Crash-input
PARTITION	Lists contents of Partition Control Blocks	Crash-input
TASK	Lists contents of the Task Control Blocks for tasks in the STD	Crash-input
/DENSITY:n	Specifies bits per inch for input device	Crash-input

Table 2–5 (Cont.): Summary of ANALYZE/CRASH_DUMP Crash-input Qualifiers

Qualifier or Argument	Function	Applies to File
/DEVICES	Lists contents of all Device Control Blocks	Crash-input
/DUMP[:(START:n,END:n,ADDRESS:n)]	Lists contents of physical addresses	Crash-input
/HEADERS	Lists contents of resident task headers	Crash-input
/KERNEL:(arg[, . . .])	Lists kernel contents	Crash-input
/KERNEL arguments:		
DATA:(START:n,END:n)	Lists contents of kernel data space	Crash-input
INSTRUCTION:(START:n,END:n)	Lists contents of kernel instruction space	Crash-input
REGISTERS	Forces assignment of values for the kernel address registers	Crash-input
/PARTITION	Lists contents of Partition Control Blocks	Crash-input
/POOL	Lists pool contents	Crash-input
/SECONDARY_POOL[:(START:n,END:n)]	Lists contents of secondary pool from START to END	Crash-input
/[NO]SYSTEM	Suppresses listing of system information	Crash-input
/TASKS:(arg[, . . .])	Lists task data	Crash-input
/TASKS arguments:		
DIRECTORY	Lists contents of the Task Control Blocks for tasks in the STD	Crash-input
ADDRESS:(NAME:name[,START:n,END:n])	Lists contents of task addresses from START to END	Crash-input
DATA:(NAME:name[,START:n,END:n])	Lists contents of task data space	Crash-input
INSTRUCTION:(NAME:[,START:n,END:n])	Lists contents of task instruction space	Crash-input

File Qualifier Descriptions:

/ACTIVE:(arg[, . . .])
DEVICES
TASKS
File: Crash-input

Effect: Lists data on active tasks and devices.

/ACTIVE arguments:

DEVICES

Effect: Lists data on the devices active in the system at the time of the crash. If you want CDA to list data on all of the devices known to the system at the time of the crash, use the /DEVICES qualifier.

TASKS

Effect: Lists the contents of the Task Control Blocks of active tasks. If you want CDA to list the contents of the Task Control Blocks of all installed tasks, both active and dormant, use the /TASKS:(DIRECTORY) qualifier.

/ALL

File: Crash-input

Effect: Analyzes all information available in the crash dump file (except the information associated with memory and task dumps). CDA lists the output in the following order:

1. System information
2. Active tasks information
3. Task headers information
4. Partition information
5. Common Block Directory entries
6. Device information
7. Clock queue contents
8. Device controller information
9. Pool contents
10. Cache region listing

/BLOCK:n

File: Crash-input

Effect: Identifies the starting block number of the crash dump file on the crash input device. The value of n must be less than 65535_{10}.

/CACHE[:[region]]

File: Crash-input

Effect: Lists information about cache memory at the time of the dump. The region is the name of the cache region as specified in the SET CACHE command or in the MOUNT command with the CACHE qualifier.

/CLOCK_QUEUE

File: Crash-input

Effect: Lists the contents of the system clock queue.

/CONTROLLERS

File: Crash-input

Effect: Lists the contents of the controller table and Controller Request Block (KRB) for each device controller in the system.

/DATA_STRUCTURES:(arg[, . . .])

COMMANDPARSER
DEVICE
PARTITION
STATUS
TASK
UNIT

File: Crash-input

Effect: Selects which system data structures CDA will format and list.

/DATA_STRUCTURES arguments:

COMMAND_PARSER

Effect: Lists the contents of the Command Line Interpreter (CLI) Parser Blocks.

PARTITION

Effect: Lists the contents of the Partition Control Blocks.

TASK

Effect: Lists the contents of the Task Control Block for every task in the System Task Directory (all installed tasks) at the time of the system crash.

DEVICE

Effect: Lists the contents of the Device Control Blocks for active devices.

STATUS

Effect: Lists the contents of the Status Control Blocks for active devices.

UNIT

Effect: Lists the contents of the Unit Control Blocks for active devices.

/DENSITY:n

File: Crash-input

Effect: Causes a crash input tape to be read at the density specified, 800 or 1600 bpi. The default is 800 bpi.

/DEVICES

File: Crash-input

Effect: Lists the contents of the control blocks for all devices in the system. To list only active devices, use the /ACTIVE:(DEVICES) qualifier.

/DUMP[:(START:a,END:b[,ADDRESS:c])]

File: Crash-input

Effect: Lists the contents of physical addresses a through b, inclusive, and labels them with their physical addresses. If you include address c, the /DUMP qualifier dumps the contents of physical addresses a through b but labels them with dummy virtual addresses, starting at c.

/HEADERS

File: Crash-input

Effect: Lists the contents of the task headers for each task resident in memory.

/KERNEL:(arg[, . . .])
DATA:(START:n,END:n)
INSTRUCTION:(START:n,END:n)
REGISTERS

File: Crash-input

Effect: Lists kernel data.

/KERNEL arguments:

DATA:(START:n,END:n)

Effect: Lists the contents of kernel data space from virtual addresses START:n to END:n.

INSTRUCTION:(START:n,END:n)

Effect: Lists the contents of kernel instruction space from virtual address START:n to END:n.

REGISTERS

Effect: Forces the assignment of the kernel address register values for the crashed system.

/PARTITION

File: Crash-input

Effect: Lists the contents of the Partition Control Blocks.

/POOL

File: Crash-input

Effect: Lists the contents of system pool in octal, Radix–50, and ASCII.

/SECONDARY_POOL:(START:n,END:n)]

File: Crash-input

Effect: Lists the contents of system secondary pool between the addresses specified by START and END.

/[NO]SYSTEM

File: Crash-input

Effect: The /NOSYSTEM qualifier suppresses the system information listing. The default action of CDA is /SYSTEM; that is, it lists the system information.

/TASKS:(arg[, . . .])
DIRECTORY
ADDRESS:(NAME:name,START:n,END:n)
DATA:(NAME:name[,START:n,END:n])
INSTRUCTION:(NAME:name[,START:n,END:n])

File: Crash-input

Effect: Lists task data.

DIRECTORY
Effect: Lists the contents of the Task Control Block for every task in the System Task Directory (all installed tasks) at the time of the system failure.

ADDRESS:(NAME:name,START:n,END:n)
Effect: Lists the contents of the task specified by NAME between the virtual addresses specified by START and END. Includes the contents of task data space if a task includes data space.

DATA:(NAME:name[,START:n,END:n])
Effect: Lists the contents of task data space for the task specified by NAME.

INSTRUCTION:(NAME:name[,START:n,END:n])
Effect: Lists the contents of task instruction space for the task specified by NAME.

2.2.2 ANALYZE/CRASH_DUMP Command Examples

The following examples illustrate the ANALYZE/CRASH_DUMP command. Assume that the user in these examples is logged in under UIC [301,356] and that the crash dump device is DR5:. In this way, you can note how CDA uses default file types. Also, assume that CDA is running as an installed task.

Example 1

```
$ ANALYZE/CRASH_DUMP/LIST:CRASH/BINARY:COPY/MEMORYSIZE:250 DR5: RET
```

This command creates:

- An output list file named CRASH.LST in the current directory for UIC [301,356].
- A binary copy of 250kb words of the crash dump from DR5: (the crash dump device). The copy is named COPY.CDA and is placed in the current directory for UIC [301,56].

CDA reads the binary crash dump input from the crash dump device and analyzes it according to the default symbol definition file. Since a symbol definition file is not specified in the command line, CDA uses the file named RSX11M.STB in the current UIC as the symbol definition file. CDA then generates a list file named CRASH.LST and spools it to the default system line printer queue. CDA also copies the specified amount of memory from the crash dump device to a binary file named COPY.CDA.

Example 2

```
$ ANALYZE/CRASH_DUMP/LIST:LP5:/PAGE_COUNT:5 DR5:/BL:100 RET
```

This command creates a list file that is printed on LP5.

CDA reads the crash input from DR5:, beginning at block 100, and analyzes it according to the default symbol definition file. CDA then prints the first five pages of its output listing on LP5.

Example 3

```
$ ANALYZE/CRASH_DUMP/LIST:TI:/SYMBOLS:[1,54] COPY.CDA RET
```

This command creates a list file that is displayed on TI (the terminal at which the command was issued).

CDA reads the previously generated binary file named COPY.CDA, analyzes it according to the file named RSX11M.STB in directory [1,54], and displays its output listings on TI.

Chapter 3
Analysis Listings

The CDA output listings in this chapter illustrate CDA operation. Each item of each listing is keyed to the brief explanatory text that precedes it.

Dumps shown in offset mode use relative addresses, rather than physical or virtual addresses, of the data. They are offset from the beginning of the displayed data.

Note

Those listings that extend across several pages in an actual dump of a crashed system are truncated here and reflect only a typical printout.

3.1 System Information

The first seven pieces of a CDA output listing normally contain the system information described in Sections 3.1.1 through 3.1.7. This system information consists of the following:

- Volatile registers
- Kernel stack
- Low Core Memory Dump
- System common
- System common alphabetized dump
- Pool statistics
- Assign table

If Group Global Event Flag Blocks are in memory when the system fails, the listing described in Section 3.1.8 appears. If error log packets are in memory at the time of the failure, the listing described in Section 3.1.9 appears.

3.1.1 Volatile Registers

Example 3–1 is a listing that reflects the state of the hardware registers at the time of the crash. Refer to the appropriate PDP–11 processor handbook for detailed information on these registers. Each item in the following list describes a correspondingly numbered item in Example 3–1.

❶ Contents of Processor Status Word (PSW) and kernel and user stack pointers after crash

❷ Contents of CPU error register that identifies the source of the abort or trap that used the vector at location 0. This field is suppressed if the processor does not have a UNIBUS map.

❸ Contents of memory system error register. This field is suppressed if the processor does not have a UNIBUS map.

❹ Contents of cache control register. This field is suppressed if the processor does not have a UNIBUS map.

❺ Program counter and PSW (that the system pushed onto the kernel stack) just prior to system crash. (These values are valid only if the system trapped.)

❻ Contents of general registers.

❼ Contents of memory management registers.

❽ Contents of Page Address and Page Description Registers. (See Section 4.1.1 for information on how to interpret this information.)

❾ Contents of UNIBUS map registers. (This field is suppressed if the processor does not have a UNIBUS map.)

Example 3–1: Volatile Registers

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0     25-SEP-86   15:26        PAGE 1
VOLATILE REGISTERS

AFTER CRASH:  PS=000344   SP(K)=001006   SP(S)=000422   SP(U)=120416  (1)
                  (2)                        (3)                      (4)
              CPU ERR = 000000   MEM SYS ERR = 000000   CACHE CTL REG = 000001

BEFORE CRASH:  PC=131530   PS=030000  (5)

R0=022254   R1=022200   R2=000000   R3=000000   R4=022254   R5=120553  (6)

MMR0=000037   MMR1=000000   MMR2=012502   MMR3=000067  (7)

                    U S E R                                U N I B U S   M A P

  I   S P A C E          D   S P A C E                      1  70200000
  PDR       PAR          PDR       PAR                     2  70210341
 017406    000000       017506    000000                   3  70210341
 077406    000742       077406    000742                   4  70210341
 077406    001142       077406    001142                   5  70210341
 077406    001342       077406    001342                   6  70210341
 077406    001542       077406    001542                   7  70210341
 077406    010667       077506    010667                   8  70210341
 077406    011067       077406    011067                   9  70210341
 077406    177600       077406    177600                  10  70210341
                                                           11  70210341
                                                           12  70210341
           S U P E R V I S O R                             13  70210341
                                                           14  70210341
  I   S P A C E          D   S P A C E                     15  70210341  (9)
  PDR       PAR          PDR       PAR                     16  70210341
 000402    003117       017406    000000  (8)              17  70210341
 000000    000000       077406    000742                   18  70210341
 000000    000000       077406    001142                   19  70210341
 000000    000000       077406    001342                   20  70210341
 000000    000000       077406    001542                   21  70210341
 000000    000000       077406    010114                   22  70210341
 077402    006160       057406    010314                   23  70210341
 077402    006360       000002    000000                   24  70210341
                                                           25  70210341
                                                           26  70210341
                K E R N E L                                27  70210341
                                                           28  70210341
  I   S P A C E          D   S P A C E                     29  70210341
  PDR       PAR          PDR       PAR                     30  70210341
 077406    000000       177506    000000                   31  70210341
 077406    000200       077506    000742
 077406    000400       077406    001142
 077406    000600       077406    001342
 077406    001000       077506    001542
 077406    010667       077406    010667
 077406    011067       077406    011067
 077406    177600       077506    177600
```

3.1.2 Kernel Stack

Example 3–2 shows the contents of the kernel stack area beginning at V$$CTR and ending at $STACK. The kernel stack pointer points to a location within this area. See Section 4.1.2 for information on interpreting the contents of the kernel stack.

Example 3–2: Kernel Stack

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26        PAGE 2
KERNEL STACK

    KERNEL STACK:
      000500    047503  051120  020056  041450  020051  044504  044507  040524
      000520    020114  034461  031470  000000  000000  000000  000000  000000
      000540    000000  000000  000000  000000  000000  000000  000000  000000
      000560    000000  000000  000000  000000  000000  000000  000000  000000
      000600    000000  000000  000000  057304  013444  000000  031730  126506
      000620    030000  126454  000000  127006  013765  000076  126732  064340
      000640    003233  003233  120544  132056  000002  133542  000001  132470
      000660    120640  000000  132545  132545  134174  000003  126522  000000
      000700    132545  000001  000012  000001  000001  003233  013765  013764
      000720    125644  020540  067130  000100  120752  003764  003440  022254
      000740    027060  026022  003764  032424  000001  000002  025230  000000
      000760    010664  124270  000010  000000  046662  032324  064650  124272
      001000    120553  131530  030000  013130  022254  022200  000000  000000
      001020    022254  120553  131532  170000
```

3.1.3 Low Core Memory Dump

The listing shown in Example 3–3 contains a dump of low core memory, alphabetized by label. Appendix B lists labels found in low core memory and their meanings.

Example 3–3: Low Core Memory

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 3
LOWCORE ALPHABETIZED DUMP

$BCERR 000104              123024       $FLTCB 000000       $SAVSP 001020
$BCFAC 000301              000000       $FMAPP 000000       $SCMOF 000742
$BCPC  131530              124270       $HEADR 001050              000003
$CPBIT 177777              006530       $HFMSK 000016       $SGFFR 000000
$CPCRM 177777              000000              000000       $SIRWF     000
$CPUER 177766              000000       $ICAVL 004404       $STAT  000000
$CPURM 177777              000135              000000              000000
$CRDEV 000104              000000       $IDLCT     000              000000
$CRPAR 077646              000000       $IDLFL     000              000000
$CRSBF 010341              000000       $IDLPT 007760       $STKDP 000000
       010341       $DXDEP 177777              007760       $STRTM 000015
       010341       $DXDK5 177777       $MPCTL 000001              125460
       010341       $DXDRL 177777       $MSTAT 000000       $SUPFL     000
$CRUNT     060      $ERLFK 021356              000000       $TKTCB 032300
$CRUPC 131530       $ERTRK 050044       $PARLV 177777       $UMRHD 000000
$CRUST 030000              000000       $POWSP 000000              170200
$CURPR 021346              120030       $PROC2     377              000034
$CXDBL     000             000000       $PROCN     377       $UMRWT 000000
$DICSV 001051              000000       $ROEND 000742              001102
       000037       $FLFRK 021362       $RQSCH 021344
       000007       $FLSTS 000000       $SAHDB 010664
       000037              000000       $SAHPT 140000
```

3.1.4 System Common

The listing in Example 3–4 provides a selective interpretation of some of the items in system common. Each item in the following describes the corresponding numbered item in Example 3–4. (Refer to the *RSX–11M–PLUS and Micro/RSX Guide to Writing an I/O Driver* for further information.)

❶ Time and date of crash, as set in the system.

❷ The task that was running at the time of the crash. (If no task was running, this field contains the null task. This condition could develop if all the active tasks are blocked at the time of the crash. For information on determining which task or driver was mapped at the time of the crash, see Section 4.1.1.)

❸ The address of the Task Control Block (TCB) of the current task.

❹ The contents of the 4-byte system ID indicating system base level.

❺ The first address available for partitions (the last address of the Executive plus 1).

❻ The system size in 32-word blocks and in total words.

❼ System UIC.

❽ Stack depth count.

❾ Contents of the global event flag words.

❿ Name of the system for which dump is generated.

⓫ Network UIC.

⓬ Device from which the system was booted.

⓭ Logical block number (LBN) of the beginning of the system image.

⓮ Size of system image file in blocks.

⓯ The octal value of the system feature masks and the meaning of each set bit.

⓰ Octal dump of system common in offset mode in numerical order by address.

Example 3–4: System Common

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 4
SYSTEM COMMON

CRASH OCCURRED AT 14:22:21  25-SEP-86  (1)
        (2)                   (3)
CURRENT TASK = BRKTO      TCB ADDRESS = 032300
      (4)                 (5)                 (6)                    (7)
$SYSID = 27     $EXSIZ = 120000     $SYSIZ = 32768./1024K     $SYUIC = [2,54]
      (8)                              (9)
$STKDP = 000000      $COMEF:  <33-48> 000000  <49-64> 100000
      (10)                        (11)
SYSTEM NAME = RLO2ID        $NTUIC = [001,054]
      (12)              (13)            (14)
LOAD DEVICE = DLO     LBN = 00023367     FILE SIZE = 1024.

SYSTEM FEATURE MASK (FIRST WORD) = 013377

      BIT SET              MEANING
      -------              -------
        EXT         22-BIT EXTENDED MEMORY SUPPORT
        MUP         MULTI-USER PROTECTION SUPPORT
        EXV         20K EXEC SUPPORTED
        DRV         LOADABLE DRIVER SUPPORT
        PLA         PLAS SUPPORT
        CAL         DYNAMIC CHECKPOINT SPACE ALLOCATION
        PKT         PREALLOCATION OF I/O PACKETS
        EXP         EXTEND TASK DIRECTIVE SUPPORTED
        OFF         PARENT/OFFSPRING TASKING SUPPORTED
        FDT         FULL DUPLEX TERMINAL DRIVER
        DYM         DYNAMIC MEMORY ALLOCATION SUPPORTED
```

(Continued on next page)

Example 3–4 (Cont.): System Common

```
SYSTEM FEATURE MASK (SECOND WORD) = 167763

      BIT SET              MEANING                                    ⓯
      -------              -------
        DAS        KERNEL DATA SPACE SUPPORT
        LIB        SUPERVISOR MODE LIBRARY SUPPORT
        ACN        ACCOUNTING SUPPORTED
        SDW        SHADOW RECORDING SUPPORTED
        POL        SECONDARY POOL SUPPORTED
        WND        SECONDARY POOL FILE WINDOWS SYSTEM
        DPR        DIRECTIVE PARTITION SYSTEM
        IRR        INSTALL, REQUEST, AND REMOVE TASK SUPPORT
        GGF        GROUP GLOBAL EVENT FLAG SUPPORT
        RAS        RECEIVE/SEND DATA PACKET SUPPORT
        RBN        ROUND ROBIN SCHEDULING SUPPORTED
        SWP        EXECUTIVE LEVEL DISK SWAPPING SUPPORTED
        STP        EVENT FLAG MASK IS IN THE TCB

SYSTEM FEATURE MASK (THIRD WORD) = 067364

      BIT SET              MEANING
      -------              -------
        EIS        SYSTEM REQUIRES THE EXTENDED INSTRUCTION SET
        UDS        USER DATA SPACE (M-PLUS ONLY)
        PRO        PROTO TCBS OUT OF POOL (M-PLUS ONLY)
        XHR        EXTERNAL HEADER SUPPORT (M-PLUS ONLY)

RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 6
SYSTEM COMMON DUMP

                                    ⓰
 ADDR   LABEL  VALUE          ADDR   LABEL  VALUE          ADDR   LABEL  VALUE

020460         113226        020620 $FMSK4 063774        020760 $DCCEL 001266
               113226                      000000               $DCCEB 000000
       $ACCLK 000015                $PRMOD 000065                      000000
       $ABTIM 125460                $CLKHD 022430               $DCNEL 000000
       $ACTHD 117500                $COPT  022026               $DCNEB 000000
              000000                $PARHD 117734                      006731
       $COMEF 000000                $LDRPT 117500               $IPUCB 000000
              100000                $TSKHD 117500               $SCDEV 000000

020500 $HRCPT 111454        020640 $CBDHD 117624        021000 $SCCTB 027476
       $DEVHD 021534                $GGEF  023264               $SCOFL 000400
       $TKNPT 116514                $GFTCB 020650               $DRAPR 002542
       $SHFPT 116310                $GEFPT 020650               $DRAP2 002731
       $CKCNT 177546                       000204               $DRAP3 003117
       $CKCSR 177546                $LDPCB 000000               $DRAP4 003233
       $CKLDC 000000                $VECTR 000474                      003412
       $SYUIC 001054                $MXEXT 177777               $DRCHE 003442
```

(Continued on next page)

Example 3–4 (Cont.): System Common

```
020520          000000      020660 $AVRHD 000000      021020          177777
        $EXSIZ  120000             $PRIFR 000000              $ENVEC  021014
        $PWRFL  000000             $ULDPT 000000              $CRKRB  115716
        $SIGFL  000000             $GNLST 000000              $CRSFM  000000
                000000             $PTCBL 001776              $CRSUN  000000
        $FXRPT  000000             $PTCPT 001770              $CRCSR  000000
        $LSTLK  000000             $PASTH 000000              $CRLCK  000001
                000003                    020674              $RNDCT  000002

020540 $CRAVL   023224      020700 $NXTLK 000000      021040 $RNDC    000006
        $K6TAB  000000             $NXTBA 000000              $RNDL   113001
        $VERTK  000000             $LOGTB 001742              $SWPCT  000010
        $LOGHD  000000                    001743              $SWPC   000036
        $CFLPT  024020             $USRLG 002167              $SWPR   000005
        $MOULS  000000                    001744              $DYPMN  020035
        $LBUIC  001454                    000000                      020037
        $NTUIC  000454             $CTXPT 000000                      020037

020560          000000      020720 $PTTCB 112760      021060          017440
                000000             $PRISZ 056750                      017440
        $HKSTS  000000             $POLST 000401                      020040
                026326             $PRIHL 003100                      000000
                026326             $PRILL 001130              $BTMSK  000001
                026326             $PFRSZ 000310                      000002
        $BTTM2  000000             $POLBP 000063                      000004
        $BTTIM  001742             $POLFL 000000                      000010

020600          000000      020740 $SHUMB 000000      021100          000020
                057252             $SHERR 111104                      000040
        $INTCT  177777             $SHLIM 000012                      000100
        $FRKHD  000000             $SHPCT 000000                      000200
                020606             $SHLOS 000000                      000400
        $FMASK  013377             $RCTPT 112510                      001000
        $FMSK2  167763             $EXCRC 000000                      002000
        $FMSK3  067364             $DCPCB 000000                      004000
```

3.1.5 System Common Alphabetized Dump

The listing in Example 3–5 represents an alphabetical list of the locations in system common that have a label associated with them. The octal numbers represent the contents of those locations, not the addresses of the labels. Appendix B lists the labels and their meanings.

Example 3–5: System Common Alphabetized Dump

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 8
SYSTEM COMMON ALPHABETIZED DUMP

$ABTIM 125460       $CPMSK 177777       $DYPMN    035       $GGEF  023264
$ACCLK 000015       $CPTBL 021464       $EFMSK 000007       $GNLST 000000
$ACNFE 000050       $CPURM 177777       $ENTSQ 000001       $HEADR 001050
$ACPTK 000000       $CPUSC 177777       $ENVEC 021014       $HFMSK 000016
$ACTHD 117500       $CRAVL 023224       $ERFLA 000017       $HKSTS 000000
$ACTPS 000074       $CRCSR 000000       $ERHEA 000000       $HRCPT 111454
$APLIM 000140       $CRFLG 000001       $ERRPT 113420       $ICAVL 004404
$APRTB 172340       $CRKRB 115716       $ERRSQ 000000       $IDLCT    000
$ATTPT 023350       $CRLCK    001       $EVBSQ 002542       $IDLFL    000
$AVRHD 000000       $CRMTB 177777       $EVDIS 002542       $IDLPT 007760
$BCMSK    001       $CRSFM 000000       $EVKS6 002542       $IIMSK 177777
$BTMSK 000001       $CRSUN 000000       $EVLEN 002542       $INTCT 177777
$BTTIM 001742       $CTLST 021522       $EVLST 002542       $IOTMP 077406
$BTTM2 000000       $CTXPT 000000       $EVSEQ 002542       $IPUCB 000000
$CBDHD 117624       $CXDBL    000       $EVTAB 002542       $K6TAB 000000
$CFLPT 024020       $DCCEB 000000       $EXCRC 000000       $KATBL 016662
$CKCNT 177546       $DCCEL 001266       $EXECL 177777       $LBUIC    054
$CKCSR 177546       $DCNEB 000000       $EXSIZ    000       $LDPCB 000000
$CKLDC 000000       $DCNEL 000000       $FLFTB 000000       $LDRPT 117500
$CKUAB 000000       $DCPCB 000000       $FMASK 013377       $LOGHD 000000
$CKURM 177777       $DEVHD 021534       $FMEND 000065       $LOGPT 112364
$CLICQ 000000       $DICSV 001051       $FMSK2 167763       $LOGTB 001742
$CLKHD 022430       $DRAP2 002731       $FMSK3 067364       $LSTLK 000000
$CNTLN 000000       $DRAP3 003117       $FMSK4 063774       $MCRPT 117124
$COMEF 000000       $DRAP4 003233       $FRKHD 000000       $MKCS1 000000
$COPT  022026       $DRAPR 002542       $FXRPT 000000       $MKCS2 177777
$CPBIT 177777       $DRCHE 003442       $GEFPT 020650       $MKCSR 177777
$CPCRM 177777       $DVSAV 000000       $GFTCB 020650       $MOULS 000000
```

(Continued on next page)

Example 3–5 (Cont.): System Common Alphabetized Dump

```
RSX-11M-PLUS CRASH DUMP ANALYZER  V4.0    25-SEP-86   15:26       PAGE 9
SYSTEM COMMON ALPHABETIZED DUMP

$MPTAB 177777       $PROC2 177777       $SCRET 027476       $SYUIC    054
$MXEXT 177777       $PROCN 177777       $SECFR 000350       $TEMP0 000774
$NCPU     000       $PRTAB 000240       $SGFFR 000000       $TEMP1 000000
$NCTPT 000000       $PTCBL 001776       $SHERR 111104       $TEMP2 026324
$NTUIC    054       $PTCPT 001770       $SHFCT 000036       $TEMP3 000003
$NXTBA 000000       $PTTCB 112760       $SHFPT 116310       $TEMP4 031730
$NXTLK 000000       $PWRFL 000000       $SHFTM 000000       $TKNPT 116514
$PARHD 117734       $RCTPT 112510       $SHLIM 000012       $TKPS  000074
$PASTH 000000       $RNDC  000006       $SHLOS 000000       $TNAME 035203
$PFRSZ 000310       $RNDCT 000002       $SHPCT 000000       $TONYL 002126
$PFURM 177777       $RNDH     226       $SHUMB 000000       $TSKHD 117500
$PKAVL 023304       $RNDL     001       $SIGFL 000000       $TTNS  000064
$PKNUM    003       $ROEND 000742       $STALR 177777       $ULDPT 000000
$PKMAX    005       $RQSCH 021344       $STENB 177777       $UMRHD 000000
$PLPAR 117734       $RQTAB 000000       $STKDP 000000       $URMST 177777
$POLBP 000063       $SABPT 002151       $STFLG 177777       $URMTB 177777
$POLFL 000000       $SAHDB 010664       $SWPC  000036       $USRLG 002167
$POLHD 002172       $SAHPT 140000       $SWPCT 000010       $VECTR 000474
$POLST 000401       $SAVSP 001020       $SWPR  000005       $VERTK 000000
$PRIFR 000000       $SCCTB 027476       $SWR   177777       $WPLST 177777
$PRIHL 003100       $SCDEV 000000       $SYALT 000000       $WPVAL 177777
$PRILL 001130       $SCERR    001       $SYLHD 000000       $XDTFL 177777
$PRISZ 056750       $SCMOF 000742       $SYSIZ 100000       $XDTIN 177777
$PRMOD 000065       $SCOFL    000       $SYUAB 002147       $XDTPR 177777
```

3.1.6 Pool Statistics

The listing in Example 3–6 contains information concerning the system pool. CDA derives Items 2, 3, and 4 by scanning the free block pointers of the pool. The minimum block size (that is, pool granularity) in Item 5 comes from the contents of $CRAVL-2. Each item in the following list describes a correspondingly numbered item in Example 3–6:

❶ Pool size in decimal bytes

❷ The largest fragment of pool space

❸ Total number of free bytes in pool

❹ Number of fragments not allocated

❺ Smallest possible block (the minimum number of bytes that may be requested at a time; the minimum block size is always four bytes)

❻ Bit map in octal

Each bit in the bit map represents one 4-byte block. If the bit is set, the block is free. The first block in the pool is bit 0 of the first octal word in the bit map. The bits are numbered as follows:

Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary	0	0	1	1	1	0	1	1	1	1	1	1	1	0	0	0
Octal	0		3			5			7			7			0	

Any bits left over in the last word of the bit map are cleared.

Example 3-6: Pool Statistics

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86    15:26        PAGE 11
POOL STATISTICS

POOL SIZE (BYTES) = 31156.   (1)

LARGEST FREE BLOCK (BYTES) = 23736.   (2)

TOTAL FREE BYTES = 24040.  (3)

B  NUMBER OF FRAGMENTS = 15.   (4)

MINIMUM BLOCK SIZE (BYTES) = 4.  (5)

POOL BITMAP (CONSTRUCTED FROM LINKED POOL, BLOCK FREE IF BIT SET):

      000000 020004 000000 000400 000000 140000 000000 000001
      000000 000000 014002 000000 000001 000000 000000 000000
      000000 000000 000000 000000 000000 000000 000000 000000
      000000 000000 000000 000000 000000 000000 000000 000000
      000000 000000 000000 000000 000000 000000 000000 000000
      000000 000000 000000 000000 000000 000000 000000 000000
      000000 037600 000174 004000 140000 001777 177770 077777
      000000 000000 000000 174000 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777   (6)
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
      177777 177777 177777 177777 177777 177777 177777 177777
```

3.1.7 Assign Table

Example 3–7 is a listing of the logical device assignment table.

Example 3–7: Logical Assignment Table

```
 RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 13
 ASSIGN TABLE

System Logicals:

     WK = LB:
     Block: 1      Status: (Final)

User Logicals:

   Terminal: TT0:
     SYS$LOGIN = DL:[7,40]
     Block: 2      Status: (Final, Privileged)
```

The Assign Table lists logical assignment table entries in two categories: system logicals and user logicals. The system logicals listing shows the logical name, the equivalence name, the logical block type, and the status of an assignment. The user logicals listing also shows the terminal from which an assignment was made.

3.1.8 Group Global Event Flags

Example 3–8 shows a group global event flag dump. If there are no group global event flags, this dump does not appear.

❶ Group number

❷ Access count

❸ Group Global Event Flag Block dump (the last two words are the group global event flags)

Example 3–8: Group Global Event Flags

```
 RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 10
 GROUP GLOBAL EVENT FLAGS


        ❶                          ❷
 GROUP NUMBER =7          ACCESS COUNT =000001

 GGEF  DUMP:     ❸

     000000     000000  000007    000001  000062  000000
```

3.1.9 Error Log Packets

The listing shown in Example 3–9 contains error logging information that resided in memory at the time of the system failure. This page does not appear if no error log packets were in memory at that time. This data is not written to the Error Log file on disk.

Address of error log buffer

❷ Error log packet entry type code

Error log packet entry type subcode

❹ Time the packet was logged

❺ Dump of error log packet in octal, Radix-50, and ASCII

Example 3-9: Error Log Packets

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    11-MAY-87   11:07       PAGE 12
ERROR LOG BUFFERS
                ❶                              ❷              ❸
    BUFFER ADDRESS = 00217700    ENTRY TYPE CODE = 000002   ENTRY TYPE SUBCODE = 000001

    TIME = 13-OCT-86   11:16:03 ❹
    000000    002202  000214  000034  000075    ! .4  CT    .  AU!!      = !
    000010    001006  033462  020040  007401    ! L8 H3J EEX BPA!!  27     !
    000020    000007  000003  000402  005126    !  G   C  FR AZF!!       V !
    000030    005415  001420  000065  000001    !AO/  SX  AM   A!!    5    !
    000040    000060  052504  000001  000001    ! AH MY.   A   A!!O DU     !
    000050    052504  000000  000000  000000    !MY.            !!DU       !   ❺
    000060    000000  000000  000000  000000    !                !!        !
    000070    000000  000000  000001  001440    !           A  T !!        !
    000100    000000  000046  000000  000000    !      8         !!  &     !
    000110    023000  000000  000000  000000    !FCH             !! &      !
    000120    000030  070511  100660  003440    !  X REA TO  AEX!!  Iq0     !
    000130    052124  001000  001000  000007    !MS.  L2  L2   G!!TT       !
    000140    000312  000000  000000  120442    ! EB          Y1J!!J     "!!
    000150    000002  000046  000040  000001    !  B   8   2   A!!  &     !
    000160    000005  032764  000001  000000    !  E HYL   A     !!  t5    !
    000170    000241  000043  000000  000000    ! DA   5         !!! #     !
    000200    057554  000003  000000  000000    !0J.   C         !!1_      !
    000210    000000  000000  000263  000000    !          DS    !!    3   !
    000220    165555  133333  165555  133333    !7Z/ /JK 7Z/ /JK!!mk[6mk[6!
```

3.2 Optional Information

CDA gives you additional information when you use the analysis switches described in Chapter 2. Examples 3-10a through 3-23 illustrate the output that CDA provides when you use these switches.

3.2.1 Active Tasks

The listing shown in Examples 3-10a and 3-10b contain active task information. The Receive Queue, AST Queue, Receive-by-Reference Queue, and Offspring Control Block sections of this example appear only if the task has them; otherwise, they are suppressed.

The following list describes the active task listing given in Example 3-10a, which includes Receive Queue, AST Queue, and Receive-by-Reference Queue listings:

❶ Task name

❷ Address of Task Control Block (TCB) for the task

❸ Name of the partition in which the task runs

❹ Address of Partition Control Block (PCB)

❺ Base address for the partition in which the task runs

❻ Device that contains task image

❼ Beginning logical block number (LBN) of the task on the device

❽ Running priority

❾ Number of outstanding QIO requests

❿ Current UIC (either the login UIC or the UIC specified with a SET command)

⓫ Physical name of task's pseudo device

⓬ Maximum size of task image in 32_{10}-word blocks

⓭ State of local event flags for task

⓮ First status word (blocking bits), using the following three-letter codes:

-EXE	Task not executing
RDN	I/O rundown in progress
CIP	Task blocked for checkpoint in progress
MSG	Abort message being output
CKR	Task has checkpoint request (RSX–11M–PLUS only)
BLC	Increment blocking count
STP	Task stopped by CLI command

⓯ Second status word (state bits), using the following three-letter codes:

AST	Asynchronous system trap (AST) in progress
SIO	Task stopped for buffered I/O
DST	AST recognition disabled
AFF	Task installed with affinity
-CHK	Task not checkpointable
SEF	Stopped for event flag
REX	Exit AST specified
HLT	Task being halted
ABO	Task marked for abort
STP	Task stopped
SPN	Task suspended
WFR	Task in wait-for state

⓰ Third status word (attribute bits), using the following three-letter codes:

ACP	Task is an Ancillary Control Processor (ACP)
-PMD	Task not dumped on synchronous abort

CMD Task is executing a CLI command

REM Remove task on exit

PRV Task is privileged

MCR Task requested as an external MCR function

SLV Task is a slave task

CLI Task is a command line interpreter

RST Task is restricted

NSD Task does not allow send data

CAL Task has checkpoint space in task image

ROV Task has resident overlays

NET Network protocol level

GFL Group global event flags are locked

SWS Reserved for Software Services

MPC Mapping change with outstanding I/O

⓱ Fourth status word, using the following three-letter codes:

MUT Task is a multiuser task

LDD Task load device is dismounted

PRO TCB is a prototype

PRV Task was privileged but has cleared TB.PRV with the GIN directive

DSP Task was built for user data space

SNC Task uses common synchronization

⓲ Octal dump of TCB in offset mode

RECEIVE QUEUE (if the task has one)

⓳ Starting address of receive block

⓴ Name of task

㉑ Octal dump of receive block in offset mode

ASYNCHRONOUS SYSTEM TRAP (AST) QUEUE (if the task has one)

㉒ An item appearing in the AST queue may be one of the following:

- Unsolicited AST
- Floating point AST
- Receive data AST
- Receive-by-reference AST

- Parity error AST
- Requested exit AST
- Power fail
- CLI command arrival AST
- Buffered I/O AST
- Offspring task AST
- Segmented buffered I/O completion AST
- Task force trace bit trap AST
- Delayed I/O completion AST
- Group-global rundown AST
- I/O request packet

㉓ Address of AST block

㉔ A 2-byte indicator (The high-order byte is an offset into the header of the AST control block; the low-order byte is the length of the AST control block in bytes)

Note

If the low-order byte is negative, the block is not an AST block, but an I/O request packet internal to the system. If the low-order byte is 0, the block is an unsolicited character AST.

㉕ Number of bytes allocated on task stack

㉖ Entry point of AST routine

㉗ Number of AST parameters

㉘ Octal dump of the AST block in offset mode (On RSX–11M–PLUS systems, two additional negative offset words appear in the dump)

RECEIVE-BY-REFERENCE QUEUE (If the task has one)

㉙ Address of Receive-by-Reference Queue Block

㉚ Address of the Task Control Block (TCB) for the task that initiated the Send by Reference

㉛ Pointer to created attachment descriptor

㉜ Offset into partition as specified in window definition

㉝ Length to be mapped

㉞ The receiving task's access rights to region being mapped

㉟ Octal dump of Receive-by-Reference Block in offset mode

Example 3–10a: Active Tasks with Receive Queue, AST Queue, and Receive-by-Reference Queue Listings

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 22
ACTIVE TASKS

TT0  ❶
------
            ❷                ❸                          ❹
 TCB ADDRESS = 032154      PAR = GEN          PCB ADDRESS = 024254
            ❺                          ❻                  ❼
 LOAD ADDRESS = 00656400      LOAD DEVICE = SY0:      LBN = 00006741
    ❽                   ❾              ❿              ⓫
 PRI = 50.      I/O COUNT = 0.      UIC = [7,40]      TI = TT0:
         ⓬                          ⓭
 MAX SIZE = 000032      EVENT FLAGS = <1-16> 000000  <17-32> 000000

 T.STAT:  000000   ⓮

 T.ST2:   120001   AST -CHK  WFR   ⓯

 T.ST3:   074005  -PMD  REM  PRV  MCR  CMD  GFL   ⓰

 T.ST4:   000002   DSP   ⓱

 TCB DUMP:

   000000    000000  000062  000000  100076  000000  002170  002171  023614
   000020    024430  000000  000000  022254  020470  000000  120001  074005
   000040    000062  006741  023102  024254  000032  020470  023216  023516   ⓲
   000060    000002  000004  000001  000001  032176  000016  000014  000000
   000100    030754  030754  002134  000000  032262  000000  000000  002165
   000120    000001

   RECEIVE QUEUE
   -------------          ⓳                    ⓴
   RECEIVE BLOCK ADDRESS = 002170   TASK NAME = TT0

   000000    002171  000025  100076  000000  064124  071551  066440  071545
   000020    060563  062547  064440  020163  062542  067151  020147  062563   ㉑
   000040    072156  073440  072151  020150  051526  040504  027044  022254
   000060    003440  035203  100660

   RECEIVE BLOCK ADDRESS = 002171   TASK NAME = TT0

   000000    000000  000025  100076  000000  064124  071551  066440  071545
   000020    060563  062547  064440  020163  062542  067151  020147  062563
   000040    072156  073440  072151  020150  051526  040504  027044  022254
   000060    003440  035203  100660

   AST QUEUE  ㉒
   ---------
                    ㉓                ㉔
   AST BLOCK ADDRESS = 023614     A.CBL = 001000

   SPECIFIED AST FROM: SFPA$, SRDA$, SRRA$, SPEA$, SREA$, OR SPFA$
```

(Continued on next page)

Example 3–10a (Cont.): Active Tasks with Receive Queue, AST Queue, and Receive-by-Reference Queue Listings

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 23
ACTIVE TASKS
          (25)                (26)                (27)
    A.BYT = 000016    A.AST = 000610    A.NPR = 000000

    023614    024430  001000  000016  000610  000000  114600  013355  000477   (28)
    023634    000000

    AST BLOCK ADDRESS = 024430    A.CBL = 001400

    SPECIFIED AST FROM: SFPA$, SRDA$, SRRA$, SPEA$, SREA$, OR SPFA$

    A.BYT = 000016    A.AST = 000702    A.NPR = 000000

    024430    000000  001400  000016  000702  000000  000000  000000  002171
    024450    031250
    RECEIVE BY REFERENCE QUEUE
    --------------------------
              (29)                (30)
    RRQ ADDRESS = 030754    SENDER TCB = 000000

    TASK = TT0
                          (31)                              (32)
    REGION ID OF RECEIVER = 023510    OFFSET IN PARTITION = 000000
                 (33)                      (34)
    LENGTH TO MAP = 000001    ACCESS RIGHTS = 000001

    000000    000000  000000  100076  000000  023510  000000  000001  000001
    000020    000000  000000  000000  000000  000000  000000  000000  000000   (35)
    000040    000001  000000
```

Example 3–10b shows an Active Tasks listing with an Offspring Control Block (OCB) list. The items found in this listing are described in the following list:

❶ Exit event flag number of offspring task

❷ Name of parent task

❸ Octal dump of offspring control block in offset mode

Example 3–10b: Active Tasks with OCB Listing

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26        PAGE 20
ACTIVE TASKS

BAPO
------

  TCB ADDRESS = 113544      PAR = GEN         PCB ADDRESS = 031474

  LOAD ADDRESS = 01266500      LOAD DEVICE = LB0:      LBN = 00032073

  PRI = 80.      I/O COUNT = 0.      UIC = [200,200]      TI = C00:

  MAX SIZE = 000473      EVENT FLAGS = <1-16> 000000  <17-32> 040000

  T.STAT:  000000

  T.ST2:   000020   STP

  T.ST3:   052000  -PMD  PRV  SLV

  T.ST4:   000000

  TCB DUMP:

    000000    000000  000120  000000  006270  135600  000000  113556  000000
    000020    113562  000000  040000  022026  112634  000000  000020  052000
    000040    000120  032073  023056  031474  000473  112634  024006  024006
    000060    000000  000003  000000  040000  113570  000302  000003  000000
    000100    000000  113644  000000  031074  031074  000001  024324  002147
    000120    000000

  OCB LIST:
  ---------
                          (1)
    EXIT EVENT FLAG (0.EFN) = 000000

    PARENT TASK NAME = QMG...   (2)

    OCB DUMP:

    000000    000000  000000  112634  121444  000000  000000  066117  131574  (3)
    000020    000000  000000  000000  000010  000000  123340
```

3.2.2 Task Headers

Example 3–11 is an example of a task header listing. The following list describes its contents:

❶ Task name

❷ Pointer to the first word in the task header

❸ Pointer to the first word in the Task Control Block (TCB)

❹ Contents of Processor Status Word (PSW) and Program Counter (PC)

❺ Contents of the general registers

❻ Initial contents of the PSW, the PC, and the stack pointer (SP)

The task header size in decimal bytes, the number of windows required to map the task, and the number of logical unit numbers assigned to the task

❸ Current and default UIC

❾ Pointer to number of window blocks

❿ Pointer to header guard word

⓫ Work area extension vector pointer

⓬ Priority difference for swapping

⓭ Directive Status Word

⓮ Address of File Control Services (FCS) impure area

⓯ Address of FORTRAN impure area

⓰ Address of overlay impure storage

LOGICAL UNIT TABLE

⓱ Logical unit number (LUN)

⓲ Physical device name before redirect

⓳ Window pointer in header

⓴ Low-order byte of this word indicates the number of map entries active; the high-order byte has the following bit assignments:

WI.RDV=400	read virtual address allowed if set
WI.WRV=1000	write virtual block allowed
WI.EXT=2000	extend allowed if set
WI.LCK=4000	set if locked against shared access
WI.DLK=10000	set if deaccess lock enabled
WI.BPS=100000	bypass access interlock if set

㉑ Address of File Control Block

㉒ File number

㉓ File sequence number

㉔ File Control Block status word

㉕ Number of accesses

㉖ Number of block locks

WINDOW BLOCKS

㉗ The name of the partition in which the task runs

㉘ The virtual limits of the task

㉙ Address of attachment descriptor

㉚ Window size in 32-word blocks

Offset into partition

㉜ Address of the first Page Description Register (PDR) used to map the window

㉝ Number of PDRs used

㉞ The contents of the last PDR used

㉟ Octal dump of task header in offset mode

Example 3-11: Task Headers (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 24
TASK HEADERS

...LDR  ❶
------

                ❷                          ❸
  HEADER ADDRESS = 117330       TCB ADDRESS = 117500

  PS=170000     PC=120462  ❹

  R0=000216  R1=012657  R2=177777  R3=032454  R4=032424  R5=032300  SP=120166   ❺

  INITIAL PS = 170017   INITIAL PC = 120230   INITIAL SP = 120166   ❻

  HEADER SIZE = 102.    NO. OF WINDOWS = 1.   NO. OF LUNS = 1.  ❼

  CURRENT UIC = [1,24]      DEFAULT UIC = [1,24]  ❽
       ❾                    ❿                   ⓫                  ⓬
  H.WND = 117432    H.GARD = 117474    H.VEXT = 000000     H.SPRI = 0.
     ⓭                   ⓮                   ⓯                  ⓰
  DSW = 000001      H.FCS = 000000     H.FORT = 000000     H.OVLY = 000000

  LOGICAL UNIT TABLE:
 ⓱ ⓲    ⓳       ⓴       ㉑       ㉒       ㉓      ㉔     ㉕   ㉖
  #  DEV  WINDOW   W.CTL   W.FCB   F.FNUM  F.FSEQ  F.STAT  NAC  NLCK
  -  ---  ------   -----   -----   ------  ------  ------  ---  ----
  1  LB0: 000000

           INSTRUCTION   SPACE
           -------------------
  WINDOW BLOCKS:
   ㉗        ㉘           ㉙          ㉚       ㉛       ㉜       ㉝      ㉞
   PAR      VIRT LIMITS   ATT DESC  WND SIZE  OFFSET  1ST PDR  NO.   LAST PDR
   ---      -----------   --------  --------  ------  -------  ---   --------
  SYSPAR  120000 124277   117250    000043    000000  177612    1    021006

  HEADER:

    000000    120166  000146  000000  000000  000424  000424  170017  120230
    000020    120166  000000  000000  000000  000000  000000  000000  000000
    000040    000000  000000  117432  000001  000000  000000  000000  000000
    000060    000000  000000  000000  000000  000000  117474  000001  023056   ㉟
    000100    000000  000001  117264  120000  124277  117250  000043  000000
    000120    000612  021006  170000  120462  032300  032424  032454  177777
    000140    012657  000216  000000
```

3.2.3 Command Line Interpreter Parser Block (CPB)

The listing shown in Example 3–12 contains the Command Line Interpreter Parser Block for MCR. The listing corresponds to the following items:

❶ Task name of the CLI

❷ Starting address of the CPB

❸ C.PSTS, which is the CPB status word

❹ Dump of the CPB in octal

Example 3–12: CLI Parser Blocks

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 124
CLI PARSER BLOCKS

CLI TASK NAME   MCR...  ❶
CPB ADDRESS = 021464    CLI NAME = MCR   ❷
C.PSTS:   LGO    ❸

CPB DUMP:

    000000    117124  050712  000000  000004  003404  005015  000076  005015   ❹
    000020    041515  037122  000000

CLI TASK NAME    ...DCL
CPB ADDRESS = 024400    CLI NAME = DCL
C.PSTS:   POL

CPB DUMP:

    000000    001752  014604  000000  003000  003405  005015  020044  006400
    000020    042012  046103  000076  000000
```

3.2.4 Partition Information

CDA outputs partition information in two segments. The listing shown in Example 3–13 contains system partition information, and the listing shown in Example 3–14 represents individual partition information. The following list describes elements of Example 3–13. Individual partitions include Attachment Descriptors and Wait Queues when they apply.

❶ Partition names

❷ Partition Control Block (PCB) address

❸ Base address of partition in memory

❹ Size of the partition

❺ Type of partition

❻ Task or tasks occupying the partition

Example 3–13: Partition Information

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 35
PARTITION INFORMATION


                          M E M O R Y    M A P

     ❶           ❷          ❸          ❹          ❺              ❻
PARTITION    PCB ADR      BASE       SIZE       TYPE         OCCUPIED BY
---------    -------      ----       ----       ----         -----------

 <EXEC>                  00000000   00074200
 <POOL>                  00077314   00074664

 SECPOL      117734      00174200   00060000   MAIN SECPOOL

 SYSPAR      117670      00254200   00147100   MAIN
             117624      00254200   00077000   SUB  RO COM    !DIR11M!
             117264      00353200   00004300   SUB  TASK      <...LDR>
             116654      00357500   00030600   SUB  TASK      <MCR...>
             116450      00410300   00007700   SUB  TASK      [TKTN  ]
             116244      00420200   00003100   SUB  TASK      [SHF...]

 DRVPAR      116164      00423300   00071600   MAIN
             116120      00423300   00006100   SUB  RO COM    !TTEXT !
             116034      00431400   00020500   SUB  RO COM    !TTCOM !
             115750      00452100   00033600   SUB  DRIVER    (TT:)
             115144      00505700   00003100   SUB  DRIVER    (DL:)
             114754      00511000   00000100   SUB  DRIVER    (NL:)
             114710      00511100   00002500   SUB  DRIVER    (VT:)
             114644      00513600   00001300   SUB  DRIVER    (RD:)

 GEN         114600      00515100   07262700   MAIN
             114534      00515100   00014700   SUB  RO COM    !PUCOM !
             023640      00532000   00000100   SUB  RW COM    !EDPREG!
                         00532100   00001700    <HOLE>
             023524      00534000   00007000   SUB  TASK      <PMT...>
             113354      00543000   00053000   SUB  RO COM    +...AT.+
             114150      00616000   00040000   SUB  RO COM    !FCSRES!
             024254      00656000   00003200   SUB  TASK      <TTO>
                         00661200   00004300    <HOLE>
             023714      00665500   00014200   SUB  TASK      <F11ACP>
             114364      00701700   00033000   SUB  RO COM    +F11ACP+
                         00734700   00036200    <HOLE>
             031540      00773100   00013200   SUB  RW COM    !XDT   !
                         01006300   00002600    <HOLE>
             024210      01011100   00034300   SUB  TASK      <QMG...>
             111410      01045400   00021000   SUB  RO COM    +...MAC+
             032424      01066400   00040300   SUB  TASK      <BRKTO >
                         01126700   00001000    <HOLE>
```

(Continued on next page)

Example 3–13 (Cont.): Partition Information

```
113134      01127700    00047000    SUB  RO COM    +...PIP+
031130      01176700    00020300    SUB  TASK      <SYSLOG>
            01217200    00047000     <HOLE>
031474      01266200    00047300    SUB  TASK      <BAPO  >
            01335500    00052000     <HOLE>
032110      01407500    00054000    SUB  RO COM    +...IND+
111630      01463500    00232200    SUB  RO COM    +...EDT+
            01715700    06046700     <HOLE>
031664      07764600    00001200    SUB  RW COM    !DLCRSH!
025244      07766000    00012000    SUB  DRIVER    (DU:)
```

Each item in the following list describes a correspondingly numbered item in Example 3–14:

PARTITION CONTROL BLOCK

❶ Partition name

❷ Pointer to the first word of the PCB

❸ Type of partition

❹ Name of main partition

❺ Physical base address of partition in 32-word blocks

❻ Partition size in 32-word blocks

❼ Pointer to the first word of the TCB of attached task

❽ Partition protection word (mapped system only)

❾ Priority of attached task or partition

❿ Resident mapped task count

⓫ Size of external header in 64-byte blocks

⓬ I/O count of attached partition

⓭ Partition status flags, using the following three-letter codes:

OUT	Partition is out of memory
CKP	Partition checkpoint in progress
CKR	Partition checkpoint is requested
CAF	Checkpoint space allocation failure
-CHK	Partition is not checkpointable
FXD	Partition is fixed
LFR	Last head of region failure
PER	Parity error in partition
LIO	Marked by Shuffler for long I/O

NSF Partition cannot be shuffled

COM Library or common block

DEL Partition should be deleted when not attached

⓮ Octal dump of PCB in offset mode

PARTITION WAIT QUEUE

⓯ Any partition with the heading WAIT QUEUE: and P.STAT containing the code OUT is awaiting access to the previous partition

ATTACHMENT DESCRIPTOR

⓰ Address of attachment descriptor

⓱ Partition to which attachment occurs

⓲ Name of attaching task

⓳ PCB attachment queue thread word

⓴ TCB attachment queue thread word

㉑ Priority of highest priority task attached to this partition

㉒ I/O count of attached partition

㉓ Number of times task is mapped through this attachment descriptor

㉔ Attachment descriptor status byte, using the following three-letter codes:

DEL Task has delete access

EXT Task has extend access

WRT Task has write access

RED Task has read access

PRO TCB is secondary pool TCB bias

SPB Cache bypass request

RBP Request to not bypass cache

㉕ Octal dump of attachment descriptors in offset mode

Example 3–14: Partition Control Blocks and Attachment Descriptors

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0     10-OCT-86   13:32        PAGE 30
PARTITION INFORMATION
   ❶          ❷       ❸     ❹     ❺      ❻       ❼       ❽       ❾    ❿   ⓫   ⓬
  NAME      PCB ADR  TYPE   MAIN   BASE    SIZE   P.OWN     PRO     PRI  RMCT HDLN PIOC
  ----      -------  ----   ----   ----    ----   -----     ---     ---  ---- ---- ----
 DLCRSH     032424   SUB    GEN    077706  000012 000000  000000  0.   0.   0.   0.

  P.STAT:  014600   -CHK  FXD -NSF  COM  ⓭

  P.ST2:   000000

    000000    000000  000000  015343  071600  000000  114600  077706  000012
    000020    000000  000000  000000  000000  014600  000000  000000  000000  ⓮
    000040    032462  000000


  NAME      PCB ADR  TYPE   MAIN   BASE    SIZE   P.OWN     PRO     PRI  RMCT HDLN PIOC
  ----      -------  ----   ----   ----    ----   -----     ---     ---  ---- ---- ----
 CROWD      025244   MAIN   CROWD  077720  000060 000000  000000  0.   0.   0.   0.

  P.STAT:  000000

  P.ST2:   000000

    000000    000000  000000  012637  110140  031020  025244  077720  000060
    000020    031744  031744  000000  000000  000000  000000  000000  000000
    000040    025302  000000

    WAIT QUEUE:  ⓯

  NAME      PCB ADR  TYPE   MAIN   BASE    SIZE   P.OWN     PRO     PRI  RMCT HDLN PIOC
  ----      -------  ----   ----   ----    ----   -----     ---     ---  ---- ---- ----
 EDPTO      031744   SUB    CROWD  000000  000033 032170  000000  60.  0.   3.   0.

  P.STAT:  110010    OUT -CHK  DEL

    000000    000000  000074  012637  110140  000000  025244  000000  000033
    000020    000000  000033  000000  032170  110010  000000  000000  000000
    000040    032002  000003

  NAME      PCB ADR  TYPE   MAIN   BASE    SIZE   P.OWN     PRO     PRI  RMCT HDLN PIOC
  ----      -------  ----   ----   ----    ----   -----     ---     ---  ---- ---- ----
 IOETO      031020   SUB    CROWD  077720  000033 031554  000000  60.  0.   3.   148.

  P.STAT:  010010   -CHK  DEL

    000000    000000  000074  012637  110140  000000  025244  077720  000033
    000020    000000  000033  000000  031554  010010  000000  000000  023334
    000040    023334  112003

    ATTACHMENT DESCRIPTORS:
      ⓰          ⓱          ⓲          ⓳       ⓴       ㉑     ㉒       ㉓
    ADDRESS   PARTITION   ATT TASK   A.PCBL   A.TCBL   PRI   IOC   MAP COUNT
    -------   ---------   --------   ------   ------   ---   ---   ---------
    023334     CROWD       IOETO     000000   000000   60.   148.      1.

    A.STAT:  WRT RED  ㉔

    000000    000000  112074  031554  000000  000403  031020  ㉕
```

3.2.5 Common Block Directory

CDA lists partition information, status words, and Partition Control Blocks for each installed, named common region. The listing in Example 3–14 shows a Common Block Directory entry. The following list describes the items in Example 3–14:

❶ Name of the installed common region partition

❷ Address of Partition Control Block (PCB)

❸ Type of partition

❹ Name of the main partition

❺ Physical base address of partition

❻ Size of partition in 32-word blocks

❼ Owning UIC of the common region

❽ Partition protection word

❾ Priority of attached task or partition

❿ Resident mapped task count

⓫ Size of external header, in 64-byte blocks

⓬ I/O count of attached partition

⓭ Partition status words, using the following three-letter codes:

OUT	Partition is out of memory
CKP	Partition checkpoint is in progress
CKR	Partition checkpoint is requested
CAF	Checkpoint space allocation failure
-CHK	Partition is not checkpointable
FXD	Partition is fixed
LFR	Last head of region failure
PER	Parity error in partition
LIO	Marked by Shuffler for long I/O
NSF	Partition cannot be shuffled
COM	Library or common block
DEL	Partition should be deleted when not attached

⓮ Octal dump of PCB

⓯ Address of PCB of the common task image file

⓰ Address of Unit Control Block (UCB) of the device on which the common resides

⓱ Starting logical block number (LBN) of the common task image file

⓲ Word that always contains a 0

Example 3–14: Common Block Directory

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 53
COMMON BLOCK DIRECTORY
 ❶          ❷        ❸     ❹      ❺       ❻       ❼       ❽       ❾     ❿    ⓫    ⓬
 NAME      PCB ADR  TYPE   MAIN   BASE    SIZE    P.OWN    PRO     PRI  RMCT HDLN PIOC
 ----      -------  ----   ----   ----    ----    -----    ---     ---  ---- ---- ----
 PUCOM     114534   SUB    GEN    005151  000147  000424  021042   0.    2.   0.   0.

  P.STAT:  014600   -CHK  FXD -NSF  COM
                                      ⓭
  P.ST2:   040106    LMA  RON  APR

    000000    000000  001000  063513  057710  023640  114600  005151  000147
    000020    114064  000147  114514  000424  014600  040106  021042  000000    ⓮
    000040    114572  000000

    COMMON TASK IMAGE FILE PCB
       ⓯          ⓰          ⓱          ⓲
      PCB ADR    P.UCB      P.LBN      P.REL
      -------    -----      -----      -----
      114514     023056  000000,026037  000000
```

3.2.6 Device Information

CDA lists information on all devices known to the system. The listing in Example 3–16 shows a disk device listing with an I/O packet. Items 14, 15, 16, and 17, the Terminal Status Words, appear only in listings for terminal devices. The codes for these items apply only to the full-duplex terminal driver. The section labeled I/O REQUEST PACKETS appears only for devices that have an I/O request in progress or an I/O request queued at the time of the system crash. The following list describes the items in Example 3–16:

❶ Device name

❷ Address of offset 0 in Unit Control Block (UCB)

❸ Address of offset 0 in Device Control Block (DCB)

❹ Address of offset 0 in Status Control Block (SCB)

❺ Device to which unit is redirected

❻ Name of Ancillary Control Processor (ACP)

❼ Name of attached task

❽ Pointer to the UCB name of the owning terminal

❾ UIC used to log into the system

❿ Unit status byte, using the following three-letter codes:

BSY	Unit is busy
-MNT	Unit is not mounted

FOR	Unit is mounted as a foreign volume
MDM	Unit is marked for dismount
PWF	Power fail occurred
WCK	Write check enabled
SPU	Unit is spinning up
VV	Volume is valid

⓫ Unit status extension byte, using the following three-letter codes:

OFL	Unit off line
-RED	Unit is not redirectable
PUB	Unit is public device
UMD	Unit attached for diagnostics
PDF	Privileged diagnostic functions only

⓬ Control Processing flags, using the following three-letter codes:

-ALG	Byte alignment not allowed
NPR	Device is a NPR device
QUE	Call driver before queuing
PWF	Always call driver at power fail entry point
ATT	Call driver on attach/detach
KIL	Always call driver at I/O kill

⓭ First device characteristics word, using the following three-letter codes:

REC	Record-oriented device
CCL	Carriage-control device
TTY	Terminal device
DIR	File-structured device
SDI	Single directory device
SQD	Sequential device
MSD	Mass storage device
EXT	Unit on extended 22-bit UNIBUS controller
UMD	User-mode diagnostics supported
MBC	MASSBUS device
SWL	Unit software write locked

ISP	Input spooled device
OSP	Output spooled device
PSE	Pseudo device
COM	Device is mountable as COM channel
F11	Device is mountable as Files–11 device
MNT	Device is mountable

⓮ With heading of U.CW2: second device characteristics word, using the following three-letter codes:

DH1	Unit is a multiplexer
DJ1	Unit is a DJ11
RMT	Unit is remote
HFF	Unit handles hardware form feeds
NEC	Solicited input not echoed
CRT	Unit is a CRT
ESC	Unit generates escape sequences
-LOG	User not logged in on terminal
SLV	Unit is a slave terminal
DZ1	Unit is a DZ11
HLD	Terminal is in hold screen mode
AT.	MCR command AT. is being processed
PRV	Unit is a privileged terminal
L3S	Unit is an LA30S terminal
VT5	Unit is a VT05B terminal
LWC	Lowercase to uppercase conversion

⓯ With heading of U.TSTA: terminal status word, using the following three-letter codes:

RST	Read with special terminators in progress
RUB	Rubout sequence (non-CRT) in progress
ESC	Escape sequence in progress
RAL	Read pass all in progress
RNE	Echo suppressed
CTO	Output disabled
OBY	Output busy

IBY	Input busy
BEL	Bell pending
DPR	Defer processing of character in buffer
DEC	Defer echo of character in buffer
DSI	Input processing disabled
CTS	Output stopped by CTRL-S
USI	Unsolicited input in progress
OBF	Buffered output in progress
IBF	Buffered input in progress

⓰ With heading of U.TSTA+2: second terminal status word, using the following three-letter codes:

ACR	Wrap-around required
CR	Trailing carriage return required on output
BRQ	Break-through write is queued
WRA	Control for wraparound
SRQ	Special request is queued
WRB	Low bit in 52-WRA bit pattern
ORQ	Output request is queued
IRQ	Input request is queued
HFL	Horizontal fill required
VFL	Vertical fill required
HHT	Hardware horizontal tab is present
HFF	Hardware form-feed is present
FLF	Force line feed before next echo
FDX	Line is full duplex mode

⓱ With heading of U.TSTA+4: fourth terminal status word, using the following three-letter codes:

RAL	Terminal is in read-pass-all mode
WES	Task waiting for escape sequence
RPO	Read with prompt output in progress
TAB	Type-ahead buffer allocation requested
8BC	Pass eight bits on input

ABD	Autobaud speed detection enabled
RCU	Restore cursor
ABP	Autobaud speed detection in progress
WAL	Terminal is in write-pass-all mode
VER	Last character in type-ahead buffer has a parity error
BCC	Last character in type-ahead buffer has a framing error
DAO	Last character in type-ahead buffer has a data overrun error
PCU	Position cursor

UNIT CONTROL BLOCK

⓲ Octal dump of Unit Control Block (UCB), including negative offsets

⓳ Octal dump of UCB extension if a UCB extension is present

DEVICE CONTROL BLOCK

⓴ Octal dump of Device Control Block

STATUS CONTROL BLOCK

㉑ Octal dump of Status Control Block

I/O REQUEST PACKETS

㉒ Address of the first word of the I/O packet

㉓ Name of the task requesting I/O

㉔ Priority of the task requesting I/O

㉕ Event flag number used to signal I/O completion

㉖ Logical unit number used by requesting task

Note

If the task was checkpointed while the packet was queued, this number may not be correct. If the address in I.LN2 is within the task header, the logical unit number is correct.

㉗ I/O function codes (for detailed information on the legal I/O function codes for each device, refer to the *RSX-11M-PLUS and Micro/RSX I/O Drivers Reference Manual*)

㉘ Status of the I/O request - current or queued

㉙ Octal dump of I/O request packet in offset mode

㉚ I.LN2 - pointer to the second word of the LUN

Example 3–16: Device Information and I/O Packet (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    10-OCT-86   13:32       PAGE 47
DEVICE INFORMATION

DUO:  (1)
-----
     (2)       (3)       (4)       (5)       (6)     (7)     (8)       (9)
  UCB ADR   DCB ADR   SCB ADR   REDIRECT   ACP     ATT     OWNER   LOGIN UIC
  -------   -------   -------   --------   ---     ---     -----   ---------
  024602    024542    025140                                NONE

  U.STS:   BSY  FOR   VV  (10)

  U.ST2:  (11)

  U.CTL:   -ALG  NPR  PWF  LGH  (12)

  U.CW1:    MNT  F11  UMD  MSD  DIR  (13)
(14)

(15)

(16)

(17)

    UNIT CONTROL BLOCK:  (18)

    024572    000000  000377  031700  000000  024542  024602  120721  000000
    024612    140710  000000  166100  001000  025140  000000  000000  000000
    024632    000000  002137  000000  024334  000000  000000  000000  000200
    024652    000000  024652  002136  000000

    UNIT CONTROL BLOCK EXTENSION:  (19)

    000000    000000  000000  000000  000000  000000  000000  000000  000000
    000020    000000  070503  144000  003336  000000  002410  000000  157000
    000040    000006  000000  000000  000000  005000  000000  000000  000000
    000060    000000  000000  000000  000000  000000  000000

    DEVICE CONTROL BLOCK:  (20)

    024542    022750  024602  052504  001400  000060  120314  177477  000070
    024562    000000  177200  000377  000000  000000  000377  031700

    STATUS CONTROL BLOCK:  (21)

    025140    042150  072210  000000  000000  025022  025140  013355  042100
    025160    074170  000000  000204  025130  000064  000000  063513  005151

    I/O REQUEST PACKETS:
       (22)          (23)          (24)    (25)    (26)        (27)          (28)
    PACKET ADR    REQUESTOR      PRI     EFN     LUN     FUNCTION CODE     STATUS
    ----------    ---------      ---     ---     ---     -------------     ------
      042150        IOETO        60.     1.      1.        IO.RLB          QUEUED

    000000    033250  000474  031554  140100  024602  001000  000000  000000
    000020    000000  000000  017400  173554  001000  000000  000000  000263  (29)
    000040    000000  000000  023334  000000
```

3.2.7 System Task Directory

CDA scans the System Task Directory and outputs the information contained in Example 3–17. The information in this format is identical to the first 17 items described in Example 3–10a of this manual.

Example 3–17: System Task Directory (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    26-SEP-86   13:51        PAGE 1
SYSTEM TASK DIRECTORY (PRIMARY POOL)

...LDR
------

  TCB ADDRESS = 117500      PAR = SYSPAR      PCB ADDRESS = 117264

  LOAD ADDRESS = 00353200      LOAD DEVICE = LB0:      LBN = 00000000

  PRI = 248.      I/O COUNT = 0.      UIC = [1,24]      TI = C00:

  MAX SIZE = 000043      EVENT FLAGS = <1-16> 000001  <17-32> 000000

  T.STAT:  000000

  T.ST2:   020020  -CHK  STP

  T.ST3:   050200  -PMD  PRV  NSD

  T.ST4:   000000

  TCB DUMP:

    000000    000000  000370  000000  131574  045662  000000  117512  000000
    000020    117516  000001  000000  022026  116514  000000  020020  050200
    000040    000370  000000  023056  117264  000043  117124  117256  117256
    000060    000000  000000  000000  000001  117522  000043  000000  000000
    000100    000000  117600  000000  000000  117606  000000  000000  002147
    000120    000000
```

3.2.8 Pool Dump

As shown in Example 3–18, CDA prints the system pool in octal, Radix–50, and ASCII. On RSX–11M–PLUS systems with secondary pool support, CDA prints a dump of the secondary pool with the /SECPOL switch. If a line is repeated more than nine times, CDA prints it once and then prints a message indicating the number of identical lines.

The symbols in Example 3–18 have the following meanings:

* Indicates that the next word is allocated.

\+ Indicates that the next word is contained in an unused, preallocated I/O packet (in $PKAVL free list).

Note

$PKAVL is a list containing fixed-size blocks. The blocks in this list are used for fast allocation, and I.LGTH determines the length of these blocks.

\- Indicates that the next word is allocated in both $CRAVL and $PKAVL. (This is an error condition.) $CRAVL is the free pool list head.

Example 3–18: Pool Dump (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 84
POOL DUMP

                        S Y S T E M     P O O L

   * = NEXT WORD ALLOCATED          FIRST FREE BLOCK ($CRAVL) = 023224
   + = NEXT WORD IS IN $PKAVL LIST                   $PKAVL  = 023304
   - = NEXT WORD IS IN $PKAVL AND ALSO IN $CRAVL

 023114    + 031024 + 000050 + 000020 + 000000  !H T  A    P    !! 2(     !
 023124    + 115272 + 001010 + 123024 + 010717  !XOJ  M  ZV6 B39!!:    &0 !
 023134    + 140024 + 000000 + 002000 + 073170  !O/L      YX R8H!! @    xv!
 023144    + 001564 + 000001 + 001564 + 001564  ! VD   A  VD  VD!!t   t t !
 023154    + 000001 + 000000 + 000000 + 000000  !  A            !!       !
 023164    * 024454 * 000000 * 112634 * 127644  !FWT     W7T .DD!!,)    $/!
 023174    * 000000 * 121452 * 000000 * 000000  !    ZDJ        !!  *#    !
 023204    * 000016 * 000646 * 000000 * 000062  !  N  JV       AJ!!  &   2 !
 023214    * 032154 * 023446 * 000403 * 024254  !HOT FJV  FS FTL!!14&'  ,(!
 023224      023300   000004 * 000402 * 032110  !FH    D  FR HNX!!@&    H4!
 023234    * 114150 * 000000 * 023414 * 001000  !XOP     FI6  L2!!h    '  !
 023244    * 112760 * 131162 * 000000 * 125230  !W9X .VB     $K2!!p r2   *!
 023254    * 000002 * 113002 * 000000 * 000560  !  B X B      IH!!       p !
 023264    * 000000 * 000007 * 000001 * 000062  !      G   A  AJ!!       2 !
 023274    * 000000 * 000000   023454   000004  !        FJ.   D!!     ,'  !
 023304    + 023114 + 000050 + 000020 + 000000  !FED  A    P    !!L&(     !
 023314    + 115272 + 001010 + 123024 + 010717  !XOJ  M  ZV6 B39!!:    &0 !
 023324    + 140024 + 000000 + 002000 + 073170  !O/L      YX R8H!! @    xv!
 023334    + 006530 + 000001 + 006530 + 000000  !BEP   A BEP     !!X   X   !
 023344    + 000000 + 000000 + 000000 + 000000  !                !!       !
 023354    * 000000 * 000224 * 112760 * 000000  !     C. W9X     !!    p   !
 023364    * 000403 * 023524 * 000000 * 002400  ! FS FK.       2 !!  T'     !
 023374    * 000022 * 124260 * 000002 * 000144  !  R Z9X   B  BT!!  0(  d !
 023404    * 000000 * 073376 * 100660 * 000000  !    SAV TO     !!  ~v0   !
 023414    * 023570 * 000000 * 112364 * 145201  !FLX     W3L 2PA!!x'  t  J!
 023424    * 000000 * 124402 * 000000 * 000000  !    $AZ        !!   )    !
 023434    * 000000 * 000477 * 023510 * 000062  !     G9 FKP  AJ!!  ? H'2 !
 023444    * 032154 * 023516 * 000401 * 023640  !HOT FKV  FQ FMX!!14N'   '!
 023454      023704   000004 * 000000 * 000113  !FNT   D      A5!!D'    K !
 023464    * 112634 * 023502 * 000403 * 024210  !W7T FKJ  FS FSP!!  B'   (!
 023474    * 000000 * 000113 * 112634 * 000000  !     A5 W7T     !!  K     !
 023504    * 000401 * 114150 * 000000 * 000062  ! FQ XOP       AJ!!  h   2 !
 023514    * 032154 * 000000 * 000001 * 023640  !HOT        A FMX!!14     '!
 023524    * 113354 * 000224 * 026226 * 000000  !XE6  C. GEN     !!1    ,  !
 023534    * 113354 * 114600 * 005340 * 000070  !XE6 XVP A/X  AP!!1   ' 8 !
 023544    * 000000 * 000070 * 000000 * 112760  !     AP      W9X!!  8   p !
 023554    * 000010 * 000000 * 000000 * 023354  !  H          FID!!      l&!
 023564    * 023354 * 000002 * 000000 * 000000  !FID   B         !!l&      !
 023574    * 112364 * 104641 * 000003 * 124504  !W3L V 3   C $CL!!t !   D)!
 023604    * 000000 * 000000 * 023400 * 177776  !         FIX  8N!!      '~ !
 023614    * 024430 * 001000 * 000016 * 000610  !FW   L2   N  I2!! )       !
 023624    * 000000 * 114600 * 013355 * 000477  !    XVP CZ/  G9!!     m ? !
 023634    * 000000 * 000477 * 000000 * 000463  !     G9       G$!!  ?   3 !
 023644    * 017760 * 070517 * 023524 * 114600  !EDP REG FK. XVP!!p OqT'   !
 023654    * 005320 * 000001 * 114150 * 000001  !A/H   A XOP    A!!P   h    !
 023664    * 000000 * 003440 * 000210 * 000200  !    AEX  CP  CH!!         !
```

(Continued on next page)

Example 3–18 (Cont.): Pool Dump (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26       PAGE 101
SECONDARY POOL DUMP


                    S E C O N D A R Y   P O O L

  NUMBER OF FREE BYTES = 00035000      FIRST FREE BYTE = 00217200

  LENGTH = 00060000 BYTES

  START ADDRESS = 00174200            ENDING ADDRESS = 00254177

00174200    000000  000000  000000  000000    !               !!         !
00174210    000000  000000  000000  000000    !               !!         !
00174220    000000  000000  000000  000000    !               !!         !
00174230    000000  000000  000000  000000    !               !!         !
00174240    000000  000000  000000  000000    !               !!         !
00174250    000000  000000  000000  000000    !               !!         !
00174260    000000  000000  001745  000000    !         X7    !!    e    !
00174270    000000  000000  000000  000000    !               !!         !

                [ABOVE LINE REPEATED 16. TIMES]

00174500    000000  000400  000001  000000    !     FP   A    !!         !
00174510    001402  045527  041114  000072    ! SJ LBG JXL  AR!!   WKLB: !
00174520    000000  000000  000000  000000    !               !!         !
00174530    000000  000000  000000  000000    !               !!         !
00174540    000000  000000  000000  000000    !               !!         !
00174550    000000  000000  000000  000000    !               !!         !
00174560    000000  000000  000000  000000    !               !!         !
00174570    000000  000000  000000  000000    !               !!         !
00174600    000000  000101  113670  131574    !     AY XK  ...!!  A 8 |3!
00174610    003273  000000  140012  000000    !ACC     O/B    !!;    @  !
00174620    140016  000000  000000  000000    !O/F            !! @       !
00174630    002030  100000  020000  050000    ! ZH TSH ED2 L2 !!        P!
00174640    000101  030660  023056  114600    ! AY G8  FDN XVP!!A 01.&   !
00174650    000311  000000  000000  140054    ! EA          O/6!!I     ,@!
00174660    000010  000003  000000  000000    !  H   C        !!         !
00174670    000000  000120  000003  000000    !     B    C    !!  P      !
00174700    000000  140100  000000  000000    !    OOP         !!  @@     !
00174710    140106  000000  000000  000000    !OOV             !!F@       !
00174720    000000  000000  000000  000000    !               !!         !
00174730    000000  000000  000000  000000    !               !!         !
00174740    000000  000000  000000  000000    !               !!         !
00174750    000000  000000  000000  000000    !               !!         !
00174760    000000  000000  000000  000000    !               !!         !
00174770    000000  000000  000000  000000    !               !!         !
00175000    000000  000062  000000  131574    !     AJ      ...!!  2   |3!
00175010    007537  000000  140012  000000    !BRO     O/B    !!_    @  !
00175020    140016  000000  000000  000000    !O/F            !! @       !
00175030    001756  100000  020000  050000    ! YF TSH ED2 L2 !!n       P!
00175040    000062  026527  023056  114600    ! AJ GJG FDN XVP!!2 W-.&   !
00175050    000363  000000  000000  140054    ! FC          O/6!!s     ,@!
00175060    000010  000003  000000  000000    !  H   C        !!         !
00175070    000000  000360  000003  000000    !     F    C    !!  p      !
00175100    000000  140100  000000  000000    !    OOP         !!  @@     !
```

3.2.9 Task Dump

CDA prints all or a portion of the task's virtual address space if the /TASK switch is specified. Example 3–19 and the following list illustrate this output.

❶ Task name

❷ Address of the first word of the Task Control Block for the task

❸ Address of the first word of the task's header

WINDOW BLOCKS

❹ Name of the partition to which the task is mapped

❺ Task virtual address limits

❻ Address of the attachment descriptor

❼ Size of window in 32-word blocks

❽ Offset to memory region within partition in 32-word blocks

❾ First Page Description Register (PDR) used to map the task

❿ Number of PDRs used to map task

⓫ Contents of the last PDR used to map the task

⓬ Task virtual address limits

⓭ Physical starting address of the memory region being dumped

⓮ Dump of the data within the window, formatted in octal, Radix–50, and ASCII

Example 3–19: Task Dump (Truncated)

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    26-SEP-86   13:50        PAGE 1
TASK DUMP

                          TASK DUMP OF TT0 ❶
                          ------------------

          INSTRUCTION   SPACE
          ------------------

                    ❷                                ❸
           TCB ADDRESS = 032154        HEADER ADDRESS = 006560

 WINDOW BLOCKS:
  ❹          ❺           ❻         ❼         ❽        ❾     ❿      ⓫
  PAR     VIRT LIMITS   ATT DESC  WND SIZE  OFFSET  1ST PDR  NO.  LAST PDR
  ---     -----------   --------  --------  ------  -------  ---  --------
 GEN      000000 000777   023210    000010  000004  177600    1    003406

 WINDOW #1 -- TASK VIRTUAL LIMITS 000000-000777 ⓬
 ----------------------------------------------

   PHYSICAL STARTING ADDRESS = 00656400 ⓭
   -----------------------------------

   000000    000000  000232  140315  160400    !     C4 037 6  !!     M@ a!
   000010    003440  003440  174017  000314    !AEX AEX 9$W  ED!!     xL !
   000020    001314  000000  000000  000000    ! Q6             !!L        !
   000030    000000  000000  000000  000000    !                !!         !
   000040    000000  140232  000126  000000    !    02Z  BF     !!   @V    !
   000050    000000  000000  000000  000000    !                !!         !
   000060    000000  000000  000000  000000    !                !!         !
   000070    000000  000230  000006  000000    !     C2   F     !!         !
   000100    000000  000000  000000  000000    !                !!         !
   000110    000000  000000  000000  000000    !                !!         !
   000120    000000  000000  000000  000003    !                C!!        !  ⓮
   000130    000000  000000  000000  000000    !                !!         !
   000140    000000  000000  000000  000000    !                !!         !
   000150    000000  000000  000000  000000    !                !!         !
   000160    000000  000000  000000  000000    !                !!         !
   000170    000000  000000  000000  000000    !                !!         !
   000200    000000  000000  000000  000000    !                !!         !
   000210    000000  000000  000002  160400    !          B 6   !!        a!
   000220    140315  000000  000000  000000    !037             !!M@       !
   000230    000000  000000  000000  000000    !                !!         !
   000240    000000  000000  000000  000000    !                !!         !
   000250    000000  000000  000000  000000    !                !!         !
   000260    000000  000000  000000  000000    !                !!         !
   000270    000000  000000  000000  000000    !                !!         !
   000300    000000  000000  000000  000000    !                !!         !
```

3.2.10 Clock Queue

The example in Example 3–20 shows a clock queue listing. The following list explains the example:

❶ Address of the clock queue entry

❷ One of the following types of time schedule requests:

Type 0	Mark time request
Type 2	Request with periodic rescheduling
Type 4	Single-shot task request
Type 6	Single-shot internal system subroutine with system subroutine identification
Type 10	Single-shot internal system subroutine without system subroutine identification
Type 12	Clear stop bit (Shuffler)

❸ Task Control Block address or system subroutine identification

❹ Task issuing the clock request

❺ The hour, minute, and second that time request comes due

❻ This field varies with each type of time schedule request

For a mark time request, the labels are:

C.AST	AST address
C.SRC	Event flag mask word
C.DST	Event flag mask address
	Event Flag Number

For a periodic rescheduling request, the labels are:

C.RSI	Reschedule internal
C.UIC	Scheduling UIC

The field for a single-shot task request contains only one label:

C.UIC	Scheduling UIC

The field for a single-shot internal subroutine (both with and without system subroutine identification) contains:

C.SUB	Subroutine address
C.AR5	Relocation base address (for loadable drivers)

❼ Octal dump of clock queue in offset mode

Example 3–20: Clock Queue

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    25-SEP-86   15:26        PAGE 76
CLOCK QUEUE

          ❶                ❷                ❸              ❹
ADDRESS = 023164    REQUEST TYPE = 0    TCB = 112634    TASK = QMG...

TIME REQUEST BECOMES DUE = 14:22:40.8  ❺

C.AST = 121452    C.SRC = 000000    C.DST = 000000    EVENT FLAG = 0.  ❻

000000    024454  000000  112634  127644  000000  121452  000000  000000  ❼
000020    000016

ADDRESS = 024454    REQUEST TYPE = 0    TCB = 032154    TASK = TT0

TIME REQUEST BECOMES DUE = 14:22:43.5

C.AST = 000000    C.SRC = 000001    C.DST = 032176    EVENT FLAG = 1.

000000    023240  000400  032154  130105  000000  000000  000001  032176
000020    122526

ADDRESS = 023240    REQUEST TYPE = 0    TCB = 112760    TASK = PMT...

TIME REQUEST BECOMES DUE = 14:22:52.8

C.AST = 125230    C.SRC = 000002    C.DST = 113002    EVENT FLAG = 2.

000000    023414  001000  112760  131162  000000  125230  000002  113002
000020    000000
```

3.2.11 Controller Information

Example 3–21 shows the information associated with a device controller. The following list explains the items in Example 3–21:

❶ Name of the device controller

❷ Address of the Controller Table (CTB)

❸ Address of the Device Controller Block (DCB) for this device

❹ CTB status byte, which may contain any of the following three-letter status codes:

CLK Clock block appears at the top of the CTB

MDC Multidriver CTB

CBL Clock block is linked into the clock queue

CIN Controller uses the common interrupt dispatch table

NET DECnet device

Octal dump of the CTB

❻ Common Interrupt Address

DCB for each device interfaced by this controller

❽ Name of each device interfaced by this controller

❾ Address of the Controller Request Block (KRB)

❿ Controller status; the following is a list of possible status values and their meanings:

OFL	Controller is off line
MOF	Controller is marked for off line
UOP	Controller supports overlapped operations
MBC	Device is a MASSBUS controller
SDX	Seek operations allowed during data transfers
POE	Parallel operations enabled
UCB	UCB table present
DIP	Data transfer in progress
PDF	Privileged diagnostic functions only

⓫ Octal dump of KRB in one or two parts

Example 3-21: Controller Information

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    10-OCT-86   14:22       PAGE 1
CONTROLLER INFORMATION

  RH  ❶
  --

                ❷                    ❸
    CTB ADDRESS = 022216       L.DCB = 022274

    L.STS:  MDC CIN   ❹

    CONTROLLER TABLE BLOCK:

    022214    000000  022316  044122  022274  005020  000000  110404  114474
    022234    105614  000000  110460  114554  105664  000000  110532  114636   ❺
    022254    107202  000000  110614  114722  105734

    DEVICES INTERFACED BY THIS CONTROLLER:

    COMMON INTERRUPT ADDRESS = 023300   ❻

      ❼          ❽
      DCB      DEVICE NAME
      ---      -----------
    113270         DB
    107314         DR
    107030         EM
    104574         MM

    KRB ADDRESS = 110404   ❾

    K.STS:  SDX UOP MBC POE UCB   ❿

    CONTROLLER REQUEST BLOCK:

    110366    000000  110532  110614  000000  015240  000002  000174  176300
    110406    000034  000003  000000  000000  110414  000001  000000  000000
    110426    000000  000000  000000  000000

    110424    000000  000000  000000  000000  000000  000050  000000  000000
    110444    000000  107544
```

3.2.12 Kernel Data Space

Example 3–22 shows a dump of kernel data space from the specified starting virtual address to the specified ending virtual address.

Example 3–22: Kernel Data Space

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    26-SEP-86   13:51       PAGE 1
KERNEL DATA SPACE DUMP


        VIRTUAL ADDRESS:  020000      PHYSICAL ADDRESS:  00074200

020000    000000  000000  000000  000000   !               !!        !
020010    000000  000000  000000  000000   !               !!        !
020020    000000  000000  000000  000000   !               !!        !
020030    000000  000000  000000  000000   !               !!        !
020040    000000  000000  000000  000000   !               !!        !
020050    000000  000000  020052  000000   !        EE4    !!    *   !
020060    000000  020060  000000  000000   !    EF         !!  0     !
020070    000000  000000  000000  000000   !               !!        !
020100    000000  000000  000000  000000   !               !!        !
```

3.2.13 Kernel Instruction Space

Example 3–23 shows a dump of kernel instruction space from the specified starting virtual address to the specified ending virtual address.

Example 3–23: Kernel Instruction Space

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    26-SEP-86   13:51       PAGE 1
KERNEL INSTRUCTION SPACE DUMP


        VIRTUAL ADDRESS:  020000      PHYSICAL ADDRESS:  00020000

020000    012021  012021  011011  020127   !CHQ CHQ B5Q EF9!!      W !
020010    172276  001405  012700  177660   !9FN  SM CSH  6P!!>t  @ O !
020020    012701  172260  000760  012667   !CSI 9F   LP CR9!!A Otp 7 !
020030    152342  012603  000207  010246   !4BR CQ$  CO BZV!!bT    & !
020040    012702  172200  052703  100000   !CSJ 9D2 M.5 TSH!!B  tCU  !
020050    106303  006203  103006  016261   !VT5 B C UQO DWY!!C     1 !
020060    000040  000040  012221  005720   !  2   2 CKY A5X!!      P !
020070    000405  016061  000040  000040   ! FU DTQ   2   2!!  1     !
020100    012021  005722  105703  100361   !CHQ A5Z VNS TYI!!  R C q !
```

3.2.14 Cache Dump

The following example shows a sample cache dump listing, produced with the /CACHE switch. Noted sections are described as follows:

❶ Device for which data exists in cache memory

❷ Address of the Cache Extent Descriptor (CED) that describes the region of the disk read into cache

❸ Starting logical block number on disk from which the data was read

❹ Length (in octal) of the data read into cache memory from the disk, in 512-byte blocks

❺ Physical memory address where the disk data is located now in cache memory

Example 3–24: Cache Information

```
RSX-11M-PLUS CRASH DUMP ANALYZER   V4.0    8-OCT-86   16:20       PAGE 1
CACHE INFORMATION

                DUMP OF CACHE REGION:  DEMO
               ----------------------------

❶
DEVICE:  DUO:
     ❷ DESCRIPTOR     ❸ STARTING LOGICAL     ❹ LENGTH          ❺ PHYSICAL
       ADDRESS          BLOCK NUMBER           (DISK BLOCKS)     ADDRESS
       01471310         00000003               10                02102500
       01471250         00013352               1                 02101500
       01470350         00073041               1                 02063500
       01470410         00073042               1                 02064500
       01470450         00073043               1                 02065500
       01470250         00073044               1                 02061500
       01470510         00073045               1                 02066500
       01470550         00073046               1                 02067500
       01470610         00073047               1                 02070500
       01471350         00073050               1                 02103500
       01471410         00073051               1                 02104500
       01471450         00073052               1                 02105500
       01471510         00073053               1                 02106500
       01471550         00073054               1                 02107500
       01471610         00073055               1                 02110500
       01471650         00073056               1                 02111500
       01471710         00073057               1                 02112500
       01471750         00073060               1                 02113500
       01472010         00073061               1                 02114500
       01470650         00073062               1                 02071500
       01470710         00073063               1                 02072500
       01470750         00073104               1                 02073500
       01471010         00073105               1                 02074500
       01471050         00073123               1                 02075500

CACHE INFORMATION

       01471110         00073124               1                 02076500
       01471150         00073125               1                 02077500
       01471210         00073255               1                 02100500
       01470310         00073325               1                 02062500
```

Chapter 4
Interpreting a Crash Dump Listing

This chapter introduces basic concepts that can help you use CDA output listings to analyze the cause of a system failure. However, this chapter is not intended as a complete guide to interpreting a crash dump.

4.1 Helpful Concepts

Two concepts are helpful in using CDA output listings to determine the cause of a system failure:

- Determining what was mapped at the time of the system failure
- Interpreting stack depth and the kernel stack

4.1.1 Determining What Was Mapped

To determine what was mapped at the time of the failure, look at the dump of the Kernel Page Address Registers on the first page of the CDA listing (instruction and data space registers on RSX-11M-PLUS systems). This listing, titled Volatile Registers, is shown in Example 3-1.

Example 4-1 is an example of the listing of the contents of the sixth and seventh words from a Kernel Page Address Registers dump. The contents of the sixth and seventh words are the block numbers of the task or driver that was mapped at the time of the system failure. You can determine what occupied that portion of memory from the memory map on the first page of partition information in the CDA output listing. Look under the BASE heading (for the base address of the mapped partition).

You can use this information, along with data from the kernel stack and system common listings, to analyze the state of the system at the time of the failure.

Example 4–1: Kernel Page Address Registers

```
                KERNEL

 I  SPACE            D  SPACE
 PDR       PAR       PDR       PAR
077406    000000    177506    000000
077406    000200    077506    000742
077406    000400    077406    001142
077406    000600    077406    001342
077406    001000    077506    001542
077406    010667    077406    010667
077406    011067    077406    011067
077406    177600    077506    177600
```

4.1.2 Interpreting The Kernel Stack

The system stack depth indicator, $STKDP, appears on the first page of the system common listing. The value of the stack depth indicator shows which state the system was in when it failed. The following lists shows the possible values of the stack depth indicator and the corresponding system states:

$STKDP = 1	Indicates User state
$STKDP = 0	Indicates System state
$STKDP = -1,-2,-3,-4, . . .	Indicates Interrupt state

Note that when the system is in Interrupt state, the stack depth value is negative. Two types of interrupt conditions can decrement stack depth:

- A synchronous system trap (SST), which can be caused by any of the following:
  - A directive
  - A TRAP instruction
  - An illegal instruction
  - Another SST
- The interrupt save routine ($INTSV)

When an SST occurs, the following information from the current task is pushed onto the stack:

The Processor Status Word (PSW)
The Program Counter (PC)
The address of return to DIRSV
SST specific information
R5
R4
R3
R2
R1
R0

When an interrupt occurs, the following are pushed onto the stack:

PSW
PC

If a call to $INTSV is then issued, the following are pushed onto the stack after the PC:

Address of return to $INTSV
R5
R4

The stack depth is then decremented. If the value of $STKDP is 0 and the currently mapped driver issues a call to the $FORK routine, the following items are pushed onto the stack:

Return to $FORK
R3 (Which replaces the return to $INTSV)
R2
R1
R0

If the driver issues a call to $FORK when $STKDP is not 0, the registers are saved in a fork block, which is queued for later execution.

Since interrupts can still occur, more of both basic types of stack contents (interrupt or $INTSV) can be pushed. If an SST occurs in the Executive, SST information will be pushed onto the stack, possibly followed by a system failure.

This information, along with the kernel stack pointer SP(K), which appears on the first page of the CDA listing, can be used to interpret the contents of the dump of the kernel stack.

The contents of the PC and the PSW before the failure appear on the first page of the CDA listing. You can compare this information to the contents of the PC and the PSW on the stack to help locate the cause of the failure.

Refer to the *RSX-11M-PLUS and Micro/RSX Guide to Writing an I/O Driver* for a further description of the contents of the kernel stack.

Appendix A
CDA Messages

CDA displays a message on your terminal when it detects one of the error conditions described below. These messages reflect operational conditions. Do not confuse these messages with the diagnostic analysis messages that CDA generates during the analysis and prints in the analysis listing.

Not all of the messages listed below terminate CDA analysis. Some are simply informational messages, and others warn you of nonfatal errors. Also, this list includes some ANALYZE /CRASH_DUMP command error messages.

ANALYZE—Illegal crash input specification,

Severity: Fatal

Explanation: You must include a crash-input device or file in the command line. Also, you must use the correct command line syntax when you specify the crash input.

ANALYZE—Illegal crash output specification,

Severity: Fatal

Explanation: You must specify at least one output file or device specification. Also, you must use the correct command line syntax when you specify the crash output.

ANALYZE—Illegal, contradictory, or ambiguous qualifiers,

Severity: Fatal

Explanation: You must use qualifiers with the correct file or device specification. Refer to the qualifier tables in Chapter 2 if you are unsure which file or device specification a qualifier modifies. Also, when you specify qualifier arguments, be sure to enclose them in parentheses. If you specify more than one argument for the same qualifier, separate the arguments with commas.

ANALYZE—Sorry, task not installed,

Severity: Fatal

Explanation: CDA is not presently installed in the system. If you are a nonprivileged user, refer to Chapter 1 for an explanation of how you can run CDA as an uninstalled task, or ask a privileged user (such as your system manager) to install CDA. If you are a privileged user, you can install CDA and then reenter your command line.

CDA—ACP out of memory or not in execution,

Severity: Analysis diagnostic

Explanation: The partition containing the File Control Block (FCB) for the current logical unit number (LUN) was not in memory.

CDA—Address out of range,

Severity: Fatal

Explanation: CDA was unable to read a block from the crash-input file. Possible causes for this are:

- A device failure
- A bad block on the volume
- The crashed system had a corrupted data base
- The binary file does not contain all of the crashed system's memory

CDA—Analysis output must be directed to an explicit device or file,

Severity: Fatal

Explanation: CDA requires an explicit output file specification. There are no default output file names.

CDA—Analysis terminated after n. pages,

Severity: Informational

Explanation: CDA terminated the analysis after generating n pages of analysis output. If you have not specified the /LIMIT switch in the CDA command string, this message indicates that CDA has generated more than 300_{10} pages of output.

CDA—Cache region is not in memory,

Severity: Warning

Explanation: The requested cache region was not in memory at the time of the system failure.

CDA—Cache region was not found,

Severity: Warning

Explanation: No part of memory has been defined as the requested region.

CDA—Command I/O error,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to read a command line.

CDA—Command line syntax error,

Severity: Fatal

Explanation: CDA detected an error in the syntax of a CDA command line. CDA will point to the beginning of the error within the command line.

CDA—Crash dump must be input from an explicit device or file,

Severity: Fatal

Explanation: The crash dump input file specification must be explicit. There is no default file specification for the crash dump input.

CDA—Device driver missing,

Severity: Fatal

Explanation: You have not loaded the driver for the crash dump input device.

CDA—Dump aborted - kernel PARs clobbered,

Severity: Fatal

Explanation: This message appears on mapped systems only. It indicates that the values contained in the kernel Page Address Registers are invalid. To restart the analysis, you must specify the /KMR switch. This switch forces CDA to use standard mapping values when converting kernel virtual addresses to physical memory addresses.

CDA—Error reading crash dump,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to read the crash dump file. This could be caused by:

- A device failure
- A bad block on the volume
- On RSX–11M–PLUS systems, the device might not be mounted foreign

CDA—Error reading file filename,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to read the crash dump file. This could be caused by:

A device failure

A bad block on the volume

CDA—Error reading symbol file filename,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to read the symbol table file indicated. Possible causes for the error are:

- A device failure
- A bad block on the volume
- The specified symbol file was not an STB file

CDA—Errors detected: n.,

Severity: Informational

Explanation: CDA has detected n analysis errors during the run.

CDA—Error writing analysis file,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to write a line into the analysis listing file. This could be caused by:

- A full volume
- A problem with the device
- A bad block on the volume

CDA—Error writing dump file filename,

Severity: Fatal

Explanation: The system returned an error when CDA attempted to write into the binary output file. This condition could be caused by:

- A full volume
- A problem with the device
- A bad block on the volume

CDA—Exiting due to illegal trap - snapshot dump being attempted,

Severity: Fatal

Explanation: CDA has aborted after detecting an odd address or some other type of fault. If Postmortem Dump (PMD) is installed in the system, the system will generate a snapshot dump. This is an indication of a software problem. If you send a Software Performance Report (SPR) to DIGITAL for this type of failure, you should include any available dumps. Also, preserve the following until your SPR is answered:

From the crashed system:

- All applicable user task images
- RSX11M.SYS
- RSX11M.STB

- RSXMC.MAC
- All applicable privileged task images
- Crash dump volume

- From the system used for analysis:
  - RSX11M.SYS
  - RSX11M.STB
  - CDA.TSK

CDA—Failed to assign LUN to input device ddu,

Severity: Fatal

Explanation: The Assign LUN (ALUN$) directive failed when CDA attempted to use it to attach the specified input device before reading the crash dump from the device. ALUN$ fails if the device name in the CDA command line is invalid.

CDA—Failed to extend page buffer - n. pages available,

Severity: Informational

Explanation: The Extend Task (EXTK$) directive failed when CDA attempted to use it to expand the page buffer. This problem causes the analysis to take longer, but the analysis continues with a buffer of n pages, each 256 words long.

Note

CDA uses the Extend Task directive to obtain additional buffering space. CDA does not use space allocated by a /INC= qualifier on the install command.

CDA—Failed to open input file filename,

Severity: Fatal

Explanation: One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specified directory does not exist.
- The specified file does not exist.
- You do not have read access privileges.

CDA—Failed to open output file filename,

Severity: Fatal

Explanation: One of the following conditions exists:

- The specified device does not exist.
- The volume is not mounted.
- A problem exists with the device.
- The specified directory does not exist.
- The volume is full or the device is write-protected.
- You do not have write access privilege to directory.

CDA—Illegal switch,

Severity: Fatal

Explanation: You specified an unknown switch or used a legal switch with the wrong file specification. CDA will point to the error within the command line.

CDA—Inconsistency in dynamic storage,

Severity: Informational

Explanation: CDA detected an inconsistency while scanning the pool pointers. This condition could be associated with the crash. However, it may mean that you specified the wrong executive symbol table file.

CDA—Indirect command syntax error,

Severity: Fatal

Explanation: The name of the indirect command file (@filename) is syntactically incorrect.

CDA—Indirect file open failure,

Severity: Fatal

Explanation: CDA could not open an indirect command file specified as "@filename" in the CDA command line.

CDA—Invalid address range,

Severity: Fatal

Explanation: You specified an address that was not consistent with Active Page Register (APR) mapping.

CDA—List count expired,

Severity: Analysis diagnostic

Explanation: The linked list of data structures has too many entries. The list may be corrupted or contain a loop.

CDA—Loop found in cache extent descriptors at nnnnn,

Severity: Analysis diagnostic

Explanation: A loop was found in cache extent descriptors at physical address nnnnn.

CDA—Loop found in cache statistics blocks at nnnnn,

Severity: Analysis diagnostic

Explanation: A loop was found in the cache statistics blocks at physical address nnnnn.

CDA—Loop found in linked list in pool at nnnnn,

Severity: Analysis diagnostic

Explanation: A loop was found in a linked list in kernel data space at location nnnnn.

CDA—Maximum indirect file depth exceeded,

Severity: Fatal

Explanation: You exceeded the maximum allowable number of nested indirect command files. (CDA only permits one indirect command level.)

CDA—No devices have caching associated with this region,

Severity: Informational

Explanation: The region specified has no devices specified for caching.

CDA—No input file specified,

Severity: Fatal

Explanation: You did not specify a crash dump input file.

CDA—No output file specified,

Severity: Fatal

Explanation: You did not specify an output file.

CDA—Output dump filename must be explicit,

Severity: Fatal

Explanation: You did not specify a valid output file.

CDA—Pool link error found - continuing,

Severity: Analysis diagnostic

Explanation: CDA detected a link error while scanning the pool free block pointers ($CRAVL). This condition can be associated with the crash. It can also mean that you specified the wrong Executive symbol table file. If the latter is true, the entire analysis will be meaningless and you should abort CDA.

CDA—Premature end of dump input - filename being zero-filled,

Severity: Informational

Explanation: CDA reached the end of the medium (or the end-of-file mark on a magnetic tape) before the crash dump output file had been completely filled. If you expected the file to be completely filled by the dump, this condition could indicate a problem.

CDA—Processor n failed to dump its registers,

Severity: Informational

Explanation: On a multiprocessor system, when the system crashes, each on-line processor is notified by an interrupt. If the processor does not respond to the interrupt (for example, if it halted or is off line), it won't dump its registers into the crash buffer. CDA notes this and prints the informational message.

CDA—Redirect error (U.RED=0),

Severity: Analysis diagnostic

Explanation: CDA detected an error in the pointer to the Unit Control Block (UCB) of a redirected unit. This condition may be associated with the cause of the crash.

CDA—Requested partition is not a cache region,

Severity: Warning

Explanation: The partition you specified is not a part of the cache region.

CDA—Symbol symbolname not defined in symbol file,

Severity: Fatal

Explanation: CDA did not find a symbol that it required for the analysis in the specified Executive symbol table file. You have probably entered the wrong file name or have mistakenly used the default file name.

CDA—Symbol file filename has illegal format,

Severity: Fatal

Explanation: The specified Executive symbol table file has an improper format, probably caused by entry of the wrong file name. However, this message could also indicate a problem with the device or medium on which the file is located.

CDA—Task 'taskname' not in memory,

Severity: Analysis diagnostic

Explanation: This message can be caused by two conditions:

- You have requested a dump of a task which does not have an entry in the System Task Directory.

  The task has an entry in the System Task Directory, but it is marked out of memory.

You can verify the state of the task by examining a dump of the Task Control Blocks.

CDA—Transfer complete - ddu may be unloaded,

Severity: Informational

Explanation: The transfer of the crash dump to the output file is finished; you may unload the crash dump device. This message occurs only when you have specified a binary file in the output of the command string to CDA.

CDA—Unknown AST type,

Severity: Analysis diagnostic

Explanation: CDA has detected an AST block that is not one of the following valid types of ASTs:

- Unsolicited character AST
- Buffered I/O AST
- Emit status AST
- Completion AST from:

  QIO$
  MRKT$
  SPWN$
  CNCT$
  CINT$

- Specified AST from:

  SFPA$
  SRDA$
  SRRA$
  SPFA$

CDA—Unknown get command line error,

Severity: Fatal

Explanation: An unrecognized error occurred when CDA attempted to read a command line.

CDA—$PKVAL link error at n –FWD PTR = n,

Severity: Analysis diagnostic

Explanation: CDA detected a link error while scanning the pool free-packet list $PKVAL. This condition can be associated with the crash. It can also mean that you specified the wrong executive symbol table file.

Appendix B

Definitions of Values in Low Core Memory Dump and System Common Alphabetized Dump

B.1 Low Core Memory Dump

The following is a source-code listing defining the values in the Low Core Memory Dump portion of a crash dump.

```
;---------------------------------------------------------------
;               Lowcore data area
;---------------------------------------------------------------

;
; PROCESSOR IMPURE DATA AREA
;

        .IF DF  M$$PRO

$PROC2::.BYTE   0               ;PROCESSOR NUMBER * 2
$PROCN::.BYTE   0               ;PROCESSOR NUMBER

        .ENDC

$STKDP::.WORD   0               ;STACK DEPTH INDICATOR
$TKTCB::.WORD   0               ;POINTER TO CURRENT TASK TCB
$CURPR::.WORD   $PRTAB          ;POINTER TO CURRENT TASK PRIORITY


$SGFFR:: .WORD  0               ;POINTER INTO STACK FOR $SGFIN

$HFMSK::.WORD   HF.EIS          ;HARDWARE MASK WORD
        .WORD   H2.BRG          ;SECOND HARDWARE MASK WORD
                                ;H2.BRG IS SET BECAUSE THE VIRGIN BOOT
                                ;OF A PROFESSIONAL 300 SERIES MACHINE
                                ;IS CURRENTLY ALWAYS DONE ON A BRIDGE
                                ;SYSTEM AND INITL DOES NOT SET THIS BIT
$HFEND::                        ;REF. LABLE TO DELIMIT HARDWARE FEATURE
                                ;MASK WORDS.

; $RBTAD IS THE ADDRESS OF THE RE-BOOT ENTRY TO THE BOOT ROM
```

```
        .IF DF B$$OOT
$RBTAD::.WORD   B$$OOT          ;ADDRESS OF REBOOT ROM ENTRY
        .ENDC   ; DF B$$OOT

;
; THE FOLLOWING BYTE IS A FLAG SET WHENEVER SUPERVISOR I SPACE IS
; MAPPED READ/WRITE, MEANING THE PDR'S MUST BE CLEARED ON CONTEXT SWITCH
;
$SIRWF::.BYTE   0               ;SUPER I SPACE R/W FLAG
$CXDBL::.BYTE   0               ;CONTEXT SWITCH DISABLE FLAG
$SAHDB::                        ;BIAS OF CURRENT TASK HEADER
        .IF DF X$$HDR
        .WORD   0
$SAVSP::.WORD   $UMR4           ;SAVED SP - ALWAYS POINTS TO UMR4
        .IFTF
$SAHPT::                        ;VIRTUAL ADDRESS OF CURRENT TASK HEADER
        .IFT
        .WORD   0
        .ENDC
$HEADR::.WORD   0               ;POINTER TO CURRENT TASK HEADER
        .IF DF F$$MAP
$FMAPP::.WORD   0               ;POINTER TO FAST MAP AREA OF HEADER
        .ENDC ; DF F$$MAP
$RQSCH::.WORD   $RQTAB          ;POINTER TO CURRENT RESCHEDULE POINTER
;
; THE FOLLOWING CELLS CONDITIONALLY DESCRIBE THE EXECUTIVE CONFIGURATION
; IF KERNEL DATA SPACE IS SUPPORTED.
;
        .IF DF  K$$DAS
$ROEND::.WORD   <<<$EXEND+77>&177700>/100>&1777 ;LENGTH IN 32 WD BLOCKS
                                                ;OF R/O SECTION OF EXECUTIVE
$SCMOF::.WORD   0               ;OFFSET TO DATA SPACE OR 0 IF NOT LOADED
        .WORD   3               ;ALLOCATION SIZE MASK
$ICAVL::.WORD   $ICBEG          ;BEGINNING OF ALLOCATED ICB POOL
        .WORD   0               ;DUMMY SIZE FOR FIRST BLOCK
        .IFF
$SCMOF::                        ;MAKE SURE SYMBOL IS ALWAYS DEFINED
        .ENDC

;
; UMR ALLOCATION LISTHEAD AND WAIT QUEUE LISTHEAD.  THE ALLOCATION
; LISTHEAD DOUBLES AS A DESCRIPTOR OF THE UMRS STATICALLY ALLOCATED TO
; THE EXEC AND ANYONE ELSE.
;
```

```
        .IF DF  M$$EXT&U$$UMR
$UMRHD::.WORD   0               ;MAPPING ASSIGNMENT BLOCK LISTHEAD
        .WORD   UBMPR           ;ADDRESS OF FIRST ASSIGNED UMR
        .WORD   N$$UMR          ;NUMBER OF UMR'S STATICLY ASSIGNED * 4
$UMRWT::.WORD   0,.-2           ;UMR WAIT QUEUE LISTHEAD
        .ENDC   ; DF M$$EXT&U$$UMR

;
; DATA STRUCTURES FOR EXECUTIVE IDLE CODE
;
$IDLCT::.BYTE   0               ;IDLE PATTERN COUNT BYTE
$IDLFL::.BYTE   0               ;IDLE PATTERN FLAG BYTE
$IDLPT::.WORD   7760            ;IDLE PATTERN WORD
        .WORD   0               ;WORKING STORAGE FOR IDLE PATTERN

;
; SYSTEM POWER FAIL STACK
;

        .IF DF  P$$RFL

        .IF DF  M$$PRO
$PWKAO::.BLKW   M$$PRO          ;;;STORAGE FOR KINARO IN CPA'S AREA
$PWCSR::.WORD   FKCSR           ;;;POINTER TO IIST CSR
FKCSR:  .WORD   0               ;;;CPA CSR (WHEN NO IIST PRESENT)
        .ENDC

$PWBTM::.BLKW   7               ;RO THRU R5 AND SP

        .IF DF  M$$EXT
        .BLKW   31.*2           ;UNIBUS MAPPING REGISTERS
        .ENDC

        .BLKW   25.             ;MEMORY MANAGEMENT REGS & USER SP

        .IF DF  F$$LPP
        .BLKW   27.             ;FLOATING POINT REGISTERS
        .ENDC

        .IF DF  K$$DAS
        .BLKW   8.              ;KERNEL D SPACE APR'S
        .ENDC

        .IF DF  S$$LIB
        .BLKW   16.             ;SUPER I SPACE PARS AND PDRS
        .BLKW   16.             ;SUPER D SPACE PARS AND PDRS
        .BLKW   1               ;SUPER SP
        .ENDC
```

```
        .IF DF U$$DAS
        .BLKW   16.             ;USER D SPACE REGISTERS
        .ENDC   ; DF U$$DAS

$PWSTK==.-2                     ;REF LABEL

$POWSP:: .BLKW  1               ;SAVED SP FOR POWERFAIL STACK

        .ENDC

;
; FLOATING POINT SUPPORT WORK AREA
;

        .IF DF  F$$LPP
$FLSTS::.BLKW   2               ;FLOATING POINT STATUS
$FLFRK::.WORD   $FLFTB          ;POINTER FOR FLOATING POINT FORK BLOCK
$FLTCB::.BLKW   1               ;ADDRESS OF TCB CAUSING EXCEPTION TRAP
        .ENDC

;
; SAVE AREA FOR ERROR LOGGING NONSENSE INTERRUPT HANDLING
;

        .IF DF  E$$LOG&E$$NSI

$VID::  .BLKB   1               ;VECTOR ID STORAGE
$NSI::  .BYTE   -1              ;RECURSION COUNTER
$OPS::  .BLKW   1               ;OLD PS STORAGE
$OPC::  .BLKW   1               ;OLD PC STORAGE
$ERLFK::.WORD   $PBEZ           ;POINTER TO ERROR LOGGING FORK BLOCK

        .ENDC   ; DF E$$LOG&E$$NSI

;
; DATA STRUCTURES FOR PARITY SUPPORT
;
; EXEC PARITY ERROR MESSAGE
;

        .IF DF P$$RTY
        .NLIST  BEX
$EXMSG::.ASCIZ  <15><12>/***EXEC PARITY ERROR STOP***/<15><12><12> ;
        .LIST   BEX
        .EVEN

;
; INTERRUPT RECURSION LEVEL COUNTER
;

$PARLV::.WORD   -1              ;

;
; PARITY CONTROL STATUS REGISTER ADDRESS TABLE
;
```

```
;       ******* NOTE WELL ! *******
;
; THE FOLLOWING TABLES ARE REFERENCED IN INITL AND PARTY, AS WELL AS
; BY SAVE AND HRC.  THE ADJACENCY OF THE ITEMS STARTING AT $PARTB AND
; INCLUDING $MEMR1, $MPCSR, AND $CPUER IS NOT ONLY ASSUMED, IT IS
; CRUCIAL AND MUST BE PRESERVED.  NOTE ALSO THAT ANY CHANGE IN THE
; STRUCTURE OF $PARTB SHOULD PROBABLY BE REFLECTED IN $MKCSR AND $MKCS2,
; BOTH FOUND IN SYSCM.

$CSRTB::                        ;REFERENCE LABEL
$MEMRG::.BYTE   16.,<172100&177400>/400,172100&377 ;MEMORY PARITY CSR'S
        .BYTE   6. ,<177740&177400>/400,177740&377 ;CACHE CONTROL CSR'S
        .BYTE   1. ,<177766&177400>/400,177766&377 ;CPU ERROR REGISTER
        .BYTE   0               ;END OF TABLE

        .BLKW   1               ;DUMMY STORAGE FOR MEMORY ERROR REGISTER

        .IFTF   ; DF P$$RTY

$PARTB::.WORD   1               ;DUMMY CSR FOR NONEXISTANT REGISTERS

        .IFT    ; DF P$$RTY

;
; TABLE OF CSR ADDRESSES. IF THE CSR DOES NOT EXIST, THE ENTRY POINTS
; TO $PARTB ABOVE. THIS MUST BE THE CASE SO THAT CODE REFERENCING THE
; CSRS PRIOR TO THE CSR SCAN (SUCH AS THE LOCK$ $EXECL IN INITL) WILL
; ACCESS A DUMMY LOCATION, NOT LOCATION 0.
;

$MEMR1::                        ;MEMORY PARITY CSR TABLE
        .REPT   16.
        .WORD   $PARTB          ;POINT TO DUMMY CSR
        .ENDR
$MPCSR::                        ;VECTOR OF CACHE CSR ADDRESSES
        .REPT   6.
        .WORD   $PARTB          ;POINT TO DUMMY CSR
        .ENDR

        .IFTF   ; DF P$$RTY

$CPUER::                        ;CPU ERROR REGISTER
        .WORD   $PARTB          ;POINT TO DUMMY CSR

        .IFT    ; DF P$$RTY


;
; PDP-11/70 CACHE PARITY STATUS TABLE
;

        .IF DF  P$$D70

$MSTAT::.BLKW   2               ;FIRST TWO PARITY CSR'S
$STAT:: .BLKW   1               ;MEMORY STATUS REGISTER
        .BLKW   3               ;LAST THREE PARITY CSR'S
$ERTRK::.BYTE   44,120          ;ADDRESS/DATA GROUP 0
        .WORD   0               ;TIME OF LAST ERROR
        .BYTE   30,240          ;ADDRESS/DATA GROUP 1
        .WORD   0               ;TIME OF LAST ERROR
        .WORD   0               ;END OF TABLE
```

```
;
; NEW CACHE PARITY CSR CONTENTS
;
$MPCTL::.WORD   1               ;

        .ENDC

        .IFF    ; DF P$$RTY

$CSRTB::                        ;REFERENCE LABEL

        .ENDC   ; DF P$$RTY

; DATA AREA FOR DYNAMIC EXEC DEBUGGER INTERFACE

        .IF DF  D$$DXD

$DXDK5::.WORD   0               ;SAVED KINAR5 FOR DXD
$DXDRL::.WORD   0               ;RELOCATION BIAS FOR DXD
$DXDEP::.WORD   0               ;ENTRY POINT TO DXD

        .ENDC   ; D$$DXD

;
; DATA AREA FOR BUGCHECK SUPPORT
;
$BCFAC::.WORD   0               ;FACILITY CODE
$BCERR::.WORD   0               ;ERROR CODE

;
; DATA AREA FOR SYSTEM CRASH MODULE
;

        .IF DF C$$CDA

$CRBAE::.WORD   0               ;DEFINE EXISTENCE OF BAE REGISTER

        .IIF EQ C$$CDA-12 CDA$MS=0      ;MS RESIDENT CRASH DRIVER
        .IIF EQ C$$CDA    CDA$MS=0      ;LOADABLE CRASH DRIVERS

        .IF DF CDA$MS                   ;MS RESIDENT OR LOADABLE CRASH DRIVER

$MSCR1::                        ;COMMAND BUFFER FOR MS CRASH DUMP
        .BLKW   5               ;NEED FOUR WORDS ON DOUBLE WORD BOUNDARY
$MSCR2::                        ;CHARACTERISTICS DATA
        .WORD   $MSCR3          ;LOW ORDER 16 BITS OF MESSAGE
                                ;BUFFER ADDRESS
        .WORD   0               ;HIGH ORDER 2 BITS
        .WORD   14.             ;SIZE OF THE MESSAGE BUFFER
        .WORD   0               ;FLAG WORD

$MSCR3::.BLKW   7               ;MESSAGE BUFFER

        .ENDC   ;DF CDA$MS

        .IIF EQ C$$CDA-16       CDA$DU=0
        .IIF EQ C$$CDA-17       CDA$MU=0

        .IF DF  CDA$DU!CDA$MU!H$$GEN
```

```
;
; CRASH DATA BASE FOR DU DEVICES
;
; *** NOTE WELL ***
; ANY CHANGES TO THIS DATABASE MUST ALSO BE MADE IN CRASH.MAC

; COMMAND AND RESPONSE PACKETS

        .WORD   60              ;PACKET LENGTH
        .WORD   1               ;VIRTUAL CIRCUIT AND CREDIT/DEBIT FIELD
CMDPKT: .WORD   1               ;COMMAND PACKET AREA
        .BLKW   23.             ;
        .BLKW   2               ;ENVELOPE
RSPPKT: .BLKW   24.             ;RESPONSE PACKET AREA
        .BLKW   2               ;

; COMMAND RINGS

RINGS:
RSP:    .WORD   RSPPKT          ;RESPONSE PACKET RING
        .WORD   100000          ; PACKET "OWNED" BY CONTROLLER
CMD:    .WORD   CMDPKT          ;COMMAND PACKET RING
        .WORD   100000          ; PACKET "OWNED" BY CONTROLLER

; INITIALIZATION TABLE

$MUCSH::
$DUCSH::.WORD   CMDPKT, RSPPKT  ;ADDRESSES OF PACKETS
        .WORD   100000, RINGS   ;STEPS 1 AND 2
        .WORD   0, 1            ;STEPS 3 AND 4
        .WORD   RSP             ;ADDRESS OF RESPONSE RING

        .ENDC

        .IFF    ; DF C$$CDA

$BCPC:: .WORD   0               ;BUGCHECK PC FOR TYPE DIRECT

        .IFT    ; DF C$$CDA

$BCPC::                         ;ALTERNATE LABEL FOR CRASH PC

$CRUPC::.WORD   0               ;USER PC IS STORED HERE
$CRUST::.WORD   0               ;USER PS IS STORED HERE

        .NLIST  BEX
$CRMSO::.ASCII  <15><12><12>/CRASH -- CONT WITH SCRATCH MEDIA ON /

$CRDEV::                ;REFERENCE SYMBOL
```

```
.IIF EQ C$$CDA-1,       .ASCII  /DT/
.IIF EQ C$$CDA-2,       .ASCII  /DK/
.IIF EQ C$$CDA-3,       .ASCII  /MT/
.IIF EQ C$$CDA-4,       .ASCII  /MM/
.IIF EQ C$$CDA-5,       .ASCII  /DB/
.IIF EQ C$$CDA-6,       .ASCII  /DM/
.IIF EQ C$$CDA-7,       .ASCII  /DL/
.IIF EQ C$$CDA-10,      .ASCII  /DD/
.IIF EQ C$$CDA-11,      .ASCII  /DR/
.IIF EQ C$$CDA-12,      .ASCII  /MS/
.IIF EQ C$$CDA-13,      .ASCII  /DX/    ;NOT SUPPORTED
.IIF EQ C$$CDA-14,      .ASCII  /DY/    ;NOT SUPPORTED
.IIF EQ C$$CDA-15,      .ASCII  /MF/    ;NOT SUPPORTED
.IIF EQ C$$CDA-16,      .ASCII  /DU/
.IIF EQ C$$CDA-17       .ASCII  /MU/

.IIF EQ <.-$CRDEV>      .ASCII  /XX/    ;ALWAYS ALLOW FOR A DEVICE NAME

$CRUNT::.BYTE   <60+C$$RUN>
        .ASCIZ  <15><12><12>

        .IF DF  M$$PRO

$CRMS1::.ASCIZ  <15><12>/CRASH -- WAITING FOR PROCESSORS TO QUIESCE/
$CRMS2::.ASCIZ  <15><12>/CRASH -- PROCESSOR REGISTERS HAVE BEEN SAVED/
$CRMS4::.ASCIZ  <15><12>/CRASH -- SANITY TIMER EXPIRED ON PROCESSOR CP/
$CRMS5::.ASCIZ  <15><12>/CRASH -- CRASH REQUESTED BY PROCESSOR CP/
$CRCPU::.ASCIZ  /?/

        .ENDC

$CRMS3::.ASCIZ  <15><12>\CRASH -- I/O ERROR ON CRASH DUMP DEVICE\
$CRMS6::.ASCIZ  <15><12>/CRASH -- ILLEGAL CRASH DEVICE/
$CRMS7::.ASCIZ  <15><12>/CRASH -- CRASH DRIVER NOT LOADED/

; ALL THE SECTIONS OF THE BUGCHECK MESSAGE MUST BE CONTIGUOUS

$BCMSG::.ASCIZ  <15><12>/SYSTEM FAULT DETECTED AT PC=/
        .ASCIZ  /  FACILITY=/
        .ASCIZ  /  ERROR CODE=/
        .ASCIZ  <15><12><12>

        .BLKB   6               ;BUFFER FOR OCTAL CONVERSION
$BCBUF::.BYTE   0               ;OF VALUES IN CRASH

        .LIST   BEX

        .ENDC   ; DF C$$CDA

        .IF DF  A$$CNT

$BILNG::.BYTE   1               ;BILLING IS INITIALLY OFF

        .IFF

$BILNG::

        .ENDC


        .IF DF  S$$LIB

$SUPFL::.BYTE   0               ;SUPERVISOR WINDOW FLAG
```

```
        .ENDC   ; DF S$$LIB

        .EVEN

                                ; *** THE FOLLOWING MUST BE ADJACENT

        .IF DF C$$CDA

        .BLKW   6               ;STACK AREA FOR SUBROUTINE CALLS
$CRSBF::.BLKW   177.            ;CRASH STACK IS THIS SIZE (NOT
                                ;169.+4 AS PREVIOUSLY THOUGHT)
                                ;(SEE CRASH.MAC FOR FORMAT)
$CRSST==.-2                     ;TOP OF CRASH STACK
                                ; *** ABOVE MUST BE ADJACENT
        .IF EQ, C$$CDA          ;LOADABLE CRASH DRIVER SUPPORT

        PBNH = 0                ;STARTING DEVICE ADDRESS
        PBNL = 0                ;UNKNOWN

        .IFTF

$CRSBN::.WORD   PBNH,PBNL       ;STARTING DEVICE ADDRESS
$CRSCS::.WORD   PBNH+PBNL       ;CHECKSUM OF DEVICE ADDRESS
$CRDMP::.BYTE   0               ;FLAG INDICATING REGS HAVE BEEN SAVED
        .BYTE   0               ;PRESERVE WORD ALIGNMENT

$CRPAR::

        .IFT    ; EQ, C$$CDA

        .WORD   0               ;P.REL OF LOADED CRASH DRIVER

        .ENDC   ; EQ, C$$CDA

        .ENDC   ; DF C$$CDA


        .IF DF  A$$CNT

$STRTM::.BLKW   2               ;ABSOLUTE TIME OF CONTEXT SWITCH TO TASK

        .IFF

$STRTM::

        .ENDC


; HERE WE DEFINE A SINGLE "FAKE WORD" WHICH RECEIVES ALL THE ADDRESSES
; FOR THINGS WHICH ARE NOT GENNED INTO THE SYSTEM, BUT WHICH REQUIRE
; THAT THE ADDRESSES ALWAYS BE RESOLVED FOR TASKBUILDING MCR, ETC.
; THIS ONE (IN LOWCR) IS TO BE USED FOR I SPACE REFERENCES AND CPU
; SPECIFIC INFO (ALSO THINGS WHICH MUST BE ACCESSED WHEN D SPACE IS
; TURNED OFF), AND THE ONE IN SYSCM IS USED FOR D SPACE LOCATIONS

        .IF NDF R$$WPT

$WPBR::

        FAKE = 0

        .ENDC   ; NDF R$$WPT

        .IF NDF P$$RTY

$PARLV::
```

```
        FAKE = 0
        .ENDC   ; NDF P$$RTY

        .IF NDF D$$DXD
$DXDEP::
$DXDK5::
$DXDRL::
        FAKE = 0
        .ENDC   ; NDF D$$DXD
        .IF DF FAKE
        .WORD   -1              ;FAKE DATA LOCATION FOR ALL OF THE
                                ;ABOVE
        .ENDC   ; DF FAKE

;+
; **-$NS0,$NS1,$NS2,$NS3,$NS4,$NS5,$NS6,$NS7-
;       NONSENSE INTERRUPT IDENTIFIER ROUTINES
;
; EACH OF THESE ROUTINES IS VECTORED TO BY ONE OF A GROUP OF 16
; UNUSED VECTORS.  THE VECTORS ARE SUB-CODED IN THE PS CONDITION CODES.
;-

        .IF DF  M$$PRO

;
; IN A MULTIPROCESSING SYSTEM, SOME STACK SPACE IS NECESSARY WHEN
; THE CPU IS INTERRUPTED.  THE M9312MP BOOTSTRAP SETS R6 TO THE
; VALUE POINTED TO BY PHYSICAL ZERO.  THIS IS THAT AREA.
;
        .BLKW   20.             ;STACK SPACE FOR IIST
        .ENDC

$NS0::  NSI                     ;;;CALL COMMON ROUTINE
        .WORD   0               ;;;GROUP 0-17
$NS1::  NSI                     ;;;
        .WORD   20              ;;;GROUP 20-37
$NS2::  NSI                     ;;;
        .WORD   40              ;;;GROUP 40-57
$NS3::  NSI                     ;;;
        .WORD   60              ;;;GROUP 60-77
$NS4::  NSI                     ;;;
        .WORD   100             ;;;GROUP 100-117
$NS5::  NSI                     ;;;
        .WORD   120             ;;;GROUP 120-137
$NS6::  NSI                     ;;;
        .WORD   140             ;;;GROUP 140-157
$NS7::  NSI                     ;;;
        .WORD   160             ;;;GROUP 160-177

;
; INDIRECT POINTERS INTO MULTIPROCESSOR TABLES
;
```

```
        .IF DF  M$$PRO

$TKPTR::.WORD   $TKTAB          ;POINTER INTO CURRECT TASK TABLE
$MPSWT::.WORD   $MPTAB          ;POINTER INTO CPU STATUS TABLE
$CPURM::.WORD   $URMTB          ;POINTER INTO CPU URM TABLE
$CPCRM::.WORD   $CRMTB          ;POINTER INTO COMPLEMENTED RUN MASK TBL
$TTUQP::.WORD   $TTUQ           ;TTDRV MULTIPROCESSOR POINTER

;
; MULTIPROCESSOR LOCAL MEMORY WORDS.
;

$CPBIT::.WORD   1               ;PROCESSOR BIT FOR USE IN MASKS
$FRKCT::.WORD   0               ;NUMBER OF FORK BLOCKS DEQUEUED

        .ENDC
;
; KXJ11 SPECIFIC LOCATIONS
;
        .IF DF C$$KXJ           ; KXJ SUPPORTED?

$KXVC1::.BLKW   1               ; KXJ VECTOR 1 - DRIVER COMMUNICATION
$CSFSV::.BLKW   1               ; SAVED KXJ CSR F - FULLY SHARED MEMORY
$CSHSV::.BLKW   1               ; SAVED KXJ CSR H -   "     "      "
$KXPTR::.BLKW   1               ; POINTER TO KXJ DRIVER'S DCB
$KXBAS::.BLKW   1               ; BASE ADDRESS OF KXDRV

        .IFF

$KXVC1::
$CSFSV::
$CSHSV::
$KXPTR::
$KXBAS::

        .ENDC  ; C$$KXJ
;
; REMOTE SYSTEM SPECIFIC LOCATIONS
;
        .IF DF C$$RMT           ; REMOTE HOST

$XXLOW::.BLKW   1               ; FIRST GENERIC DCB ADDRESS
$XXHGH::.BLKW   1               ; LAST GENERIC DCB ADDRESS
                                ; USED TO IDENTIFY A GENERIC  DEVICE

        .IFF

$XXLOW::.BLKW   1
$XXHGH::.BLKW   1

        .ENDC   ; C$$RMT

        .END
```

B.2 System Common Alphabetized Dump

The following is a source-code listing defining the values in the System Common Alphabetized Dump portion of a crash dump.

```
;---------------------------------------------------------------
;       System common data area
;---------------------------------------------------------------
;
; NULL TASK CONTROL BLOCK
;
; THIS TCB TERMINATES THE SYSTEM AND ACTIVE TASK LISTS.  IT MUST HAVE
; A PRIORITY OF ZERO AND ALWAYS BE BLOCKED.  REQUIRED POSITIONS ARE
; ENFORCED BY THE "ASSUME" MACRO.
;
$CMBEG::                        ;BEGINNING OF SYSCM AREA FOR CDA
;
; $ACCLK MUST REFERENCE THE SAME TWO WORDS AS $ABTIM FOR ACCOUNTING TO
; WORK, UNLESS ACCOUTING HAS ITS OWN CLOCK.  IN THIS CASE, $ACCLK WILL
; BE THE ADDRESS OF WHERE ACCOUNTING CAN ACCESS IT.  ACCOUNTING WILL
; USE A TWO WORD $ABTIM IF A PRIVATE CLOCK IS NOT AVAILABLE
;
$ACCLK::                        ;ABSOLUTE TIME CLOCK FOR ACCOUTING
$ABTM2::.WORD   0               ;ADDITIONAL WORD FOR $ABTIM
$ABTIM::.WORD   0               ;ABSOLUTE TIME COUNTER
$ACTHD::.WORD   .               ;T.LNK-ACTIVE TASK LIST LISTHEAD
        ASSUME  .-$ACTHD,T.PRI  ;T.PRI MUST BE ZERO
        .BYTE   0               ;T.PRI-NULL TASK PRIORITY IS ZERO

        .IF DF  M$$PRO

$NCPU:: .BYTE   M$$PRO          ;T.IOC - NUMBER OF PROCESSORS IN SYSTEM

        .IFF

$NCPU:: .BYTE   0               ;T.IOC - NOT MULTIPROCESSOR SYSTEM

        .ENDC

$COMEF::.WORD   0               ;T.DPCB-COMMON EVENT FLAGS 1-16
        .WORD   0               ;T.NAM-COMMON EVENT FLAGS 17-32
$HRCPT::.WORD   0               ;T.NAM+2-POINTER TO HRC... TCB
$DEVHD::.WORD   $DEVTB          ;T.RCVL-PTR TO FIRST DEVICE CONTROL BLOCK
$TKNPT::.WORD   0               ;T.RCVL+2-POINTER TO TKTN TCB
$SHFPT::.WORD   0               ;T.ASTL-POINTER TO SHUFFLER TCB
$CKCNT::.WORD   K$$CNT          ;T.ASTL+2-ADDRESS OF CLOCK COUNT REG
$CKCSR::.WORD   K$$CSR          ;T.EFLG-ADDR OF CLOCK CNTRL STATUS REG

        .IF EQ  K$$CSR-177546

$CKLDC::.WORD   0               ;T.EFLG+2-CLOCK LOAD COUNT

        .IFF

$CKLDC::.WORD   K$$LDC          ;T.EFLG+2-CLOCK LOAD COUNT

        .ENDC
```

```
$SYUIC::.BYTE   54,1            ;T.UCB-DEFAULT MAPPED SYSTEM UIC
        ASSUME  .-$ACTHD,T.TCBL ;T.TCBL MUST BE ZERO
        .WORD   0               ;T.TCBL-LINK TO NEXT TCB
        ASSUME  .-$ACTHD,T.STAT ;T.STAT MUST BE NONZERO
$EXSIZ::.WORD   $SYTOP          ;T.STAT-ADDR OF LAST BYTE IN EXEC
$PWRFL::.WORD   0               ;T.ST2-POWERFAIL RECOVERY REQUEST FLAG
$SIGFL::.WORD   0               ;T.ST3-TASK WAITING FOR SIG EVENT
        .WORD   0               ;T.DPRI MUST BE 0, T.LBN - UNUSED
$FXRPT::.WORD   0               ;T.LBN+1-POINTER TO PARITY ERROR TASK
$LSTLK::.WORD   0               ;T.LDV-LOCK WORD (TCB ADDRESS OF OWNER)
        .WORD   3               ;T.PCB-CONSTANT FOR ALLOCATION ROUTINES

        .IF NDF K$$DAS

$ICAVL::                        ;ICB POOL SAME AS CORE POOL IF NO D SPACE

        .ENDC

$CRAVL::.WORD   $SYBEG          ;T.MXSZ-ACTIVE TASK LIST LISTHEAD
        ASSUME  .-$ACTHD,T.ACTL ;T.ACTL MUST BE ZERO

        .IF NDF M$$PRO

$K6TAB::                        ;REFERENCE LABEL FOR MAPPING IN
                                ;NON-MULTIPROCESSOR SYSTEM

        .ENDC

        .WORD   0               ;T.ACTL-NEXT ACTIVE TASK-DUMMY BLK SIZE

;
; END OF SUPERIMPOSED AREA-REMAINING SYSTEM COMMON AREA
;

$ACPTK::

        .IF DF  R$$AMD

        .WORD   0               ;TCB POINTER FOR DEFAULT F11ACP

        .ENDC ; DF R$$AMD

$VERTK::.WORD   0               ;TCB POINTER FOR VERIFICATION TASK

$LOGHD::.WORD   0               ;LOGICAL DEVICE ASSIGNMENT LISTHEAD
$CFLPT::.WORD   0               ;POINTER TO FIRST CHECKPOINT FILE PCB
$MOULS::.WORD   0               ;MOUNT LIST
$LBUIC::.BYTE   54,3            ;LIBRARY UIC
$NTUIC::.BYTE   54,1            ;NETWORK UIC

$SPMAX::.WORD   0               ;POINTER TO AUXILARY MONITOR POOL NODE
$SPMVC::.BLKW   1               ;POINTER TO SPM NODE IN ICB POOL
$HKSTS::.BLKW   1               ;SPM MONITOR STATUS WORD

; SPM ENTRY VECTORS - EXTERNAL TASKS (LOADER,F11ACP) CALL THROUGH THESE
;                     HOOKS. THEY POINT TO THE RETURN IN $GTPKT IF
;                     SPM IS TURNED OFF, AND SPM INTERCEPTS THEM WHEN IT
;                     IS ENABLED.
;
        .IF DF K$$DAS
```

```
$SPV01::.WORD   0               ;SPM VECTOR FOR ACP HOOKPOINT ENTRY
$SPV02::.WORD   0               ;SPM VECTOR FOR LOADR HOOKPOINT ENTRY
$SPV03::.WORD   0               ;SPM VECTOR FOR LOADR HOOKPOINT ENTRY

        .IFF ; DF K$$DAS

$SPV01::.WORD   $SPVEX          ;SPM VECTOR FOR ACP HOOKPOINT ENTRY
$SPV02::.WORD   $SPVEX          ;SPM VECTOR FOR LOADR HOOKPOINT ENTRY
$SPV03::.WORD   $SPVEX          ;SPM VECTOR FOR LOADR HOOKPOINT ENTRY

        .ENDC ; DF K$$DAS

;
; $BTTIM HOLDS THE ABSOLUTE TIME OF WHEN THE SYSTEM WAS BOOTED.
; THE TWO WORDS MUST BE CONSECUTIVE, THE LABEL IS ON THE SECOND ONE
; FOR COMPATABILITY WITH $ABTIM
;
$BTTM2::.WORD   0               ;HIGH ORDER WORD OF $BTTIM
$BTTIM::.WORD   0               ;VALUE OF $ABTIM WHEN SYS BOOTED

        .IF DF  D$$MND

$DICSR::.WORD   D$$MND          ;ADDRESS OF DIAMOND REGISTER
$DITCB::.WORD   0               ;ADDRESS OF TCB FOR TASK BEING WATCHED

        .ENDC

        .IF DF  M$$PRO

$CKURM::.WORD   1               ;URM OF PROCESSOR THAT KEEPS CLOCK

        .ENDC

        .BLKW   1               ;CLOCK INTERRUPT FORK BLOCK LINK
        .BLKW   1               ;CLOCK INTERRUPT FORK BLOCK PC
$INTCT::.WORD   -1              ;CLOCK INTERRUPT TICKS COUNT
$FRKHD::.WORD   0               ;FORK QUEUE LISTHEAD
        .WORD   $FRKHD          ;(LAST POINTS TO FIRST INITIALLY)
$FMASK::.WORD   FMASK           ;SYSTEM FEATURE MASK
$FMSK2::        .WORD   F2MASK  ;SECOND WORD OF SYSTEM FEATURE MASK
$FMSK3::        .WORD   F3MASK  ;THIRD WORD OF SYSTEM FEATURE MASK
$FMSK4::        .WORD   F4MASK  ;FOURTH WORD OF SYSTEM FEATURE MASK
$FMSK5::        .WORD   F5MASK  ;FIFTH WORD OF SYSTEM FEATURE MASK
$FMEND::                        ;REF. LABLE DELIMITING FEATURE MSK WDS
        .IF DF C$$RMT                                                   ; KLN011
                                                                        ; KLN011
$FMSKR::.WORD   FMASKR          ; THIS WORD BECOMES $FMSK5 IF HF.RMT IS SET     ; KLN011
                                                                        ; KLN011
        .IFF ; C$$RMT                                                   ; KLN011
                                                                        ; KLN011
$FMSKR::                                                                ; KLN011
                                                                        ; KLN011
        .ENDC ; C$$RMT                                                  ; KLN011
$PRMOD::.WORD   0               ;PROCESSOR MODEL NUMBER
$CLKHD::.WORD   0               ;CLOCK QUEUE
$COPT:: .WORD   .C00            ;POINTER TO COMMAND OUTPUT UCB
$PARHD::.WORD   0               ;POINTER TO PARTITION LIST
$LDRPT::.WORD   0               ;POINTER TO LOADER TCB
$TSKHD::.WORD   $ACTHD          ;POINTER TO SYSTEM TASK DIRECTORY
$CBDHD::.WORD   0               ;COMMON BLOCK DIRECTORY LISTHEAD
```

```
        .IF DF G$$GEF
$GGEF:: .WORD   0               ;GROUP GLOBAL EVENT FLAGS
$GFTCB::.WORD   $GEFPT+2        ;GRP GLOBAL USER TCB POINTER
$GEFPT::.WORD   $GEFPT+2        ;GROUP GLOBAL MASK ADDRESS POINTER
        .WORD   0               ;DUMMY WORD FOR NON-GRP GLOBAL USE

        .ENDC   ; DF G$$GEF

$LDPCB::.WORD   0               ;CURRENT LOADER PCB POINTER
$VECTR::.WORD   V$$CTR-4        ;HIGHEST VECTOR ADDRESS
$MXEXT::.WORD   177777          ;INITIALLY NO LIMIT FOR TASK EXTENSION
$AVRHD::.WORD   0               ;AUTOMATIC VOLUME RECOGNITION LISTHEAD
$PRIFR::.WORD   0               ;CURRENT AMOUNT OF FREE POOL
$ULDPT::.WORD   0               ;MICROCODE LOADER TASK TCB ADDRESS
$GNLST::.WORD   0               ;GENERAL USE POOL PACKET LISTHEAD
$PTCBL::.WORD   0               ;PROTOTYPE TCB LISTHEAD
$PTCPT::.WORD   0               ;KISAR5 BIAS OF PROTOTYPE TCB
$PASTH::.WORD   0,.-2           ;PARTITION AST LISTHEAD
;
; P/OS $NXTSK LOCK AND BASE FROM WHICH TO START SEARCH
;
$NXTLK::.WORD   0               ;$NXTSK RECURSION LOCK
$NXTBA::.WORD   0               ;$NXTSK BASE SEARCH BIAS

;
; LOGICAL NAME TRANSLATION SUPPORT DATA
;
        .IF DF  L$$GCL

$LOGTB::.WORD   0               ;APR BIAS PTR TO SYSTEM LOGICAL HASH TABLE
        .WORD   0               ;APR BIAS PTR TO GROUP LOGICAL HASH TABLE
$USRLG::.WORD   0               ;APR BIAS PTR TO USER LOGICAL HASH TABLE
        .WORD   0               ;APR BIAS PTR TO TASK LOGICAL HASH TABLE
        .WORD   0               ;RESERVED WORD FOR P/OS LT.USR (ALWAYS 0)

        .IFF    ; DF L$$GCL
$LOGTB::
$USRLG::

        .ENDC   ; DF L$$GCL

;
; NAMED DIRECTORY CONTEXT BLOCK POINTER
;
$CTXPT::

        .IF DF  N$$DIR

        .WORD   0

        .ENDC   ;DF N$$DIR

;
; THE FOLLOWING WORDS ARE USED FOR THE POOL MONITOR INTERFACE.
; THE ORDERING OF THESE WORDS CANNOT BE CHANGED BECAUSE PMT AND VMR
; ARE WRITTEN TO DEPEND ON IT.
;
        .IF DF  P$$CTL
```

```
$PTTCB::.WORD   0               ;TCB ADDRESS OF POOL RECOVERY TASK
$PRISZ::.WORD   0               ;TOTAL SIZE OF FREE POOL
$POLST::.WORD   0               ;EXEC/POOL TASK COMMUNICATIONS WORD
$PRIHL::.WORD   P$$HIL          ;HIGH WATER MARK FOR POOL MONITORING
$PRILL::.WORD   P$$LOL          ;LOW WATER MARK FOR POOL MONITORING
$PFRSZ::.WORD   P$$FRS          ;MINIMUM SIZE OF LARGEST FRAGMENT
$POLBP::.WORD   P$$BPR          ;MINIMUM  PRIORITY OF NONPRIVILEGED
                                ;TASKS TO EXECUTE DURING LOW POOL
$POLFL::.WORD   0               ;EXECUTIVE POOL USAGE CONTROL FLAGS

        .IFF

$PTTCB::                        ;ALWAYS DEFINE THESE LABELS
$PRISZ::
$POLST::
$PRIHL::
$PRILL::
$PFRSZ::
$POLBP::
$POLFL::

        .ENDC

;
; SHADOW RECORDING IMPURE DATA AREA
;

$SHUMB::.WORD   0               ;ROOT FOR UMB LIST
$SHERR::.WORD   0               ;POINTS TO TCB OF SHADOW ERROR TASK
$SHLIM::.WORD   10.             ;ERROR PACKET LIMIT
$SHPCT::.WORD   0               ;CURRENT SHADOW ERROR COUNT
$SHLOS::.WORD   0               ;NUMBER OF PACKETS LOST FROM SATURATION

$RCTPT::.WORD   0               ;POINTER TO CURRENT RCT... TCB

$EXCRC::.WORD   0               ;EXECUTIVE RO CODE CRC16

;                                                                       ; BM387
;       WINDOW I/O COUNT FLAG. THE FOLLOWING WORD IS USED BY DREIF,     ; BM387
;       IOFIN, AND DEACC IN THE ACP TO DETERMINE WHETHER TO MANAGE      ; BM387
;       THE WINDOW I/O COUNT. IF SET TO DV.F11 THE BEHAVIOR IS TO       ; BM387
;       MANAGE THE I/O COUNT (NEW). IF SET TO ZERO THE I/O COUNT IS     ; BM387
;       LEFT ALONE AND THE WINDOW MAY BE DEALLOCATED WITH I/O           ; BM387
;       PENDING. THIS MAY CAUSE CRASHES LATER.                          ; BM387
;                                                                       ; BM387
$WCFLG::.WORD   0 ; DV.F11      ; WINDOW I/O COUNT MANAGEMENT FLAG      ; BM387
                                                                        ; BM387
        .IF DF  D$$CHE          ;DISK DATA CACHING SUPPORT

;
; DATA CACHING IMPURE DATA AREA
;

$DCPCB::.WORD   0               ;PCB ADDRESS OF SYSTEM DEFAULT CACHE PAR

;
;       THESE CELLS ARE FILLED IN BY $MPVBN (MDSUB) FOR
;       VIRTUAL TO LOGICAL I/O TRANSLATIONS.  THIS INFORMATION
;       IS USED BY THE CACHER WHEN PROCESSING THE RESULTING
;       I/O PACKET.
;
```

```
$DCCEL::.WORD   0               ;CURRENT EXTENT LENGTH
$DCCEB::.WORD   0,0             ;CURRENT EXTENT BASE LBN
$DCNEL::.WORD   0               ;NEXT EXTENT LENGTH
$DCNEB::.WORD   0,0             ;NEXT EXTENT BASE LBN
$DCSTS::.WORD   0               ;STATUS INFORMATION FROM WINDOW BLOCK          ;JRK332
        .ENDC                   ;D$$CHE

;
; POINTER TO FIRST IP11 UCB, USED BY IP11 POWERFAIL CODE (IN POWER)
;
$IPUCB::.WORD   0               ; POINTER TO FIRST IP11 UCB

;
; PARAMETER AREA FOR STATUS CHANGE ROUTINES IN OLRSR.MAC
;
$SCDEV::.BLKW   1               ;UCB OR KRB FOR STATUS CHANGE
$SCRET::                        ;REFERENCE LABEL (ALIAS FOR $SCCTB)
$SCCTB::.BLKW   1               ;CTB IF $SCDEV CONTAINS KRB
$SCOFL::.BLKB   1               ;ONLINE OR OFFLINE PARAMETER
$SCERR::.BLKB   1               ;ERROR RETURN FROM DRIVER

;
; EVENT TRACE READ/WRITE DATA AREA
;
        .IF DF  E$$VNT

;
; CURRENT BUFFER INFORMATION
;
$EVKS6::.WORD   0               ;KISAR6 OFFSET TO BUFFER
$EVDIS::.WORD   0               ;BUFFER POSITION FOR NEXT EVENT
$EVLEN::.WORD   0               ;PTR TO WORD BEYOND END OF BUFFER
$EVSEQ::.WORD   0               ;EVENT SEQUENCE NUMBER
$EVLOS::.WORD   0               ;NUMBER OF EVENTS LOST THROUGH SATURATION
$EVTCB::.WORD   0               ;TCB ADDRESS OF EVENT LOGGER TASK

;
; EVENT SWITCH TABLE
;
;       INDEXED BY EVENT NUMBER
;       IF BYTE NEGATIVE, EVENT DISABLED
;       IF BYTE POSITIVE, VALUE IS LENGTH OF EVENT
;
; EVERY EVENT MUST CONTAIN AN ENTRY IN THIS TABLE, OF THE FORM:
;
;       .BYTE   -EV.XXX         ;THIS IS A SAMPLE EVENT
;
;   WHERE XXX IS THE NUMBER OF THE EVENT.
;
$EVTAB::                        ;REFERENCE LABEL

        .REPT   E$$VNT
        .BYTE   -1              ;SET ALL EVENTS TO NULL INITIALLY
        .ENDR
        .BYTE   0               ;THERE IS NO EVENT ZERO
```

```
        .EVEN
;
; EVENT BUFFER MANAGEMENT
;
$EVBSQ::.WORD   1               ;BUFFER SEQUENCE NUMBER (NEVER = 0)
$EVLST::                        ;REFERENCE LABEL

        .REPT   E$$BUF+1
        .WORD   0               ;KISAR6 POINTER TO BUFFER
        .ENDR
        .IFF
$EVKS6::                        ;REFERENCE LABEL
$EVDIS::
$EVLEN::
$EVSEQ::
$EVTAB::
$EVBSQ::
$EVLST::
        .ENDC

; WATCHPOINT DEBUG INFO
        .IF DF R$$WPT
$WPLST::.WORD   0               ;LAST SYSTEM STATE ROUTINE CALLED
$WPVAL::.WORD   0               ;VALUE TO WATCH FOR OR AGAINST
$WPADR::.WORD   $WPVAL          ;PLACE TO WATCH FOR IT
        .ENDC   ; DF R$$WPT

;
; ADDITIONAL ARGUMENT TO $POWER FOR MULTIPROCESSOR SYSTEMS.
;
        .IF DF  M$$PRO
$PFURM::.WORD   0               ;URM TO POWERFAIL
        .ENDC

                                                                        ;LBM050
$SWR::  .WORD   0               ;SOFTWARE SWITCH REGISTER IF NONE IN HARDWARE   ;LBM050
                                                                        ;LBM050
                                                                        ;**-5
; EXECUTIVE COMMON APR TABLE
;
;               * * *   N O T E   * * *
;
; THIS TABLE IS ORDER DEPENDENT, SINCE THERE IS A TABLE IN VMR OF THE
; STANDARD COMMON NAMES THAT OCCUR IN THE DIRECTIVE PARTITION. A SIDE
; EFFECT OF THIS IS THAT THE ENTRIES BELOW MAY NOT BE CONDITIONALIZED
;
        .IF DF  D$$PAR
```

```
$DRAPR::.WORD   0               ;APR VALUE TO MAP DIRECTIVE PARTITION
$DRAP2::.WORD   0               ;BIAS OF SECOND DIRECTIVE COMMON
$DRAP3::.WORD   0               ;BIAS OF THIRD DIRECTIVE COMMON
$DRAP4::.WORD   0               ;BIAS OF FOURTH DIRECTIVE COMMON
$DRAPV::.WORD   0               ;BIAS OF EXEC VECTORING DIRECTIVE
$DRCHE::.WORD   0               ;BIAS OF DATA CACHE MANAGER COMMON
        .WORD   -1              ;FLAG FOR END OF COMMON BIAS TABLE

        .IFF

$DRAPR::                        ;ALWAYS DEFINE THIS SYMBOL

        .ENDC

; $ENVEC - TABLE OF ENTRY POINTS FOR VECTORING CODE WHICH CAN'T
;          USE $DRGIN
;
;       THIS TABLE CONTAINS A *SMALL* NUMBER OF ADDRESSES AND IS
;       INTENDED TO PROVIDE THE MINIMUM AMOUNT OF INFO TO ACCESS THEN
;       SYSTEM WITHOUT BEING LINKED TO IT.
;
$ENVEC::.WORD   $DRAPV          ;POINTER TO APR BIAS OF VECTORING
                                ;COMMON

        .IF DF C$$CDA

$CRKRB::.WORD   0               ;CRASH DUMP DEVICE KRB ADDRESS
$CRSFM::.WORD   0               ;CRASH FORMATTER NUMBER (RH TAPE ONLY)
$CRSUN::.WORD   C$$RUN          ;CRASH PHSICAL UNIT NUMBER.  NOTE:  C$$RUN IS
                                ;...A LOGICAL UNIT NUMBER.  INITL CONVERTS
                                ;...$CRSUN TO A PHYSICAL UNIT NUMBER.
$CRCSR::.WORD   0               ;CRASH DEVICE CSR ADDRESS WHEN NO KRB
$CRLCK::.BYTE   1               ;ONLY ONE CPU SHOULD DUMP MEMORY
        .BYTE   0               ;PRESERVE WORD ALIGNMENT

        .ENDC   ; DF C$$CDA


        .IF DF  R$$NDC

$RNDCT::.WORD   R$$NDC          ;CLOCK TICKS TO NEXT SCHEDULE INTERVAL
$RNDC:: .WORD   R$$NDC          ;CLOCK TICKS PER SCHEDULING INTERVAL
$RNDL:: .BYTE   R$$NDL          ;LOWEST PRIORITY CLASS TO CONSIDER
$RNDH:: .BYTE   R$$NDH          ;HIGHEST PRIORITY CLASS

        .ENDC


        .IF DF  S$$WPC&D$$ISK

$SWPCT::.WORD   S$$WPC          ;CLOCK TICKS TO NEXT SWAPPING INTERVAL
$SWPC:: .WORD   S$$WPC          ;CLOCK TICKS PER SWAPPING INTERVAL
$SWPR:: .WORD   S$$WPR          ;SWAPPING PRIORITY

        .ENDC

;
; DAYS PER MONTH TABLE (ENTRY CONTAINS DAYS PER MONTH + 1)
;
```

```
$DYPMN::.BYTE   29.,32.         ;FEBRUARY, MARCH
        .BYTE   31.,32.         ;APRIL, MAY
        .BYTE   31.,32.         ;JUNE, JULY
        .BYTE   32.,31.         ;AUGUST, SEPTEMBER
        .BYTE   32.,31.         ;OCTOBER, NOVEMBER
        .BYTE   32.,32.         ;DECEMBER, JANUARY

;
; BIT MASK TABLE
;

; NOTE: DUE TO THE ORGANIZATION OF THE SUPERVISOR MODE MAPPING CONTROL
;       SUPPORT, NAMELY THE FACT THAT BIT 0 OF THE BYTE CORRESPONDS TO
;       APR 1, NOT APR 0, THERE MUST BE A ZERO PRECEDING $BTMSK.
;
        .WORD   0               ;DUMMY FOR APR0 BITMASK
$BTMSK::.WORD   1               ;BIT 0.
        .WORD   2               ;BIT 1.
        .WORD   4               ;BIT 2.
        .WORD   10              ;BIT 3.
        .WORD   20              ;BIT 4.
        .WORD   40              ;BIT 5.
        .WORD   100             ;BIT 6.
        .WORD   200             ;BIT 7.
        .WORD   400             ;BIT 8.
        .WORD   1000            ;BIT 9.
        .WORD   2000            ;BIT 10.
        .WORD   4000            ;BIT 11.
        .WORD   10000           ;BIT 12.
        .WORD   20000           ;BIT 13.
        .WORD   40000           ;BIT 14.
        .WORD   100000          ;BIT 15.

;       BIT MASK TABLE FOR FORMING ADDRESS MASKS IN $ACHR0 IN
;       USER I/D SYSTEMS

        .IF DF  U$$DAS!S$$LIB

        .BYTE   0
$BCMSK::.BYTE   1,3,7,17,37,77,177,377

        .ENDC   ; DF U$$DAS!S$$LIB

;       TABLE OF APR ADDRESSES USED BY GIN SF.APR FUNCTION

        .EVEN
$APRTB::.WORD   KINAR0,KINDR0   ;KERNEL I SPACE REGISTERS

        .IF DF  K$$DAS

        .WORD   KDSAR0,KDSDR0   ;KERNEL D SPACE REGISTERS

        .ENDC   ; DF K$$DAS

        .WORD   UINAR0,UINDR0   ;USER I SPACE REGISTERS

        .IF DF  U$$DAS

        .WORD   UDSAR0,UDSDR0   ;USER D SPACE REGISTERS

        .ENDC   ; DF U$$DAS

        .IF DF  S$$LIB
```

```
        .WORD   SISARO,SISDRO   ;SUPER I SPACE REGISTERS
        .WORD   SDSARO,SDSDRO   ;SUPER D SPACE REGISTERS

        .ENDC   ; DF S$$LIB

        .WORD   0               ;END OF TABLE

;
; SHUFFLER TASK REQUEST LIMIT FLAGS.  $SHFTM CONTAINS THE NUMBER
; OF CLOCK TICKS THAT MUST EXPIRE BEFORE THE SHUFFLER TASK CAN
; BE REQUESTED BY THE EXECUTIVE.  IT IS INITIALIZED TO THE VALUE
; CONTAINED IN $SHFCT EACH TIME THE SHUFFLER EXECUTES.
;
$SHFTM:: .WORD  0               ;TIME REMAINING BEFORE NEXT POSSIBLE REQUEST

$SHFCT:: .WORD  S$$HFC          ;MINIMUM TICKS BETWEEN SHUFFLER REQUESTS

;
; ONLINE ERROR LOGGING DATA BASE
;

        .IF DF  E$$LOG

;
;       ERROR LOG FEATURE MASK SUPPORT
;
EFEAT = 0
        .IIF    DF,E$$ICM       EFEAT = EFEAT!EL.ICM
        .IIF    DF,E$$MOU       EFEAT = EFEAT!EL.MOU
        .IIF    DF,E$$SEF       EFEAT = EFEAT!EL.SEF
$EFMSK::.WORD   EFEAT           ;SELECTED FEATURE MASK WORDS

$ERHEA::.WORD   0, .-2          ;MESSAGE QUEUE LISTHEAD
$ENTSQ::.WORD   0               ;ENTRY SEQUENCE NUMBER
$ERFLA::.WORD   0               ;ERROR LOGGER FLAG WORD
$ERRX:: .WORD   0               ;POINTER TO SECONDARY POOL ERRLOG DATA

        .IFF

$ERFLA::                        ;REFERENCE LABEL

        .ENDC   ; DF E$$LOG

$ERRPT::.WORD   0               ;POINTER TO ERROR LOGGER TCB
$ERRSQ::.WORD   0               ;ERROR SEQUENCE NUMBER

$NCTPT::
        .IF DF  T$$TSA
        .WORD   0               ;APR BIAS OF NT.NCT
        .WORD   0               ;APR5 ADDRESS OF QUEUE ROUTINE
        .WORD   0               ;NCT... TCB ADDRESS
        .ENDC   ; T$$TSA

        .IF DF R$$PRO
```

```
;
; $NVRTM CONTAINS THE NUMBER OF TICKS THAT MUST ELAPSE BEFORE THE
; O/S UPTIME CELL IN THE PROFESSIONAL 3XX NVR IS UPDATED. IN GENERAL
; IT IS SET TO THE NUMBER OF TICKS PER SECOND TIMES 60. THIS RESULTS
; IN UPTIME IN MINUTES BEING RECORDED. IT IS INITIALIZED WITH $NVRCT
; EACH TIME THAT NVR IS UPDATED.
;
$NVRTM::.WORD   0               ;TICKS TILL NEXT NVR UPDATE
$NVRCT::.WORD   N$$VRC          ;TICKS PER NVR UPDATE CYCLE

;
; TASK NAME OF THE P/OS DISPATCHER.
; USED TO SEND MESSAGES AND AST'S TO IT.
;
$CTBNM::.RAD50  /C$CTEX/        ;NAME OF THE P/OS DISPATCHER

        .ENDC   ; DF R$$PRO
$DVSAV::.BLKW   1               ;SAVED CSR CONTENTS

;
; SYSTEM BOOTSTRAP AND SAVE CONFIGURATION VECTOR
;

        .IF NDF K$$DAS
$SYALT::                        ;REFERENCE LABEL
        .ENDC

$SYSIZ::.WORD   S$$YSZ          ;SIZE OF MEMORY IN 32W BLOCKS
        .BLKB   1               ;LOGICAL UNIT NUMBER OF LOAD DEVICE
        .BLKB   3               ;LBN OF LOAD/SAVE IMAGE ON DISK
        .BLKW   1               ;NAME OF LOAD DEVICE IN ASCII
        .BLKW   1               ;SIZE OF LOAD IMAGE IN 256W BLOCKS

;
; TIME LIMIT PARAMETERS
;
        .WORD   -1              ;YEARS PER UNIVERSE
        .WORD   13.             ;MONTHS PER YEAR
        .WORD   32.             ;DAYS PER MONTH (CALCULATED)
        .WORD   24.             ;HOURS PER DAY
        .WORD   60.             ;MINUTES PER HOUR
        .WORD   60.             ;SECONDS PER MINUTE
;
; ACCOUNTING USES $ACTPS TO GET THE NUMBER OF CLOCK TICKS PER SECOND.
; ON SYSTEMS WHERE ACCOUNTING IS USING THE SYSTEM CLOCK, $ACTPS MUST
; REFERENCE THE SAME WORD AS $TKPS.  IF ACCOUNTING HAS ITS OWN CLOCK,
; IT WILL CONTAIN THE TICK RATE OF THAT CLOCK.
;
$ACTPS::                        ;CLOCK RATE FOR ACCOUNTING

        .IF EQ  K$$CSR-177546
$TKPS:: .WORD   H$$RTZ          ;TICKS PER SECOND
        .IFF
```

```
$TKPS:: .WORD   K$$TPS          ;TICKS PER SECOND
        .ENDC

;
; CURRENT TIME VECTOR
;
        .WORD   82.             ;YEAR OF UNIVERSE - HAPPY 30TH
        .WORD   3.              ;MONTH OF YEAR    - BIRTHDAY
        .WORD   30.             ;DAY OF MONTH     - MS. LAWLER
        .WORD   0               ;HOUR OF DAY
        .WORD   0               ;MINUTE OF HOUR
        .WORD   0               ;SECOND OF MINUTE
$TTNS:: .WORD   0               ;TICK OF SECOND
        .IF DF  P$$3XX
$CKCSA::.WORD   $CKCSA          ;THIS WORD IS SET EITHER TO POINT
                                ;TO ITSELF (PDP-11 HARDWARE) OR
                                ;TO THE CLOCK CSR (SAME AS $CKCSR)
                                ;(PRO HARDWARE) TO MAKE TDSCH CLOCK
                                ;TICKLING RUN TIME CONDITIONAL
        .ENDC   ; DF P$$3XX

;
; LIFO SEND AND I/O PREALLOCATION LIST POINTER AND PARAMETERS
;
        .IF DF  Q$$OPT
$PKAVL::.WORD   0               ;POINTER TO FIRST PACKET IN LIST
$PKNUM::.BYTE   0               ;NUMBER OF PACKETS CURRENTLY IN LIST
$PKMAX::.BYTE   Q$$OPT          ;MAXIMUM NUMBER ALLOWED IN LIST
        .IFF
$PKNUM::                        ;REF LABEL FOR MCR
$PKMAX::                        ;REF LABEL FOR MCR
        .ENDC

;
; SAVE AREA FOR CALCULATED MULTI-USER TASK NAMES.
;
        .IF DF  R$$DSP
$TNAME::.BLKW   2               ;(2ND WORD IS ALSO FLAG FOR RQST & RUN)
        .ENDC

;
; TEMPORARY STORAGE VARIABLES FOR DIRECTIVE SERVICES
;
```

```
$TEMP0::.BLKW   1               ;
$TEMP1::.BLKW   1               ;
$TEMP2::.BLKW   1               ;COMPLETION ROUTINE ADDR FOR $GSPKT
$TEMP3::.WORD   0               ;SCRATCH SPACE FOR SPWN$ AND RPOI$
$TEMP4::.WORD   0               ;SCRATCH SPACE FOR SPWN$ AND RPOI$
$TONYL::.WORD   0               ;SCRATCH STORAGE FOR SEEK OPTIMIZATION
$IOTMP::.WORD   0               ;SCRATCH WD FOR ADDR CHECKING AND
                                ;ACP QIO TEMPORARY WORK
$ATTPT::.WORD   0               ;POINTER TO I.AADA IN CURRENT PACKET

$CTLST::.WORD   $CTBO           ;START OF THE CTB LIST


; KERNEL AST DISPATCH TABLE

; THIS TABLE IS FILLED IN BY INITL AT BOOT TIME. ANY CHANGES MADE
; HERE MUST BE REFLECTED IN INITL AT DBTBL.
;
; IF THE SYSTEM DOES NOT SUPPORT KERNEL I/D SPACE, THEN THE REFERENCES
; ARE RESOLVED DIRECTLY.
;
$KATBL::.WORD   $FINBF          ;$FINBF-BUFF I/O FINISH  (AK.BUF)
        .WORD   $FINXT          ;$FINXT-OFF. TASK EXIT   (AK.OCB)
        .WORD   $GENBF          ;$GENBF-GEN. BUFF. I/O   (AK.GBI)
        .WORD   $DBTRP          ;$DBTRP-FORCE T-BIT TRAP (AK.TBT)
        .WORD   $FINDI          ;$FINBF-DELAYED I/O FIN  (AK.DIO)

        .IF DF  G$$GEF

        .WORD   $GGFRN          ;$GGFRN-GROUP GBL RUNDWN (AK.GGF)

        .IFF    ; DF G$$GEF

;       .WORD   0               ;IF OTHER KASTS ARE INVENTED, MAKE
                                ;THIS A PLACEHOLDER IN THE TABLE.

        .ENDC   ; DF G$$GEF

        .IF DF  M$$PRO

$URMST::.WORD   0               ;UNIBUS RUN MASK STATUS TABLE


        .IF DF  X$$DBT

$XDTFL::.WORD   1               ;XDT INITIALIZATION TABLE
$XDTPR::.WORD   0               ;FLAG FOR PROMPTS OF XDT

        .IFF

$XDTFL::.WORD   177777          ;ALWAYS LOOKS INITIALIZED WHEN NOT THERE
$XDTPR::                        ;REFERENCE LABEL -- UNUSED IF XDT ALREADY
                                ;INITIALIZED

        .ENDC


OFF=177777
        .REPT   M$$PRO
OFF=OFF+OFF
        .ENDR
```

```
$CPMSK::.WORD   OFF             ;PROCESSOR BIT CLEAR MASK
$IIMSK::.WORD   0               ;IIST INTERRUPT MASK WORD
$IIPND::.WORD   0               ;PENDING URM WORK WORD. LOCKED WITH $FORKL
$IICPU::.WORD   0               ;MASK OF URMS THAT HAVE BEEN INTERRUPTED
$IINXT::.WORD   $BTMSK          ;ROUND ROBIN WORD FOR $IISVC
$STENB::.WORD   0               ;SANITY TIMER ENABLED ON VARIOUS CPU'S
$STFLG::.WORD   0               ;SECONDARY SANITY TIMER FLAG
$STALR::.WORD   0               ;SANITY TIMER ALARM ENABLED ON CPU'S
$PWRMK::.WORD   0               ;MASK OF CPUS IN THEIR POWERFAIL CODE
$CPPAR::.WORD   0               ;POINTER TO PARTITION FOR CPU LOCAL MEM

; NOTE THAT THE NEXT TWO TABLES FOLLOW AN INDEX STRUCTURE LIKE THAT
; OF $PARTB+2 INS LOWCR, AND THE TWO SHOULD BE UPDATED IN HARMONY WITH
; HRC (MODULE HRBOX)

        .IF DF  M$$K11

$MKCS1::.WORD   1               ;CONTENTS FOR CSR 1 OF MK11'S
        .WORD   8.              ;NUMBER OF MK11'S POSSIBLE
$MKCSR::.BLKW   8.              ;CSR ADDRESS TABLE FOR MK11 PARITY
$MKCS2::.BLKW   8.              ;CSR CONTENTS FOR MK11'S

        .ENDC   ; DF M$$K11

;
; MULTIPROCESSOR LOCKS
;
;       BYTE 0  -- LOCK VALUE (0 IF LOCKED, 1 IF FREE)
;       BYTE 1  -- IF BYTE 0 EQUALS:
;                       0, THEN BYTE 1 EQUALS PROCESSOR ID OF OWNER
;                       1, THEN BYTE 1 EQUALS COMPLEMENT OF ID OF
;                          PREVIOUS OWNER
;
; IF LOCK TYPE IS "WAIT" THEN:
;
;       BYTE 2,3 = SECONDARY LOCK WORD (ABOVE FORMAT) FOR WAIT MASK.
;       BYTE 4,5 = MASK OF WAITING PROCESSORS.
;
; NOTE: BYTES 4+5 ARE REFERENCED AS A WORD.
;

$EXECL::LCKDF$  WAIT            ;SERIALIZE ACCESS TO EXECUTIVE DATA
$FORKL::LCKDF$  SPIN            ;SERIALIZE ACCESS TO FORK LIST
$IIFNL::LCKDF$  SPIN            ;SERIALIZE ACCESS TO $MPTAB
$PWRLK::LCKDF$  SPIN            ;SERIALIZE ACCESS TO $PWRMK

;
; MULTIPROCESSOR TABLE
;

$TKTAB::                        ;CURRENT TASK TCB TABLE
        .REPT   M$$PRO
        .WORD   $ACTHD          ;INITIALIZE TO NULL TASK TCB
        .ENDR

        .IFTF

$RQTAB::.WORD   $ACTHD          ;RESCHEDULE POINTER TCB TABLE

        .IFT
```

```
        .REPT   M$$PRO-1
        .WORD   $ACTHD          ;INITIALIZE TO NULL TASK TCB
        .ENDR

        .IFTF

$PRTAB::.WORD   0               ;PROCESSOR CURRENT TASK PRIORITY TABLE

        .IFT

        .REPT   M$$PRO-1
        .WORD   0               ;INITIALIZE TO PRIORITY ZERO
        .ENDR

$MPTAB::                        ;PROCESSOR STATUS TABLE
        .REPT   M$$PRO
        .WORD   MP.STP          ;INITIALLY STOPPED
        .ENDR

OFF=0                           ;START K6 OFFSETS AT ZERO
$K6TAB::                        ;PROCESSOR IMPURE AREA APR OFFSETS
        .REPT   M$$PRO
        .WORD   OFF
OFF=OFF+40                      ;INCREASE OFFSET BY 1K
        .ENDR

OFF=1                           ;START PROCESSOR URMS AT 1
$URMTB::                        ;PROCESSOR URM CONNECTIVITY TABLE
        .REPT   M$$PRO
        .WORD   OFF
OFF=OFF+OFF                     ;POOR MAN'S LEFT SHIFT
        .ENDR

OFF=1                           ;START PROCESSOR URMS AT 1
$CRMTB::                        ;COMPLEMENT OF CPU URM CONNECTIVITY TBL
        .REPT   M$$PRO
        .WORD   ^C<OFF>
OFF=OFF+OFF                     ;POOR MAN'S LEFT SHIFT
        .ENDR

        .IFTF

$CRFLG::.WORD   0               ;FLAG INDICATING REGISTERS HAVE BEEN SAVED

        .IFT

        .REPT   M$$PRO-1
        .WORD   0
        .ENDR

$CRFPR::.WORD   377             ;$PROCN OF FIRST PROCESSOR TO CRASH

        .IFT

        .WORD   1               ;URM IN FORK BLOCK FOR ERRLOG

        .IFTF

        .BLKW   2               ;ERROR LOGGING FORK BLOCK
$PBEZ::                         ;REFERENCE LABEL

        .IFT
```

```
URM=2
        .REPT   M$$PRO-1
        .WORD   URM             ;URM IN FORK BLOCK
URM=URM*2
        .BLKW   3               ;3 WORD FORK BLOCK
        .ENDR

        .ENDC

        .IF DF  F$$LPP

        .IF DF  M$$PRO

        .WORD   1               ;URM IN FORK BLOCK

        .IFTF

        .BLKW   2               ;2 WORD FORK BLOCK
$FLFTB::                        ;REFERENCE LABEL

        .IFT

URM=2
        .REPT   M$$PRO-1
        .WORD   URM
URM=URM*2
        .BLKW   2
        .ENDR

        .ENDC

        .ENDC


;
; DEFINITIONS FOR NON MK11 MEMORY SYSTEMS
;
        .IF NDF M$$K11

$MKCS1::
        .WORD   0               ;# OF MK11 REGISTERS (FOR BOO,SAV,HRC)
$MKCSR::
$MKCS2::

        .ENDC   ; DF M$$K11

        .IF NDF M$$PRO
```

```
;
;
; DEFINITIONS FOR NON-MULTIPROCESSOR SYSTEMS
;
$CKURM::                        ;REFERENCE LABELS
$CPBIT::
$CPCRM::                        ;COMPLEMENTED RUN MASK
$CPMSK::
$CPURM::
$CPUSC::                        ;CPU STATUS CHANGE ENTRY
$CRMTB::                        ;COMPLEMENTED RUN MASK TABLE
$EXECL::
$IBXMT::                        ;IIST INTERCPU BOOT TRANSMIT
$IIMSK::
$IISTM::                        ;IIST SUBROUTINE ENTRY POINT
$IIXMT::                        ;IIST TRANSMIT SUBROUTINE
$MPTAB::
$PROCN::
$PROC2::
$PFURM::
$STENB::                        ;SANITY TIMER ENABLE
$STFLG::
$STALR::
$TKTAB::                                                        ;**-1
$URMST::
$URMTB::
$XDTFL::
$XDTPR::
$XDTIN::

        .ENDC

; HERE WE DEFINE A SINGLE "FAKE WORD" WHICH RECEIVES ALL THE ADDRESSES
; FOR THINGS WHICH ARE NOT GENNED INTO THE SYSTEM, BUT WHICH REQUIRE
; THAT THE ADDRESSES ALWAYS BE RESOLVED FOR TASKBUILDING MCR, ETC.
; THIS ONE (IN SYSCM) IS TO BE USED FOR D SPACE LOCATIONS, AND THE
; ONE IN LOWCR IS USED FOR I SPACE REFERENCES AND CPU SPECIFIC INFO
; (ALSO THINGS WHICH MUST BE ACCESSED WHEN D SPACE IS TURNED OFF)

        .IF NDF R$$WPT

$WPVAL::
$WPADD::
$WPLST::

        FAKE = 0

        .ENDC   ; NDF R$$WPT

        .IF NDF E$$LOG

$ERHEA::
$ENTSQ::

        FAKE = 0

        .ENDC   ; NDF E$$LOG

        .IF DF FAKE

        .WORD   -1              ;FAKE DATA LOCATION FOR ALL OF THE
                                ;ABOVE
```

```
        .ENDC   ; DF FAKE

        .IF DF  P$$OOL
$PLPAR::.WORD   0               ;POINTER TO SECONDARY POOL PCB
$POLHD::.WORD   0               ;LIST HEAD FOR SECONDARY POOL FREE LIST
$SECFR::.WORD   0               ;NUMBER OF FREE BLOCKS IN SECONDARY POOL
        .IFF
$PLPAR::
$POLHD::
$SECFR::
        .ENDC

        .IF DF  A$$CNT
$SYLHD::.WORD   0               ;LISTHEAD FOR SYSLOG INPUT QUEUE
        .WORD   $SYLHD          ;END OF LIST POINTER
$SABPT::.WORD   0               ;POINTER TO SYSTEM ACCOUNT BLOCK

        .IF DF  X$$ACC
$ACNFE::.WORD   BF.TRN!BF.XAC   ;EXTENDED ACCOUNTING
        .IFF    ;X$$ACC
$ACNFE::.WORD   BF.TRN          ;NORMAL SUPPORTED ACCOUNTING
        .ENDC   ;X$$ACC

$APLIM::.WORD   0               ;FREE SEC POOL SPACE ACNT MUST RESERVE
$SYUAB::.WORD   0               ;ADDRESS OF UAB FOR SYSTEM TASKS
$CKUAB::.WORD   0               ;UAB FOR TASK RUN FROM CLOCK QUEUE
        .IFF
$SABPT::
$SYLHD::
$ACNFE::
$APLIM::
$SYUAB::
        .IFTF
$LOGPT::.WORD   0               ;POINTER TO TCB OF SYSLOG TASK

        .ENDC

;
; CLI DATABASE AREA
;
$CLICQ::.WORD   0               ;COMMAND QUEUE LISTHEAD
        .WORD   $CLICQ
$CNTLN::.WORD   0               ;CONTINUATION LINE SEGMENT LISTHEAD
        .WORD   $CNTLN
$CPTBL::.WORD   $MCRPT          ;POINTER TO CPB FOR MCR
        .IF DF  A$$CLI
        .REPT   A$$CLI-1
```

```
        .WORD   0               ;TABLE FOR CPB ADDRESSES
        .ENDR
$NMCLI==A$$CLI                  ;NUMBER OF CLIS SYSTEM SUPPORTS
        .IFTF   ;A$$CLI
;
; CPB FOR MCR
;
$MCRPT::.WORD   0               ;POINTER TO MCR'S TCB, START OF MCR'S CPB
        .RAD50  /MCR   /        ;CLI NAME
        .WORD   CP.LGO          ;STATUS WORD
        .BYTE   MDPL            ;LENGTH OF DEFAULT PROMPT STRING
        .BYTE   MCPL            ;LENGTH OF CONTROL/C PROMPT STRING
$$$=.
        .IF DF R$$MIC
        .ASCIZ  <15><12>/$ /    ;MICRO DEFAULT PROMPT
        .IFF    ; DF R$$MIC
        .ASCIZ  <15><12>/>/     ;DEFAULT PROMPT STRING
        .ENDC   ; DF R$$MIC
MDPL=.-$$$
$$$=.
        .IF DF R$$MIC
        .ASCIZ  <15><12>/DCL>/  ;CONTROL/C PROMPT
        .IFF    ; DF R$$MIC
        .ASCIZ  <15><12>/MCR>/  ;CONTROL/C PROMPT STRING
        .ENDC   ; DF R$$MIC
MCPL=.-$$$
        .EVEN
        .IFF    ;A$$CLI
$NMCLI==0                       ;ALTERNATE CLI NOT SUPPORTED
        .ENDC   ;A$$CLI

$CMFIN::                        ; END OF SYSCM AREA FOR CDA
;
; PARAMETER WORD TO CONTROL TERMINAL DRIVER SYSTEM-WIDE BEHAVIOR:
;
; BIT 0 = 0 FOR REMOTE LINES, DON'T HANGUP IMMEDIATELY ON CARRIER LOSS
;         1 FOR REMOTE LINES, HANGUP IMMEDIATELY ON CARRIER LOSS
; BIT 1 = 0 FOR REMOTE LINES, DON'T ENABLE DTR UNTIL RING IS SEEN
;         1 FOR REMOTE LINES, ENABLE DTR BEFORE RING IS SEEN
;
$TTPRM::.WORD   2               ;DEFAULT TERMINAL DRIVER BEHAVIOR
;
; TERMINAL DRIVER DATA BUFFER - NOT DUMPED BY CDA
;
        .IF DF  K$$DAS&T$$COM
```

```
$DALED::.BLKW   1               ;TERMINAL DRIVER DATA
        .ENDC   ;DF K$$DAS&T$$COM

        .IF DF  T$$ACD
$ACDHD::.WORD   0               ;ANCILLARY CONTROLL DRIVER BLOCK LISTHEAD
        .ENDC   ; DF T$$ACD
        .END
```

Appendix C
System Data Structures and Symbolic Definitions

This appendix describes the RSX-11M-PLUS and Micro/RSX system macros that supply symbolic offsets for data structures listed in Table C-1.

The data structures are defined by macros in the Executive macro library. To reference any of the data structure offsets from your code, include the macro name in an .MCALL directive and invoke the macro. For example:

```
.MCALL   DCBDF$
DCBDF$                    ;Define DBC offsets
```

Note

All physical offsets and bit definitions are subject to change in future releases of the operating system. Code that accesses system data structures should always use the symbolic offsets rather than the physical offsets.

The first two arguments, <:> and <=>, make all definitions global. If they are left blank, the definitions will be local.

All of these macros are in the Executive macro library LB:[1,1]EXEMC.MLB. All except F11DF$, ITBDF$, MTADF$, OLRDF$, and SHDDF$ are also in the Executive definition library LB:[1,1]EXELIB.OLB.

Table C-1: Summary of System Data Structure Macros

Macro Arguments	Data Structures
ABODF$ <:>,<=>	Task abort and termination notification message codes
ACNDF$ <:>,<=>	Accounting data structures (user account block, task account block, system account block)
CLKDF$ <:>,<=>	Clock queue control block
CTBDF$ <:>,<=>	Controller table
DCBDF$ <:>,<=>	Device Control Block

Table C–1 (Cont.): Summary of System Data Structure Macros

Macro Arguments	Data Structures
EPKDF$ <:> , <=>	Error message block
EVNDF$ <:> , <=>	Terminal Software Architecture (TSA) event packet definitions
F11DF$ <:> , <=>	FILES-11 data structures (Volume Control Block, mount list entry, File Control Block, file window block, locked block list node)
HDRDF$ <:> , <=>	Task header and window block
HWDDF$ <:> , <=>	Hardware register addresses and feature mask definitions
ITBDF$ <:> , <=>	Interrupt transfer block
KRBDF$ <:> , <=>	Controller request block
LCBDF$ <:> , <=>	Logical assignment control block
LNMDF$ <:> , <=>	Logical name block
MTADF$ <:> , <=>	ANSI magtape data structures (volume set control block)
OLRDF$	On-line reconfiguration interface
PCBDF$ <:> , <=>	Partition Control Block and attachment descriptor
PKTDF$ <:> , <=>	I/O packet, AST control block, offspring control block, group global event flag control block, and CLI parser block
SCBDF$ <:> , <=>	Status Control Block and UMR assignment block
SHDDF$ <:> , <=>	Shadow recording linkage block
TCBDF$ <:> , <=>	Task Control Block
UCBDF$ <:> , <=> ,TTDEF	Unit Control Block

```
        .MACRO  ABODF$,L,B
;+
; TASK ABORT CODES
;
; NOTE: S.COAD-S.CFLT ARE ALSO SST VECTOR OFFSETS
;-
S.CACT='B'-4.                   ;TASK STILL ACTIVE
S.CEXT='B'-2.                   ;TASK EXITED NORMALLY
S.COAD='B'0.                    ;ODD ADDRESS AND TRAPS TO 4
S.CSGF='B'2.                    ;SEGMENT FAULT
S.CBPT='B'4.                    ;BREAK POINT OR TRACE TRAP
S.CIOT='B'6.                    ;IOT INSTRUCTION
S.CILI='B'8.                    ;ILLEGAL OR RESERVED INSTRUCTION
S.CEMT='B'10.                   ;NON RSX EMT INSTRUCTION
S.CTRP='B'12.                   ;TRAP INSTRUCTION
S.CFLT='B'14.                   ;11/40 FLOATING POINT EXCEPTION
S.CSST='B'16.                   ;SST ABORT-BAD STACK
S.CAST='B'18.                   ;AST ABORT-BAD STACK
S.CABO='B'20.                   ;ABORT VIA DIRECTIVE
S.CLRF='B'22.                   ;TASK LOAD REQUEST FAILURE
S.CCRF='B'24.                   ;TASK CHECKPOINT READ FAILURE
S.IOMG='B'26.                   ;TASK EXIT WITH OUTSTANDING I/O
S.PRTY='B'28.                   ;TASK MEMORY PARITY ERROR
S.CPMD='B'30.                   ;TASK ABORTED WITH PMD REQUEST
S.CELV='B'32.                   ;TI: VIRTUAL TERMINAL WAS ELIMINATED
S.CINS='B'34.                   ;TASK INSTALLED IN 2 DIFFERENT SYSTEMS
S.CAFF='B'36.                   ;TASK ABORTED DUE TO BAD AFFINITY (REQUIRED BUS
                                ;RUNS ARE OFFLINE OR NOT PRESENT)
S.CCSM='B'38.                   ;BAD CSM PARAMETERS OR BAD STACK
S.COTL='B'40.                   ;TASK HAS RUN OVER ITS TIME LIMIT
S.CTKN='B'42.                   ;ABORT VIA DIRECTIVE WITH NO TKTN MESSAGE

;
; TERMINATION CODES FOR BOM$
;
; NOTE:
;
; THE NORMAL TKTN ERROR CODES SPAN -4 THROUGH 42. THE BOM CODES,
; ALTHOUGH DEFINED FOR THE TASK, ETC, AS 0 THROUGH N, ARE PASSED TO
; TKTN AS -127 THROUGH -127+N. AN UNRECOGNIZED CODE IS PASSED AS -128,
; UNKNOWN ERROR.
;
;               * * *   W A R N I N G   * * *
;
; THESE CODES ARE CURRENTLY ALSO DEFINED IN BOM.MAC IN BOMDF$. UNTIL
; THAT IS CHANGED, KEEP THE TWO SETS OF DEFINITIONS IN SYNC.
;
S.BUNK='B'-128.                 ;UNKNOWN BOM$ ERROR
S.BFEI='B'0.                    ;ERROR IN HIGH LEVEL LANGUAGE INTERFACE
S.BOVL='B'1.                    ;LOAD OVERLAY FAILURE

;
; KEEP THE FOLLOWING DEFINED AS THE HIGHEST CODE IN USE
;
S.BHI='B'1.                     ;HIGHEST ACCEPTABLE BOM$ CODE
```

ABODF$

```
;
; BIT DEFINITIONS FOR BOM$ FLAGS WORD
;
S.BBIF='B'1.                    ;ENABLE CONDITIONAL BREAKPOINT

;
; TASK TERMINATION NOTIFICATION MESSAGE CODES
;
T.NDNR='B'0                     ;DEVICE NOT READY
T.NDSE='B'2                     ;DEVICE SELECT ERROR
T.NCWF='B'4                     ;CHECKPOINT WRITE FAILURE
T.NCRE='B'6                     ;CARD READER HARDWARE ERROR
T.NDMO='B'8.                    ;DISMOUNT COMPLETE
T.NUER='B'10.                   ;UNRECOVERABLE ERROR
T.NLDN='B'12.                   ;LINK DOWN (NETWORKS)
T.NLUP='B'14.                   ;LINK UP (NETWORKS)
T.NCFI='B'16.                   ;CHECKPOINT FILE INACTIVE
T.NUDE='B'18.                   ;UNRECOVERABLE DEVICE ERROR
T.NMPE='B'20.                   ;MEMORY PARITY ERROR
T.NKLF='B'22.                   ;UCODE LOADER NOT INSTALLED
T.NAAF='B'24.                   ;ACCOUNTING ALLOCATION FAILURE
T.NTAF='B'26.                   ;ACCOUTING TAB ALLOCATION FAILURE
T.NDEB='B'28.                   ;TASK HAS NO DEBUGGING AID
T.NRCT='B'30.                   ;REPLACEMENT CONTROL TASK NOT INSTALLED
T.NWBL='B'32.                   ;WRITE BACK CACHING DATA LOST. UNIT WRITE LOCKED
T.NVER='B'34.                   ;MOUNT VERIFICATION TASK NOT INSTALLED
T.NIOS='B'36.                   ;I/O STALLED TO DEVICE
T.NIOR='B'38.                   ;I/O RESUMING ON DEVICE

        .MACRO  ABODF$  X,Y
        .ENDM
        .ENDM
```

```
;
        .MACRO  ACNDF$,L,B

;+
; ACCOUNTING  BLOCK OFFSET AND STATUS DEFINITIONS
; FOR EACH TRANSACTION TYPE.
;
;
; HEADER COMMON TO ALL TRANSACTIONS
;-

        .ASECT
.=0

B.LNK:'L'.BLKW  1               ;LINK TO NEXT IN SYSLOG QUEUE
B.TYP:'L'.BLKB  1               ;TRANSACTION TYPE
B.LEN:'L'.BLKB  1               ;TRANSACTION LENGTH
B.TIM:'L'.BLKW  3               ;ENDING TIME OF TRANSACTION
B.HID='B'.                      ;START OF HEADER IDENTIFICATION AREA
B.UID:'L'.BLKW  2               ;UNIQUE SESSION IDENT
                                ; FIRST WORD-RAD50, SECOND-BINARY
B.ACN:'L'.BLKW  1               ;ACCOUNT NUMBER
B.TID:'L'.BLKB  1               ;ASCII TERMINAL TYPE (V,T,B OR C)
                                ;(VIRTUAL,REAL,BATCH, OR CONSOLE)
        .BLKB   1               ;UNIT NUMBER
B.HEND='B'.                     ;END OF HEADER ID AREA
$$$HLN=.                        ;HEADER LENGTH


;+
; ACCUMULATION FIELDS FOR TAB, UAB, AND SAB
;-

B.CPU:'L'.BLKW  2               ;TOTAL CPU TIME USED
B.DIR:'L'.BLKW  2               ;TOTAL DIRECTIVE COUNT
B.QIO:'L'.BLKW  2               ;TOTAL QIO$ COUNT
B.TAS:'L'.BLKW  2               ;TOTAL TASK COUNT
B.MEM:'L'.BLKW  3               ;RESERVED
B.BEG:'L'.BLKW  3               ;BEGINNING/LOGIN TIME
B.CPUL:'L'.BLKW 2               ;CPU LIMIT
B.PNT:'L'.BLKW  1               ;POINTER TO HIGHER LEVEL TOTALS
B.STM:'L'.BLKB  1               ;STATUS MASK
$$$TLN=.                        ;TOTAL'S LENGTH

;+
; USER ACCOUNT BLOCK (UAB)
;       NOTE:  UAB'S MUST END ON A WORD BOUNDRY
;-
```

ACNDF$

```
.=$$$TLN                        ;START AFTER TOTALS
B.USE:'L'.BLKB  1               ;USE COUNT
B.ACT:'L'.BLKW  1               ;NUMBER OF CURRENTLY ACTIVE TASKS
B.UUIC:'L'.BLKW 1               ;LOGIN UIC
B.UCB:'L'.BLKW  1               ;POINTER TO UCB
B.LGO:'L'.BLKW  3               ;LOGOFF TIME
B.ULNK:'L'.BLKW 1               ;LINK TO NEXT UAB
B.RNA:'L'.BLKW  3               ;LOC IN SYSTEM ACCNT FILE (OFFSET,VBN-HI,VBN-LO)
B.NAM:'L'.BLKB  14.             ;LAST NAME OF USER
        .BLKB   1               ;FIRST INITIAL OF USER
        .BLKB   1               ;FLAG BYTE FOR UAB (bs.sil) etc.
B.LDS:'L'.BLKB  10.             ;LOGIN DIRECTORY STRING
B.ULEN='B'.                     ;UAB LENGTH
$$$=    <.+77>/100              ;UAB LENGTH (ROUNDED UP TO 32 WORD BOUND)

;+
; TASK ACCOUNT BLOCK (TAB)
;       NOTE: THE TAB MUST END ON A WORD BOUNDRY
;-

.=$$$TLN                        ;STARTS AFTER TOTALS
B.PRI:'L'.BLKB  1               ;HIGHEST RUNNING PRIORITY
B.TNAM:'L'.BLKW 2               ;TASK NAME
B.TCB:'L' .BLKW 1               ;TCB ADDRESS
B.TST3:'L'.BLKW 1               ;T.ST3 FROM TASK'S TCB
          .BLKW 1               ;RESERVED FOR FUTURE STATUS BITS
B.CUIC:'L'.BLKW 1               ;CURRENT UIC OF TASK
B.PUIC:'L'.BLKW 1               ;PROTECTION UIC OF TASK
B.CTXT:'L'.BLKW 2               ;NUMBER OF CONTEXT LOADS
B.TCKP:'L'.BLKW 2               ;TIMES TASK HAS BEEN CHECKPOINTED
B.OVLY:'L'.BLKW 2               ;NUMBER OF DISK OVERLAY LOADS
B.EXST:'L'.BLKW 2               ;EXIT STATUS AND ABORT CODE
B.TLEN='B'.                     ;TAB LENGTH
B.TBLK='B'<.+77>/100            ;NUMBER OF SEC POOL BLOCKS IN TAB

;+
; SYSTEM ACCOUNT BLOCK (SAB)
;-

.=$$$TLN                        ;START AFTER TOTALS
B.SHDN:'L'.BLKB 1               ;ACCOUNTING SHUTDOWN REASON CODE
B.UHD:'L'.BLKW  1               ;UAB LISTHEAD
B.ULO:'L'.BLKW  1               ;NUMBER OF USERS CURRENTLY LOGGED ON
B.ULT:'L'.BLKW  2               ;TOTAL NUMBER OF LOGONS
B.CKP:'L'.BLKW  2               ;TOTAL NUMBER OF CHECKPOINTS
B.SHF:'L'.BLKW  2               ;TOTAL NUMBER OF SHUFFLER RUNS
B.RND:'L'.BLKW  2               ;NUMBER OF CPU INTERVALS ROUNDED UP TO 1
B.FID:'L'.BLKW  3               ;FILE-ID OF TRANSACTION FILE
B.DVNM:'L'.BLKB 2               ;DEVICE  OF TRANSACTION FILE
B.UNIT:'L'.BLKW 1               ;UNIT    OF TRANSACTION FILE
B.EXTS:'L'.BLKW 1               ;EXTEND SIZE FOR TRANSACTION FILE
B.LSCN:'L'.BLKW 3               ;TIME OF LAST SCAN
B.SCNR:'L'.BLKW 1               ;SCAN RATE IN SECONDS
B.DSCN:'L'.BLKW 1               ;STATISTICAL SCAN RATE (IN SEC)
B.STSP:'L'.BLKW 2               ;RESERVED
B.SYSM:'L'.BLKW 1               ;RESERVED
B.CKUS:'L'.BLKW 3               ;RESERVED
B.CKSP:'L'.BLKW 2               ;RESERVED
B.CKAL:'L'.BLKW 1               ;RESERVED
B.SLEN='B'.                     ;SAB LENGTH
```

```
; NEW FIELDS FOR EXTENDED ACCOUNTING

B.CPUT:'L'.BLKW 8.              ;CPU TIME USED PER PROCESSOR
B.CTXP:'L'.BLKW 8.              ;NUMBER OF CONTEXT SWITCHES (PER PROC)
B.IDCT:'L'.BLKW 8.              ;NUMBER OF IDLE LOOP ENTRIES (PER PROC)
B.QIOC:'L'.BLKW 8.              ;NUMBER OF I/O INITIATIONS (PER PROC)
B.MIOC:'L'.BLKW 8.              ;MASS STORE I/O COMPLETIONS (PER PROC)
B.AIOC:'L'.BLKW 8.              ;ALL I/O COMPLETIONS (PER PROC)
B.IPSN:'L'.BLKW 8.              ;IP INTERRUPTS SENT (PER PROC)
B.IPRC:'L'.BLKW 8.              ;IP INTERRUPTS RCVD (PER PROC)
B.CKEX:'L'.BLKW 2               ;CHECKPOINT DUE TO EXTEND TASKS
B.CFCL:'L'.BLKW 2               ;CALLS TO CFORK
B.CFRK:'L'.BLKW 2               ;CFORK FORKS
B.TLOD:'L'.BLKW 2               ;TASK LOADS
B.RLOD:'L'.BLKW 2               ;REGION LOADS
        .BLKB 82.               ;BUMP SIZE TO NEXT 32 WORD BLOCK
B.SSBL=.-B.SLEN                 ;EXTRA LENGTH OF SYSTEM STATISTICS BLOCK
$$$=    <.+77>/100              ;SAB LENGTH (ROUNDED UP TO 32 WORD BOUND)

;+
; SYSLOG STARTUP TRANSACTION
;-

.=      $$$HLN                  ;START AFTER HEADER
B.SSLN='B'                      ;TRANSACTION LENGTH

;+
; CRASH RECOVERY TRANSACTION
;-

.=      $$$HLN                  ;START AFTER STANDARD HEADER
B.CTLS:'L'.BLKW 3               ;TIME OF LAST SCAN BEFORE CRASH
B.CSRT:'L'.BLKW 1               ;SCAN RATE BEFORE CRASH
B.CRSN:'L'.BLKB 60.             ;ASCII TEXT EXPLAINING CRASH
B.CLEN='B'.                     ;TRANSACTION LENGTH

;+
; INVALID LOGIN TRANSACTION
;-

.=      $$$HLN                  ;
B.INAM:'L'.BLKB 14.             ;NAME FROM LOGIN LINE
B.IUIC:'L'.BLKB 6.              ;UIC FROM LOGIN LINE
B.IPSW:'L'.BLKB 6.              ;PASSWORD FROM LOGIN LINE
B.ILEN='B'.                     ;TRANSACTION LENGTH

;+
; DEVICE TRANSACTIONS (ALLOCATION, DEALLOCATION, MOUNT, AND DISMOUNT)
;-

.=      $$$HLN                  ;
B.DNAM:'L'.BLKW 1               ;ASCII DEVICE NAME
B.DUNT:'L'.BLKB 1               ;OCTAL DEVICE UNIT NUMBER
B.DLEN='B'.                     ;TRANSACTION LENGTH FOR ALL, DEA, AND DMO
        .BLKB   1               ;UNUSED BYTE
B.DLBL:'L'.BLKW 6               ;VOLUME LABEL
B.DMST:'L'.BLKW 1               ;MOUNT STATUS BITS
B.DUIC:'L'.BLKW 1               ;OWNER UIC
B.DVPR:'L'.BLKW 1               ;VOLUME PROTECTION CODE
B.DACP:'L'.BLKW 2               ;NAME OF ACP FOR DEVICE
B.MLEN='B'.                     ;LENGTH OF MOUNT TRANSACTION
```

ACNDF$

```
;+
; STATUS BITS FOR MOUNT STATUS MASK (B.DMST)
;-
BM.SHR='B'1                     ;DEVICE IS MOUNTED SHARED
BM.NOS='B'2                     ;DEVICE IS MOUNTED NOSHARE
BM.SYS='B'4                     ;DEVICE IS MOUNTED FOR THE SYSTEM (PUBLIC)
BM.FOR='B'10                    ;DEVICE IS MOUNTED FOREIGN

;+
; SYSTEM TIME CHANGE TRANSACTION
;-
.=      $$$HLN                  ;
B.TOLD:'L'.BLKB 6               ;OLD TIME (YR, MON, DAY, HR, MIN, SEC)
B.TNEW:'L'.BLKB 6               ;NEW TIME (YR, MON, DAY, HR, MIN, SEC)
B.TMLN='B'.                     ;TRANSACTION LENGTH

;+
; PRINT DESPOOLER TRANSACTION
;-
.=      $$$HLN                  ;START AFTER HEADER
B.PNAM:'L'.BLKW 3               ;PRINT JOB NAME  (RAD50)
B.PPGS:'L'.BLKW 1               ;PAGE COUNT
B.PNFI:'L'.BLKW 1               ;NUMBER OF FILES PRINTED
B.PFRM:'L'.BLKB 1               ;FORM NUMBER
B.PPRI:'L'.BLKB 1               ;PRINT PRIORITY
B.PDEV:'L'.BLKW 1               ;PRINT DEVICE NAME  (ASCII)
B.PPUN:'L'.BLKB 1               ;UNIT NUMBER OF PRINT DEVICE
B.PLEN='B'.                     ;TRANSACTION LENGTH

;+
; CARD READER SPOOLING TRANSACTION
;-
.=      $$$HLN                  ;START AFTER HEADER
B.RNAM:'L'.BLKW 3               ;BATCH OR PRINT JOB NAME
B.RCDS:'L'.BLKW 1               ;NUMBER OF CARDS READ
B.RDEV:'L'.BLKW 1               ;READER DEVICE NAME (ASCII)
B.RUNT:'L'.BLKB 1               ;UNIT NUMBER OF READER DEVICE
B.RSOP:'L'.BLKB 1               ;SUBMIT OR PRINT (0=SUBMIT, 1=PRINT)
B.RLEN='B'.                     ;TRANSACTION LENGTH

;+
; LOGIN TRANSACTION
;-
.=      $$$HLN                  ;START AFTER HEADER
B.LUIC:'L'.BLKW 1               ;LOGIN UIC
B.LNAM:'L'.BLKB 14.             ;USER'S LAST NAME
        .BLKB   1               ;AND FIRST INITIAL
B.LLEN='B'.                     ;TRANSACTION LENGTH

;+
;  RESET TRANSACTION PARAMETERS
;-
```

```
.=$$$HLN                                ;AFTER HEADER
B.OFID:'L'.BLKW 3                       ;FILE-ID OF OLD TRN. FILE
B.ODNM:'L'.BLKB 2                       ;DEVICE OF OLD TRN. FILE
B.OUNT:'L'.BLKW 1                       ;UNIT OF OLD TRN. FILE
B.NFID:'L'.BLKW 3                       ;FILE-ID OF NEW TRN. FILE
B.NDNM:'L'.BLKB 2                       ;DEVICE OF NEW TRN. FILE
B.NUNT:'L'.BLKW 1                       ;UNIT OF NEW TRN. FILE
B.OEXS:'L'.BLKW 1                       ;EXT. SIZE FOR OLD TRN. FILE
B.NEXS:'L'.BLKW 1                       ;EXT. SIZE FOR NEW TRN. FILE
B.OSCR:'L'.BLKW 1                       ;OLD SCAN RATE IN SECONDS
B.NSCR:'L'.BLKW 1                       ;NEW SCAN RATE IN SECONDS
B.ODSC:'L'.BLKW 1                       ;OLD STATISTICAL SCAN RATE
B.NDSC:'L'.BLKW 1                       ;NEW STATISTICAL SCAN RATE
B.RTLN='B'.

;+
; TRANSACTION TYPES
;
;       000 THRU 127            RESERVED FOR DEC USE
;       128 THRU 255            RESERVED FOR CUSTOMER USE
;
;-

BT.SAB='B'1                     ;SYSTEM ACCOUNT BLOCK (SAB)
BT.UAB='B'2                     ;USER ACCOUNT BLOCK (UAB)
BT.TAB='B'3                     ;TASK ACCOUNT BLOCK (TAB)
BT.SS='B'11                     ;SYSLOG STARTUP TRANSACTION
BT.INV='B'12                    ;INVALID LOGIN TRANSACTION
BT.TIM='B'13                    ;SYSTEM TIME CHANGE TRANSACTION
BT.ALL='B'14                    ;ALLOCATE DEVICE TRANSACTION
BT.DEA='B'15                    ;DEALLOCATE DEVICE TRANSACTION
BT.MOU='B'16                    ;MOUNT DEVICE TRANSACTION
BT.DMO='B'17                    ;DISMOUNT DEVICE TRANSACTION
BT.PRT='B'20                    ;PRINT DESPOOLER TRANSACTION
BT.DIR='B'21                    ;DISK ACCOUNTING BY DIRECTORY   (UNSUPPORTED)
BT.VOL='B'22                    ;DISK ACCOUNTING BY VOLUME      (UNSUPPORTED)
BT.LOG='B'23                    ;LOGIN TRANSACTION
BT.CRH='B'24                    ;CRASH RECOVERY TRANSACTION
BT.DST='B'25                    ;DEVICE STATISTICS (UCB EXTENSION)
BT.RTP='B'26                    ;RESET TRANSACTION PARAMETERS
BT.INP='B'27                    ;CARD READER SPOOLING TRANSACTION


;+
; STATUS MASK BIT DEFINITIONS (B.STM)
;-

BS.ACT='B'200                   ;CONTROL BLOCK ACTIVE
BS.CRH='B'100                   ;RECORD FROM "TMP" FILE AFTER SYSTEM CRASH
BS.LGO='B'40                    ;LOGGED OFF WITH OUTSTANDING ACTIVITY (UAB)
BS.CO='B'40                     ;TASK'S TI: IS CO: (TAB ONLY)
BS.TML='B'20                    ;TAB EXISTS ONLY FOR TIME LIMIT (TAB ONLY)
BS.SIL='B'20                    ;SILENT LOGIN/LOGOUT (UAB ONLY)
BS.ZER='B'10                    ;LAST CPU INTERVAL WAS OF LENGTH ZERO
BS.SCN='B'4                     ;TRANSACTION READY FOR WRITE TO SCAN FILE

;+
; ACCOUNTING FEATURE MASK ($ACNFE)
;-
```

ACNDF$

```
BF.DST='B'40000                 ;STATISTICAL SCAN RATE
BF.WRT='B'2000                  ;FORCE SYSLOG TO WRITE ITS BUFFER
BF.SCN='B'1000                  ;SCAN REQUESTED
BF.SLR='B'400                   ;SYSLOG IS RUNNING (NOT STOPPED)
BF.ERR='B'200                   ;ACCOUNTING STOPPED DUE TO FATAL ERROR
BF.STR='B'100                   ;ACCOUNTING IS STARTING UP / SHUTTING DOWN
BF.LSS='B'40                    ;ACCUMULATE SYSTEM STATISTICS (POINT UAB TO SAB)
BF.TRN='B'10                    ;OUTPUT TO TRANSACTION FILE
BF.XTK='B'4                     ;CHECKPOINT REQUEST IS DUE TO EXTK$
BF.TSK='B'2                     ;TASK ACCOUNTING TURNED ON
BF.XAC='B'1                     ;EXTENDED ACCOUNTING ASSEMBLED IN

;+
; SHUTDOWN CODES (B.SHDN)
;-

; 1             MAINTENANCE
; 2             REBOOT
; 3             SCHEDULED_SHUTDOWN
; 4             ACCOUNTING SHUTDOWN BY TASK "SHUTUP"
; 5             OTHER


;*********************************************************************
B.MAXL='B'128.                  ;MAXIMUM TRANSACTION LENGTH
B.MINL='B'$$$HLN                ;MINIMUM TRANSACTION LENGTH
;*********************************************************************

        .PSECT
        .MACRO  ACNDF$  X,Y
        .ENDM
        .ENDM

        .MACRO  ACTDF$,L,B
        .ASECT
.=0
A.GRP:'L'       .BLKB   3               ; GROUP CODE (ASCII)
A.MBR:'L'       .BLKB   3               ; MEMBER CODE
A.PSWD:'L'      .BLKB   6               ; PASSWORD
A.LNM:'L'       .BLKB   14.             ; LAST NAME
A.FNM:'L'       .BLKB   12.             ; FIRST NAME
A.LDAT:'L'      .BLKB   6               ; DATE OF LAST LOG ON (DD/MM/YY HH:MM:SS
A.NLOG:'L'      .BLKB   2               ; TOTAL NUMBER OF LOGONS
A.SYDV:'L'      .BLKB   4               ; DEFAULT SYSTEM DEVICE
A.ACN:'L'       .BLKW   1               ;ACCOUNT NUMBER (BINARY)
A.CLI:'L'        .BLKW  2               ; RAD50 USER CLI
                .BLKW   2               ; UNUSED
A.LPRV:'L'      .BLKW   1               ;LOGIN PRIVILEGE WORD
A.SID:'L'       .BLKW   1               ; SESSION IDENTIFIER
A.DDS:'L'       .BLKB   11.             ;DEFAULT DIRECTORY STRING
                .BLKB   1               ;UNUSED BYTE
A.FPRO:'L'      .BLKW   1               ;DEFAULT FILE PROTECTION
A.RLVL:'L'      .BLKW   1               ;ACCOUNT RECORD REV. LEVEL
AR.LVL='B'401
A.SALT:'L'      .BLKW   1               ;16-BIT ENCRYPTION SALT VALUE
A.ENCT:'L'      .BLKB   1               ;ENCRYPTION TYPE
                                        ;       0 = PLAIN TEXT OR ENCRPT
                                        ;       1 = PURDY-V ALGORITHM
                .BLKB   1               ;UNUSED
```

```
A.HPW:'L'       .BLKW   4               ;HASHED PASSWORD
                .IF DF  A$$LOG
A.TTY:'L'       .BLKB   5               ;TERMINAL TTNNN FOR AUTO LOGIN
A.PRID:'L'      .BLKB   1               ;PRIMARY DAYS MASK
A.SECD:'L'      .BLKB   1               ;SECONDARY DAYS MASK
A.PRIT:'L'      .BLKW   1               ;PRIMARY DAYS TIME
A.SECT:'L'      .BLKW   1               ;SECONDARY DAYS TIME
A.RLEN  ='B'    .
                .ENDC   ; DF A$$LOG
A.LEN   ='B'    128.                    ; LENGTH OF CONTROL BLOCK
;
;
; BIT DEFINITIONS ON A.LPRV - LOGIN PRIVILEGE BITS
;
AL.SLV='B'      1                       ;SLAVE TERMINAL ON LOGIN
AL.DDS='B'      2                       ;INDICATOR FOR PROLOGUE 2 FORMAT
AL.SIL='B'      4                       ;SILENT LOGIN/LOGOUT
                .IF DF A$$LOG
AL.AUT='B'      10                      ;AUTO LOGIN ENABLED     ('*)
AL.BND='B'      20                      ;BINDING ENABLED        ('Y)
AL.RMT='B'      40                      ;REMOTE DIALUP 1=NO
AL.NET='B'      100                     ;NETWORK LOGIN 1=NO
AL.DIS='B'      200                     ;DISABLE THIS ACCOUNT FROM LOGIN
AL.PRI='B'      400                     ;PRIMARY DAYS LIMIT SET
AL.SEC='B'      1000                    ;SECONDARY DAYS LIMIT SET
                .ENDC ; DF A$$LOG
;
        .PSECT
        .ENDM
```

CLKDF$

```
        .MACRO  CLKDF$,L,B
;+
; CLOCK QUEUE CONTROL BLOCK OFFSET DEFINITIONS
;
; CLOCK QUEUE CONTROL BLOCK
;
; THERE ARE FIVE TYPES OF CLOCK QUEUE CONTROL BLOCKS. EACH CONTROL BLOCK HAS
; THE SAME FORMAT IN THE FIRST FIVE WORDS AND DIFFERS IN THE REMAINING THREE.
;
; THE FOLLOWING CONTROL BLOCK TYPES ARE DEFINED:
;-
C.MRKT='B'0                     ;MARK TIME REQUEST
C.SCHD='B'2                     ;TASK REQUEST WITH PERIODIC RESCHEDULING
C.SSHT='B'4                     ;SINGLE SHOT TASK REQUEST
C.SYST='B'6                     ;SINGLE SHOT INTERNAL SYSTEM SUBROUTINE (IDENT)
C.SYTK='B'8.                    ;SINGLE SHOT INTERNAL SYSTEM SUBROUTINE (TASK)
C.CSTP='B'10.                   ;CLEAR STOP BIT (CONDITIONALIZED ON SHUFFLING)

;
; CLOCK QUEUE CONTROL BLOCK TYPE INDEPENDENT OFFSET DEFINTIONS
;
        .ASECT
.=0
C.LNK:'L' .BLKW 1               ;CLOCK QUEUE THREAD WORD
C.RQT:'L' .BLKB 1               ;REQUEST TYPE
C.EFN:'L' .BLKB 1               ;EVENT FLAG NUMBER (MARK TIME ONLY)
C.TCB:'L' .BLKW 1               ;TCB ADDRESS OR SYSTEM SUBROUTINE IDENTIFICATION
C.TIM:'L' .BLKW 2               ;ABSOLUTE TIME WHEN REQUEST COMES DUE

;
; CLOCK QUEUE CONTROL BLOCK-MARK TIME DEPENDENT OFFSET DEFINITIONS
;
.=C.TIM+4                       ;START OF DEPENDENT AREA
C.AST:'L' .BLKW 1               ;AST ADDRESS
C.SRC:'L' .BLKW 1               ;FLAG MASK WORD FOR 'BIS' SOURCE
C.DST:'L' .BLKW 1               ;ADDRESS OF 'BIS' DESTINATION
        .BLKW   1               ;UNUSED

;
; CLOCK QUEUE CONTROL BLOCK-PERIODIC RESCHEDULING DEPENDENT OFFSET DEFINITIONS
;
.=C.TIM+4                       ;START OF DEPENDENT AREA
C.RSI:'L' .BLKW 2               ;RESCHEDULE INTERVAL IN CLOCK TICKS
C.UIC:'L' .BLKW 1               ;SCHEDULING UIC
C.UAB:'L'.BLKW  1               ;POINTER TO ASSOCIATED UAB

;
; CLOCK QUEUE CONTROL BLOCK-SINGLE SHOT DEPENDENT OFFSET DEFINITIONS
;
.=C.TIM+4                       ;START OF DEPENDENT AREA
        .BLKW   2               ;TWO UNUSED WORDS
        .BLKW   1               ;SCHEDULING UIC
        .BLKW   1               ;C.UAB
```

```
;
; CLOCK QUEUE CONTROL BLOCK-SINGLE SHOT INTERNAL SUBROUTINE OFFSET DEFINITIONS
;
; THERE ARE TWO TYPE CODES FOR THIS TYPE OF REQUEST:'L'
;
;       TYPE 6=SINGLE SHOT INTERNAL SUBROUTINE WITH A 16 BIT VALUE AS AN IDENTIFIER.
;       TYPE 8=SINGLE SHOT INTERNAL SUBROUTINE WITH A TCB ADDRESS AS AN IDENTIFIER.
;
.=C.TIM+4                       ;START OF DEPENDENT AREA
C.SUB:'L' .BLKW 1               ;SUBROUTINE ADDRESS
C.AR5:'L' .BLKW 1               ;RELOCATION BASE (FOR LOADABLE DRIVERS)
C.URM:'L'.BLKW  1               ;URM TO EXECUTE ROUTINE ON
                                ;(MP SYSTEMS, C.SYST ONLY)
        .BLKW   1               ;UNUSED
C.LGTH='B'.                     ;LENGTH OF CLOCK QUEUE CONTROL BLOCK

;
;       NAMED DIRECTORY SUPPORT
;
; OFFSET C.EFN WILL BE REUSED IN SCHEDULING REQUESTS TO INDICATE IF C.UIC
; HAS A VALID UIC (C.NAM=0) OR IF C.UIC POINTS TO A CONTEXT BLOCK WITH A DDS.
; IN SCHEDULE REQUESTS, C.EFN WILL BE REFERRED TO AS C.NAM.
;
C.NAM='B'C.EFN                  ;FLAG WORD FOR USE WITH NAME DIRECTORIES
;
; DEPENDING ON THE VALUE IN C.NAM, C.UIC WILL CONTAIN A UIC OR A POINTER
; TO A CONTEXT BLOCK.  C.UIC WILL BE REFERED TO AS C.CTX WHEN IT CONTAINS
; A POINTER TO A CONTEXT BLOCK.
;
C.CTX='B'C.UIC                  ;POINTER TO CONTEXT BLOCK


        .PSECT

        .MACRO  CLKDF$  X,Y
        .ENDM
        .ENDM
```

CTBDF$

```
;------------------------------------------------------------------------
;       CTBDF$
;------------------------------------------------------------------------

        .MACRO      CTBDF$,L,B,SYSDEF

;+
; CONTROLLER TABLE (CTB)
;
; THE CONTROLLER TABLE IS A CONTROL BLOCK THAT CONTAINS A VECTOR
; OF KRB ADDRESSES.  THIS VECTOR MAY BE ADDRESSED BY THE CONTROLLER
; INDEX TAKEN FROM THE INTERRUPT PS BY $INTSV/$INTSE.
;-

        .ASECT
.=177756

L.CLK:'L' .BLKW 8.              ;START OF CLOCK BLOCK (IF ANY)
L.ICB:'L' .BLKW 1               ;ICB CHAIN FOR THIS CTB
L.LNK:'L' .BLKW 1               ;CTB LINK WORD
L.NAM:'L' .BLKW 1               ;GENERIC CONTROLLER NAME (ASCII)
L.DCB:'L' .BLKW 1               ;DCB ADDRESS OF THIS DEVICE
L.NUM:'L' .BLKB 1               ;NUMBER OF KRB ADDRESSES IN TABLE
L.STS:'L' .BLKB 1               ;CTB STATUS BYTE
L.KRB:'L' .BLKW 1               ;START OF KRB ADDRESSES.

;
; NOTE: THE SYMBOL $XYCTB:: IS DEFINED FOR EACH CTB, WHERE THE
; CHARACTERS XY ARE THE SAME AS THOSE STORED IN L.NAM.  THE
; SYMBOL IS NOT THE START OF THE CTB, BUT INSTEAD THE START OF
; THE KRB TABLE AT THE END OF THE CTB (L.KRB).
;

        .PSECT

;+
; CONTROLLER TABLE STATUS BYTE BIT DEFINITIONS
;-

LS.CLK='B'1                     ;CLOCK BLOCK AT TOP OF CTB (1=YES)
LS.MDC='B'2                     ;MULTIDRIVER CTB.  (1=YES)
LS.CBL='B'4                     ;CLOCK BLK LINKED INTO CLK Q (1=YES)
LS.CIN='B'10                    ;CONT. USE COMMON INT TABLE (1=YES)
LS.NET'B'=20            ;THIS IS DECNET DEVICE.  ICB'S IN K.PRM. (1=YES)

;
; COMMON INTERRUPT TABLE DISPATCH ENTRY POINTS
;

CI.CSR='B'-6                    ;CSR TEST ENTRY POINT
CI.KRB='B'-4                    ;KRB STATUS CHANGE ENTRY POINT
CI.PWF='B'-2                    ;POWERFAIL ENTRY POINT
CI.INT='B'0                     ;COMMON INTERRUPT ADDRESS
CI.DCB='B'2                     ;START OF DCB TABLE (0 ENDS TABLE)

        .MACRO  CTBDF$,X,Y,Z
        .ENDM
        .ENDM
```

```
        .MACRO  DCBDF$,L,B,SYSDEF

;+
;
; DEVICE CONTROL BLOCK
;
; THE DEVICE CONTROL BLOCK (DCB) DEFINES GENERIC INFORMATION ABOUT A DEVICE
; TYPE AND THE LOWEST AND HIGHEST UNIT NUMBERS. THERE IS AT LEAST ONE DCB
; FOR EACH DEVICE TYPE IN A SYSTEM. FOR EXAMPLE, IF THERE ARE TELETYPES IN A
; SYSTEM, THEN THERE IS AT LEAST ONE DCB WITH THE DEVICE NAME 'TT'. IF PART
; OF THE TELETYPES WERE INTERFACED VIA DL11-A'S AND THE REST VIA A DH11, THEN
; THERE WOULD BE TWO DCB'S. ONE FOR ALL DL11-A INTERFACED TELETYPES, AND ONE
; FOR ALL DH11 INTERFACED TELETYPES.
;-

        .ASECT
.=0
D.LNK:'L' .BLKW 1               ;LINK TO NEXT DCB
D.UCB:'L' .BLKW 1               ;POINTER TO FIRST UNIT CONTROL BLOCK
D.NAM:'L' .BLKW 1               ;GENERIC DEVICE NAME
D.UNIT:'L' .BLKB 1              ;LOWEST UNIT NUMBER COVERED BY THIS DCB
        .BLKB   1               ;HIGHEST UNIT NUMBER COVERED BY THIS DCB
D.UCBL:'L' .BLKW 1              ;LENGTH OF EACH UNIT CONTROL BLOCK IN BYTES
D.DSP:'L' .BLKW 1               ;POINTER TO DRIVER DISPATCH TABLE
D.MSK:'L' .BLKW 1               ;LEGAL FUNCTION MASK  CODES 0-15.
        .BLKW   1               ;CONTROL FUNCTION MASK CODES 0-15.
        .BLKW   1               ;NOP'ED FUNCTION MASK CODES 0-15.
        .BLKW   1               ;ACP FUNCTION MASK CODES 0-15.
        .BLKW   1               ;LEGAL FUNCTION MASK CODES 16.-31.
        .BLKW   1               ;CONTROL FUNCTION MASK CODES 16.-31.
        .BLKW   1               ;NOP'ED FUNCTION MASK CODES 16.-31.
        .BLKW   1               ;ACP FUNCTION MASK CODES 16.-31.
D.PCB:'L' .BLKW 1               ;LOADABLE DRIVER PCB ADDRESS

        .PSECT

;+
; DRIVER DISPATCH TABLE OFFSET DEFINITIONS
;-

D.VDEB='B'-2                    ;DEALLOCATE BUFFER(S)
D.VCHK='B'-4                    ;ADDRESS OF ROUTINE CALLED TO VALIDATE
                                ;AND CONVERT THE LBN.  USED BY DRIVERS
                                ;THAT SUPPORT SEEK OPTIMIZATION.
D.VNXC='B'-4                    ;ADDRESS OF ROUTINE IN TTDRV CALLED TO
                                ;HAVE IT SEND THE NEXT COMMAND IN THE
                                ;TYPEAHEAD BUFFER TO MCR...
D.VTOU='B'-10                   ;ADDRESS OF ROUTINE IN TTDRV CALLED
                                ;FOR OUTPUT COMPLETION
D.VTIN='B'-6                    ;ADDRESS OF ROUTINE IN TTDRV CALLED
                                ;FOR INPUT FROM THE CT FIRMWARE TASK
D.VINI='B'0                     ;DEVICE INITIATOR
D.VCAN='B'2                     ;CANCEL CURRENT I/O FUNCTION
D.VOUT='B'4                     ;DEVICE TIMEOUT
D.VPWF='B'6                     ;POWERFAIL RECOVERY
D.VKRB='B'10                    ;CONTROLLER STATUS CHANGE ENTRY
D.VUCB='B'12                    ;UNIT STATUS CHANGE ENTRY

        .IF NB  SYSDEF

D.VINT='B'14                    ;BEGINNING OF INTERRUPT STUFF
```

DCBDF$

```
.ENDC

.MACRO  DCBDF$,X,Y,Z
.ENDM
.ENDM
```

```
        .MACRO  EPKDF$,L,B
;+
; Error Message Block Definitions
;-
        .ASECT
;
; Header Subpacket
;
;
;       +-----------------------------------------------+
;       | Subpacket Length in Bytes                     |
;       +-----------------------------------------------+
;       | Subpacket Flags                               |
;       +-----------------------+-----------------------+
;       | Format Identification | Operating System Code |
;       +-----------------------+-----------------------+
;       | Operating System Identification               |
;       |                                               |
;       +-----------------------+-----------------------+
;       | Flags                 | Context Code          |
;       +-----------------------+-----------------------+
;       | Entry Sequence                                |
;       +-----------------------------------------------+
;       | Error Sequence                                |
;       +-----------------------+-----------------------+
;       | Entry Type Subcode    | Entry Type Code       |
;       +-----------------------+-----------------------+
;       | Time Stamp                                    |
;       |                                               |
;       |                                               |
;       +-----------------------+-----------------------+
;       | Reserved              | Processor Type        |
;       +-----------------------+-----------------------+
;       | Processor Identification (URM)                |
;       +-----------------------------------------------+
;
.=0
E$HLGH:'L'      .BLKW   1       ; Subpacket length in bytes
E$HSBF:'L'      .BLKW   1       ; Subpacket Flags
E$HSYS:'L'      .BLKB   1       ; Operating System Code
E$HIDN:'L'      .BLKB   1       ; Format Identification
E$HSID:'L'      .BLKB   4       ; Operating System Identification
E$HCTX:'L'      .BLKB   1       ; Context Code
E$HFLG:'L'      .BLKB   1       ; Flags
E$HENS:'L'      .BLKW   1       ; Entry Sequence Number
E$HERS:'L'      .BLKW   1       ; Error Sequence Number
E$HENC:'L'                      ; Entry Code
E$HTYC:'L'      .BLKB   1       ; Entry Type Code
E$HTYS:'L'      .BLKB   1       ; Entry Type Subcode
E$HTIM:'L'      .BLKB   6       ; Time Stamp
E$HPTY:'L'      .BLKB   1       ; Processor Type
                .BLKB   1       ; Reserved
E$HURM:'L'      .BLKW   1       ; Processor Identification (URM)
```

EPKDF$

```
                .EVEN
E$HLEN:'L'                      ; Length
;
; Subpacket Flags for E$HSBF
;
        SM.ERR  ='B'         1  ; Error Packet
        SM.HDR  ='B'         1  ; Header Subpacket
        SM.TSK  ='B'         2  ; Task Subpacket
        SM.DID  ='B'         4  ; Device Identification Subpacket
        SM.DOP  ='B'        10  ; Device Operation Subpacket
        SM.DAC  ='B'        20  ; Device Activity Subpacket
        SM.DAT  ='B'        40  ; Data Subpacket
        SM.MBC  ='B'     20000  ; 22-bit massbus controller present
        SM.CMD  ='B'     40000  ; Error Log Command Packet
        SM.ZER  ='B'    100000  ; Zero I/O Counts
;
; Codes for field E$HIDN
;
        EH$FOR  ='B'         2  ; Current packet format

;
; Flags for the error log flags byte ($ERFLA) in the exec.
;
        ES.INI  ='B'         1  ; Error log initialized
        ES.DAT  ='B'         2  ; Error log receiving data packets
        ES.LIM  ='B'         4  ; Error limiting enabled
        ES.LOG  ='B'        10  ; Error logging enabled
;
; Type and Subtype Codes for fields E$HTYC and E$HTYS
;
;       Symbols with names E$Cxxx are type codes for field E$HTYC,
;       symbols with names E$Sxxx are subtype codes for field E$HTYS.
;
        E$CCMD  ='B'         1  ; Error Log Control
        E$SSTA  ='B'         1  ;       Error Log Status Change
        E$SSWI  ='B'         2  ;       Switch Logging Files
        E$SAPP  ='B'         3  ;       Append File
        E$SBAC  ='B'         4  ;       Declare Backup File
        E$SSHO  ='B'         5  ;       Show
        E$SCHL  ='B'         6  ;       Change Limits

        E$CERR  ='B'         2  ; Device Errors
        E$SDVH  ='B'         1  ;       Device Hard Error
        E$SDVS  ='B'         2  ;       Device Soft Error
        E$STMO  ='B'         3  ;       Device Interrupt Timeout (HARD)
        E$SUNS  ='B'         4  ;       Device Unsolicited Interrupt
        E$STMS  ='B'         5  ;       Device Interrupt Timeout (SOFT)

        E$CDVI  ='B'         3  ; Device Information
        E$SDVI  ='B'         1  ;       Device Information Message

        E$CDCI  ='B'         4  ; Device Control Information
        E$SMOU  ='B'         1  ;       Device Mount
        E$SDMO  ='B'         2  ;       Device Dismount
        E$SRES  ='B'         3  ;       Device Count Reset
        E$SRCT  ='B'         4  ;       Block Replacement

        E$CMEM  ='B'         5  ; Memory Detected Errors
        E$SMEM  ='B'         1  ;       Memory Error
```

```
        E$CSYS  ='B'         6  ; System Control Information
        E$SPWR  ='B'         1  ;       Power Recovery

        E$CCTL  ='B'         7  ; Control Information
        E$STIM  ='B'         1  ;       Time Change
        E$SCRS  ='B'         2  ;       System Crash
        E$SLOA  ='B'         3  ;       Device Driver Load
        E$SUNL  ='B'         4  ;       Device Driver Unload
        E$SHRC  ='B'         5  ;       Reconfiguration Status Change
        E$SMES  ='B'         6  ;       Message

        E$CCPU  ='B'        10  ; CPU Detected Errors
        E$SINT  ='B'         1  ;       Unexpected Interrupt
;       E$SINT  ='B'         2  ;       Unexpected Interrupt
; Subtype code 2 is reserved.  Use 3 for the next following Subtype code

        E$CSDE  ='B'        11  ; Software Detected Events
        E$SABO  ='B'         1  ;       Task Abort

;
; Codes for Context Code entry E$HCTX
;
        EH$NOR  ='B'         1  ; Normal Entry
        EH$STA  ='B'         2  ; Start Entry
        EH$CRS  ='B'         3  ; Crash Entry
;
; Codes for Flags entry E$HFLG
;
        EH$VIR  ='B'         1  ; Addresses are virtual
        EH$EXT  ='B'         2  ; Addresses are extended
        EH$COU  ='B'         4  ; Error counts supplied
        EH$QBS  ='B'        10  ; Q-BUS CPU
        EH$LMR  ='B'        20  ; Limit reached
;
; Task Subpacket
;
;       +-----------------------------------------------+
;       | Task Subpacket Length                         |
;       +-----------------------------------------------+
;       | Task Name in RAD50                            |
;       |                                               |
;       +-----------------------------------------------+
;       | Task UIC                                      |
;       +-----------------------------------------------+
;       | Task TI: Device Name                          |
;       +----------------------+------------------------+
;       | Flags                | Task TI: Unit Number   |
;       +----------------------+------------------------+

.=0

E$TLGH:'L'      .BLKW   1       ; Task Subpacket Length
E$TTSK:'L'      .BLKW   2       ; Task Name in RAD50
E$TUIC:'L'      .BLKW   1       ; Task UIC
E$TTID:'L'      .BLKB   2       ; Task TI: Device Name
E$TTIU:'L'      .BLKB   1       ; Task TI: Unit
E$TFLG:'L'      .BLKB   1       ; Flags

        .EVEN
E$TLEN:'L'
```

EPKDF$

```
;
; Flags for entry E$TFLG
;
        ET$PRV  ='B'            1   ; Task is Privileged
        ET$PRI  ='B'            2   ; Terminal is Privileged
;
; Device Identification Subpacket
;
;       +-----------------------------------------------+
;       | Device Identification Subpacket Length        |
;       +-----------------------------------------------+
;       | Device Mnemonic Name                          |
;       +-----------------------+-----------------------+
;       | Controller Number     | Device Unit Number    |
;       +-----------------------+-----------------------+
;       | Physical Subunit #    | Physical Unit #       |
;       +-----------------------+-----------------------+
;       | Physical Device Mnemonic (RSX-11M-PLUS only)  |
;       +-----------------------+-----------------------+
;       | Reserved              | Flags                 |
;       +-----------------------+-----------------------+
;       | Volume Name of Mounted Volume                 |
;       |                                               |
;       |                                               |
;       |                                               |
;       |                                               |
;       |                                               |
;       +-----------------------------------------------+
;       | Pack Identification                           |
;       |                                               |
;       +-----------------------------------------------+
;       | Device Type Class                             |
;       +-----------------------------------------------+
;       | Device Type                                   |
;       |                                               |
;       +-----------------------------------------------+
;       | I/O Operation Count Longword                  |
;       |                                               |
;       +-----------------------+-----------------------+
;       | Hard Error Count      | Soft Error Count      |
;       +-----------------------+-----------------------+
;       | Blocks Transferred Count (RSX-11M-PLUS only)  |
;       |                                               |
;       +-----------------------------------------------+
;       | Cylinders Crossed Count (RSX-11M-PLUS only)   |
;       |                                               |
;       +-----------------------------------------------+
;
.=0
E$ILGH:'L'      .BLKW   1       ; Device Identification Subpacket Length
E$ILDV:'L'      .BLKW   1       ; Device Mnemonic Name
E$ILUN:'L'      .BLKB   1       ; Device Unit Number
E$IPCO:'L'      .BLKB   1       ; Controller Number
E$IPUN:'L'      .BLKB   1       ; Physical Unit Number
E$IPSU:'L'      .BLKB   1       ; Physical Subunit Number

                .IF DF R$$MPL

E$IPDV:'L'      .BLKW   1       ; Physical Device Mnemonic
```

```
                .ENDC ; R$$MPL
E$IFLG:'L'      .BLKB   1       ; Flags
                .BLKB   1       ; Reserved
E$IVOL:'L'      .BLKB   12.     ; Volume Name
E$IPAK:'L'      .BLKB   4       ; Pack Identification
E$IDEV:'L'                      ; Device Type
E$IDCL:'L'      .BLKW   1       ; Device Type Class
E$IDTY:'L'      .BLKW   2       ; Device Type
E$IOPR:'L'      .BLKW   2       ; I/O Operation Count Longword
E$IERS:'L'      .BLKB   1       ; Soft Error Count
E$IERH:'L'      .BLKB   1       ; Hard Error Count

                .IF DF R$$MPL

E$IBLK:'L'      .BLKW   2       ; Blocks transferred count
E$ICYL:'L'      .BLKW   2       ; Cylinders crossed count

                .ENDC ; R$$MPL

        .EVEN

E$ILEN:'L'                      ; Subpacket Length
;
; Flags for field E$IFLG
;
        EI$SUB  ='B'        1   ; Subcontroller device

                .IF DF R$$MPL

        EI$NUX  ='B'        2   ; No UCB extension, data invalid

                .ENDC ; R$$MPL
;
; Device Operation Subpacket
;
;       +-----------------------------------------------+
;       | Device Operation Subpacket Length             |
;       +-----------------------------------------------+
;       | Task Name in RAD50                            |
;       |                                               |
;       +-----------------------------------------------+
;       | Task UIC                                      |
;       +-----------------------------------------------+
;       | Task TI: Logical Device Mnemonic              |
;       +-----------------------+-----------------------+
;       | Reserved              | Task TI: Device Unit  |
;       +-----------------------+-----------------------+
;       | I/O Function Code                             |
;       +-----------------------+-----------------------+
;       | Reserved              | Operation Flags       |
;       +-----------------------+-----------------------+
;       | Transfer Operation Address                    |
;       |                                               |
;       +-----------------------------------------------+
;       | Transfer Operation Byte Count                 |
;       +-----------------------+-----------------------+
;       |    Maximum Retries    |     Retries Left      |
;       +-----------------------+-----------------------+
;
.=0
```

EPKDF$

```
E$OLGN:'L'      .BLKW   1       ; Subpacket Length
E$OTSK:'L'      .BLKW   2       ; Task Name in RAD50
E$OUIC:'L'      .BLKW   1       ; Task UIC
E$OTID:'L'      .BLKB   2       ; Task TI: Logical Device Mnemonic
E$OTIU:'L'      .BLKB   1       ; Task TI: Logical Device Unit
                .BLKB   1       ; Reserved
E$OFNC:'L'      .BLKW   1       ; I/O Function Code
E$OFLG:'L'      .BLKB   1       ; Operation Flags
                .BLKB   1       ; Reserved
E$OADD:'L'      .BLKW   2       ; Transfer Operation Address
E$OSIZ:'L'      .BLKW   1       ; Transfer Operation Byte Count
E$ORTY:'L'      .BLKB   1       ; Retries Left
                .BLKB   1       ; Maximum Retries

        .EVEN

E$OLEN:'L'                      ; Device Operation Subpacket Length
;
; Flags for field E$OFLG
;
        EO$TRA  ='B'         1  ; Transfer Operation
        EO$DMA  ='B'         2  ; DMA Device
        EO$EXT  ='B'         4  ; Extended Addressing Device
        EO$PIP  ='B'        10  ; Device is positioning
        EO$IIO  ='B'        20  ; Internal I/O operation
;
; I/O Activity Subpacket
;
;       +---------------------------------------------+
;       | I/O Activity Subpacket Length               |
;       +---------------------------------------------+
;

.=0

E$ALGH:'L'      .BLKW   1       ; Subpacket Length
```

```
;
; I/O Activity Subpacket Entry
;
;       +-----------------------------------------------+
;       | Logical Device Name Mnemonic                  |
;       +-----------------------+-----------------------+
;       | Controller Number     | Logical Device Unit   |
;       +-----------------------+-----------------------+
;       | Physical Subunit #    | Physical Unit Number  |
;       +-----------------------+-----------------------+
;       | Physical Device Mnemonic (RSX-11M-PLUS only)  |
;       +-----------------------+-----------------------+
;       | Task TI: logical unit | Device flags          |
;       +-----------------------+-----------------------+
;       | Requesting Task Name in RAD50                 |
;       |                                               |
;       +-----------------------------------------------+
;       | Requesting Task UIC                           |
;       +-----------------------------------------------+
;       | Task TI: Logical Device Name                  |
;       +-----------------------------------------------+
;       | I/O Function Code                             |
;       +-----------------------+-----------------------+
;       | Reserved              | Flags                 |
;       +-----------------------+-----------------------+
;       | Transfer Operation Address                    |
;       |                                               |
;       +-----------------------------------------------+
;       | Transfer Operation Byte Count                 |
;       +-----------------------------------------------+
;
.=0
E$ALDV:'L'      .BLKW   1       ; Logical Device Name Mnemonic
E$ALUN:'L'      .BLKB   1       ; Logical Device Unit
E$APCO:'L'      .BLKB   1       ; Controller Number
E$APUN:'L'      .BLKB   1       ; Physical Unit Number
E$APSU:'L'      .BLKB   1       ; Physical Subunit Number
                .IF DF R$$MPL
E$APDV:'L'      .BLKW   1       ; Physical Device Mnemonic
                .ENDC
E$ADFG:'L'      .BLKB   1       ; Device flags
E$ATIU:'L'      .BLKB   1       ; Task TI: Logical Unit
E$ATSK:'L'      .BLKW   2       ; Requesting Task Name in RAD50
E$AUIC:'L'      .BLKW   1       ; Requesting Task UIC
E$ATID:'L'      .BLKW   1       ; Task TI: Logical Device Name
E$AFNC:'L'      .BLKW   1       ; I/O Function Code
E$AFLG:'L'      .BLKB   1       ; Flags
                .BLKB   1       ; Reserved
E$AADD:'L'      .BLKW   2       ; Transfer Operation Address
E$ASIZ:'L'      .BLKW   1       ; Transfer Operation Byte Count
        .EVEN
```

EPKDF$

```
E$ALEN:'L'                      ; Subpacket Entry Length
;
; Flags for field E$ADFG
;
        EA$SUB  ='B'         1  ; Subcontroller device

                .IF DF R$$MPL

        EA$NUX  ='B'         2  ; No UCB extension, data invalid

                .ENDC ; R$$MPL
;
; Flags for field E$AFLG
;
        EA$TRA  ='B'         1  ; Transfer Operation
        EA$DMA  ='B'         2  ; DMA Device
        EA$EXT  ='B'         4  ; Device has Extended Addressing
        EA$PIP  ='B'        10  ; Device is positioning
        EA$IIO  ='B'        20  ; Internal I/O operation
        .PSECT
;
; FLAG DEFINITIONS FOR ERROR LOG FEATURE MASK
;
        EL.ICM  ='B'    1       ;SET - INHIBIT OPERATOR CONSOLE MESSAGES
        EL.SEF  ='B'    2       ;SET - SPECIAL FILE FORMATS ENABLED
        EL.MOU  ='B'    4       ;SET - PROCESS MOU/DMO IN SPECIAL FILES

        .MACRO  EPKDF$  X,Y
        .ENDM

        .ENDM
```

```
        .SBTTL  EVNDF$ -- TSA Event Packet Definitions

;+
; EVNDF$ -- Event Packet Definitions
;
;       This macro defines symbols for Event Packet offsets and fields
;       needed to support Digital's Terminal Software Architecture on RSX.
;
; Explicit Inputs:
;
;       L       ":" for global offset definitions
;       B       "=" for global bit/value definitions
;       LST     "LIST" for macro expansion listing
;
; Implicit Inputs:
;
;       NONE
;
; Outputs:
;
;       Symbols defined as described above.
;       Listing as described above.
;-
;
        .MACRO  EVNDF$  L,B,LST
        .IIF NB LST     .LIST   ; Conditionally list macro expansion
;
; General packet header format
;
        .ASECT                  ; Define offsets absolutely
        .=0

E.VLNK:'L'      .BLKW   1       ; Link word
E.VSIZ:'L'      .BLKB   1       ; Packet size
E.VTYP:'L'      .BLKB   1       ; Packet type
E.VUCB:'L'      .BLKW   1       ; Terminal UCB address

;
; E.VTYP Values
;
ET.LOW='B'0                     ; Lowest valid type code

ET.QIO='B'0                     ; QIO (distinguishes QIO packet from TEP)
ET.BND='B'2                     ; Bind Request
ET.UNB='B'4                     ; Unbind Request
ET.BCP='B'6                     ; Bind Complete
ET.REJ='B'10                    ; Bind Reject
ET.DIS='B'12                    ; Disconnect Notification
ET.DCP='B'14                    ; Disconnect Complete
ET.ICS='B'16                    ; Input Count State Change
ET.OOB='B'20                    ; Out-of-Band (OOB)
ET.ONO='B'22                    ; Abnormal Termination Request
ET.PHO='B'24                    ; Physical Terminal Disconnected

ET.HI='B'24                     ; Highest valid type code
```

EVNDF$

```
;
;       The following definitions are for packet types that require
;  passing additional information in the packets.  All other packet
;  types use the general packet format described above.
;

;
; Bind Request packet (Terminal Management Mode --> Network)
;
        .=E.VUCB+2
E.VBCT:'L'      .BLKW   1       ; Count of nodes (One for now)
E.VBND:'L'      .BLKB   6       ; Node name
E.VBLN:'L'                      ; Length of bind request

;
; Input Count State Change, Out-Of-Band packets (TTDRV --> Network)
; And Modem Hang-up packets (TSA... --> Network)
;
        .=E.VUCB+2
E.VAPR:'L'      .BLKW   1       ; Doubleword address of packet...
E.VADR:'L'      .BLKW   1       ; ...queueing routine
E.VFLG:'L'      .BLKW   1       ; Flag

;
; Input Count State Change
;
        .=E.VFLG+2
E.VSLN:'L'                      ; Length of Input state message

;
; OOB
;
        .=E.VFLG+2
E.VOBM:'L'      .BLKW   6       ; Out-of-Band bitmasks
E.VHDR:'L'      .BLKW   2       ; Type-ahead buffer header
E.VTAB:'L'      .BLKB   10.     ; Type-ahead buffer
E.VOLN:'L'                      ; Length of OOB packet

;
; Terminal Management Switch Characters
;
        .=E.VFLG+2
E.VSWC:'L'      .BLKW   1       ; Terminal management switch characters
E.VTLN:'L'                      ; Length of Switch Character packet

;
; Bit values in flag word (E.VFLG).  For convenience some bits have
;       corresponding bits in the AST Control Block flag word (A.PRM+5).
;
EF.NCO='B'1                     ; All non-control characters are out-of band
EF.NOI='B'2                     ; All non-control OOB are include-OOB
EF.AST='B'10                    ; Reserved bit synonymous with TF.AST
EF.LCK='B'40                    ; Reserved bit synonymous with AF.LCK
EF.QUE='B'100                   ; TEP is queued
EF.MDE='B'200                   ; TEP is marked for delete
```

```
;
; Unbind Request packet (TMM --> Network)
;
        .=E.VUCB+2
E.VULN:'L'                      ; Length of Unbind message
;
; Connect Reject notification packet (Network --> TMM)
;
        .=E.VUCB+2
E.VRR:'L'       .BLKW   1       ; Reason for Rejection
E.VRLN:'L'                      ; Length of Reject message
;
; Disconnect Notification packet (Network --> TMM)
;
        .=E.VUCB+2
E.VRD:'L'       .BLKW   1       ; Reason for Disconnect
E.VDLN:'L'                      ; Length of Disconnect message
;
; Disconnect Complete packet (TMM --> Network)
;
        .=E.VUCB+2
E.VDCL:'L'                      ; Length of Disconnect Complete message

        .PSECT
.IF NB LST
        .NLIST                  ; Turn listing back off
.IFF
        .MACRO  EVNDF$          ; If not listing, redefine
        .ENDM                   ;  macro to nothing
.ENDC
        .ENDM   EVNDF$
```

F11DF$

```
        .MACRO  F11DF$,L,B,SYSDEF

;
; VOLUME CONTROL BLOCK
;

        .ASECT
.=0

V.TRCT:'L'.BLKW 1       ; TRANSACTION COUNT
V.TYPE:'L'.BLKB 1       ; VOLUME TYPE DESCRIPTOR
        VT.FOR='B' 0    ; Foreign volume structure
        VT.SL1='B' 1    ; Files-11 Structure level 1
        VT.SL2='B' 2    ; Files-11 Structure level 2
        VT.ANS='B' 10   ; ANSI labeled tape
        VT.UNL='B' 11   ; Unlabeled tape
V.VCHA:'L'.BLKB 1       ; Volume characteristics
        VC.SLK='B' 1    ; Clear volume valid on dismount
        VC.HLK='B' 2    ; Unload the volume on dismount
        VC.DEA='B' 4    ; Deallocate the volume on dismount
        VC.PUB='B' 10   ; Set (clear) US.PUB on dismount
        VC.DUP='B' 20   ; Duplicate volume name; don't delete logicals
        VC.SIL='B' 40   ; Silent mode; suppress DISMOUNT COMPLETE message
V.LABL:'L'.BLKB 14      ; Volume label (ASCII)
V.PKSR:'L'.BLKW 2       ; Pack serial number for error logging
V.SLEN:'L'              ; Length of short VCB
V.IFWI:'L'.BLKW 1       ; INDEX FILE WINDOW
V.FCB:'L'.BLKW  2       ; FILE CONTROL BLOCK LIST HEAD
V.IBLB:'L'.BLKB 1       ; INDEX BIT MAP 1ST LBN HIGH BYTE
V.IBSZ:'L'.BLKB 1       ; INDEX BIT MAP SIZE IN BLOCKS
        .BLKW   1       ; INDEX BITMAP 1ST LBN LOW BITS
V.FMAX:'L'.BLKW 1       ; MAX NO. OF FILES ON VOLUME
V.WISZ:'L'.BLKB 1       ; DEFAULT SIZE OF WINDOW IN RTRV PTRS
                        ; VALUE IS < 128.
V.SBCL:'L'.BLKB 1       ; STORAGE BIT MAP CLUSTER FACTOR
V.SBSZ:'L'.BLKW 1       ; STORAGE BIT MAP SIZE IN BLOCKS
V.SBLB:'L'.BLKB 1       ; STORAGE BIT MAP 1ST LBN HIGH BYTE
V.FIEX:'L'.BLKB 1       ; DEFAULT FILE EXTEND SIZE
        .BLKW   1       ; STORAGE BIT MAP 1ST LBN LOW BITS
;
;  WARNING
;
;  THE FOLLOWING CELLS OF THE VCB ARE ORDER DEPENDANT.
;  THEY ARE RETURNED BY A READ ATTRIBUTES FUNCTION AND
;  MUST BE KEPT CONTIGUOUS.  IF NOT, THINGS MAY BREAK
;  SOMEWHERE ALONG THE LINE.
;
V.VOWN:'L'.BLKW 1       ; VOLUME OWNER'S UIC
V.VPRO:'L'.BLKW 1       ; VOLUME PROTECTION
V.FPRO:'L'.BLKW 1       ; VOLUME DEFAULT FILE PROTECTION
V.FRBK:'L'.BLKB 1       ; NUMBER OF FREE BLOCKS ON VOLUME HIGH BYTE
V.LRUC:'L'.BLKB 1       ; COUNT OF AVAILABLE LRU SLOTS IN FCB LIST
        .BLKW   1       ; NUMBER OF FREE BLOCKS ON VOLUME LOW BITS
;
;  WARNING
;
;  THE ABOVE CELLS OF THE VCB ARE ORDER DEPENDANT.
;  THEY ARE RETURNED BY A READ ATTRIBUTES FUNCTION AND
;  MUST BE KEPT CONTIGUOUS.  IF NOT, THINGS MAY BREAK
;  SOMEWHERE ALONG THE LINE.
```

```
;
V.STS:'L'.BLKB  1       ; VOLUME STATUS BYTE, CONTAINING THE FOLLOWING
        VS.IFW='B' 1    ; INDEX FILE IS WRITE ACCESSED
        VS.BMW='B' 2    ; STORAGE BITMAP FILE IS WRITE ACCESSED
V.FFNU:'L'.BLKB 1       ; FIRST FREE INDEX FILE BITMAP BLOCK
V.EXT:'L'.BLKW  1       ; POINTER TO VCB EXTENSION
V.HBLB:'L'.BLKW 2       ; LBN of home block
V.HBCS:'L'.BLKW 2       ; Home block checksums
V.LGTH:'L'              ; SIZE IN BYTES OF VCB

;
; MOUNT LIST ENTRY
;
; EACH ENTRY ALLOWS ACCESS TO A SPECIFIED USER FOR A NON-PUBLIC DEVICE
;
; TO ALLOW EXPANSION, ONLY THE ONLY TYPE CODE DEFINED IS "1" FOR
; DEVICE ACCESS BLOCKS
;

        .ASECT
.=0

M.LNK:'L'.BLKW  1       ; LINK WORD
M.TYPE:'L'.BLKB 1       ; TYPE OF ENTRY
        MT.MLS='B' 1    ; Mounted volume user access list
M.ACC:'L'.BLKB  1       ; NUMBER OF ACCESSES
M.DEV:'L'.BLKW  1       ; DEVICE UCB
M.TI:'L'.BLKW   1       ; ACCESSOR TI: UCB
M.LEN:'L'               ; LENGTH OF ENTRY

;
; FILE CONTROL BLOCK
;

        .ASECT
.=0

F.LINK:'L'.BLKW 1       ; FCB CHAIN POINTER
F.FNUM:'L'.BLKW 1       ; FILE NUMBER
F.FSEQ:'L'.BLKW 1       ; FILE SEQUENCE NUMBER
        .BLKB   1       ; NOT USED
F.FSQN:'L'.BLKB 1       ; FILE SEGMENT NUMBER
F.FOWN:'L'.BLKW 1       ; FILE OWNER'S UIC
F.FPRO:'L'.BLKW 1       ; FILE PROTECTION CODE
F.UCHA:'L'.BLKB 1       ; USER CONTROLLED CHARACTERISTICS
F.SCHA:'L'.BLKB 1       ; SYSTEM CONTROLLED CHARACTERISTICS
F.HDLB:'L'.BLKW 2       ; FILE HEADER LOGICAL BLOCK NUMBER
                        ; BEGINNING OF STATISTICS BLOCK
F.LBN:'L'.BLKW  2       ; LBN OF VIRTUAL BLOCK 1 IF CONTIGUOUS
                        ; 0 IF NON CONTIGUOUS
F.SIZE:'L'.BLKW 2       ; SIZE OF FILE IN BLOCKS
F.NACS:'L'.BLKB 1       ; NO. OF ACCESSES
F.NLCK:'L'.BLKB 1       ; NO. OF LOCKS
S.STBK='B'.-F.LBN       ; SIZE OF STATISTICS BLOCK
```

F11DF$

```
F.STAT:'L'              ; FCB STATUS WORD
F.NWAC:'L'.BLKB 1       ; NUMBER OF WRITE ACCESSORS
        .BLKB   1       ; STATUS BITS FOR FCB CONSISTING OF
        FC.WAC='B' 100000 ; SET IF FILE ACCESSED FOR WRITE
        FC.DIR='B' 40000  ; SET IF FCB IS IN DIRECTORY LRU
        FC.CEF='B' 20000  ; SET IF DIRECTORY EOF NEEDS UPDATING
        FC.FCO='B' 10000  ; SET IF TRYING TO FORCE DIRECTORY CONTIG
F.DREF:'L'.BLKW 1       ; DIRECTORY EOF BLOCK NUMBER
F.DRNM:'L'.BLKW 1       ; 1ST WORD OF DIRECTORY NAME
F.FEXT:'L'.BLKW 1       ; POINTER TO EXTENSION FCB
F.FVBN:'L'.BLKW 2       ; STARTING VBN OF THIS FILE SEGMENT
F.LKL:'L'.BLKW  1       ; POINTER TO LOCKED BLOCK LIST FOR FILE
F.WIN:'L'.BLKW  1       ; WINDOW BLOCK LIST FOR THIS FILE
F.LGTH:'L'              ; SIZE IN BYTES OF FCB

;
; WINDOW
;
        .ASECT
.=0
W.ACT:'L'               ; NUMBER OF ACTIVE MAPPING POINTERS
                        ;  WHEN NO SECONDARY POOL
W.BLKS:'L'              ; BLOCK SIZE OF SECONDARY POOL SEGMENT
                        ;  WHEN SECONDARY POOL
W.CTL:'L'.BLKW  1       ; LOW BYTE = # OF MAP ENTRIES ACTIVE
                        ; HIGH BYTE CONSISTS OF CONTROL BITS
        WI.RDV='B' 400    ; READ VIRTUAL BLOCK ALLOWED IF SET
        WI.WRV='B' 1000   ; WRITE VIRTUAL BLOCK ALLOWED IF SET
        WI.EXT='B' 2000   ; EXTEND ALLOWED IF SET
        WI.LCK='B' 4000   ; SET IF LOCKED AGAINST SHARED ACCESS
        WI.DLK='B' 10000  ; SET IF DEACCESS LOCK ENABLED
        WI.PND='B' 20000  ; WINDOW TURN PENDING BIT
        WI.EXL='B' 40000  ; SET IF MANUAL UNLOCK DESIRED
        WI.WCK='B' 100000 ; Data check all writes to file
W.IOC:'L'.BLKB  1       ; COUNT OF I/O THROUGH THIS WINDOW
W.STS:'L'.BLKB  1       ; STATUS BYTE
        WS.MDL='B'1     ; FILE DEFINED BY THIS WINDOW IS MARKED FOR DELETE
W.FCB:'L'.BLKW  1       ; FILE CONTROL BLOCK ADDRESS
W.TCB:'L'.BLKW  1       ; TCB address of accessor
W.UCB:'L'.BLKW  1       ; Original UCB address of device
W.LKL:'L'.BLKW  1       ; POINTER TO LIST OF USERS LOCKED BLOCKS
W.WIN:'L'.BLKW  1       ; WINDOW BLOCK LIST LINK WORD

        .IF     NB,SYSDEF  ; IF SYSDEF SPECIFIED IN CALL

        .IF     NDF,P$$WND ; IF SECONDARY POOL WINDOWS NOT ALLOWED

;
; NON-SECONDARY POOL WINDOW BLOCK
;       IF SECONDARY POOL WINDOWS ARE NOT ENABLED, THE WINDOW BLOCK
;       CONTAINS THE CONTROL INFORMATION AND RETRIEVAL POINTERS.
;
W.VBN:'L'.BLKB  1       ; HIGH BYTE OF 1ST VBN MAPPED BY WINDOW
W.MAP:'L'               ; DEFINE LABEL WITH ODD ADDRESS TO CATCH BAD REFS
W.WISZ:'L'.BLKB 1       ; SIZE IN RTRV PTRS OF WINDOW (7 BITS)
        .BLKW   1       ; LOW ORDER WORD OF 1ST VBN MAPPED
W.RTRV:'L'              ; OFFSET TO 1ST RETRIEVAL POINTER IN WINDOW
```

```
W.SLEN='B'-4            ; Dummy definition to prevent incorrect reference
                        ; (-4 when rounded "up" is a VERY large block)
        .IFF            ; IF WINDOWS IN SECONDARY POOL
;
; SECONDARY POOL WINDOW CONTROL AND MAPPING BLOCK
;       IF SECONDARY POOL WINDOW BLOCKS ARE ENABLED, LUTN2 POINTS
;       TO A CONTROL BLOCK IN SYSTEM POOL WHICH CONTAINS THE
;       FOLLOWING CONTROL FIELDS AND THE MAPPING INFORMATION
;       FOR THE SECONDARY POOL WINDOW.
;
W.MAP:'L'.BLKW  1       ; ADDR TO THE MAPPING PTRS IN SECONDARY POOL
W.SLEN:'L'              ; Length of primary pool stub
;
; SECONDARY POOL WINDOW
;       IF SECONDARY POOL WINDOW BLOCKS ARE ENABLED, THE RETRIEVAL
;       POINTERS ARE MAINTAINED IN SECONDARY POOL IN THE FOLLOWING
;       FORMAT.
;
.=0
        ASSUME  W.CTL,0
        .BLKB   1       ; NUMBER OF ACTIVE MAPPING POINTERS
W.USE:'L'.BLKB  1       ; STATUS OF BLOCK
W.VBN:'L'.BLKB  1       ; HIGH BYTE OF 1ST VBN MAPPED BY WINDOW
W.WISZ:'L'.BLKB 1       ; SIZE IN RTRV PTRS OF WINDOW (7 BITS)
        .BLKW   1       ; LOW ORDER WORD OF 1ST VBN MAPPED
W.RTRV:'L'              ; OFFSET TO 1ST RETRIEVAL POINTER IN WINDOW
        .ENDC           ; END SECONDARY POOL WINDOW CONDITIONAL
        .ENDC           ; END SYSDEF CONDITIONAL
;
; LOCKED BLOCK LIST NODE
;
        .ASECT
.=0
L.LNK:'L'.BLKW  1       ; LINK TO NEXT NODE IN LIST
L.WI1:'L'.BLKW  1       ; POINTER TO WINDOW FOR FIRST ENTRY
L.VB1:'L'.BLKB  1       ; HIGH ORDER VBN BYTE
L.CNT:'L'.BLKB  1       ; COUNT FOR ENTRY
        .BLKW   1       ; LOW ORDER VBN
L.LKSZ:'L'
;
; END OF DEFINITIONS
;
        .PSECT
        .MACRO  F11DF$  X,Y,Z
        .ENDM   F11DF$
        .ENDM   F11DF$
```

HDRDF$

```
        .MACRO  HDRDF$,L,B

;+
; TASK HEADER OFFSET DEFINITIONS
;-

        .ASECT
.=0
H.CSP:'L'.BLKW  1               ;CURRENT STACK POINTER
H.HDLN:'L'.BLKW 1               ;HEADER LENGTH IN BYTES
H.SMAP:'L'.BLKB 1               ;SUPERVISOR D SPACE OVERMAP MASK
H.DMAP:'L'.BLKB 1               ;USER D SPACE OVERMAP MASK
H.FMAP:'L'.BLKW 1               ;POINTER TO FAST MAP SECTION OF HDR
H.CUIC:'L'.BLKW 1               ;CURRENT TASK UIC
H.DUIC:'L'.BLKW 1               ;DEFAULT TASK UIC
H.IPS:'L'.BLKW  1               ;INITIAL PROCESSOR STATUS WORD (PS)
H.IPC:'L'.BLKW  1               ;INITIAL PROGRAM COUNTER (PC)
H.ISP:'L'.BLKW  1               ;INITIAL STACK POINTER (SP)
H.ODVA:'L'.BLKW 1               ;ODT SST VECTOR ADDRESS
H.ODVL:'L'.BLKW 1               ;ODT SST VECTOR LENGTH
H.TKVA:'L'.BLKW 1               ;TASK SST VECTOR ADDRESS
H.TKVL:'L'.BLKW 1               ;TASK SST VECTOR LENGTH
H.PFVA:'L'.BLKW 1               ;POWER FAIL AST CONTROL BLOCK ADDRESS
H.FPVA:'L'.BLKW 1               ;FLOATING POINT AST CONTROL BLOCK ADDRESS
H.RCVA:'L'.BLKW 1               ;RECIEVE AST CONTROL BLOCK ADDRESS
H.EFSV:'L'.BLKW 1               ;EVENT FLAG ADDRESS SAVE ADDRESS
H.FPSA:'L'.BLKW 1               ;POINTER TO FLOATING POINT/EAE SAVE AREA
H.WND:'L'.BLKW  1               ;POINTER TO NUMBER OF WINDOW BLOCKS
H.DSW:'L'.BLKW  1               ;TASK DIRECTIVE STATUS WORD
H.FCS:'L'.BLKW  1               ;FCS IMPURE POINTER
H.FORT:'L'.BLKW 1               ;FORTRAN IMPURE POINTER
H.OVLY:'L'.BLKW 1               ;OVERLAY IMPURE POINTER
H.VEXT:'L'.BLKW 1               ;WORK AREA EXTENSION VECTOR POINTER
H.SPRI:'L'.BLKB 1               ;PRIORITY DIFFERENCE FOR SWAPPING
H.NML:'L'.BLKB  1               ;NETWORK MAILBOX LUN
H.RRVA:'L'.BLKW 1               ;RECEIVE BY REFERENCE AST CONTROL BLOCK ADDRESS
H.X25:'L'.BLKB  1               ;FOR USE BY X25 SOFTWARE
        .BLKB   1               ;3 RESERVED BYTES
        .BLKW   1               ;
H.LUTE:'L'.BLKW 1               ;POINTER TO LUT EXTENSION OF HEADER
H.GARD:'L'.BLKW 1               ;POINTER TO HEADER GUARD WORD
H.NLUN:'L'.BLKW 1               ;NUMBER OF LUN'S
H.LUN:'L'.BLKW  2               ;START OF LOGICAL UNIT TABLE

;+
; LENGTH OF FLOATING POINT SAVE AREA
;-

H.FPSL='B'25.*2                 ;

;+
; WINDOW BLOCK OFFSETS
;-
```

```
.=0
W.BPCB:'L'.BLKW 1               ;PARTITION CONTROL BLOCK ADDRESS
W.BLVR:'L'.BLKW 1               ;LOW VIRTUAL ADDRESS LIMIT
W.BHVR:'L'.BLKW 1               ;HIGH VIRTUAL ADDRESS LIMIT
W.BATT:'L'.BLKW 1               ;ADDRESS OF ATTACHMENT DESCRIPTOR
W.BSIZ:'L'.BLKW 1               ;SIZE OF WINDOW IN 32W BLOCKS
W.BOFF:'L'.BLKW 1               ;PHYSICAL MEMORY OFFSET IN 32W BLOCKS
W.BFPD:'L'.BLKB 1               ;FIRST PDR ADDRESS
W.BNPD:'L'.BLKB 1               ;NUMBER OF PDR'S TO MAP
W.BLPD:'L'.BLKW 1               ;CONTENTS OF LAST PDR
W.BLGH:'L'                      ;LENGTH OF WINDOW DESCRIPTOR

;
; BIT DEFINITION FOR W.BLPD
;

WB.NBP='B'20                    ;CACHE BYPASS IS NOT DESIRED FOR THIS WINDOW
WB.BPS='B'40                    ;ALWAYS BYPASS THE CACHE FOR THIS WINDOW

        .PSECT

        .MACRO  HDRDF$  X,Y
        .ENDM
        .ENDM
```

HWDDF$

```
        .MACRO HWDDF$,L,B,SYSDEF

;+
; MACROS FOR DEFINING MAPPING REGISTER DEFINITIONS
;-

        .MACRO  CRESET  NAM,ADDR
$$$=0
        .REPT   8.
        CRENAM  NAM,ADDR+<$$$*2>,\$$$
$$$=$$$+1
        .ENDR
        .ENDM

        .MACRO  CRENAM  NAM,ADDR,N
'NAM''N'==ADDR
        .ENDM

;+
; HARDWARE REGISTER ADDRESSES AND STATUS CODES
;-

MPCSR='B'177746                 ;ADDRESS OF PDP-11/70 MEMORY PARITY REGISTER
MPAR='B'172100                  ;ADDRESS OF FIRST MEMORY PARITY REGISTER
PIRQ='B'177772                  ;PROGRAMMED INTERRUPT REQUEST REGISTER
PR0='B'0                        ;PROCESSOR PRIORITY 0
PR1='B'40                       ;PROCESSOR PRIORITY 1
PR4='B'200                      ;PROCESSOR PRIORITY 4
PR5='B'240                      ;PROCESSOR PRIORITY 5
PR6='B'300                      ;PROCESSOR PRIORITY 6
PR7='B'340                      ;PROCESSOR PRIORITY 7
PS='B'177776                    ;PROCESSOR STATUS WORD
SWR='B'177570                   ;CONSOLE SWITCH AND DISPLAY REGISTER
TPS='B'177564                   ;CONSOLE TERMINAL PRINTER STATUS REGISTER
KXCSRA='B'177520                ;KXJ CSR A                                      ; KLNO23
KXCSRD='B'177530                ;KXJ CSR D                                      ; KLNO23
KXJQIR='B'177532                ;KXJ QIR REGISTER                               ; KLNO23
KXCSRF='B'177534                ;KXJ CSR F                                      ; KLNO23
KXCSRH='B'177536                ;KXJ CSR H                                      ; KLNO23

;+
; EXTENDED ARITHMETIC ELEMENT REGISTERS
;-

        .IF DF  E$$EAE

AC='B'177302                    ;ACCUMULATOR
MQ='B'177304                    ;MULTIPLIER-QUOTIENT
SC='B'177310                    ;SHIFT COUNT

        .ENDC

;+
; MEMORY MANAGEMENT HARDWARE REGISTERS AND STATUS CODES
;-

        .IF NB  B
```

```
        CRESET  KINAR,172340    ;KERNEL I PAR'S
        CRESET  KINDR,172300    ;KERNEL I PDR'S
        CRESET  KDSAR,172360    ;KERNEL D PAR'S
        CRESET  KDSDR,172320    ;KERNEL D PDR'S
        CRESET  SISAR,172240    ;SUPERVISOR I PAR'S
        CRESET  SISDR,172200    ;SUPERVISOR I PDR'S
        CRESET  SDSAR,172260    ;SUPERVISOR D PAR'S
        CRESET  SDSDR,172220    ;SUPERVISOR D PDR'S
        CRESET  UINAR,177640    ;USER I PAR'S
        CRESET  UINDR,177600    ;USER I PDR'S
        CRESET  UDSAR,177660    ;USER D PAR'S
        CRESET  UDSDR,177620    ;USER D PDR'S

        .ENDC

        .IF NB  SYSDEF

        .IF DF  K$$DAS

        CRESET  KISAR,172360    ;KERNEL D PAR'S
        CRESET  KISDR,172320    ;KERNEL D PDR'S

        .IFF

        CRESET  KISAR,172340    ;KERNEL I PAR'S
        CRESET  KISDR,172300    ;KERNEL I PDR'S

        .ENDC

        .IF DF  U$$DAS

        CRESET  UISAR,177660    ;USER D PAR'S
        CRESET  UISDR,177620    ;USER D PDR'S

        .IFF    ; DF U$$DAS

        CRESET  UISAR,177640    ;USER I PAR'S
        CRESET  UISDR,177600    ;USER I PDR'S

        .ENDC   ; DF U$$DAS

        .ENDC

UBMPR='B'170200                 ;UNIBUS MAPPING REGISTER 0
CMODE='B'140000                 ;CURRENT MODE FIELD OF PS WORD
PMODE='B'30000                  ;PREVIOUS MODE FIELD OF PS WORD
CSMODE='B'40000                 ;CURRENT MODE = SUPERVISOR PS WORD BITS
PSMODE='B'10000                 ;PREVIOUS MODE = SUPERVISOR PS WORD BITS
SR0='B'177572                   ;SEGMENT STATUS REGISTER 0
SR3='B'172516                   ;SEGMENT STATUS REGISTER 3
CPUERR='B'177766                ;CPU ERROR REGISTER
MEMERR='B'177744                ;MEMORY SYSTEM ERROR REGISTER
MEMCTL='B'177746                ;MEMORY CONTROL REGISTER

;+
; DEFINE THE LOCATIONS USED IN THE NON-VOLATIL RAM (NVR)
; FOR XT SYSTEMS
;-
```

HWDDF$

```
N.KEY='B'173054                 ;NUMBER OF KEYS PRESSED
N.UPT='B'173064                 ;UPTIME IN MINUTES
N.DZA='B'173074                 ;NUMBER OF I/OS DONE ON THE DZ
N.DWA='B'173104                 ;NUMBER OF I/OS DONE ON THE DW
N.DAY='B'173114                 ;DATE THAT THE NVR WAS LAST INITIALIZED
N.MON='B'173116                 ;...
N.YEA='B'173120                 ;...

;+
; FEATURE SYMBOL DEFINITIONS
;-

FE.EXT='B'1                     ;22-BIT EXTENDED MEMORY SUPPORT
FE.MUP='B'2                     ;MULTI-USER PROTECTION SUPPORT
FE.EXV='B'4                     ;EXECUTIVE IS SUPPORTED TO 20K
FE.DRV='B'10                    ;LOADABLE DRIVER SUPPORT
FE.PLA='B'20                    ;PLAS SUPPORT
FE.CAL='B'40                    ;DYNAMIC CHECKPOINT SPACE ALLOCATION
FE.PKT='B'100                   ;PREALLOCATION OF I/O PACKETS
FE.EXP='B'200                   ;EXTEND TASK DIRECTIVE SUPPORTED
FE.LSI='B'400                   ;PROCESSOR IS AN LSI-11
FE.OFF='B'1000                  ;PARENT/OFFSPRING TASKING SUPPORTED
FE.FDT='B'2000                  ;FULL DUPLEX TERMINAL DRIVER SUPPORTED
FE.X25='B'4000                  ;X.25 CEX IS LOADED
FE.DYM='B'10000                 ;DYNAMIC MEMORY ALLOCATION SUPPORTED
FE.CEX='B'20000                 ;COM EXEC IS LOADED
FE.MXT='B'40000                 ;MCR EXIT AFTER EACH COMMAND MODE
FE.NLG='B'100000                ;LOGINS DISABLED - MULTI-USER SUPPORT

;+
; FEATURE MASK DEFINITIONS (SECOND WORD)
;-

F2.DAS='B'1                     ;KERNEL DATA SPACE SUPPORTED
F2.LIB='B'2                     ;SUPERVISOR MODE LIBRARIES SUPPORTED
F2.MP='B'4                      ;SYSTEM SUPPORTS MULTIPROCESSING
F2.EVT='B'10                    ;SYSTEM SUPPORTS EVENT TRACE FEATURE
F2.ACN='B'20                    ;SYSTEM SUPPORTS CPU ACCOUNTING
F2.SDW='B'40                    ;SYSTEM SUPPORTS SHADOW RECORDING
F2.POL='B'100                   ;SYSTEM SUPPORTS SECONDARY POOLS
F2.WND='B'200                   ;SYSTEM SUPPORTS SECONDARY POOL FILE WINDOWS
F2.DPR='B'400                   ;SYSTEM HAS A SEPARATE DIRECTIVE PARTITION
F2.IRR='B'1000                  ;INSTALL, RUN, AND REMOVE SUPPORT
F2.GGF='B'2000                  ;GROUP GLOBAL EVENT FLAG SUPPORT
F2.RAS='B'4000                  ;RECEIVE/SEND DATA PACKET SUPPORT
F2.AHR='B'10000                 ;ALT. HEADER REFRESH AREA SUPPORT
F2.RBN='B'20000                 ;ROUND ROBIN SCHEDULING SUPPORT
F2.SWP='B'40000                 ;EXECUTIVE LEVEL DISK SWAPPING SUPPORT
F2.STP='B'100000                ;EVENT FLAG MASK IS IN THE TCB(1=YES)

;+
; THIRD FEATURE MASK SYMBOL DEFINITIONS
;-
```

```
F3.CRA='B'1                     ;SYSTEM SPONTANEOUSLY CRASHED (1=YES)
F3.XCR='B'2                     ;SYSTEM CRASHED FROM XDT (1=YES)
F3.EIS='B'4                     ;SYSTEM REQUIRES EXTENDED INSTRUCTION SET
F3.STM='B'10                    ;SYSTEM HAS SET SYSTEM TIME DIRECTIVE
F3.UDS='B'20                    ;SYSTEM SUPPORTS USER DATA SPACE
F3.PRO='B'40                    ;SYSTEM SUPPORTS SEC. POOL PROTO TCBS
F3.XHR='B'100                   ;SYSTEM SUPPORTS EXTERNAL TASK HEADERS
F3.AST='B'200                   ;SYSTEM HAS AST SUPPORT
F3.11S='B'400                   ;RSX-11S SYSTEM
F3.CLI='B'1000                  ;MULTIPLE CLI SUPPORT
F3.TCM='B'2000                  ;SYSTEM HAS SEPARATE TERMINAL DRIVER POOL
F3.PMN='B'4000                  ;SYSTEM SUPPORTS POOL MONITORING
F3.WAT='B'10000                 ;SYSTEM HAS WATCHDOG TIMER SUPPORT
F3.RLK='B'20000                 ;SYSTEM SUPPORTS RMS RECORD LOCKING
F3.SHF='B'40000                 ;SYSTEM SUPPORTS SHUFFLER TASK

;+
; FOURTH FEATURE MASK BITS
;-

F4.CXD='B'1                     ;COMM EXEC IS DEALLOCATED (NON-I/D ONLY)
F4.XT='B'2                      ;SYSTEM IS AN XT SYSTEM (1=YES)
F4.ERL='B'4                     ;SYSTEM SUPPORTS ERROR LOGGING (1=YES)
F4.PTY='B'10                    ;SYSTEM SUPPORTS PARITY MEMORY (1=YES)
F4.DVN='B'20                    ;SYSTEM SUPPORTS DECIMAL VERSIONS (1=YES)
F4.LCD='B'40                    ;SYSTEM SUPPORTS LOADABLE CRASH (1=YES)
F4.NIM='B'100                   ;SYSTEM SUPPORTS DELETED TASK IMAGES (1=YES)
F4.CHE='B'200                   ;SYSTEM SUPPORTS DISK DATA CACHING (1=YES)
F4.LOG='B'400                   ;SYSTEM SUPPORTS LOGICAL NAMES (1=YES)
F4.NAM='B'1000                  ;SYSTEM SUPPORTS NAMED DIRECTORIES (1=YES)
F4.FMP='B'2000                  ;SYSTEM SUPPORTS FAST MAP DIRECTIVE
F4.DCL='B'4000                  ;DCL IS DEFAULT CLI (1=YES)
F4.DDS='B'10000                 ;NAMED DIRECTORY MODE IS THE DEFAULT (1=YES)
F4.ACD='B'20000                 ;SYSTEM SUPPORTS ACD'S (1=YES)
F4.NCT='B'40000                 ;SYSTEM HAS NCT SUPPORT (1=YES)
F4.LSD='B'100000                ;SYSTEM HAS LUT SCAN DISABLED
                                                                        ; KLN011
;+                                                                      ; KLN011
; FIFTH FEATURE MASK BITS                                               ; KLN011
;-                                                                      ; KLN011
                                                                        ; KLN011
F5.PRO='B'1                     ;SYSTEM SUPPORTS PROFESSIONAL 3XX SERIES
F5.DFB='B'2                     ;SYSTEM HAS DEFERRED BINDING            ; KLN011
F5.RTB='B'4                     ;RUN TIME BINDING                       ; KLN011
F5.ODB='B'10                    ;USER CAN OVERRRIDE DEFERRED BINDING    ; KLN011
F5.XDJ='B'20                    ;XDT IS USING KXJ INTERFACE FOR I/O     ; KLN018
F5.NSY='B'40                    ;NO LOCAL SYSTEM DISK                   ; KLN011
F5.NCO='B'100                   ;NO LOCAL CONSOLE                       ; KLN011
F5.RTK='B'200                   ;REMOTE TASK SERVICES                   ; KLN011
F5.RDR='B'400                   ;REMOTE DIRECTORY STORAGE               ; KLN011
F5.RLG='B'1000                  ;REMOTE LOGICAL SUPPORT                 ; KLN011

;+
; HARDWARE FEATURE MASK BIT DEFINITIONS
;
;       HF.CIS,HF.FPP DEFINED AS SIGN BITS FOR RUN TIME SPEED
;-
```

HWDDF$

```
HF.UBM='B'1                     ;PROCESSOR HAS A UNIBUS MAP (1=YES)
HF.EIS='B'2                     ;PROCESSOR HAS EXTENDED INSTRUCION SET
HF.QB='B'4                      ;SYSTEM HAS A QBUS (1=YES)
HF.DSP='B'10                    ;HARDWARE SUPPORTS DATA SPACE
HF.SWR='B'20                    ;SYSTEM HAS SWITCH REGISTER HARDWARE PRESENT    ;LBM050
HF.KXJ='B'40                    ;PROCESSOR TYPE IS KXJ11                        ; KLN011
HF.RMT='B'100                   ;REMOTE SERVICES REQUIRED                       ; KLN011
HF.CIS='B'200                   ;PROCESSOR SUPPORTS COMMERCIAL INSTRUCTION SET
HF.FPP='B'100000                ;(1=PROC. HAS NO FLOATING POINT UNIT)

;+
; SECOND HARDWARE FEATURE MASK BIT DEFINITIONS
; THIS WORD IS RESERVED FOR XT HARDWARE FEATURES
;-

H2.NVR='B'1                     ;XT NON-VOLATILE RAM PRESENT (1=YES)
H2.INV='B'2                     ;NON-VOLATILE RAM IS INVALID (1=YES)
H2.CLK='B'4                     ;XT CLOCK IS PRESENT (1=YES)
H2.ITF='B'10                    ;INVALID TIME FORMAT IN NON-VOLATILE RAM (1=YES)
H2.PRO='B'20                    ;RUNNING ON PRO/3XX HARDWARE
H2.BRG='B'100000                ;XT BRIDGE MODULE PRESENT (1=YES)

;+
; SYSGEN FEATURE SELECTIONS MASK. THIS IS INTENDED TO RECORD IN A
; BIT MASK THE CHOICES THE USER HAS MADE AT SYSGEN TIME. FEATURES WILL
; BE LISTED HERE WHEN THEY ARE BEING RECORDED FOR OUR INFORMATIONAL
; PURPOSES ONLY. THEY CANNOT BE TESTED LIKE BITS IN THE FEATURE MASK
; SINCE THIS ONLY EXISTS IN THE RSX11M.STB FILE. NO BITS IN MEMORY
; ARE USED. THEY ARE ONLY INTENDED TO BE PRINTED FROM THE STB FILE BY CDA.
;-

SF.STD='B'1                     ;STANDARD EXEC SELECTED
SF.PGN='B'2                     ;SYSTEM WAS PRE-GENERATED (EX. RL02/RC25 SYSTEM)

;+
; MULTIPROCESSOR STATUS TABLE DEFINITIONS (TEMPORARY)
;-

MP.CRH='B'100000                ;CRASH PROCESSOR IMMEDIATELY
MP.PWF='B'40000                 ;POWERFAIL ON ONE CPU
MP.RSM='B'20000                 ;RESET INTERRUPT MASKS
MP.NOP='B'10000                 ;NOP FUNCTION FOR TRANSMISSION CHECK
MP.STP='B'4                     ;STOP PROCESSOR IN ORDERLY FASHION
MP.INT='B'7777                  ;BIC MASK FOR INTERRUPT LVL FUNCTIONS

        .MACRO  HWDDF$  X,Y,Z
        .ENDM
        .ENDM
```

```
        .MACRO  ITBDF$  L,B,SYSDEF
;
;+
; INTERRUPT TRANSFER BLOCK (ITB) OFFSET DEFINITIONS
;-

        .IF DF  A$$TRP

        .MCALL  PKTDF$
        PKTDF$                  ; DEFINE AST BLOCK OFFSETS

        .ENDC

        .ASECT
        .=0
X.LNK:'L' .BLKW 1               ; LINK WORD FOR ITB LIST STARTING IN TCB
X.JSR:'L' JSR   R5,@#0          ; CALL $INTSC
X.PSW:'L' .BLKB 1               ; LOW BYTE OF PSW FOR ISR
        .BLKB   1               ; UNUSED
X.ISR:'L' .BLKW 1               ; ISR ENTRY POINT (APR5 MAPPING)
X.FORK:'L'                      ; FORK BLOCK
        .BLKW   1               ; THREAD WORD
        .BLKW   1               ; FORK PC
        .BLKW   1               ; SAVED R5
        .BLKW   1               ; SAVED R4

        .IF DF  M$$MGE

X.REL:'L' .BLKW 1               ; RELOCATION BASE FOR APR5

        .ENDC

X.DSI:'L' .BLKW 1               ; ADDRESS OF DIS.INT. ROUTINE
X.TCB:'L' .BLKW 1               ; TCB ADDRESS OF OWNING TASK

        .IF NB  SYSDEF

        .IF DF  A$$TRP

        .BLKW   1               ; A.DQSR FOR AST BLOCK
X.AST:'L' .BLKB A.PRM           ; AST BLOCK

        .ENDC

X.VEC:'L' .BLKW 1               ; VECTOR ADDRESS (IF AST SUPPORT,
                                ; THIS IS FIRST AND ONLY AST PARAMETER)
X.VPC:'L' .BLKW 1               ; SAVED VECTOR PC
X.LEN:'L'                       ; LENGTH IN BYTES OF ITB

        .ENDC

        .PSECT
        .MACRO  ITBDF$  X,Y,Z
        .ENDM   ITBDF$
        .ENDM   ITBDF$
```

KRBDF$

```
        .MACRO  KRBDF$,L,B,SYSDEF

;+
; CONTROLLER REQUEST BLOCK (KRB)
;
; THE CONTROLLER REQUEST BLOCK DEFINES THE ENVIRONMENT OF A DEVICE
; CONTROLLER.  EXACTLY ONE KRB EXISTS FOR EVERY DEVICE CONTROLLER
; IN AN RSX-11M+ SYSTEM.  THE KRB CONTAINS CERTAIN DEVICE STATUS
; INCLUDING THE CSR AND VECTOR ADDRESS FOR THE CONTROLLER.
;-

        .ASECT
.=177770

K.PRM:'L' .BLKW 1               ;DEVICE DEPENDANT PARAMETER WORD
K.PRI:'L' .BLKB 1               ;CONTROLLER PRIORITY
K.VCT:'L' .BLKB 1               ;INTERRUPT VECTOR ADDRESS
K.CON:'L' .BLKB 1               ;CONTROLLER INDEX WITHIN THE SYSTEM
K.IOC:'L' .BLKB 1               ;CONTROLLER I/O COUNT
K.STS:'L' .BLKW 1               ;CONTROLLER STATUS
K.CSR:'L' .BLKW 1               ;ADDRESS OF CONTROL STATUS REGISTER
;
; NOTE: K.CSR MUST BE THE ZERO OFFSET!
;
K.OFF:'L' .BLKW 1               ;OFFSET TO UCB/UMR/RHBAE TABLE
K.HPU:'L' .BLKB 1               ;HIGHEST PHYSICAL UNIT NUMBER
        .BLKB   1               ;UNUSED BYTE
K.OWN:'L' .BLKW 1               ;OWNER OF CONTROLLER
K.CRQ:'L' .BLKW 2               ;CONTROLLER REQUEST QUEUE
K.URM:'L' .BLKW 1               ;CONTROLLER UNIBUS RUN MASK
K.FRK:'L' .BLKW 1               ;POSSIBLE KRB FORK BLOCK

;+
; OFFSETS FOR THE KRB EXTENSION REACHED BY ADDING (K.OFF) TO
; THE STARTING ADDRESS OF THE KRB.
;-

;
; DEFINE OFFSETS IN SCB/KRB FOR DISK MSCP CONTROLLERS
;

.=-20.

KE.UMH:'L'.BLKW 2               ;LIST HEAD FOR UMR WAITING ASSIGNMENT BLOCK(S)
KE.UMC:'L'.BLKW 1               ;COUNT OF AVAILABLE UMR WAITING ASSIGNMENT BLOCK(S)

.=177776

KE.RHB:'L'.BLKW 1               ;OFFSET TO RHBAE REGISTER (IF ANY)
;
; WHEN ONE ADDS (K.OFF) TO THE KRB ADDRESS, IT YIELDS AN ADDRESS
; WHICH POINTS TO HERE.
;
KE.UCB:'L'.BLKW 1               ;OFFSET TO UCB TABLE (IF KS.UCB SET)

        .PSECT

;+
; CONTROLLER REQUEST BLOCK (KRB) STATUS BIT DEFINITIONS
;-
```

```
KS.OFL='B'1                     ;CONTROLLER OFFLINE (1=YES)
KS.MOF='B'2                     ;CONTROLLER MARKED FOR OFFLINE (1=YES)
KS.UOP='B'4                     ;SUPPORTS OVERLAPPED OPERATION (1=YES)
KS.MBC='B'10                    ;DEVICE IS MASSBUS CONTROLLER (1=YES)
KS.SDX='B'20                    ;SEEKS ALLOWED DURING DATA XFERS (1=YES)
KS.POE='B'40                    ;PARALLEL OPERATION ENABLED (1=YES)
KS.UCB='B'100                   ;UCB TABLE PRESENT (1=YES)
KS.DIP='B'200                   ;DATA TRANSFER IN PROGRESS (1=YES)
KS.PDF='B'400                   ;PRIVILEGED DIAGNOSTIC FUNCTIONS ONLY (1=YES)
KS.EXT='B'1000                  ;EXTENDED 22-BIT UNIBUS CONTROLLER (1=YES)
KS.SLO='B'2000                  ;CONTROLLER IS SLOW COMING ONLINE (1=YES)

;+
; DEFINE THE CONTIGUOUS SCB OFFSETS
;-

        .ASECT
.=177762
S.PRI:'L' .BLKB 1               ;CONTROLLER PRIORITY
S.VCT:'L' .BLKB 1               ;INTERRUPT VECTOR ADDRESS
S.CON:'L' .BLKB 1               ;CONTROLLER INDEX
        .BLKB   1
        .BLKW   1
S.CSR:'L' .BLKW 1               ;CONTROL AND STATUS REGISTER
        .BLKW   1
        .BLKB   1
        .BLKB   1
S.OWN:'L' .BLKW 1               ;DISTRIBUTED CNTBL


;+
; SUBCONTROLLER REQUEST BLOCK (KRB1)
;
; THE SUBCONTROLLER REQUEST BLOCK DEFINES THE ENVIRONMENT OF A DEVICE
; SUBCONTROLLER.  EXACTLY ONE KRB1 EXISTS FOR EVERY DEVICE SUBCONTROLLER
; IN AN RSX-11M+ SYSTEM.
;-

        .ASECT

.=-4
K1.CON:'L'.BLKB 1               ;SUBCONTROLLER INDEX WITHIN THE SYSTEM
        .BLKB   1               ;UNUSED BYTE
K1.STS:'L'.BLKW 1               ;SUBCONTROLLER STATUS
K1.MAS:'L'.BLKW 1               ;UCB ADDRESS OF THE MASTER UNIT
;
; NOTE: K1.MAS MUST BE THE ZERO OFFSET
;
K1.OWN:'L'.BLKW 1               ;OWNER OF SUBCONTROLLER
K1.CRQ:'L'.BLKW 2               ;SUBCONTROLLER REQUEST QUEUE
K1.UCB:'L'                      ;START OF THE UCB TABLE (IF ANY)

        .PSECT

        .MACRO  KRBDF$,X,Y,Z
        .ENDM
        .ENDM
```

LCBDF$

```
        .MACRO  LCBDF$,L,B
;+
; LOGICAL ASSIGNMENT CONTROL BLOCK
;
; THE LOGICAL ASSIGNMENT CONTROL BLOCK (LCB) IS USED TO ASSOCIATE A
; LOGICAL NAME WITH A PHYSICAL DEVICE UNIT.  LCB'S ARE LINKED TOGETHER
; TO FORM THE LOGICAL ASSIGNMENTS OF A SYSTEM.  ASSIGNMENTS MAY BE ON
; A SYSTEM WIDE OR LOCAL (TERMINAL) BASIS.
;-
        .ASECT
.=0
L.LNK:'L' .BLKW 1               ;LINK TO NEXT LCB
L.NAM:'L' .BLKW 1               ;LOGICAL NAME OF DEVICE
L.UNIT:'L' .BLKB 1              ;LOGICAL UNIT NUMBER
L.TYPE:'L' .BLKB 1              ;TYPE OF ENTRY (0=SYSTEM WIDE)
L.UCB:'L' .BLKW 1               ;TI UCB ADDRESS
L.ASG:'L' .BLKW 1               ;ASSIGNMENT UCB ADDRESS
L.LGTH='B'.-L.LNK               ;LENGTH OF LCB
        .PSECT

        .MACRO  LCBDF$,X,Y
        .ENDM
        .ENDM
```

```
        .MACRO  LNMDF$,L,B,SYSDEF
        .SAVE
;+
;       LOGICAL NAME BLOCK (LNB)
;-
        .ASECT
.=0
L.NLNK:'L'      .BLKW   1       ;LINK WORD
L.NTBL:'L'      .BLKB   1       ;LOGICAL TABLE NUMBER
L.NBLK:'L'      .BLKB   1       ;LOGICAL NAME BLOCK TYPE
L.NSTS:'L'      .BLKB   1       ;STATUS WORD
L.NGRP:'L'      .BLKB   1       ;CREATOR GROUP NUMBER (GROUP TABLE ONLY)
L.NUCB:'L'
L.NTCB:'L'      .BLKW   1       ;CREATOR UCB (USER) OR TCB (TASK)
L.NLNS:'L'      .BLKB   1       ;SIZE OF LOGICAL NAME STRING
L.NENS:'L'      .BLKB   1       ;LENGTH OF EQUIVALENCE NAME STRING
L.NNAM:'L'                      ;VARIABLE LENGTH LOGICAL NAME STRING
L.NHSZ='B'.                     ;SIZE OF LOGICAL NAME BLOCK HEADER
;+
;  TABLE NUMBER DEFINITIONS
;-
LT.SYS='B'0                     ;SYSTEM WIDE LOGICAL NAME
LT.GRP='B'1                     ;GROUP LOGICAL NAME (UIC GROUP NUMBER)
LT.USR='B'2                     ;USER SPECIFIC LOGICAL NAME (P/OS)
LT.TSK='B'3                     ;TASK SPECIFIC LOGICAL NAME
LT.SES='B'4                     ;SESSION WIDE LOGICAL NAME (RSX LOCAL/LOGIN)
LT.INV='B'6                     ;TASK/PROCESS INVOCATION LOGICAL NAME
LT.APP='B'5                     ;APPLICATION WIDE LOGICAL NAME
;+
; STATUS BIT DEFINITIONS
;-
LS.TRM='B'1                     ;LOGICAL HAS A TERMINAL STATUS
LS.PRV='B'2                     ;USER MUST BE PRIVILEGED TO DELETE LOGICAL
;+
; STANDARD BLOCK TYPE DEFINITIONS
;-
LB.LOC='B'1                     ;BLOCK NUMBER FOR A STANDARD LOGICAL CREATION
LB.LOG='B'2                     ;SPECIAL BLOCK NUMBER FOR LOGIN LOGICALS
;+
; INHIBIT MASK BIT DEFINITIONS
;-
IN.TSK='B'1                     ;INHIBIT THE TASK LOGICAL TABLE
IN.SES='B'20                    ;INHIBIT THE SESSION LOGICAL TABLE
IN.GRP='B'4                     ;INHIBIT THE GROUP LOGICAL TABLE
IN.SYS='B'10                    ;INHIBIT THE SYSTEM LOGICAL TABLE

;+
;  CONTEXT BLOCK OFFSETS FOR DDS STORAGE
;-
.=0
C.REF:'L'       .BLKB   1       ;REFERENCE COUNT
C.DDSL:'L'      .BLKB   1       ;LENGTH OF DDS
C.CUIC:'L'      .BLKW   1       ;UIC FOR CLOCK BLOCK PROPAGATION
C.FIXL='B'                      ;FIXED LENGTH OF CONTEXT BLOCK
C.DDS:'L'
```

LNMDF$

```
;+
; Definitions of the offsets for the parse block.
;
;       NOTE:
;
;       These definitions are the same as those in DRPFN.MAC.
;       Any change here must be reflected there.
;-
.=0
O$STAT:'L'      .BLKW   1               ;Status value
O$FLAG:'L'      .BLKW   1               ;Output flags
O$NODS:'L'      .BLKW   1               ;Node spec size
O$NODA:'L'      .BLKW   1               ;Node spec address
O$DEVS:'L'      .BLKW   1               ;Device spec size
O$DEVA:'L'      .BLKW   1               ;Device spec address
O$DIRS:'L'      .BLKW   1               ;Dir spec size
O$DIRA:'L'      .BLKW   1               ;Dir spec address
O$NAMS:'L'      .BLKW   1               ;Nam spec size
O$NAMA:'L'      .BLKW   1               ;Nam spec address
O$TYPS:'L'      .BLKW   1               ;Typ spec size
O$TYPA:'L'      .BLKW   1               ;Typ spec address
O$VERS:'L'      .BLKW   1               ;Version spec size
O$VERA:'L'      .BLKW   1               ;Version spec address
O$TRLS:'L'      .BLKW   1               ;Trailing portion size
O$TRLA:'L'      .BLKW   1               ;Trailing portion address
O$ACCS:'L'      .BLKW   1               ;Access spec size
O$ACCA:'L'      .BLKW   1               ;Access spec address
O$LTYP:'L'      .BLKB   1               ;Logical type
                .BLKB   1               ;Reserved
O$PLEN='B'.                             ;Length of parse block

;+
; Definitions of the values for the parse block status field O$STAT.
;
;       NOTE:
;
;       These definitions are the same as those in RMSLIB.MLB.
;       Any change here must be reflected there.
;-
SU$SUC  ='B'1                   ;Operation successful
ER$NOD  ='B'175550              ;Error in node name
ER$DEV  ='B'177100              ;Bad device, or inappropriate device typ
ER$DIR  ='B'177060              ;Error in directory name
ER$FNM  ='B'176420              ;Error in file name
ER$TYP  ='B'174540              ;Error in file type extension
ER$VER  ='B'174460              ;Error in version number
ER$ESS  ='B'176640              ;Expanded string area too short
ER$XTR  ='B'174340              ;Extraneous field detected during parse
ER$BEQ  ='B'177510              ;Illegal logical expansion
ER$TRN  ='B'174550              ;Logical exceeded recursion count
ER$FTB  ='B'176364              ;bad size
```

```
;+
; Definitions of the flag bits in the parse block flags field O$FLAG.
;
;       NOTE:
;
;       These definitions are the same as those in RMSLIB.MLB.
;       Any change here must be reflected there.
;-
FS$NOD  ='B'400                 ;Node present
FS$DEV  ='B'200                 ;Device present
FS$DIR  ='B'100                 ;Directory present
FS$QUO  ='B'2000                ;Quoted filename present
FS$NAM  ='B'4                   ;Filename present
FS$TYP  ='B'2                   ;File type present
FS$VER  ='B'1                   ;File version present
FS$WCH  ='B'4000                ;Wildcard character present
FS$WDI  ='B'1000                ;Wild directory
FS$WNA  ='B'40                  ;Wild filename
FS$WTY  ='B'20                  ;Wild file type
FS$WVE  ='B'10                  ;Wild file version
FS$NDF  ='B'10000               ;Suppress system defaulting

;+
; Definitions of the values for the parse block logical type field O$LTYP.
;
;       NOTE:
;
;       These definitions are the same as those in DRPFN.MAC.
;       Any change here must be reflected there.
;-
P.LNON  ='B'0           ;No logical name present
P.LNAM  ='B'1           ;The filename may be a logical name
P.LDEV  ='B'2           ;The device name may be a logical name
P.LNOD  ='B'3           ;The node specification may be a logical name

        .RESTORE
        .MACRO  LNMDF$  X,Y,Z
        .ENDM
        .ENDM
```

MTADF$

```
        .MACRO  MTADF$,L,B
        .ASECT
;
; ANSI MAGTAPE SPECIFIC DATA STRUCTURES
;
; VOLUME SET CONTROL BLOCK OFFSET DEFININTIONS (VSCB)
;
; VOLUME SET AND PROCESS CONTROL SECTION
;
.=0
V.TCNT:'L' .BLKW 1              ;TRANSACTION COUNT
V.TYPE:'L' .BLKB 1              ;VOLUME TYPE DESCRIPTOR
V.VCHA:'L' .BLKB 1              ;VOLUME CHARACTERISTICS
V.LABL:'L' .BLKB 12.            ;FILE SET ID (FIRST SIX BYTES)
V.NXT:'L' .BLKW 1               ;PTR TO NEXT VSCB NODE
V.MVL:'L' .BLKW 1               ;PTR TO MOUNTED VOL LIST
V.UVL:'L' .BLKW 1               ;PTR TO UNMOUNTED VOL LIST
V.ATL:'L' .BLKW 1               ;ATL ADDR OF ACCESSING TASK
                                ; TCB IN RSX11M
V.UCB:'L' .BLKW 1               ;ADDR OF CURRENT UCB OR PUD
V.RVOL:'L' .BLKB 1              ;CURRENT RELATIVE VOL #
V.MOU:'L' .BLKB 1               ;MOUNT MODE BYTE
V.TCHR:'L' .BLKW 1              ;UINT CHAR. FOR ALL UNITS USED FOR VOL SET
V.SEQN:'L' .BLKW 1              ;CURRENT FILE SEQUENCE #
V.SECN:'L' .BLKW 1              ;CURRENT FILE SECTION #
V.TPOS:'L' .BLKB 1              ;POSITION OF TAPE IN TM'S TO NXT HDR1
V.PSTA:'L' .BLKB 1              ;PROCESS STATUS BYTE
V.TIMO:'L' .BLKW 1              ;BLOCKED PROCESS TIMEOUT COUNTER
V.STAT:'L' .BLKW        3       ; STATUS WORDS USED BY COMMAND
                                ;EXECUTION MODULES
V.TRTB:'L' .BLKB 1              ;TRANSLATION CONTROL BYTE
V.EFTV:'L' .BLKB 1              ;FOR MAG TO RETURN IE.EOF, EOT, EOV
;
; LABEL DATA SECTION
;
V.BLKL:'L' .BLKW 1              ;BLOCK LENGTH
V.RECL:'L' .BLKW 1              ;RECORD LENGTH
V.FNAM:'L' .BLKW 3              ;FILE NAME
V.FTYP:'L' .BLKW 1              ;FILE TYPE
V.FVER:'L' .BLKW 1              ;FILE VERSION #
V.CDAT:'L' .BLKW 2              ;CREATION DATE
V.EDAT:'L' .BLKW 2              ;EXPRIATION DATE
V.BLKC:'L' .BLKW 2              ;BLOCK COUNT FOR FILE SECTION
V.RTYP:'L' .BLKB 1              ;RECORD TYPE
V.FATT:'L' .BLKB 1              ;FILE ATTRIBUTES FOR CARRIAGE CONTROL
           .BLKB 30.            ;REMAINDER OF FILE ATTRIBUTES
;
; NULL WINDOW SECTION
;
V.WIND:'L' .BLKW  4.            ; NULL WINDOW
       ;*****
V.MST2:'L' .BLKW  1             ;MAGTAPE STATUS BITS
V.FABY:'L' .BLKB  1             ;FILE ACCESSIBILITY BYTE (HDR1)
           .BLKB  1             ;SPARE
V.ANSN:'L' .BLKB 17.            ;ANSI 17 CHARACTER FILE NAME
V.BOFF:'L' .BLKB  1.            ;BUFFER OFFSET
V.DENS:'L' .BLKB  1.            ;REQUESTED UNIT DENSITY
V.DRAT:'L' .BLKB  1.            ;DEFAULT RECORD ATTRIBUTES
V.DBLK:'L' .BLKW  1.            ;DEFAULT BLOCK SIZE
```

```
V.DREC:'L' .BLKW  1.            ;DEFAULT RECORD SIZE
        ;*****
S.VSCB='B'.                     ;SIZE OF VSCB
        .PSECT
;
; DEFINE OFFSETS INTO NULL WINDOW SECTION
;
        .ASECT
.=0
        W.CTL:'L'.BLKW 1        ;CONTROL WORD IN WINDOW
        V.WINC='B'V.WIND+W.CTL  ; CNTRL WORD IN NULL WINDOW
        .PSECT
                                ; RELATIVE TO THE VSCB
;
; MOUNTED VOLUME LIST OFFSET DEFININTIONS (MVL)
;
        .ASECT
.=0
        .IF     DF      R$$11M
M.NXT:'L' .BLKW 1               ;PTR TO NXT MVL NODE (11M)
        .ENDC
M.UIC:'L' .BLKW 1               ;OWNER UIC FROM RVOL #1
M.CH:'L' .BLKW 1                ; U.CH/U.VP (11D)
M.PROT:'L' .BLKW 1              ;PROTECTION U.AR IN 11D
        .IF     NDF     R$$11M
                .BLKW 2         ; ACP WORDS 11D
M.NXT:'L' .BLKW 1               ;PTR TO NEXT MVL NODE (11D)
        .ENDC
M.RVOL:'L' .BLKB 1              ;RELATIVE VOL # OF MOUNTED VOLUME
M.STAT:'L' .BLKB 1              ;VOLUME STATUS
M.VIDP:'L' .BLKW 1              ;VOLUME ID POINTER
M.UCB:'L' .BLKW 1               ;ADDR OF ASSOC UCB OR PUD
S.MVL='B'.                              ;SIZE OF MVL NODE
        .PSECT
;
; UNMOUNTED VOLUME AND VOLUME  LIST OFFSET DEFINITIONS (UVL)
;
        .ASECT
.=0
L.NXT:'L' .BLKW 1               ;PTR TO NXT UVL NODE
L.VOL1:'L' .BLKB 1              ;REL VOL # OF 1'ST VOL IN NODE
L.VOL2:'L' .BLKB 1              ;REL VOL # OF 2'ND VOL IN NODE
L.VID1:'L' .BLKB 6              ;VOL ID OF 1'ST VOL IN NODE
L.VID2:'L' .BLKB 6              ;VOL ID OF 2'ND VOL IN NODE
S.UVL='B'.                      ;SIZE OF UVL NODE
        .PSECT
;
;
; SYSTEM DATA STRUCTURE CONTENT VALUES
;

; VSCB VALUES

; V.MOU VALUES
VM.OLD  ='B'    200             ;OLD .FL300 VOLUME -- VM.BYP WILL ALSO BE SET
VM.BYP  ='B'    100             ;BYPASS LABEL PROCESSING
VM.ULB  ='B'    40              ;UNLABELED TAPE
VM.FSC  ='B'    20              ;OVERRIDE FILE SET ID CHECK
VM.EXC  ='B'    10              ;OVERRIDE EXPRIATION DATE CHECK
```

MTADF$

```
; V.MST2 VALUES

V2.INI  ='B'    1               ;MAG WANTS US TO INITIALIZE NEXT OUTPUT
V2.XH2  ='B'    2               ;THIS FILE HAS NO HDR2, DON'T WRITE EOF2
V2.XH3  ='B'    4               ;THIS FILE HAS NO HDR3, DON'T WRITE EOF3
V2.NH3  ='B'    10              ;DON'T WRITE HDR3/EOX3 LABELS
V2.OAC  ='B'    20              ;OVERRIDE FILE/VOLUME ACCESIBILITY

; V.PSTA VALUES - UNBLOCKED TRANSITION STATE

VP.RM   ='B'    2               ;READ DATA MODE
VP.WM   ='B'    4               ;WRITE DATA MODE
VP.UCM  ='B'    6               ;UNLABELLED CREATE POSITIONING MODE
VP.SM   ='B'    10              ;SEARCH MODE
VP.MOU  ='B'    20              ;MOUNT MODE
VP.RWD  ='B'    40              ;REWIND OR VOL VERIFICATION WAIT
VP.VFY  ='B'    VP.RWD
VP.POS  ='B'    100             ;PROCESS IN POSITIONING MODE (MULTI-SECTION FILE)
;
; BLOCKED STATE = -(UNBLOCKED TRANSITION STATE VALUES)
;
;PROCESS TIMED OUT BIT 0 = 1
        VP.TO='B'1
;
;
; NULL WINDOW CONTROL BIT DEFINITIONS
;
WI.RDV  ='B'    400             ;ACCESSED FOR READ
WI.WRV  ='B'    1000            ;ACCESSED FOR WRITE
WI.EXT  ='B'    2000            ;ACCESSED FOR EXTEND
WI.LCK  ='B'    4000            ;LOCKED
;

;
; MVL VALUES IN THE M.STAT FIELD
;
MS.VER  ='B'    200             ;VOL ID NOT VERIFIED
MS.RID  ='B'    1               ;VOL ID TO BE READ NOT CHECKED
MS.NMO  ='B'    2               ;MOUNT MESSAGE NOT GIVEN YET
MS.TMO  ='B'    4               ;ONE TIMEOUT ALREADY EXPRIED
MS.EXP  ='B'    10              ;EXPIRATION DATE MESSAGE GIVEN
;
;
; MISC BITS USED IN MOUNT (STORED IN V.STS)
;
MO.OVR  ='B'    1               ; OVER RIDE VOL NAME SWITCH
MO.UIC  ='B'    2               ; EXPLICIT UIC GIVEN
MO.PRO  ='B'    4               ; EXPLICIT PROTECTION GIVEN
MO.160  ='B'    10              ; 1600 BPI SPECIFIED
;
        .ENDM
```

```
        .MACRO  OLRDF$  $$$GBL
        .MCALL  .WORD.,DEFIN$
        .IF IDN <$$$GBL>,<DEF$G>
...GBL=1
        .IFF
...GBL=0
        .ENDC

;
; THE FOLLOWING MACRO DEFINES THE SUB-FUNCTION CODES FOR EACH OF THE OPERATIONS
; PERFORMED BY THE HRC TASK AND A PARAMETER DESCRIBING THE ARGUMENTS REQUIRED
; FOR EACH FUNCTION.
; IN A MACRO CALL THE FOLLOWING ARE THE LEGAL COMBINATIONS FOR THE 'MASK'
; PARAMETER:
;               <>        SIGNIFYING NO PARAMETERS
;               <D>       SIGNIFYING ONE BUFFER DESCRIPTOR
;               <D,D>     SIGNIFYING TWO BUFFER DESCRIPTORS
;               <D,CT>    SIGNFIYING ONE DESCRIPTOR AND 'CT' BYTES OF PARAMETERS
;               <CT>      SIGNIFYING 'CT' BYTES OF PARAMETERS
;

        .MACRO FUNC     NAME,SUBF,FUN,MASK
        .WORD.  IO.'NAME,SUBF,FUN
       FUNCA NAME,<MASK>
        .ENDM

        .MACRO FUNCA NAME,MSK
       PARCT=0
       DESCT=0
        .IRP X,<MSK>
        .IIF IDN <X>,<P> PARCT=PARCT+1
        .IIF IDN <X>,<D> DESCT=DESCT+1
        .IIF GT <PARCT-17> .ERROR INVALID PARAMETER COUNT
        .IIF GT <DESCT-17> .ERROR INVALID DESCRIPTOR COUNT
        .ENDR

       TEMP=<DESCT*4>+<PARCT*2>
        .WORD.  IO$'NAME,<<DESCT*20+PARCT>>,TEMP
        .ENDM

;
; DEFINE ONLINE RECONFIGURATION I/O FUNCTIONS
;

        .WORD.  IO.MFC,000,001  ; MULTI-FUNCTION MODIFY CONFIGURATION
        .WORD.  IO.RSC,000,002  ; READ SYSTEM CONFIGURATION
        .WORD.  IO.WSC,000,006  ; MODIFY DEVICE CONFIGURATION

;
; DEFINE SUBFUNCTIONS TO MODIFY DEVICE CONFIGURATION
;
```

OLRDF$

```
        FUNC    ONL,001,006,<D,D>       ; SET DEVICE ONLINE
        FUNC    OFL,002,006,<D,D>       ; SET DEVICE OFFLINE
        FUNC    MAI,003,006,<D,D>       ; SET DEVICE IN MAINTAINENCE MODE
        FUNC    CAC,004,006,<>          ; CACHE CONTROL
        FUNC    MEM,005,006,<>          ; MIND CONTROL
        FUNC    STN,006,006,<P,P>       ; RECONFIGURATION CONTROL,
                                        ; SPECIFY TASK NAME
        FUNC    HRC,007,006,<P,P>       ; RECONFIGURATION CONTROL,
                                        ; HRC OPERATING MODE
        FUNC    ONE,010,006,<P,P>       ; ON <CONDITION> <COMMAND>
        FUNC    STA,011,006,<D>         ; RETURN DEVICE STATE
        FUNC    IF ,012,006,<P,P>       ; IF <CONDITION> <COMMAND>
        FUNC    RLI,013,006,<D,D,D,D>   ; LINK UNIBUS RUN
        FUNC    RUL,014,006,<D,D,D,D>   ; UNLINK UNIBUS RUN
        FUNC    MBO,015,006,<P,P,D,D,D,D,D,D,D,D,D> ; MEMORY BOX ONLINE
        FUNC    RSW,016,006,<D,D,D,D>   ; SWITCH BUS
        FUNC    WAT,017,006,<D>         ; WRITE ATTRIBUTES
        FUNC    RAT,020,006,<D,D>       ; READ ATTRIBUTES
        FUNC    MBF,021,006,<P,P,D,D,D,D,D,D,D,D,D> ; MEMORY BOX OFFLINE

        IO$MAX=21                       ; DEFINE MAXIMUM SUBFUNCTION

        DEFIN$  IS.HRG,6.       ; STOP PROCESSING CONDITION ENCOUNTERED
                                ; SECOND STATUS WORD IS ARGUMENT


;
; DEFINE A MACRO, WHICH WHEN EXPANDED WITH THE APPROPRIATE DEFINITION
; FOR .IOER. WILL DEFINE THE PRIVATE ERROR CODES USED BY HRC AND CON.
;
        .MACRO  OLREM$

        $$$VAL=-256.            ; DEFINE INITIAL ERROR NUMBER VALUE

        .IOER.  IE$DAL,<DEVICE already linked>
        .IOER.  IE$DNL,<DEVICE not linked>
        .IOER.  IE$PRM,<Parameter error>
        .IOER.  IE$SYN,<Syntax error>
        .IOER.  IE$AFE,<Attribute format error>
        .IOER.  IE$TMU,<HRC... Internal tables insufficient for this system>
        .IOER.  IE$CAB,<Unable to access busrun>
        .IOER.  IE$TRP,<HRC... internal addressing error>
        .IOER.  IE$ALG,<Memory box parameter error>
        .IOER.  IE$TQU,<Timeout on unit quieting operation>
        .IOER.  IE$EPO,<ONLINE CPU failure>
        .IOER.  IE$EUO,<ONLINE UNIT failure>
        .IOER.  IE$ECO,<ONLINE CONTROLLER failure>
        .IOER.  IE$EPF,<OFFLINE CPU failure>
        .IOER.  IE$EUF,<OFFLINE UNIT failure>
        .IOER.  IE$ECF,<OFFLINE CONTROLLER failure>
        .IOER.  IE$CFU,<Attempt to quiet unit for controller failed>
        .IOER.  IE$CSR,<CSR for controller not present in I/O page>
        .IOER.  IE$SWF,<Unable to switch unit away from current controller>
        .IOER.  IE$ICE,<HRC... detected I/O database consistancy error>
        .IOER.  IE$SCE,<Executive or Driver status change error>
        .IOER.  IE$MDE,<HRC... Memory descriptor format error>
        .IOER.  IE$NFW,<No path to target device is available>
        .IOER.  IE$CXT,<Unable to take unit with context offline.>
        .IOER.  IE$IDU,<Invalid device descriptor>
        .IOER.  IE$UNK,<Device is unknown in this configuration>
```

```
        .IOER.  IE$SZE,<HRC... Unable to access device to size drive>
        .IOER.  IE$POB,<HRC... Can't take box offline. Partition overmaps box>
        .IOER.  IE$NLB,<HRC... Can't take box offline. Not last box in memory>
        .IOER.  IE$OMP,<HRC... Can't modify partition size. Overmap exists>
        .IOER.  IE$POC,<HRC... Can't modify partition size. Occupied>
        .IOER.  IE$DFE,<HRC... Request format error.>
        .IOER.  IE$IDS,<HRC... Invalid device specification.>
        .IOER.  IE$UOE,<HRC... Unkown error from online/offline call>
        .ENDM

;
; CONDITION CODES FOR CONDITIONS TESTED BY IO.ONE AND IO.IF FUNCTIONS
;

        CO$ONL = 1              ; IF DEVICE NOW ONLINE
        CO$OFL = 2              ; IF DEVICE NOW OFFLINE
        CO$UNK = 3              ; UNKNOWN DEVICE
        CO$ACC = 4              ; ACCESSABLE (ACCESS PATH EXISTS)
        CO$ANY = 5              ; ANY ERROR CONDITION
        CO$MAI = 6              ; MAINTENANCE MODE

        CO$MAX = 6              ; MAXIMUM CODE

;
; CONDITION COMMAND CODES FOR IO.ONE AND IO.IF FUNCTIONS
;

        CD$STO = 2              ; 'STOP' COMMAND
        CD$GOT = 4              ; 'GOTO'
        CD$CON = 6              ; 'CONTINUE'

        CD$MAX = 6              ; MAXIMUM CONDITION DEFINED

;
; ARGUMENT DEFINITION FOR IO.HRC FUNCTION
;

        M$LOG = 1               ; SUPRESS CONFIGURATION TRANSMISSION TO ERRLOG
        M$INIT = 2              ; INITALIZE HRC
        M$DEBG = 4              ; SET HRC INTO DEBUG MODE. (DEVELOPMENT ONLY)
        M$EXIT = 10             ; EXIT REQUEST (FROM ABORT AST REQUEST)

;
; DEFINE TABLE OFFSETS AND STATUS BITS RETURNED IN RESPONSE TO
; A 'READ CONFIGURATION' QIO
;

        .ASECT
        .=0
C$DTYP: .BLKB   1               ; ENTRY TYPE FIELD

;
; ENTRY TYPE CODES ARE AS FOLLOWS
;

        ET$HDR = 1              ; CONFIGURATION HEADER ENTRY
        ET$END = 2              ; END OF CONFIGURATION DATA

        ET$DEV = 'A             ; MINIMUM VALUE FOR DEVICE SPECIFICATION ENTRY
```

OLRDF$

```
C$DECT: .BLKB   1               ; COUNT OF TABLE ENTRIES (CPUS+SWITCHED
                                ; BUS RUNS+CONTROLLERS+UNITS)
C$DVER: .BLKB   1               ; VERSION OF RECONFIGURATION TASK PROTOCAL
C$DSTD: .BLKB   1               ; SIZE OF HEADER
C$DMUB: .BLKB   1               ; MAXIMUM UNIBUS RUNS SUPPORTED
C$DMCT: .BLKB   1               ; MAXIMUM CONTROLLERS OF A GIVEN TYPE SUPPORTED
        .EVEN
C$DFAC: .BLKW   2               ; FACILITES SUPPORTED IN HOST SYSTEM
C$DIDN: .BLKW   9.              ; HRC VERSION AND BUILD TIMESTAMP
C$STD:                          ; SIZE OF THE TABLE HEADER

;
; OFFSETS WITHIN THE FIXED PORTION OF A GIVEN ENTRY
;
        .=0

C$DTYP:                         ; ENTRY TYPE CODE
C$DNAM: .BLKW   1               ; TWO ASCII CHARACTER UNIT OR CONTROLLER NAME
C$DPUN: .BLKB   1               ; CONTROLLER NUMBER (0-255.)
C$DLUN: .BLKB   1               ; LOGICAL UNIT NUMBER IF THIS DEVICE IS A UNIT
C$DSCT: .BLKB   1               ; SUB-CONTROLLER NUMBER
C$DEVT: .BLKB   1               ; DEVICE TYPE CODE
C$DSTS: .BLKW   1               ; DEVICE STATUS MASK

;
; FLAG VALUES FOR C$DSTS
;

        CS$ATR=1                ; VARIABLE LENGTH ATTRIBUTE INFO IS APPENDED
        CS$EXF=76               ; FIELD IN C$DSTS CONTAINING COUNT OF ADDITIONAL
                                ; BYTES IN THIS DEVICE ENTRY
        CS$SUB=100              ; THIS IS A SUB-CONTROLLER DEVICE
        ;CS$XXX=200             ; UNUSED
        CS$OFL=400              ; 1=>DEVICE IS OFFLINE, 0=>DEVICE IS ONLINE
        CS$PDF=1000             ; DEVICE IS RESTRICTED TO PRIVILEGED DIAG FNS
        CS$POR=2000             ; THIS IS A MULTIPORT DEVICE
        CS$MBD=4000             ; DEVICE IS A MASS BUS DEVICE
        CS$UNK=10000            ; DEVICE IS UNKNOWN
        CS$ACC=20000            ; AN ONLINE ACCESS PATH EXISTS TO THIS DEVICE
        CS$MTD=40000            ; DEVICE IS MOUNTED(DISK) OR LOGGED IN (TERM)
        CS$DRV=100000           ; A DRIVER IS LOADED FOR THIS DEVICE

C$DST2: .BLKW   1               ; STATUS EXTENSION

        CS$PUN=20               ; 1=> THIS DEVICE SPECIFIED WITH PHYSICAL UNIT NUMBER
        CS$CRD=40               ; 1=> THIS IS A CONTROLLER RELATIVE DEVICE SPEC
        CS$PRC=100              ; 1=> THIS IS A PORT RELATIVE CONTROLLER SPEC
        CS$CTL=200              ; DEVICE IS A CONTROLLER (MUST BE SIGN BIT)
        CS$DCL=3400             ; DEVICE CLASS CODE FIELD. MUST BE LOW ORDER BITS
                                ; OF HIGH BYTE.

;
; DEVICE CLASS VALUES
;
```

```
        DC$UNI = 0              ; UNIT
        DC$CTL = 1              ; CONTROLLER
        DC$MKU = 2              ; MEMORY BOX UNIT
        DC$MKC = 3              ; MEMORY BOX CONTROLLER
        DC$SBU = 4              ; SWITCHED BUS UNIT
        DC$SBC = 5              ; SWITCHED BUS CONTROLLER
        DC$CPU = 6              ; CPU
        ;DC$XXX = 7             ; UNUSED

C$DDAT: .BLKW   2               ; DEVICE DEPENDANT DATA
C$SME:                          ; SIZE IF A MINIMUM ENTRY

;
; VARIABLE PORTION OF A GIVEN ENTRY
;

;
; FOR CONTROLLERS
;
        .=C$SME

C$DKPO: .BLKW   1               ; PORT-STATUS-WORD. THIS DESCRIBES THE BUS RUN
                                ; CPU OR SWITCHED BUS, TO WHICH THIS
                                ; CONTROLLER IS CONNECTED.
C$SCT:                          ; MIMIMUM SIZE OF A CONTROLLER ENTRY

;
; FOR UNIT ENTRIES
;
        .=C$SME

C$DCTN: .BLKW   1               ; CONTROLLER NAME. TWO CHARACTER ASCII CODE
                                ; OF THE CONTROLLER TO WHICH THIS UNIT IS
                                ; ATTACHED.
C$DUPO: .BLKW   1               ; PORT-STATUS-WORD. THIS IS THE
                                ; FIRST OF THE PSWS DESCRIBING THE CONTROLLER(S)
                                ; TO WHICH THIS UNIT IS CONNECTED.
C$SUN:                          ; MIMIMUM SIZE OF A UNIT ENTRY

;
; FOR CPU-S
;
        .=C$SME

C$DCPO: .BLKW   1               ; PORT-STATUS-WORD. THIS IS THE BUS
                                ; NUMBER FOR THIS CPU.
C$SCP:                          ; MINIMUM SIZE OF A CPU ENTRY

;
; FOR MEMORY BOXES
;
        .=C$SME

C$DCTN: .BLKW   1               ; CONTROLLER NAME.
        .BLKW   4               ; MAXIMUM OF 4 PORTS FOR MEMORY CONTROLLERS

C$SMB:                          ; MAXIMUM SIZE OF A MEMORY BOX ENTRY

;
; STATUS BIT DEFINITIONS FOR THE PORT STATUS WORD
;
```

OLRDF$

```
        CP$OFL=400              ; 1=> PORT IS OFFLINE
        CP$XXX=1000             ; UNUSED
        CP$CUR=2000             ; THIS PORT IS THE CURRENT PORT (S.KRB
                                ; REFERENCES THIS PORT
        CP$XXX=4000             ; UNUSED
        CP$XXX=10000            ; UNUSED
        CP$ACC=20000            ; THIS PORT HAS AN ACCESS PATH
        CP$MTD=40000            ; PORT HAS CONTEXT OR SERVICES A DEVICE HAVING
                                ; CONTEXT
        CP$XXX=100000           ; UNUSED


;
; DEVICE ATTRIBUTES CODES
;
        .MACRO ATT NAME,SIZ
        $$$TMP=$$$TMP+1
        DEFIN$ DA$'NAME,$$$TMP!<400*SIZ>
        .ENDM

        $$$TMP=0

        ATT     CSR,2           ; CSR ADDRESS
        ATT     VEC,2           ; VECTOR ADDRESS
        ATT     UBR,2           ; UNIBUS RUN
        ATT     TYP,2           ; DEVICE TYPE, READ ONLY
        ATT     VOL,12.         ; MOUNTED VOLUME NAME, READ ONLY
        ATT     ERR,10          ; DEVICE ERROR COUNTERS, READ/WRITE
        ATT     PRI,2           ; DEVICE INTERRUPT PRIORITY
        ATT     MBP,6           ; MEMORY BOX PARAMETER
        ATT     STE,2           ; SANITY TIMER ENABLE/DISABLE
        ATT     SAL,2           ; ALARM ENABLE/DISABLE
        ATT     DSN,2           ; DEVICE SERIAL NUMBER
        ATT     CSN,10          ; CPU SERIAL NUMBERS

;
; MEMORY BOX ATTRIBUTE BUFFER
;
        .ASECT
        .=0

C$MBAS: .BLKW   1               ; BASE ADDRESS OF BOX
C$MINT: .BLKB   1               ; INTERLEAVE FACTOR
        .BLKB   1               ; FREE BYTE
C$MSIZ: .BLKW   1               ; SIZE OF BOX IN 32 WORD BLOCKS
C$MGRN: .BLKW   1               ; BOX GRANULARITY. "BYTES-PER-UNIT"

C$MDSC:                         ; SIZE OF BOX ATTRIBUTE BUFFER


        .PSECT

;
; REDEFINE MACRO TO NULL
;
        .MACRO  OLRDF$  X
        .ENDM
        .MACRO ATT X
        .ENDM
```

```
        .ENDM
;
; MACRO FOR THE DEFINITION OF DEVICE TYPE CODES
;
        .MACRO DEVCD$ $$$GBL
        .MCALL DEFIN$
        .IF IDN <$$$GBL>,<DEF$G>
...GBL=1
        .IFF
...GBL=0
        .ENDC
        .MACRO DEV X
        DEFIN$ D$'X,$$$TMP
        $$$TMP=$$$TMP+1
        .ENDM
        $$$TMP = 0
        DEV UDET        ; UNDETERMINED DEVICE TYPE
        DEV UKNO        ; UNKNOWN DEVICE TYPE
        DEV RK03        ; RK03
        DEV RK05        ; RK05
        DEV RK5F        ; RK05-F (DUAL DENSITY FIXED CARTRIDGE)
        DEV RX01        ; RX01
        DEV RX02        ; RX02 (DUAL DENSITY RX01)
        DEV RL01        ; RL01
        DEV RL02        ; RL02
        DEV RP02        ; RP02
        DEV RP03        ; RP03
        DEV RP04        ; RP04
        DEV RP05        ; RP05
        DEV RP06        ; RP06
        DEV RP07        ; RP07
        DEV RK06        ; RK06
        DEV RK07        ; RK07
        DEV RM02        ; RM02
        DEV RM03        ; RM03
        DEV RM05        ; RM05
        DEV RM80        ; RM80
        DEV RS03        ; RS03
        DEV RS04        ; RS04 (DUAL DENSITY RS03)
        DEV RF11        ; RF11/RS08
```

OLRDF$

```
DEV TK25        ; TK25
DEV TK50        ; TK50
DEV TU10        ; TU10
DEV TU16        ; TU16
DEV TU45        ; TU45
DEV TU77        ; TU77
DEV TU78        ; TU78
DEV TS11        ; TS11
DEV TSU0        ; TSU05
DEV TSV0        ; TSV05
DEV TU80        ; TU80
DEV TU81        ; TU81

DEV TM02        ; TM02
DEV TM03        ; TM03
DEV TM78        ; TM78

DEV TU56        ; TU56
DEV TU58        ; TU58
DEV TU60        ; TU60

DEV MSCP        ; UDA50
DEV RA60        ; RA60
DEV RA80        ; RA80
DEV RA81        ; RA81
DEV RC25        ; RC25 (AZTEC)

DEV RD50        ; RD50
DEV RD51        ; RD51
DEV RX50        ; RX50

DEV ML11        ; ML11

DEV TERM        ; TERMINAL

$$$TMP=370
DEV USR0        ; USER TYPE 0
DEV USR1        ; USER TYPE 1
DEV USR2        ; USER TYPE 2
DEV USR3        ; USER TYPE 3
DEV USR4        ; USER TYPE 4
DEV USR5        ; USER TYPE 5
DEV USR6        ; USER TYPE 6
DEV USR7        ; USER TYPE 7


.MACRO DEVCD$
.ENDM
.MACRO DEV X
.ENDM

.ENDM
```

```
        .MACRO  PCBDF$  L,B,SYSDEF

;+
; MAIN PARTITION PCB
;-

        .ASECT
.=0
P.LNK:'L'.BLKW  1               ;LINK TO NEXT MAIN PARTITION PCB
        .BLKW   1               ;(UNUSED)
P.NAM:'L'.BLKW  2               ;PARTITION NAME IN RAD50
P.SUB:'L'.BLKW  1               ;POINTER TO FIRST SUBPARTITION
P.MAIN:'L'.BLKW 1               ;POINTER TO SELF
P.REL:'L'.BLKW  1               ;STARTING PHYSICAL ADDRESS IN 32W BLOCKS
P.BLKS:'L'
P.SIZE:'L'.BLKW 1               ;SIZE OF PARTITION IN 32W BLOCKS
P.WAIT:'L'.BLKW 2               ;PARTITION WAIT QUEUE LISTHEAD
        .BLKW   2               ;(UNUSED)
P.STAT:'L'.BLKW 1               ;PARTITION STATUS FLAGS
P.ST2:'L' .BLKW 1               ;STATUS EXTENSION FOR COMMON AND MAIN PCB'S
        .BLKW   3               ;(UNUSED)
P.HDLN:'L'.BLKB 1               ;SIZE OF EXTERNAL HEADER IN 32W BLOCKS
P.IOC:'L' .BLKB 1               ;PARTITION I/O COUNT
$$$=.
P.RRM:'L'.BLKW  1               ;REQUIRED RUN MASK

        .IF NDF M$$PRO
.=$$$
        .ENDC

        .IF NB  SYSDEF
P.LGTH='B'.                     ;PARTITION CONTROL BLOCK LENGTH
        .ENDC


;+
; TASK REGION PCB
;-
```

PCBDF$

```
.=0
P.LNK:'L'.BLKW  1               ;UTILITY LINK WORD
P.PRI:'L'.BLKB  1               ;PRIORITY OF PARTITION
P.RMCT:'L'.BLKB 1               ;RESIDENT MAPPED TASKS COUNT
P.NAM:'L'.BLKW  2               ;PARTITION NAME IN RAD50
P.SUB:'L'.BLKW  1               ;POINTER TO NEXT SUBPARTITION
P.MAIN:'L'.BLKW 1               ;POINTER TO MAIN PARTITION
P.REL:'L'.BLKW  1               ;STARTING PHYSICAL ADDRESS IN 32W BLOCKS
P.BLKS:'L'
P.SIZE:'L'.BLKW 1               ;SIZE OF PARTITION IN 32W BLOCKS
        .BLKW   1               ;(UNUSED)
P.SWSZ:'L'.BLKW 1               ;PARTITION SWAP SIZE
P.DPCB:'L'.BLKW 1               ;CHECKPOINT ALLOCATION PCB
P.TCB:'L'.BLKW  1               ;TCB ADDRESS OF OWNER TASK
P.STAT:'L'.BLKW 1               ;PARTITION STATUS FLAGS
P.HDR:'L' .BLKW 1               ;POINTER TO HEADER CONTROL BLOCK
        .BLKW   1               ;(UNUSED)
P.ATT:'L' .BLKW 2               ;ATTACHMENT DESCRIPTOR LISTHEAD

P.HDLN:'L'.BLKB 1               ;SIZE OF EXTERNAL HEADER IN 32W BLOCKS
P.IOC:'L' .BLKB 1               ;PARTITION I/O COUNT

$$$=.

P.RRM:'L'.BLKW  1               ;REQUIRED RUN MASK

        .IF NDF M$$PRO

.=$$$

        .ENDC

;+
; COMMON REGION PCB
;-

.=0
P.LNK:'L'.BLKW  1               ;UTILITY LINK WORD
P.PRI:'L'.BLKB  1               ;PRIORITY OF PARTITION
P.RMCT:'L'.BLKB 1               ;RESIDENT MAPPED TASKS COUNT
P.NAM:'L'.BLKW  2               ;PARTITION NAME IN RAD50
P.SUB:'L'.BLKW  1               ;POINTER TO NEXT SUBPARTITION
P.MAIN:'L'.BLKW 1               ;POINTER TO MAIN PARTITION
P.REL:'L'.BLKW  1               ;STARTING PHYSICAL ADDRESS IN 32W BLOCKS
P.BLKS:'L'
P.SIZE:'L'.BLKW 1               ;SIZE OF PARTITION IN 32W BLOCKS
P.CBDL:'L'.BLKW 1               ;COMMON BLOCK DIRECTORY LINK
P.CSBA:'L'                      ;CACHE STATISTICS BLOCK LISTHEAD
                                ;...(IF P2.CHE IS SET, PARTITION WON'T SWAP)
P.SWSZ:'L'.BLKW 1               ;PARTITION SWAP SIZE
P.DPCB:'L'.BLKW 1               ;POINTER TO DISK PCB
P.OWN:'L' .BLKW 1               ;OWNING UIC OF REGION
P.STAT:'L'.BLKW 1               ;PARTITION STATUS FLAGS
P.ST2:'L' .BLKW 1               ;STATUS EXTENSION FOR COMMON AND MAIN PCB'S
P.PRO:'L' .BLKW 1               ;PROTECTION WORD [DEWR,DEWR,DEWR,DEWR]
P.ATT:'L' .BLKW 2               ;ATTACHMENT DESCRIPTOR LISTHEAD

P.HDLN:'L'.BLKB 1               ;SIZE OF EXTERNAL HEADER IN 32W BLOCKS
P.IOC:'L' .BLKB 1               ;PARTITION I/O COUNT

$$$=.
```

```
P.RRM:'L'.BLKW  1               ;REQUIRED RUN MASK

        .IF NDF M$$PRO
.=$$$
        .ENDC

        .PSECT

;+
; PARTITION STATUS WORD BIT DEFINITIONS
;-

PS.OUT='B'100000                ;PARTITION IS OUT OF MEMORY(1=YES)
PS.CKP='B'40000                 ;PARTITION CHECKPOINT IN PROGRESS (1=YES)
PS.CKR='B'20000                 ;PARTITION CHECKPOINT IS REQUESTED (1=YES)
PS.CHK='B'10000                 ;PARTITION IS NOT CHECKPOINTABLE (1=YES)
PS.FXD='B'4000                  ;PARTITION IS FIXED (1=YES)
PS.CAF='B'2000                  ;CHECKPOINT SPACE ALLOCATION FAILURE (1=YES)
PS.LIO='B'1000                  ;MARKED BY SHUFFLER FOR LONG I/O (1=YES)
PS.NSF='B'400                   ;PARTITION IS NOT SHUFFLEABLE (1=YES)
PS.COM='B'200                   ;LIBRARY OR COMMON BLOCK (1=YES)
PS.LFR='B'100                   ;LAST LOAD OF REGION FAILED (1=YES)
PS.PER='B'40                    ;PARTIY ERROR OCCURED IN THIS REGION (1=YES)
PS.NWB='B'20                    ;COMMON SHOULDN'T BE WRITTEN BACK
PS.DEL='B'10                    ;PARTITION SHOULD BE DELETED WHEN NOT ATTACHED (1=YES)
PS.AST='B'4                     ;PARTITION HAS REGION LOAD AST PENDING
PS.CSA='B'2                     ;CHECKPOINT SPACE ALLOCATED

;+
; REQUIRED RUN MASK
;-

PR.UBT='B'100000                ;UNIBUS RUN T
PR.UBS='B'40000                 ;UNIBUS RUN S
PR.UBR='B'20000                 ;UNIBUS RUN R
PR.UBP='B'10000                 ;UNIBUS RUN P
PR.UBN='B'4000                  ;UNIBUS RUN N
PR.UBM='B'2000                  ;UNIBUS RUN M
PR.UBL='B'1000                  ;UNIBUS RUN L
PR.UBK='B'400                   ;UNIBUS RUN K
PR.UBJ='B'200                   ;UNIBUS RUN J
PR.UBH='B'100                   ;UNIBUS RUN H
PR.UBF='B'40                    ;UNIBUS RUN F
PR.UBE='B'20                    ;UNIBUS RUN E
PR.CPD='B'10                    ;PROCESSOR D
PR.CPC='B'4                     ;PROCESSOR C
PR.CPB='B'2                     ;PROCESSOR B
PR.CPA='B'1                     ;PROCESSOR A

;+
; STATUS EXTENSION WORD BIT DEFINITIONS
;       (THESE BITS CAN ONLY BE EXAMINED IN COMMON OR MAIN PCB'S)
;-
```

PCBDF$

```
P2.LMA='B'40000                 ;DON'T SHUFFLE,DELETE SPINDLE OR MUTILATE
                                ;THIS PARTITION
P2.CPC='B'20000                 ;CPCR INITIATED CHECKPOINT PENDING
P2.CHE='B'10000                 ;CACHE PARTITION
P2.SEC='B'4000                  ;THIS IS RO SECTION OF MU TASK
                                ;WITH TCB IN SEC. POOL
P2.PAR='B'2000                  ;THE FIXER TASK HAS HANDLED A PARITY ERROR
P2.POL='B'1000                  ;SECONDARY POOL PARTITION
P2.CPU='B'400                   ;MULTIPROCESSOR CPU PARTITION
P2.PIC='B'200                   ;POSITION INDEPENDENT LIBRARY OR COMMON (1=YES)
P2.RON='B'100                   ;READ-ONLY COMMON (1=YES)
P2.DRV='B'40                    ;DRIVER COMMON PARTITION (1=YES)
P2.LDD='B'20                    ;LOAD DEVICE DISABLED/DISMOUNTED
P2.APR='B'7                     ;STARTING APR NUMBER MASK FOR NON-PIC COMMON

;+
; CHECKPOINT FILE PCB
;-

        .ASECT
.=0
P.LNK:'L' .BLKW 1               ;LINK WORD OF CHECKPOINT FILE PCB'S
P.UCB:'L' .BLKW 1               ;UCB ADDRESS OF CHECKPOINT FILE DEVICE
P.LBN:'L' .BLKW 1               ;HIGH PART OF STARTING LBN
        .BLKW 1                 ;LOW PART OF STARTING LBN
P.SUB:'L' .BLKW 1               ;POINTER TO FIRST CHECKPOINT ALLOCATION PCB
P.MAIN:'L' .BLKW 1              ;MUST BE 0 (FOR $RLPR1)
P.REL:'L' .BLKW 1               ;CONTAINS 0 IF FILE IN USE, 1 IF NOT IN USE
P.SIZE:'L' .BLKW 1              ;SIZE OF CHECKPOINT FILE IN 256W BLOCKS
P.DLGH='B'.                     ;LENGTH OF ALL DISK PCB'S

;+
; CHECKPOINT ALLOCATION PCB
;-

.=0
        .BLKW 4                 ;(UNUSED)
P.SUB:'L' .BLKW 1               ;LINK TO NEXT CHECKPOINT ALLOCATION PCB
P.MAIN:'L' .BLKW 1              ;ADDRESS OF CHECKPOINT FILE PCB
P.REL:'L' .BLKW 1               ;RELATIVE POSITION IN FILE IN 256W BLOCKS
P.SIZE:'L' .BLKW 1              ;SIZE ALLOCATED IN 256W BLOCKS

;+
; COMMON TASK IMAGE FILE PCB
;-

.=0
P.FID1:'L' .BLKW 1              ;FILE ID WORD FOR SAVE
P.UCB:'L' .BLKW 1               ;UCB ADDRESS OF DEVICE ON WHICH COMMON RESIDES
P.LBN:'L' .BLKW 1               ;HIGH PART OF STARTING LBN
        .BLKW 1                 ;LOW PART OF STARTING LBN
P.IID='B' P.LBN+2               ;IMAGE INDEX FOR CPR SYSTEMS
P.FID2:'L' .BLKW 1              ;FILE ID WORD FOR SAVE
P.MAIN:'L'.BLKW 1               ;POINTER TO SELF
P.REL:'L' .BLKW 1               ;ALWAYS CONTAINS A 0
P.FID3:'L' .BLKW 1              ;FILE ID WORD FOR SAVE
```

```
;+
; ATTACHMENT DESCRIPTOR OFFSETS
;
;
; NOTE:  THE DETACH REGION ROUTINE DEPENDS ON THE FACT THAT:
;
;       A.TCB IS AT OFFSET 4 FROM A.PCBL
;       A.PCB IS AT OFFSET 4 FROM A.TCBL
;
; REFER TO $DETRG IN MODULE PLSUB FOR MORE DETAILS
;
;-

        .ASECT
.=0
A.PCBL:'L'.BLKW 1               ;PCB ATTACHMENT QUEUE THREAD WORD
A.PRI:'L'.BLKB  1               ;PRIORITY OF ATTACHED TASK
A.IOC:'L'.BLKB  1               ;I/O COUNT THROUGH THIS DESCRIPTOR
A.TCB:'L'.BLKW  1               ;TCB ADDRESS OF ATTACHED TASK
A.TCBL:'L'.BLKW 1               ;TCB ATTACHMENT QUEUE THREAD WORD
A.STAT:'L'.BLKB 1               ;STATUS BYTE
A.MPCT:'L'.BLKB 1               ;MAPPING COUNT OF TASK THRU THIS DESCRIPTOR
A.PCB:'L'.BLKW  1               ;PCB ADDRESS OF ATTACHED TASK
A.LGTH='B'.                     ;LENGTH OF ATTACHMENT DESCRIPTOR

;+
; ATTACHMENT DESCRIPTOR STATUS BYTE BIT DEFINITIONS
;-

        .PSECT
AS.PRO='B'100                   ;A.TCB IS SEC POOL TCB BIAS (1=YES)
AS.SBP='B'20                    ;CACHE BYPASS REQUESTED
AS.RBP='B'40                    ;REQUEST TO NOT BYPASS CACHE
AS.DEL='B'10                    ;TASK HAS DELETE ACCESS (1=YES)
AS.EXT='B'4                     ;TASK HAS EXTEND ACCESS (1=YES)
AS.WRT='B'2                     ;TASK HAS WRITE ACCESS (1=YES)
AS.RED='B'1                     ;TASK HAS READ ACCESS (1=YES)

        .MACRO  PCBDF$  X,Y,Z
        .ENDM
        .ENDM
```

PKTDF$

```
        .MACRO  PKTDF$,L,B,SYSDEF

;+
; ASYNCHRONOUS SYSTEM TRAP CONTROL BLOCK OFFSET DEFINITIONS
;
; SOME POSITIONAL DEPENDENCIES BETWEEN THE OCB AND THE AST CONTROL BLOCK
; ARE RELIED UPON IN THE ROUTINE $FINXT IN THE MODULE SYSXT.
;-

        .ASECT
.=177774
A.KSR5:'L' .BLKW 1              ;SUBROUTINE KISAR5 BIAS (A.CBL=0)
A.DQSR:'L' .BLKW 1              ;DEQUEUE SUBROUTINE ADDRESS (A.CBL=0)
        .BLKW   1               ;AST QUEUE THREAD WORD
A.CBL:'L' .BLKW 1               ;LENGTH OF CONTROL BLOCK IN BYTES
                                ;IF A.CBL = 0, THE AST CONTROL BLOCK IS
                                ;TO BE DEALLOCATED BY THE DEQUEUE SUBROUTINE
                                ;POINTED TO BY A.DQSR MAPPED VIA APR 5
                                ;VALUE A.KSR5.  THIS IS CURRENTLY USED ONLY
                                ;BY THE FULL DUPLEX TERMINAL DRIVER FOR
                                ;UNSOLICITED CHARACTER ASTS.
                                ;IF THE LOW BYTE OF A.CBL = 0, AND THE
                                ;HIGH BYTE IS NOT = 0, THE AST CONTROL BLOCK
                                ;IS A SPECIFIED AST, WITH LENGTH, C.LGTH.
                                ;IF THE HIGH BYTE OF A.CBL=0
                                ;AND THE LOW BYTE > 0, THEN
                                ;THE LOW BYTE IS THE LENGTH OF THE
                                ;AST CONTROL BLOCK.
                                ;IF HIGH BYTE = 0 AND LOW BYTE IS NEGATIVE,
                                ;THEN THE BLOCK IS A KERNEL AST
                                ;BIT 6 IS SET IF $SGFIN SHOULD
                                ;NOT BE CALLED PRIOR TO DISPATCHING
                                ;THE AST, AND THE LOW SIX BITS (5-0)
                                ;REPRESENT THE INDEX/2 INTO THE
                                ;KERNEL AST DISPATCH TABLE ($KATBL)
A.BYT:'L' .BLKW 1               ;NUMBER OF BYTES TO ALLOCATE ON TASK STACK
A.AST:'L' .BLKW 1               ;AST TRAP ADDRESS
A.NPR:'L' .BLKW 1               ;NUMBER OF AST PARAMETERS
A.PRM:'L' .BLKW 1               ;FIRST AST PARAMETER
AS.FPA='B'1                     ;CODE FOR FLOATING POINT AST
AS.RCA='B'2                     ;CODE FOR RECEIVE DATA AST
AS.RRA='B'3                     ;CODE FOR RECEIVE BY REFERENCE AST
AS.PEA='B'4                     ;CODE FOR PARITY ERROR AST
AS.REA='B'5                     ;CODE FOR REQUESTED EXIT AST
AS.PFA='B'6                     ;CODE FOR POWER FAIL AST
AS.CAA='B'7                     ;CODE FOR CLI COMMAND ARRIVAL AST
;
; BIT VALUES IN A.PRM+5
;
AF.XCC='B'1                     ;ATTACHED FOR ALL BUT CONTROL-C (TF.XCC)
AF.NOT='B'2                     ;ATTACHED FOR ALL NOTIFICATION (TF.NOT)
AF.OOB='B'4                     ;ACB IS FOR OUT-OF-BAND AST
AF.AST='B'10                    ;ACB HANDLES UNSOL. INPUT CHAR AST'S (TF.AST)
AF.ESQ='B'20                    ;ATTACHED FOR ESCAPE SEQUENCES (TF.ESQ)
AF.LCK='B'40                    ;ACB IS LOCKED
AF.QUE='B'100                   ;ACB IS QUEUED
AF.MDE='B'200                   ;ACB IS MARKED FOR DELETE
;
;
```

```
; ABORTER SUBCODES FOR ABORT AST (AS.REA) TO BE RETURNED ON USER'S STACK
;
AB.NPV='B'1                     ;ABORTER IS NONPRIVILEGED (1=YES)
AB.TYP='B'2                     ;ABORT FROM DIRECTIVE (0=YES)
                                ;ABORT FROM CLI COMMAND (1=YES)
AB.CTC='B'4                     ;ABORT FROM CONTROL-C GIN PROCESSING (1=YES)
A.PLGH='B'70                    ;SIZE OF PARITY ERROR AST CONTROL BLOCK
A.DUCB='B'10                    ;UCB OF TERM ISSUING DEBUG COMMAND
A.DLGH='B'10.                   ;LENGTH OF DEBUG (AK.TBT) AST BLOCK

;       KERNEL AST CONTROL CODES (A.CBL)

AK.BUF='B'200                   ;BUFFERED I/O COMPLETION
                                ;THIS CODE MUST BE 200 UNTIL ALL
                                ;REFERENCES IN TTDRV ARE FIXED
AK.OCB='B'201                   ;OFFSPRING TASK EXIT
AK.GBI='B'202                   ;SEGMENTED BUFFERED I/O COMPLETION
AK.TBT='B'203                   ;TASK FORCE T-BIT TRAP (DEBUG CMD)
AK.DIO='B'204                   ;DELAYED I/O COMPLETION
AK.GGF='B'205                   ;GRP. GBL. RUNDWN
;+
; BIT DEFINITIONS FOR THE GET/SET INFORMATION DIRECTIVE.
;-
SF.PRV='B'100000                ;FUNCTION IS PRIVILEGED
SF.IN='B' 40000                 ;FUNCTION IS AN INPUT FUNCTION

;+
; GROUP GLOBAL EVENT FLAG BLOCK OFFSETS
;-

.=0
G.LNK:'L'.BLKW  1               ;LINK WORD
G.GRP:'L'.BLKB  1               ;GROUP NUMBER
G.STAT:'L'.BLKB 1               ;STATUS BYTE
G.CNT:'L'.BLKW  1               ;ACCESS COUNT
G.EFLG:'L'.BLKW 2               ;EVENT FLAGS
G.LGTH='B'.                     ;LENGTH OF GROUP GLOBAL EVENT FLAG
                                ;BLOCK
GS.DEL='B'1                     ;STATUS BIT -- MARKED FOR DELETE

;+
; EXECUTIVE POOL MONITOR CONTROL FLAGS
;-

; $POLST IS THE SYNCHRONIZATION WORD BETWEEN THE EXEC AND POOL MONITOR

PC.HIH='B'1                     ;HIGH POOL LIMIT CROSSED (1=YES)
PC.LOW='B'2                     ;LOW POOL LIMIT CROSSED (1=YES)
PC.ALF='B'4                     ;POOL ALLOCATION FAILURE (1=YES)
PC.XIT='B'200                   ;FORCE POOL MONITOR TASK TO EXIT (MUST
                                ;BE COUPLED WITH SETTING FE.MXT IN THE
                                ;FEATURE MASK)
PC.NRM='B'PC.HIH*400            ;POOL TASK INHIBIT BIT FOR HIGH POOL
PC.ALM='B'PC.LOW*400            ;POOL TASK INHIBIT BIT FOR LOW POOL

; $POLFL IS THE POOL USAGE CONTROL WORD

PF.INS='B'40                    ;REJECT NONPRIVILEGED INS/RUN/REM
PF.LOG='B'100                   ;NONPRIVILEGED LOGINS ARE DISABLED
PF.REQ='B'200                   ;STALL REQUEST OF NONPRIV. TASKS
```

PKTDF$

```
PF.ALL='B'177777                ;TAKE ALL POSSIBLE ACTIONS TO SAVE POOL

;+
; OFFSPRING CONTROL BLOCK DEFINITIONS
;
; SOME POSITIONAL DEPENDENCIES ARE DEPENDED ON BETWEEN THE OCB AND THE
; AST BLOCK IN THE ROUTINE $FINXT IN THE MODULE SYSXT.
;-

.=0
O.LNK:'L'.BLKW  1               ;OCB LINK WORD
O.MCRL:'L'.BLKW 1               ;ADDRESS OF MCR COMMAND LINE
O.PTCB:'L'.BLKW 1               ;PARENT TCB ADDRESS
O.AST:'L'.BLKW  1               ;EXIT AST ADDRESS
O.EFN:'L'.BLKW  1               ;EXIT EVENT FLAG
O.ESB:'L'.BLKW  1               ;EXIT STATUS BLOCK VIRTUAL ADDRESS
O.STAT:'L'.BLKW 8.              ;EXIT STATUS BUFFER
O.LGTH='B'.                     ;LENGTH OF OCB

;+
; I/O PACKET OFFSET DEFINITIONS
;
;
; NOTE:  THE ROUTINE $QRMVT IN MODULE QUEUE DEPENDS ON THE FACT THAT
;       I.TCB IS AT OFFSET 4 IN AN I/O PACKET.
;
;-

        .ASECT
.=0
I.LNK:'L' .BLKW 1               ;I/O QUEUE THREAD WORD
I.PRI:'L' .BLKB 1               ;REQUEST PRIORITY
I.EFN:'L' .BLKB 1               ;EVENT FLAG NUMBER
I.TCB:'L' .BLKW 1               ;TCB ADDRESS OF REQUESTOR
I.LN2:'L' .BLKW 1               ;POINTER TO SECOND LUN WORD
I.UCB:'L' .BLKW 1               ;POINTER TO UNIT CONTROL BLOCK
I.FCN:'L' .BLKW 1               ;I/O FUNCTION CODE
I.IOSB:'L' .BLKW        1               ;VIRTUAL ADDRESS OF I/O STATUS BLOCK
        .BLKW   1               ;I/O STATUS BLOCK RELOCATON BIAS
        .BLKW   1               ;I/O STATUS BLOCK ADDRESS
I.AST:'L' .BLKW 1               ;AST SERVICE ROUTINE ADDRESS
I.PRM:'L' .BLKW 1               ;RESERVED FOR MAPPING PARAMETER #1
        .BLKW   6               ;PARAMETERS 1 TO 6
        .BLKW   1               ;USER MODE DIAGNOSTIC PARAMETER WORD

;
; FOLLOWING ARE DEFINITIONS FOR FLAG BITS IN I.PRM+11
; (DSA DRIVERS INTERNAL USE ONLY)
;

IP.FAK ='B' 20                  ;IOP IS PSEUDO IOP
IP.ABO ='B' 40                  ;(MUDRV)ABORT COMMAND MUST BE ISSUED FOR IOP
IP.PND ='B' 100                 ;(MUDRV)ABORT COMMAND WAS ISSUED FOR IOP
IP.UMR ='B'200                  ;A UMR WAIT BLOCK IS IN USE FOR THIS I/O

I.ATTL='B'.                     ;MINIMUM LENGTH OF I/O PACKET (USED BY
                                ;FILE SYSTEM TO CALCULATE MAXIMUM
                                ;NUMBER OF ATTRIBUTES)
I.AADA:'L' .BLKW 2              ;STORAGE FOR ATT DESCR PTRS WITH I/O
I.LGTH='B'.                     ;LENGTH OF I/O REQUEST CONTROL BLOCK
I.ATRL='B'6*8.                  ;LENGTH OF FILE SYSTEM ATTRIBUTE BLOCK
```

```
;
; DEFINE OFFSETS IN I/O PACKET EXTENSION (IOPX)
;
        .ASECT
.=      0

I.XLNK:'L' .BLKW  1             ;LINK WORD
I.XIOP:'L' .BLKW  1             ;I/O PACKET ADDRESS
I.XTCB:'L' .BLKW  1             ;TCB ADDRESS OF REQUESTING TASK
I.XMOD:'L' .BLKW  2             ;MODIFIER WORDS (NOTE: 2ND WORD MUST BE
                                ;SPECIFIED AND MUST BE ZERO.)
I.XRBF:'L' .BLKW  2             ;READ DATA BUFFER ADDRESS APR BIAS
                                ;READ DATA BUFFER VIRTUAL ADDRESS
I.XRBL:'L' .BLKW  1             ;READ DATA BUFFER LENGTH
I.XTMO:'L' .BLKW  1             ;READ TIME-OUT INTERVAL
I.XPBF:'L' .BLKW  2             ;PROMPT BUFFER ADDRESS APR BIAS
                                ;PROMPT BUFFER VIRTUAL ADDRESS
I.XPBL:'L' .BLKW  1             ;PROMPT BUFFER LENGTH
I.XPBV:'L' .BLKW  1             ;PROMPT BUFFER VERTICAL FORMS CONTROL
I.XTTB:'L' .BLKW  2             ;TERMINATOR TABLE ADDRESS APR BIAS
                                ;TERMINATOR TABLE VIRTUAL ADDRESS
I.XTTL:'L' .BLKW  1             ;TERMINATOR TABLE LENGTH
I.XDBF:'L' .BLKW  2             ;DEFAULT INPUT BUFFER ADDRESS APR BIAS
                                ;DEFAULT INPUT BUFFER VIRTUAL ADDRESS
I.XDBL:'L' .BLKW  1             ;DEFAULT INPUT BUFFER LENGTH

I.XLEN='B'.                     ;LENGTH OF IOPX
;+
; CLI PARSER BLOCK (CPB) DEFINITIONS
;-

.=0

C.PTCB:'L' .BLKW 1              ;ADDRESS OF CLI'S TCB
C.PNAM:'L' .BLKW 2              ;CLI NAME
C.PSTS:'L' .BLKW 1              ;STATUS MASK
C.PDPL:'L' .BLKB 1              ;LENGTH OF DEFAULT PROMPT
C.PCPL:'L' .BLKB 1              ;LENGTH 0 CNTRL/C PROMPT
C.PRMT:'L'                      ;START OF PROMPT STRINGS. DEFAULT
                                ;IS CONCATENATED WITH CONTROL C PROMPT

;
; STATUS BIT DEFINITIONS
;

CP.NUL='B'1                     ;PASS EMPTY COMMANDS TO CLI
CP.MSG='B'2                     ;CLI DESIRES SYSTEM MESSAGES
CP.LGO='B'4                     ;CLI WANTS COMMANDS FROM LOGGED OFF TTYS
CP.DSB='B'10                    ;CLI IS DISABLED
CP.PRV='B'20                    ;USER MUST BE PRIV TO SET TTY TO THIS CLI
CP.SGL='B'40                    ;DON'T HANDLE CONTINUATIONS (M-PLUS ONLY)
CP.NIO='B'100                   ;MCR..., HEL, BYE DO NO I/O TO TTY
                                ;HEL, BYE DO NOT SET CLI ETC.
CP.RST='B'200                   ;ABILITY TO SET TO THIS CLI IS RESTRICTED
                                ;TO THE CLI ITSELF
CP.EXT='B'400                   ;PASS TASK EXIT PROMPT REQUESTS TO CLI
CP.POL='B'1000                  ;CLI TCB IS IN SECONDARY POOL
CP.CTC='B'2000                  ;^C NOTIFICATION PACKETS ARE WANTED

;+
; SECONDARY POOL COMMAND BUFFER BLOCKS
;-
```

PKTDF$

```
.=0
C.CLK:'L'   .BLKW       1       ;LINK WORD
C.CTCB:'L'  .BLKW       1       ;TCB ADDRESS OF TASK TO RECEIVE COMMAND
C.CUCB:'L'  .BLKW       1       ;UCB ADDRESS OF RESPONSIBLE TERMINAL
C.CCT:'L'   .BLKW       1       ;CHARACTER COUNT, EXCLUDING TRAILING CR
C.CSTS:'L'  .BLKW       1       ;STATUS MASK
C.CMCD:'L'                      ;SYSTEM MESSAGE CODE
C.CSO:'L'   .BLKW       1       ;STARTING OFFSET OF VALID COMMAND TEXT
C.CTR:'L'   .BLKB       1       ;TERMINATOR CHARACTER
C.CBLK:'L'  .BLKB       1       ;SIZE OF PACKET IN SEC POOL (32 WD.) BLOCKS
C.CTXT:'L'                      ;COMMAND TEXT, FOLLOWED BY CR

;+
; STATUS BITS FOR COMMAND BLOCKS
;-

CC.MCR='B'1                     ;FORCE COMMAND TO MCR
CC.PRM='B'2                     ;ISSUE DEFAULT PROMPT
CC.EXT='B'4                     ;TASK EXIT PROMPT REQUEST
CC.KIL='B'10                    ;DELETE ALL CONTINUATION PIECES FROM THIS TTY
CC.CLI='B'20                    ;COMMAND TO BE RETREIVED BY GCCI$ ONLY
CC.MSG='B'40                    ;PACKET CONTAINS SYSTEM MESSAGE TO CLI
CC.TTD='B'100                   ;COMMAND CAME FROM TTDRV
CC.CTC='B'200                   ;^C NOTIFICATION PACKET

; IDENTIFIER CODES FOR SYSTEM TO CLI MESSAGES
;
; CODES 0-127. ARE RESERVED FOR USE BY DIGITAL
; CODES 128.-255. ARE RESERVED FOR USE BY CUSTOMERS
;
CM.INE='B'1                     ;CLI INITIALIZED ENABLED
CM.IND='B'2                     ;CLI INITIALIZED DISABLED
CM.CEN='B'3                     ;CLI ENABLED
CM.CDS='B'4                     ;CLI DISABLED
CM.ELM='B'5                     ;CLI BEING ELIMINATED
CM.EXT='B'6                     ;CLI MUST EXIT IMMEDIATELY
CM.LKT='B'7                     ;NEW TERMINAL LINKED TO CLI
CM.RMT='B'8.                    ;TERMINAL REMOVED FROM CLI
CM.MSG='B'9.                    ;GENERAL MESSAGE TO CLI

;+
; ANCILLARY CONTROL BLOCK (ACB) DEFINITIONS
;-
```

```
.=0
A.REL:'L' .BLKW 1               ;ACD RELOCATION BIAS
A.DIS:'L' .BLKW 1               ;ACD DISPATCH TABLE POINTER
A.MAS:'L' .BLKW 2               ;ACD FUNCTION MASK WORDS
A.NUM:'L' .BLKB 1               ;ACD IDENTIFICATION NUMBER
A.FLEN:'L'.BLKB 1               ;LENGTH IN BYTES OF FULL ACB
A.LIN:'L' .BLKW 1               ;ACD LINK WORD
A.ACC:'L' .BLKB 1               ;ACD ACCESS COUNT
A.STA:'L' .BLKB 1               ;ACD STATUS BYTE
A.PLEN='B'.                     ;LENGTH IN BYTES OF PROTOTYPE ACB
;
.=A.LIN                         ;FULL ACB OVERLAPS PROTOTYPE ACB
A.IMAP:'L' .BLKW 1              ;ACD INTERRUPT BUFFER RELOCATION BIAS
A.IBUF:'L' .BLKW 1              ;ACD INTERRUPT BUFFER ADDRESS
A.ILEN:'L' .BLKW 1              ;ACD INTERRUPT BUFFER LENGTH
A.SMAP:'L' .BLKW 1              ;ACD SYSTEM STATE BUFFER RELOCATION BIAS
A.SBUF:'L' .BLKW 1              ;ACD SYSTEM STATE BUFFER ADDRESS
A.SLEN:'L' .BLKW 1              ;ACD SYSTEM STATE BUFFER LENGTH
A.IOS:'L' .BLKW 2               ;ACD I/O STATUS
A.RES='B'.                      ;START OF ACB RESERVED FOR USE BY THE ACD
;
; DEFINE THE FLAG VALUES IN THE OFFSET U.AFLG
;
UA.ACC='B'1                     ;ACCEPT THIS CHARACTER
UA.PRO='B'2                     ;PROCESS THIS CHARACTER
UA.ECH='B'4                     ;ECHO THIS CHARACTER
UA.TYP='B'10                    ;FORCE THIS CHARACTER INTO TYPEAHEAD
UA.SPE='B'20                    ;THIS CHARACTER HAS A SPECIAL ECHO
UA.PUT='B'40                    ;PUT THIS CHARACTER IN THE INPUT BUFFER
UA.CAL='B'100                   ;CALL THE ACD BACK AFTER THE TRANSFER
UA.COM='B'200                   ;COMPLETE THE INPUT REQUEST
;
UA.ALL='B'400                   ;ALLOW PROCESSING OF THIS I/O REQUEST
UA.TRN='B'1000                  ;TRANSLATE CHARACTERS FROM OUTPUT QIO
UA.TRA='B'2000                  ;TRANSFER CHARACTERS WHEN I/O COMPLETES
;
; DEFINE THE ACD ENTRY POINTS (OFFSETS INTO THE DISPATCH TABLE)
;
.=0
A.ACCE:'L' .BLKW 1              ;I/O REQUEST ACCEPTANCE ENTRY POINT
A.DEQU:'L' .BLKW 1              ;I/O REQUEST DEQUEUE ENTRY POINT
A.POWE:'L' .BLKW 1              ;POWER FAILURE ENTRY POINT
A.INPU:'L' .BLKW 1              ;INPUT COMPLETION ENTRY POINT
A.OUTP:'L' .BLKW 1              ;OUTPUT COMPLETION ENTRY POINT
A.CONN:'L' .BLKW 1              ;CONNECTION ENTRY POINT
A.DISC:'L' .BLKW 1              ;DISCONNECTION ENTRY POINT
A.RECE:'L' .BLKW 1              ;INPUT CHARACTER RECEPTION ENTRY POINT
A.PROC:'L' .BLKW 1              ;INPUT CHARACTER PROCESSING ENTRY POINT
A.TRAN:'L' .BLKW 1              ;OUTPUT QIO CHARACTER TRANSLATION ENTRY POINT
A.CALL:'L' .BLKW 1              ;CALL ACD BACK AFTER TRANSFER ENTRY POINT
```

PKTDF$

```
;
; DEFINE THE STATUS BITS IN A.STA OF THE PROTOTYPE ACB
;
AS.DLT='B'1                     ;ACD IS MARKED FOR DELETE
AS.DIS='B'2                     ;ACD IS DISABLED
        .PSECT

        .MACRO  PKTDF$  X,Y,Z
        .ENDM
        .ENDM
```

```
        .MACRO  SCBDF$,L,B,SYSDEF

;+
; STATUS CONTROL BLOCK
;
; THE STATUS CONTROL BLOCK (SCB) DEFINES THE STATUS OF A DEVICE CONTROLLER.
; THERE IS ONE SCB FOR EACH CONTROLLER IN A SYSTEM. THE SCB IS POINTED TO
; BY UNIT CONTROL BLOCKS. TO EXPAND ON THE TELETYPE EXAMPLE ABOVE, EACH TELE-
; TYPE INTERFACED VIA A DL11-A WOULD HAVE A SCB SINCE EACH DL11-A IS AN IN-
; DEPENDENT INTERFACE UNIT. THE TELETYPES INTERFACED VIA THE DH11 WOULD ALSO
; EACH HAVE AN SCB SINCE THE DH11 IS A SINGLE CONTROLLER BUT MULTIPLEXES MANY
; UNITS IN PARALLEL.
;-

        .IF NB  SYSDEF

        .ASECT
.=0
S.LHD:'L' .BLKW 2               ;CONTROLLER I/O QUEUE LISTHEAD
S.URM:'L'                       ;REFERENCE LABEL

        .IF DF  M$$PRO

        .BLKW   1               ;UNIBUS RUN MASK FOR THE FORK BLOCK

        .ENDC

S.FRK:'L' .BLKW 1               ;FORK BLOCK LINK WORD
        .BLKW   1               ;FORK-PC
        .BLKW   1               ;FORK-R5
        .BLKW   1               ;FORK-R4

        .IF DF  L$$DRV

S.KS5:'L' .BLKW 1               ;FORK KISAR5

        .ENDC

S.PKT:'L' .BLKW 1               ;ADDRESS OF CURRENT I/O PACKET
S.CTM:'L' .BLKB 1               ;CURRENT TIMEOUT COUNT
S.ITM:'L' .BLKB 1               ;INITIAL TIMEOUT COUNT
S.STS:'L' .BLKB 1               ;STATUS (0=FREE, NE 0=BUSY)
S.ST3:'L'.BLKB  1               ;STATUS EXTENSION BYTE
S.ST2:'L' .BLKW 1               ;STATUS EXTENSION
S.KRB:'L' .BLKW 1               ;ADDRESS OF KRB
S.RCNT:'L'.BLKB 1               ;NUMBER OF REGISTERS TO COPY
S.ROFF:'L'.BLKB 1               ;OFFSET TO FIRST DEV REG TO COPY
S.EMB:'L'.BLKW  1               ;ERROR MESSAGE BLOCK POINTER
S.KTB:'L' .BLKW 1               ;START OF MULTI-ACCESS KRBS

        .PSECT

;+
; OFFSETS FOR MSCP/TMSCP DRIVER DATA BASES (MUDRV, DUDRV)
;-

S.PORT='B'S.EMB+2               ;FIRST 3 CHAR. OF PORT NAME IN RAD50
S.PBIA='B'S.EMB+4               ;BIAS OF PORT
S.QST='B'S.EMB+6                ;ADDRESS OF QST (MU,DU CONTR. TABLE)
S.BSYU='B'S.EMB+10              ;UNIT ASSOCIATED WITH OLDEST CMD TO CONTR.

        .IFF
```

SCBDF$

```
;+
; STATUS CONTROL BLOCK  STATUS EXTENSION BIT DEFINITIONS
;-

S2.EIP='B'1                     ;ERROR IN PROGRESS (1=YES)
S2.ENB='B'2                     ;ERROR LOGGING ENABLED (0=YES)
S2.LOG='B'4                     ;ERROR LOGGING SUPPORTED (1=YES)
S2.MAD='B'10                    ;MULTIACCESS DEVICE (1=YES)
S2.LDS='B'40                    ;LOAD SHARING ENABLED (1=YES)
S2.OPT='B'100                   ;SUPPORTS SEEK OPTIMIZATION (1=YES)
S2.CON='B'200                   ;SCB AND KRB ARE CONTIGUOUS (1=YES)
S2.OP1='B'400                   ;THESE TWO BITS DEFINE THE OPTIMIZATION
S2.OP2='B'1000                  ;METHOD.
                                ;OP2,OP1=0,0 INDICATES NEAREST CYLINDER
                                ;OP2,OP1=0,1 INDICATES ELEVATOR
                                ;OP2,OP1=1,0 INDICATES C-SCAN
                                ;OP2,OP1=1,1 RESERVED
S2.ACT='B'2000                  ;DRIVER HAS OPERATION OUTSTANDING (1=YES)
S2.XHR='B'4000                  ;EXTERNAL HEADER AND NEW I.LN2 SUPPORT
S2.KRQ='B'10000                 ;SCB IS QUEUED IN CONTROLLER REQUEST QUEUE

;+
; STATUS CONTROL BLOCK STATUS EXTENSION (S.ST3) DEFINITIONS
;-

S3.DRL='B'1                     ;MULTI-ACCESS DRIVE IN RELEASED STATE (1=YES)
S3.NRL='B'2                     ;DRIVER SHOULDN'T RLS MULTI-ACCESS DRIVE (1=YES)
S3.SIP='B'4                     ;SEEK IN PROGRESS (1=YES)
S3.ATN='B'10                    ;DRIVER MUST CLEAR ATTENTION BIT (1=YES)
S3.SLV='B'20                    ;DEVICE USES SLAVE UNITS (1=YES)
S3.SPA='B'40                    ;PORT 'A' SPINNING UP
S3.SPB='B'100                   ;PORT 'B' SPINNING UP
S3.OPT='B'200                   ;SEEK OPTIMIZATION ENABLED (1=YES)
S3.SPU='B'S3.SPA!S3.SPB         ;.OR. OF PORT SPINUP BITS

;+
; KRB ADDRESS TABLE (S.KTB) PORT OFFLINE FROM THIS SCB FLAG.
;-

KP.OFL='B'1                     ;KRB ADDRESS POINTS TO OFFLINE PORT (1=YES)
```

```
;+
; MAPPING ASSIGNMENT BLOCK (FOR UNIBUS MAPPING REGISTER ASSIGNMENT)
;-

        .ASECT
.=0
M.LNK:'L'   .BLKW 1             ;LINK WORD
M.UMRA:'L'  .BLKW 1             ;ADDRESS OF FIRST ASSIGNED UMR
M.UMRN:'L'  .BLKW 1             ;NUMBER OF UMR'S ASSIGNED * 4
M.UMVL:'L'  .BLKW 1             ;LOW 16 BITS MAPPED BY 1ST ASSIGNED UMR
M.UMVH:'L'  .BLKB 1             ;HIGH 2 BITS MAPPED IN BITS 4 AND 5
M.BFVH:'L'  .BLKB 1             ;HIGH 6 BITS OF PHYSICAL BUFFER ADDRESS
M.BFVL:'L'  .BLKW 1             ;LOW 16 BITS OF PHYSICAL BUFFER ADDRESS
M.LGTH='B'.                     ;LENGTH OF MAPPING ASSIGNMENT BLOCK

        .ENDC

        .PSECT

        .MACRO  SCBDF$,X,Y,Z
        .ENDM
        .ENDM
```

SHDDF$

```
        .MACRO  SHDDF$,L,B,SYSDEF

;
; FIRST, WE MUST DEFINE THE I/O PACKET DEFINITIONS, SINCE WE
; USE THEM IN OUR DEFINITIONS.
;
        PKTDF$                  ;DEFINE I/O PACKET DEFINITIONS

;+
; SHADOW RECORDING LINKAGE BLOCK (UMB)
;
; THE UMB LINKS TOGETHER TWO UCB'S AS A SHADOW SET.  ONE IS THE
; PRIMARY UCB, THE OTHER THE SECONDARY UCB.  THE EXISTANCE OF A
; UMB SIGNALS THAT SHADOW RECORDING IS ENABLED ON A PARTICULAR
; UCB.
;-

        .ASECT

.=0
M.LNK:'L' .BLKW 1               ;LINKAGE OF ALL UMB'S IN THE SYSTEM
M.LHD:'L' .BLKW 1               ;LISTHEAD OF ALL ML NODES FOR THIS SET
M.UCBS:'L'.BLKW 2               ;PRIMARY AND SECONDARY UCB ADDRESSES
M.STS:'L' .BLKW 1               ;STATUS WORD
M.LBN:'L' .BLKB 1               ;HIGH ORDER BYTE OF FENCE
          .BLKB 1               ;UNUSED BYTE (MAYBE STATUS?)
          .BLKW 1               ;LOW ORDER WORD OF FENCE
M.LGH=.

;+
; UMB STATUS BIT DEFINITIONS
;-

        .PSECT

MS.MDA'B'=1                     ;UMB MARKED FOR DEALLOCATION (1=YES)
MS.CHP'B'=2                     ;CATCHUP IN PROGRESS (1=YES)

;+
; DEFINE THE OFFSETS FOR THE ML NODE, LINKED OFF OF THE UMB
; THROUGH CELL M.LHD.  THIS NODE CONTAINS THE SECONDARY I/O
; PACKET, AND DOUBLES AS THE ERROR PACKET TO THE ERROR MESSAGE
; TASK.
;-

        .ASECT

.=0
ML.LNK:'L' .BLKW 1              ;LINKAGE OF ALL ML NODES ON UMB
ML.LEN:'L' .BLKB 1              ;LENGTH OF ML NODE FOR DEALLOCATION
ML.TYP:'L' .BLKB 1              ;TYPE OF ML NODE FOR ERROR TASK
ML.DNC:'L' .BLKB 1              ;DONE COUNT OF PACKETS
           .BLKB 1              ;UNUSED
ML.PRI:'L' .BLKW 1              ;PRIMARY I/O PACKET ADDRESS
ML.PKT:'L' .BLKB I.LGTH         ;SECONDARY I/O PACKET
ML.LGH=.

;+
; ML NODE TYPE CODES
;-

        .PSECT
```

```
MT.PKT'B'=1                     ;ML NODE IS I/O PACKET TYPE

;+
; I/O PACKET OFFSET DEFNS FOR USE BY SHADOW RECORDING
;-

I.RO'B'=I.PRM                   ;STATUS STORAGE FOR RO STATUS
I.R1'B'=I.PRM+2                 ;STATUS STORAGE FOR R1 STATUS

;+
; DEFINE THE ERROR MESSAGE POINTERS THAT RESIDE IN THE I/O PACKET.
;-

        .PSECT

ML.FID'B'=ML.PKT+I.IOSB         ;FILE ID WHICH CONTAINS ERROR
ML.FSEQ'B'=ML.PKT+I.IOSB+2      ;FILE SEQUENCE NUMBER OF FILE IN ERROR
ML.LBN'B'=ML.PKT+I.PRM+10       ;HIGH ORDER LBN OF BLOCK(S) IN ERROR
ML.CNT'B'=ML.PKT+I.PRM+4        ;NUMBER OF BLOCKS IN BAD XFER
ML.TCB'B'=ML.PKT+I.TCB          ;TCB OF TASK WITH BAD REQUEST
ML.SRO'B'=ML.PKT+I.RO           ;RO OF SECONDARY I/O PACKET
ML.SR1'B'=ML.PKT+I.R1           ;R1 OF SECONDARY I/O PACKET
ML.PRO'B'=ML.PKT+I.PRM+14       ;RO OF PRIMARY I/O PACKET
ML.PR1'B'=ML.PKT+I.PRM+16       ;R1 OF PRIMARY I/O PACKET

        .MACRO  SHDDF$,X,Y,Z
        .ENDM
        .ENDM
```

TCBDF$

```
        .MACRO  TCBDF$,L,B,SYSDEF
;+
; TASK CONTROL BLOCK OFFSET AND STATUS DEFINITIONS
;
; TASK CONTROL BLOCK
;-
        .ASECT
.=0
T.LNK:'L' .BLKW 1               ;UTILITY LINK WORD
T.PRI:'L' .BLKB 1               ;TASK PRIORITY
T.IOC:'L' .BLKB 1               ;I/O PENDING COUNT
T.PCBV:'L' .BLKW 1              ;POINTER TO COMMON PCB VECTOR
T.NAM:'L' .BLKW 2               ;TASK NAME IN RAD50
T.RCVL:'L' .BLKW 2              ;RECEIVE QUEUE LISTHEAD
T.ASTL:'L' .BLKW 2              ;AST QUEUE LISTHEAD
T.EFLG:'L' .BLKW 2              ;TASK LOCAL EVENT FLAGS 1-32
T.UCB:'L' .BLKW 1               ;UCB ADDRESS FOR PSEUDO DEVICE 'TI'
T.TCBL:'L' .BLKW 1              ;TASK LIST THREAD WORD
T.STAT:'L' .BLKW 1              ;FIRST STATUS WORD (BLOCKING BITS)
T.ST2:'L' .BLKW 1               ;SECOND STATUS WORD (STATE BITS)
T.ST3:'L' .BLKW 1               ;THIRD STATUS WORD (ATTRIBUTE BITS)
T.DPRI:'L' .BLKB 1              ;TASK'S DEFAULT PRIORITY
T.LBN:'L' .BLKB 3               ;LBN OF TASK LOAD IMAGE
T.IID='B' T.LBN+1               ;IMAGE INDEX FOR CPR SYSTEM
T.LDV:'L' .BLKW 1               ;UCB ADDRESS OF LOAD DEVICE
T.PCB:'L' .BLKW 1               ;PCB ADDRESS OF TASK PARTITION
T.MXSZ:'L' .BLKW 1              ;MAXIMUM SIZE OF TASK IMAGE (MAPPED ONLY)
T.ACTL:'L' .BLKW 1              ;ADDRESS OF NEXT TASK IN ACTIVE LIST
T.ATT:'L' .BLKW 2               ;ATTACHMENT DESCRIPTOR LISTHEAD
T.ST4:'L' .BLKW 1               ;FOURTH TASK STATUS WORD
T.HDLN:'L' .BLKB 1              ;LENGTH OF HEADER (0 IF HDR IN POOL)
        .BLKB   1               ;UNUSED
T.GGF:'L' .BLKB 1               ;GROUP GLOBAL USE COUNT FOR TASK
T.TIO:'L' .BLKB 1               ;BUFFERED I/O IN PROGRESS COUNT
T.EFLM:'L' .BLKW 2              ;TASK WAITFOR MASK/ADDRESS
T.TKSZ:'L' .BLKW 1              ;TASK LOAD SIZE IN 32 WD BLOCKS

$$$=.                           ;MARK START OF PLAS AREA

T.OFF:'L' .BLKW 1               ;OFFSET TO TASK IMAGE IN PARTITION
        .BLKB 1                 ;RESERVED
T.SRCT:'L' .BLKB 1              ;SREF WITH EFN COUNT IN ALL RECEIVE QUEUES
T.RRFL:'L' .BLKW 2              ;RECEIVE BY REFERENCE LISTHEAD

        .IF NDF P$$LAS

.=$$$                           ;MOVE LC BACK TO START OF PLAS AREA

        .ENDC

        .IF NB  SYSDEF

$$$=.

T.CTX:'L' .BLKW 1               ;POINTER TO CONTEXT BLOCK (DDS)

        .IF NDF N$$DIR

.=$$$
```

```
        .ENDC   ; NDF N$$DIR
$$$=.                           ;MARK START OF PARENT/OFFSPRING AREA
T.OCBH:'L' .BLKW 2              ;OFFSPRING CONTROL BLOCK LISTHEAD
T.RDCT:'L' .BLKW 1              ;OUTSTANDING OFFSPRING AND VT: COUNT

        .IF NDF P$$OFF
.=$$$
        .ENDC

T.SAST:'L' .BLKW 1              ;SPECIFY AST LIST HEAD
$$$=.
T.RRM:'L'.BLKW  1               ;REQUIRED RUN MASK
T.IRM:'L'.BLKW  1               ;INITIAL RUN MASK SET UP BY INSTALL
                                ;**** THIS WORD IS NO LONGER NECESSARY
                                ;**** HOWEVER, INSTALL (INSLB), MCR (SPAWN),
                                ;**** AND VMR MUST BE MODIFIED
T.CPU:'L'.BLKB  1               ;PROCESSOR NUMBER ON WHICH TASK LAST EXECUTED
        .BLKB   1               ;(UNUSED)

        .IF NDF M$$PRO
.=$$$
        .ENDC

$$$=.
T.ACN:'L'.BLKW  1               ;POINTER TO ACCOUNTING BLOCK

        .IF NDF A$$CNT
.=$$$
        .ENDC

$$$=.
T.ISIZ:'L'.BLKW 1               ;SIZE OF ROOT I SPACE
        .IF NDF U$$DAS
.=$$$
        .ENDC   ; NDF U$$DAS

T.LGTH='B'.                     ;LENGTH OF TASK CONTROL BLOCK
T.EXT='B'0                      ;LENGTH OF TCB EXTENSION
        .IFF

;+
; TASK STATUS DEFINITIONS
;
; FIRST STATUS WORD (BLOCKING BITS)
;-
```

```
TS.EXE='B'100000                ;TASK NOT IN EXECUTION (1=YES)
TS.RDN='B'40000                 ;I/O RUN DOWN IN PROGRESS (1=YES)
TS.MSG='B'20000                 ;ABORT MESSAGE BEING OUTPUT (1=YES)
TS.CIP='B'10000                 ;TASK BLOCKED FOR CHECKPOINT IN PROGRESS (1=YES)
TS.RUN='B'4000                  ;TASK IS RUNNING ON ANOTHER PROCESSOR (1=YES)
TS.RSW='B'2000                  ;TASK IS WAITING FOR A REMOTE SERVICE          ; KLN017
TS.STP='B'1000                  ;TASK BLOCKED BY CLI COMMAND
TS.CKR='B'100                   ;TASK HAS CKP REQUEST (MP SYSTEM ONLY) (1=YES)
TS.BLC='B'37                    ;INCREMENT BLOCKING COUNT MASK

;+
; TASK BLOCKING STATUS MASK
;-

TS.BLK='B'177777

;+
; SECOND STATUS WORD (STATE BITS)
;-

T2.AST='B'100000                ;AST IN PROGRESS (1=YES)
T2.DST='B'40000                 ;AST RECOGNITION DISABLED (1=YES)
T2.CHK='B'20000                 ;TASK NOT CHECKPOINTABLE (1=YES)
T2.REX='B'10000                 ;REQUESTED EXIT AST SPECIFIED
T2.SEF='B'4000                  ;TASK STOPPED FOR EVENT FLAG(S) (1=YES)
T2.SIO='B'1000                  ;TASK STOPPED FOR BUFFERED I/O
T2.AFF='B'400                   ;TASK IS INSTALLED WITH AFFINITY
T2.HLT='B'200                   ;TASK IS BEING HALTED (1=YES)
T2.ABO='B'100                   ;TASK MARKED FOR ABORT (1=YES)
T2.STP='B'40                    ;SAVED T2.SPN ON AST IN PROGRESS
T2.STP='B'20                    ;TASK STOPPED (1=YES)
T2.SPN='B'10                    ;SAVED T2.SPN ON AST IN PROGRESS
T2.SPN='B'4                     ;TASK SUSPENDED (1=YES)
T2.WFR='B'2                     ;SAVED T2.WFR ON AST IN PROGRESS
T2.WFR='B'1                     ;TASK IN WAITFOR STATE (1=YES)

;+
; THIRD STATUS WORD (ATTRIBUTE BITS)
;-

T3.ACP='B'100000                ;ANCILLARY CONTROL PROCESSOR (1=YES)
T3.PMD='B'40000                 ;DUMP TASK ON SYNCHRONOUS ABORT (0=YES)
T3.REM='B'20000                 ;REMOVE TASK ON EXIT (1=YES)
T3.PRV='B'10000                 ;TASK IS PRIVILEGED (1=YES)
T3.MCR='B'4000                  ;TASK REQUESTED AS EXTERNAL MCR FUNCTION (1=YES)
T3.SLV='B'2000                  ;TASK IS A SLAVE TASK (1=YES)
T3.CLI='B'1000                  ;TASK IS A COMMAND LINE INTERPRETER (1=YES)
T3.RST='B'400                   ;TASK IS RESTRICTED (1=YES)
T3.NSD='B'200                   ;TASK DOES NOT ALLOW SEND DATA
T3.CAL='B'100                   ;TASK HAS CHECKPOINT SPACE IN TASK IMAGE
T3.ROV='B'40                    ;TASK HAS RESIDENT OVERLAYS
T3.NET='B'20                    ;NETWORK PROTOCOL LEVEL
T3.MPC='B'10                    ;MAPPING CHANGE WITH OUTSTANDING I/O (1=YES)
T3.CMD='B'4                     ;TASK IS EXECUTING A CLI COMMAND
T3.SWS='B'2                     ;RESERVED FOR SOFTWARE SERVICES USE
T3.GFL='B'1                     ;GROUP GLOBAL EVENT FLAG LOCK

;+
; STATUS BIT DEFINITIONS FOR FOURTH STATUS WORD (T.ST4)
;-
```

```
T4.LRW='B'2000                  ;TASK IS WAITING FOR A LOCAL RING                ; KLN017
T4.LBW='B'1000                  ;TASK IS WAITING FOR A LOCAL BUFFER              ; KLN017
T4.DFB='B'400                   ;TASK HAS DEFERRED BINDING ENABLED               ; KLF010
T4.FMP='B'200                   ;TASK HAS FAST MAP HDR EXT.
T4.CTC='B'100                   ;TASK HAS BEEN PROCESSED BY GIN ^C ABORT
T4.MUT='B'40                    ;TASK IS A MULTI-USER TASK
T4.LDD='B'20                    ;TASK'S LOAD DEVICE HAS BEEN DISMOUNTED
T4.PRO='B'10                    ;TCB IS (OR SHOULD BE) A PROTOTYPE
T4.PRV='B'4                     ;TASK WAS PRIV, BUT HAS CLEARED T3.PRV
                                ;WITH GIN (MAY RESET WITH GIN IF T4.PRV SET)
T4.DSP='B'2                     ;TASK WAS BUILT FOR USER I/D SPACE
T4.SNC='B'1                     ;TASK USES COMMONS FOR SYNCHRONIZATION

;+
; REQUIRED RUN MASK
;-

TR.UBT='B'100000                ;UNIBUS RUN T
TR.UBS='B'40000                 ;UNIBUS RUN S
TR.UBR='B'20000                 ;UNIBUS RUN R
TR.UBP='B'10000                 ;UNIBUS RUN P
TR.UBN='B'4000                  ;UNIBUS RUN N
TR.UBM='B'2000                  ;UNIBUS RUN M
TR.UBL='B'1000                  ;UNIBUS RUN L
TR.UBK='B'400                   ;UNIBUS RUN K
TR.UBJ='B'200                   ;UNIBUS RUN J
TR.UBH='B'100                   ;UNIBUS RUN H
TR.UBF='B'40                    ;UNIBUS RUN F
TR.UBE='B'20                    ;UNIBUS RUN E
TR.CPD='B'10                    ;PROCESSOR D
TR.CPC='B'4                     ;PROCESSOR C
TR.CPB='B'2                     ;PROCESSOR B
TR.CPA='B'1                     ;PROCESSOR A

        .ENDC

        .PSECT
        .MACRO  TCBDF$  X,Y,Z
        .ENDM
        .ENDM
```

UCBDF$

```
        .MACRO  UCBDF$,L,B,TTDEF,SYSDEF
;+
; UNIT CONTROL BLOCK
;
; THE UNIT CONTROL BLOCK (UCB) DEFINES THE STATUS OF AN INDIVIDUAL DEVICE
; UNIT AND IS THE CONTROL BLOCK THAT IS POINTED TO BY THE FIRST WORD OF
; AN ASSIGNED LUN. THERE IS ONE UCB FOR EACH DEVICE UNIT OF EACH DCB. THE
; UCB'S ASSOCIATED WITH A PARTICULAR DCB ARE CONTIGUOUS IN MEMORY AND ARE
; POINTED TO BY THE DCB. UCB'S ARE VARIABLE LENGTH BETWEEN DCB'S BUT ARE
; OF THE SAME LENGTH FOR A SPECIFIC DCB. TO FINISH THE TELETYPE EXAMPLE ABOVE,
; EACH UNIT ON BOTH INTERFACES WOULD HAVE A UCB.
;-
        .ASECT
.=177772

        .IF NB  SYSDEF

        .IF DF A$$CNT
.=.-2
        .ENDC   ;DF A$$CNT

        .IF DF L$$GCL
.=.-2
        .ENDC   ;DF L$$GCL

.=.-2

        .IF DF  N$$DIR
.=.-2
        .ENDC   ;DF N$$DIR

U.UAB:'L'
        .IF DF  A$$CNT
        .BLKW   1               ;POINTER TO USER ACCOUNT BLOCK
        .ENDC   ;DF A$$CNT

U.LOG:'L'
        .IF DF  L$$GCL
        .BLKW 1         ;POINTER TO USER LOGICAL HASH TABLE
        .ENDC   ;DF L$$GCL

U.FPRO:'L'
        .BLKW 1         ;DEFAULT FILE PROTECTION WORD

U.CTX:'L'
        .IF DF N$$DIR
        .BLKW 1         ;POINTER TO TERMINAL CONTEXT BLOCK
        .ENDC   ;DF N$$DIR

        .ENDC   ;NB SYSDEF
```

```
U.MUP:'L' .BLKW 1               ;MULTI-USER PROTECTION WORD
U.LUIC:'L' .BLKW 1              ;LOGIN UIC - MULTI USER SYSTEMS ONLY
U.OWN:'L' .BLKW 1               ;OWNING TERMINAL - MULTI USER SYSTEMS ONLY
U.DCB:'L' .BLKW 1               ;BACK POINTER TO DCB
U.RED:'L' .BLKW 1               ;POINTER TO REDIRECT UNIT UCB
U.CTL:'L' .BLKB 1               ;CONTROL PROCESSING FLAGS
U.STS:'L' .BLKB 1               ;UNIT STATUS
U.UNIT:'L' .BLKB 1              ;PHYSICAL UNIT NUMBER
U.ST2:'L' .BLKB 1               ;UNIT STATUS EXTENSION
U.CW1:'L' .BLKW 1               ;FIRST DEVICE CHARACTERISTICS WORD
U.CW2:'L' .BLKW 1               ;SECOND DEVICE CHARACTERISTICS WORD
U.CW3:'L' .BLKW 1               ;THIRD DEVICE CHARACTERISTICS WORD
U.CW4:'L' .BLKW 1               ;FOURTH DEVICE CHARACTERISTICS WORD
U.SCB:'L' .BLKW 1               ;POINTER TO SCB
U.ATT:'L' .BLKW 1               ;TCB ADDRESS OF ATTACHED TASK
U.BUF:'L' .BLKW 1               ;RELOCATION BIAS OF CURRENT I/O REQUEST
        .BLKW   1               ;BUFFER ADDRESS OF CURRENT I/O REQUEST
U.CNT:'L' .BLKW 1               ;BYTE COUNT OF CURRENT I/O REQUEST
U.UCBX='B'U.CNT+2               ;POINTER TO UCB EXTENSION IN SECONDARY POOL
U.ACP='B'U.CNT+4                ;ADDRESS OF TCB OF MOUNTED ACP
U.VCB='B'U.CNT+6                ;ADDRESS OF VOLUME CONTROL BLOCK
U.CBF='B'U.CNT+2                ;CONTROL BUFFER RELOCATION AND ADDRESS
U.UMB='B'U.CNT+10               ;ADDRESS OF UMB FOR SHADOW RECORDING
U.PRM='B'U.CNT+12               ;DISK SIZE PARAMETER WORDS
U.ICSR='B'U.CNT+16              ;CSR ADDRESS (P/OS)
U.SLT='B'U.CNT+20               ;SLOT ADRESS (P/OS)
U.SPRM='B'U.CNT+22              ;4 WD SAVED I/O PACKET AREA (R$$AMD)
U.UTIL='B'U.CNT+16              ;STATE WORD FOR UNIT

;
; DEFINITIONS FOR U.UTIL BITS
;

UU.SER  ='B'1           ;SERIAL MODE
UU.RCT  ='B'2           ;(DUDRV)RCT IN PROGRESS
UU.AVN  ='B'4           ;UNIT IS WAITING FOR OTHER UNITS TO SPIN DOWN
UU.GUS  ='B'10          ;UNIT MUST HAVE A GUS COMMAND ISSUED
UU.ONL  ='B'20          ;UNIT MUST HAVE A ONL COMMAND ISSUED
UU.SPC  ='B'40          ;SPECIAL ONLINE TRANSITION
UU.ATN  ='B'100         ;UNIT HAS SENT ATTENTION MESSAGE
UU.RDY  ='B'200         ;UNIT IS READY
UU.ABO  ='B'400         ;IF SET, XXCAN SET UU.SER FLAG FOR UNIT
UU.SIO  ='B'1000        ;THIS UNIT CAN STALL I/O
UU.IOS  ='B'2000        ;THIS UNIT HAS I/O STALLED
UU.BLK  ='B'4000        ;THIS UNIT DOESN'T ACCEPT DENSITY SETTINGS

U.BPKT='B'U.CNT+20              ;UNIT BAD BLOCK REPLACEMENT WAITING LIST
U.MEDI='B'U.BPKT                ;MEDIA IDENTIFIER FOR MU TAPE
U.UC2X='B'U.CNT+24              ;POINTER TO SECOND EXTENSION IN SECONDARY POOL

;
; MAGTAPE DEVICE DEPENDANT UCB OFFSETS
;

U.SNUM='B'U.CNT+10              ;SLAVE UNIT NUMBER
U.FCDE='B'U.CNT+12              ;FUNCTION CODE
U.KRB1='B'U.CNT+14              ;SUBCONTROLLER KRB1 POINTER
;
```

UCBDF$

```
;
; DEFINE SECONDARY POOL UCB EXTENSION OFFSETS (ERROR LOGGING DEVICES ONLY)
;
.=0
        .BLKW   9.              ;FIXED ACCOUNTING TRANSACTION HEADER
X.NAME:'L' .BLKW 2              ;DRIVE NAME IN RAD50
X.IOC:'L' .BLKW 2               ;I/O COUNT
X.ERSL:'L' .BLKB 1              ;SOFT ERROR LIMIT
X.ERHL:'L' .BLKB 1              ;HARD ERROR LIMIT
X.ERSC:'L' .BLKB 1              ;SOFT ERROR COUNT
X.ERHC:'L' .BLKB 1              ;HARD ERROR COUNT
X.WCNT:'L' .BLKW 2              ;WORDS TRANSFERED COUNT

;
; DEFINE OFFSETS FOR SEEK OPTIMIZATION DEVICES
;

X.CYLC:'L' .BLKW 2              ;CYLINDERS CROSSED COUNT
X.CCYL:'L' .BLKW 1              ;CURRENT CYLINDER
X.FCUR:'L' .BLKB 1              ;CURRENT FAIRNESS COUNT
X.FLIM:'L'                      ;FAIRNESS COUNT LIMIT
X.DSKD:'L' .BLKB 1              ;DISK DIRECTION (HIGH BIT 1=OUT)

X.DNAM:'L' .BLKW 1              ;DEVICE NAME FOR ACCOUNTING
X.UNIT:'L' .BLKB 1              ;UNIT NUMBER FOR ACCOUNTING
X.CSTS:'L' .BLKB 1              ;CACHE STATUS BITS
X.CPCB:'L' .BLKW 1              ;CACHE PARTITION PCB ADDRESS
X.CSBA:'L' .BLKW 1              ;CACHE STATISTICS BUFFER ADDRESS (BIAS)
X.CCED:'L' .BLKW 2              ;CACHE EXTENT DESCRIPTOR LISTHEAD
X.XDAT:'L' .BLKB 1              ;CACHE VIRTUAL EXTENT SIZE
X.XRDA:'L' .BLKB 1              ;CACHE READAHEAD EXTENT SIZE
X.XDIR:'L' .BLKB 1              ;CACHE DIRECTORY EXTENT SIZE
X.XLOG:'L' .BLKB 1              ;CACHE LOGICAL EXTENT SIZE
X.XOVR:'L' .BLKB 1              ;CACHE OVERLAY EXTENT SIZE
X.CST2:'L' .BLKB 1              ;CACHE STATUS BYTE 2
X.LGTH='B'.                     ;LENGTH OF THE UCB EXTENSION
X.DFFL='B'10.                   ;DEFAULT FAIRNESS COUNT LIMIT
X.DFSL='B'8.                    ;DEFAULT SOFT ERROR LIMIT
X.DFHL='B'5.                    ;DEFAULT HARD ERROR LIMIT

;
; CACHE STATUS BITS IN X.CSTS
;

XC.ENA='B'200                   ;AUTOCACHE ENABLED (1=YES)
XC.ACT='B'100                   ;CACHE ACTIVE FOR DEVICE (1=YES)
XC.DFR='B'040                   ;TEMPORARY FILE WRITE DEFERRED SUPPORT (1=YES)

XC.DIR='B'020                   ;CACHE DIRECTORY REQUESTS (1=YES)
XC.OVR='B'010                   ;CACHE OVERLAY REQUESTS (1=YES)
XC.DAT='B'004                   ;CACHE VIRTUAL REQUESTS (1=YES)
XC.LOG='B'002                   ;CACHE LOGICAL REQUESTS (1=YES)
XC.RDA='B'001                   ;CACHE VIRTUAL READ AHEAD (1=YES)

;
; CACHE STATUS BITS IN X.CST2
;

X2.DEA='B'1                     ;CACHE IS CURRENTLY BEING DEACTIVATED (1=YES)
```

```
;
; DEFINE CACHE MAXIMUM AND DEFAULT EXTENT SIZES
;
XX.MAX='B'127.                  ;MAXIMUM EXTENT SIZE

XX.DAT='B'5.                    ; DEFAULT VIRTUAL EXTENT SIZE
XX.RDA='B'5.                    ; DEFAULT READAHEAD EXTENT SIZE
XX.DIR='B'1.                    ; DEFAULT DIRECTORY EXTENT SIZE
XX.LOG='B'1.                    ; DEFAULT LOGICAL EXTENT SIZE
XX.OVR='B'4.                    ; DEFAULT OVERLAY EXTENT SIZE

;
; DEFINE OFFSETS FOR DISK MSCP CONTROLLERS (SECOND UCB EXTENSION)
;

;
; CHARACTERISTICS OBTAINED FROM "GET UNIT STATUS" END PACKETS
;

.=0
X.MLUN:'L'.BLKW 1               ;MULTI-UNIT CODE
X.UNFL:'L'.BLKW 1               ;UNIT FLAGS
       .BLKW   2                ;RESERVED
X.UNTI:'L'.BLKW 4               ;UNIT IDENTIFIER
X.MEDI:'L'.BLKW 2               ;MEDIA IDENTIFIER
X.SHUN:'L'.BLKW 1               ;SHADOW UNIT
X.SHST:'L'.BLKW 1               ;SHADOW UNIT STATUS
X.TRCK:'L'.BLKW 1               ;UNIT TRACK SIZE
X.GRP:'L'.BLKW  1               ;UNIT GROUP SIZE
X.CYL:'L'.BLKW  1               ;UNIT CYLINDER SIZE
X.USVR:'L'.BLKB 1               ;UNIT SOFTWARE VERSION
X.UHVR:'L'.BLKB 1               ;UNIT HARDWARE VERSION
X.RCTS:'L'.BLKW 1               ;UNIT RCT TABLE SIZE
X.RBNS:'L'.BLKB 1               ;UNIT RBN 'S / TRACK
X.RCTC:'L'.BLKB 1               ;UNIT RCT COPIES
;
; CHARACTERISTICS OBTAINED FROM "ONLINE" OR "SET UNIT CHARACTERISTICS" END
; PACKETS
;
X.UNSZ:'L'.BLKW 2               ;UNIT SIZE
X.VSER:'L'.BLKW 2               ;VOLUME SERIAL NUMBER

X.DUSZ='B'.                     ;SIZE OF DISK MSCP CONTROLLER UCB EXTENTION

       .IF NB  TTDEF

;
;
```

UCBDF$

```
; TERMINAL DRIVER DEFINITIONS
;
.=U.BUF
U.TAPR:'L'                      ;APR VALUE FOR START OF UCBX
U.TUX:'L' .BLKW 1               ;POINTER TO UCB EXTENSION (UCBX)
U.TSTA:'L' .BLKW 4              ;STATUS QUADRUPLE-WORD
U.UIC:'L' .BLKW 1               ;DEFAULT UIC
U.TFRQ:'L' .BLKW 1              ;FORK REQUEST WORD
U.TFLK:'L' .BLKW 1              ;FORK LIST LINK WORD
U.TCHP:'L' .BLKB 1              ;CURRENT HORIZONTAL POSITION
U.TCVP:'L' .BLKB 1              ;CURRENT VERTICAL POSITION
U.TTYP:'L' .BLKB 1              ;TERMINAL TYPE
U.TMTI:'L' .BLKB 1              ;MODEM TIMER
U.TTAB:'L' .BLKW 1              ;IF 0: U.TTAB+1 IS SINGLE-CHARACTER TYPE-AHEAD
                                ;       BUFFER, CURRENTLY EMPTY
                                ;IF ODD: U.TTAB+1 IS SINGLE-CHARACTER TYPE-AHEAD
                                ;       BUFFER AND HOLDS A CHARACTER
                                ;IF NON-0 AND EVEN: POINTER TO MULTI-CHARACTER
                                ;       TYPE-AHEAD BUFFER
        .=.-2                   ;THE NEXT TWO OFFSETS OVERLAP U.TTAB WHEN THE
                                ;TYPEAHEAD BUFFER IS IN SECONDARY POOL
U.TECO:'L' .BLKB 1              ;ECHO BUFFER FOR DMA OPERATIONS WHEN UCBX IS
                                ;IN SECONDARY POOL AND THUS NOT MAPPED BY A UMR
U.TBSZ:'L' .BLKB 1              ;TYPEAHEAD BUFFER SIZE
U.TLPP:'L' .BLKW 1              ;LINES PER PAGE
U.TST5:'L' .BLKW 1              ;ADDITIONAL STATUS BITS
U.TST6:'L' .BLKW 1              ;EXTENDED I/O STATUS WORD
U.TIXL:'L' .BLKW 1              ;I/O PACKET EXTENSION LISTHEAD
U.ACB:'L' .BLKW 1               ;ANCILLARY CONTROL DRIVER BLOCK ADDR
U.AFLG:'L' .BLKW 1              ;ANCILLARY CONTROL DRIVER FLAGS WORD
U.ADMA:'L' .BLKW 1              ;ANCILLARY CONTROL DRIVER DMA BUFFER

;
; LAT Host Support
;
                        ;STATUS/CONTROL INFORMATION
U.LINS:'L' .BLKB 1              ;LINK STATUS
U.CREN:'L' .BLKB 1              ;TRANSMIT CREDITS COUNTER AND STATUS EXTENSION
                        ;SERVER/CIRCUIT IDENTIFICATION
U.SRVN:'L' .BLKW 1              ;SERVER NUMBER
U.SESN:'L' .BLKB 1              ;SESSION NUMBER
U.SLSZ:'L' .BLKB 1              ;MAXIMUM SLOT SIZE ON XMT
                        ;PARAMETRS ON RECEIVE DATA
U.RSBB:'L' .BLKW 1              ;RECEIVE SLOT BIAS
U.RBHA:'L' .BLKW 1              ;RECEIVE SLOT HEADER VIRTUAL
U.RSDV:'L' .BLKW 1              ;RECEIVE DATA VIRTUAL
                        ;XMT INTERMEDIATE BUFFER CHAIN
U.TRLH:'L' .BLKW 1              ;XMT LISTHEAD ADDRESS
U.TRSC:'L' .BLKW 1              ;XMT REMAINED BYTES IN BUFFER
                        ;VIRTUAL CIRCUIT CCB
U.CCBA:'L' .BLKW 1              ;CCB ADDRESS (IN POOL)
;
; LINK STATUS WORD IN U.LINS
;
UL.TRS='B'1                     ;1-XMT STOPPED, 0-NOT
                                ;>255 CHAR. FOR SLOT. HOLD IT.
UL.TDA='B'2                     ;1-XMT DATA AVAIL, 0-NOT
                                ;DATA IN XMT QUEUE (SET BY PORT)
UL.LST='B'4                     ;1-LINE STOPPED, 0-NOT
```

```
                                ;LINE STOPPED BY USER (XOFF
                                ;CAME FROM REMOTE TERMINAL)
UL.RDA='B'10                    ;1-RCV DATA AVAIL, 0-NOT
                                ;DATA IN RCV QUEUE (SET BY PROC)
UL.ECH='B'20                    ;BUFFER ALLOCATION FAILURE
                                ;FOR 1 ECHO CHARACTER
UL.RSS='B'100                   ;1-RCV STOPPED, 0-NOT
                                ;RCV STOPPED BY TTDRV ON RECEIVE
                                ;IF NO RESOURCES AVAILABLE
UL.LEN='B'200                   ;1-LINK ENABLE, 0-NOT
                                ;TOGGLED BY START/STOP LINK CALL
;
;
; LINK STATUS EXTENSION (HIGH 4 BITS OF U.CREN)
;
LE.HIP='B'20                    ;HANGUP IN PROGRESS. SET IF LHMTIM IS
                                ;CALLED BEFORE ALL DATA HAS BEEN SENT
LE.TDC='B'40                    ;TRANSMIT DATA COMPLETE.  BIT SET DURING
                                ;.HIP WHEN ALL DATA HAS BEEN SENT
LE.CIP='B'100                   ;CONNECT IN PROGRESS.  SET WHEN A CONNECTION
                                ;TO A TERMINAL SERVER IS PENDING.
; DEFINE BITS IN STATUS WORD 1 (U.TSTA)
;
;
; I N P U T   S T A T U S
;
S1.RST='B'1                     ;READ WITH SPECIAL TERMINATORS IN PROGRESS
S1.ESC='B'2                     ;ESCAPE SEQUENCE IN PROGRESS
S1.RSP='B'4                     ;READ WITH SPECIAL PROCESSING
S1.PTH='B'10                    ;PASS THRU IS CURRENTLY ACTIVE
S1.RNE='B'20                    ;ECHO SUPPRESSED
S1.TSY='B'40                    ;TERMINAL OUTPUT SYNC IS CURRENTLY ENABLED
S1.OBY='B'100                   ;OUTPUT BUSY
S1.IBY='B'200                   ;INPUT BUSY
S1.DPR='B'400                   ;DEFER PROCESSING OF CHAR. IN U.TECB
S1.DEC='B'1000                  ;DEFER ECHO OF CHAR. IN U.TECB
S1.IBF='B'2000                  ;BUFFERED INPUT IN PROGRESS
S1.DSI='B'4000                  ;INPUT PROCESSING DISABLED
S1.RES='B'10000                 ;ESC. SEQ PROCESSING IS ENABLED FOR THE CURRENT READ
S1.RNF='B'20000                 ;READ NO FILTER IS ACTIVE (EDIT CHARS. ARE DISPLAYED)
S1.TNE='B'40000                 ;TERMINATOR NO ECHO
S1.USI='B'100000                ;UNSOLICITED INPUT IN PROGRESS
```

UCBDF$

```
; DEFINE BITS IN STATUS WORD 2 (U.TSTA+2)
;
;    O U T P U T   S T A T U S
;
S2.RCU='B'1                     ;RESTORE CURSOR (MUST = TF.RCU)
S2.WRA='B'6                     ;CONTEXT FOR WRAP-AROUND
S2.WRB='B'2                     ;LOW BIT IN S2.WRA BIT PATTERN
S2.WAL='B'10                    ;WRITE PASS ALL (MUST = TF.WAL)
S2.BRQ='B'20                    ;BREAK-THROUGH-WRITE REQUEST IN QUEUE
S2.SRQ='B'40                    ;SPECIAL REQUEST IN QUEUE
                                ;(IO.ATT, IO.DET, SF.SMC)
S2.ORQ='B'100                   ;OUTPUT REQUEST IN QUEUE (MUST = S1.OBY)
S2.IRQ='B'200                   ;INPUT REQUEST IN QUEUE (MUST = S1.IBY)
S2.FLF='B'400                   ;FORCE LINEFEED BEFORE NEXT ECHO
S2.ELF='B'1000                  ;EAT A LINEFEED (IGNORE A LEADING LF ON OUTPUT)
S2.CR='B'2000                   ;TRAILING CR REQUIRED ON OUTPUT
S2.OBF='B'4000                  ;BUFFERED OUTPUT IN PROGRESS
S2.PCU='B'10000                 ;POSITION CURSOR BEFORE WRITE
S2.BEL='B'20000                 ;BELL PENDING
S2.CTO='B'40000                 ;OUTPUT STOPPED BY CTRL-O 266.
S2.CTS='B'100000                ;OUTPUT STOPPED BY CTRL-S
; DEFINE BITS IN STATUS WORD 3 (U.TSTA+4)
;
;   TERMINAL OPERATION CHARACTERISTICS
;
S3.ACR='B'1                     ;WRAP-AROUND (AUTOMATIC CR-LF) REQUIRED
S3.TAB='B'2                     ;TYPE-AHEAD BUFFER ALLOCATION REQUESTED
S3.CTC='B'4                     ;TERMINAL WANTS CLI TO HAVE ^C NOTIFICATION
S3.RAL='B'10                    ;TERMINAL IS IN READ-PASS-ALL MODE
S3.NEC='B'20                    ;NO ECHO
S3.TSY='B'40                    ;TERMINAL SYNC
S3.8BC='B'100                   ;PASS 8 BITS ON INPUT
S3.FDX='B'200                   ;LINE IS IN FULL DUPLEX MODE
S3.MHE='B'400                   ;NOTIFY ATTACHED TASK OF MODEM HANG-UP
S3.ICE='B'1000                  ;INPUT COUNT STATE ENABLED
S3.TME='B'2000                  ;TERMINAL MANAGEMENT MODE ENABLED
S3.PTH='B'4000                  ;PASS THROUGH REQUESTED
S3.RES='B'10000                 ;TASK WANTS ESCAPE SEQUENCES
S3.PPT='B'20000                 ;TERMINAL HAS PRINTER PORT
S3.RUB='B'40000                 ;RUBOUT SEQUENCE IN PROGRESS (NON-SCOPE)
```

```
;
; DEFINE BITS IN STATUS WORD 4 (U.TSTA+6)
;
;
; TERMINAL ATTRIBUTE CHARACTERISTICS
;
S4.HFL='B'7                     ;HORIZONTAL FILL REQUIREMENT
S4.VFL='B'10                    ;VERTICAL FILL REQUIREMENT
S4.HFF='B'20                    ;HARDWARE FORM-FEED PRESENT
S4.HHT='B'40                    ;HARDWARE HORIZONTAL TAB PRESENT
S4.DLO='B'100                   ;DIAL-OUT LINE (IMPLIES U2.RMT)
S4.HSY='B'200                   ;HOST/TERMINAL SYNCHRONIZATION ENABLED (1=YES)
S4.ANI='B'400                   ;ANSI CRT TERMINAL
S4.AVO='B'1000                  ;VT100-FAMILY TERMINAL DISPLAY
S4.BLK='B'2000                  ;BLOCK MODE TERMINAL
S4.DEC='B'4000                  ;DIGITAL CRT TERMINAL
S4.EDT='B'10000                 ;TERMINAL HAS LOCAL EDITING FUNCTIONS
S4.RGS='B'20000                 ;TERMINAL SUPPORTS REGIS GRAPHICS
S4.SFC='B'40000                 ;TERMINAL SUPPORTS SOFT CHARACTERS (DRCS)
S4.ABD='B'100000                ;AUTO-BAUD SPEED DETECTION ENABLED
;
;
; DEFINE BITS IN STATUS WORD U.TST5
;
;  ADDITIONAL STATUS CHARACTERISTICS
;
;
S5.SW1='B'1                     ;FIRST TERMINAL MANAGEMENT SWITCH
                                ;CHARACTER HAS BEEN SEEN
S5.TMM='B'2                     ;TERMINAL IN TERMINAL MANAGEMENT MODE
S5.XOF='B'4                     ;SEND AN XOFF AT FIRST OPPORTUNITY
S5.XON='B'10                    ;SEND AN XON AT FIRST OPPORTUNITY
S5.HPC='B'14                    ;OUTPUT OF HIGH PRIORITY CHARACTERS REQUESTED
S5.HPO='B'20                    ;HIGH PRIORITY OUTPUT IN PROGRESS
S5.OXF='B'40                    ;XOFF HAS BEEN OUTPUT
S5.ITI='B'100                   ;IMMEDIATE TIMEOUT ON INPUT
S5.DEP='B'200                   ;DISABLE ERROR PROCESSING
S5.DMA='B'400                   ;TERMINAL SUPPORTS DMA
S5.CLN='B'1000                  ;7-BIT CHARACTER LENGTH
S5.RPO='B'2000                  ;READ W/PROMPT OUTPUT IN PROGRESS
S5.VER='B'10000                 ;LAST CHAR. IN TYPE-AHEAD BUFFER
                                ;HAS PARITY ERROR
S5.BCC='B'20000                 ;LAST CHAR. IN TYPE-AHEAD BUFFER
                                ;HAS FRAMING ERROR
S5.DAO='B'40000                 ;LAST CHAR. IN TYPE-AHEAD BUFFER
                                ;HAS DATA OVERRUN ERROR
                                ;NOTE - THE 3 BITS ABOVE MUST CORRESPOND
                                ;TO THE RESPECTIVE ERROR FLAGS IN THE
                                ;HARDWARE RECEIVE BUFFER
S5.ABP='B'100000                ;AUTO-BAUD SPEED DETECTION IN PROGRESS
;
;
; DEFINE BITS IN EXTENDED I/O STATUS WORD U.TST6
;
;
S6.LAT='B'1                     ;TERMINAL IS A LAT APPLICATION TERMINAL
S6.EIO='B'400                   ;READ WAS AN EXTENDED I/O
S6.RLU='B'1000                  ;READ WITH LOWER CASE TO UPPER CASE CONVERSION
S6.RDI='B'100000                ;READ WITH DEFAULT INPUT
```

```
;
        .ENDC
;
; VIRTUAL TERMINAL UCB DEFINITIONS
;
.=U.UNIT
U.OCNT:'L'.BLKB 1               ;OFFSPRING WITH THIS AS TI:
.=U.BUF
U.RPKT:'L'.BLKW 1               ;CURRENT OFFSPRING READ I/O PACKET
U.WPKT:'L'.BLKW 1               ;CURRENT OFFSPRING WRITE I/O PACKET
U.IAST:'L'.BLKW 1               ;INPUT AST ROUTINE ADDRESS
U.OAST:'L'.BLKW 1               ;OUTPUT AST ROUTINE ADDRESS
U.AAST:'L'.BLKW 1               ;ATTACH AST ROUTINE ADDRESS

        .IF NB  TTDEF
.IIF NE U.AAST+2-U.UIC  .ERROR  ;ADJACENCY ASSUMED
        .ENDC

.=U.AAST+4
U.PTCB:'L'.BLKW 1               ;PARENT TCB ADDRESS

;
; CONSOLE DRIVER DEFINITIONS
;
.=U.BUF+2
U.CTCB:'L' .BLKW 1              ;ADDRESS OF CONSOLE LOGGER TCB
U.COTQ:'L' .BLKW 2              ;I/O PACKET LIST QUEUE
U.RED2:'L' .BLKW 1              ;REDIRECT UCB ADDRESS

        .PSECT

;+
; DEVICE TABLE STATUS DEFINITIONS
;
; DEVICE CHARACTERISTICS WORD 1 (U.CW1) DEVICE TYPE DEFINITION BITS.
;-
DV.REC='B'1                     ;RECORD ORIENTED DEVICE (1=YES)
DV.CCL='B'2                     ;CARRIAGE CONTROL DEVICE (1=YES)
DV.TTY='B'4                     ;TERMINAL DEVICE (1=YES)
DV.DIR='B'10                    ;FILE STRUCTURED DEVICE (1=YES)
DV.SDI='B'20                    ;SINGLE DIRECTORY DEVICE (1=YES)
DV.SQD='B'40                    ;SEQUENTIAL DEVICE (1=YES)
DV.MSD='B'100                   ;MASS STORAGE DEVICE (1=YES)
DV.UMD='B'200                   ;USER MODE DIAGNOSTICS SUPPORTED (1=YES)
DV.MBC='B'400                   ;MASSBUS CONTROLLER (11M COMPATIBILITY ONLY)
DV.EXT='B'400                   ;UNIT ON EXTENDED 22-BIT UNIBUS CNTROLER (1=YES)
DV.SWL='B'1000                  ;UNIT SOFTWARE WRITE LOCKED (1=YES)
DV.ISP='B'2000                  ;INPUT SPOOLED DEVICE (1=YES)
DV.OSP='B'4000                  ;OUTPUT SPOOLED DEVICE (1=YES)
DV.PSE='B'10000                 ;PSEUDO DEVICE (1=YES)
DV.COM='B'20000                 ;DEVICE IS MOUNTABLE AS COM CHANNEL (1=YES)
DV.F11='B'40000                 ;DEVICE IS MOUNTABLE AS F11 DEVICE (1=YES)
DV.MNT='B'100000                ;DEVICE IS MOUNTABLE (1=YES)
```

```
;+
; TERMINAL DEPENDENT CHARACTERISTICS WORD 2 (U.CW2) BIT DEFINITIONS
;-

U2.DH1='B'100000                ;UNIT IS A MULTIPLEXER (1=YES)
U2.DJ1='B'40000                 ;UNIT IS A DJ11 (1=YES)
U2.RMT='B'20000                 ;UNIT IS REMOTE (1=YES)
U2.HFF='B'10000                 ;UNIT HANDLES HARDWARE FORM FEEDS (1=YES)
U2.L8S='B'10000                 ;OLD NAME FOR U2.HFF
U2.NEC='B'4000                  ;DON'T ECHO SOLICITED INPUT (1=YES)
U2.CRT='B'2000                  ;UNIT IS A CRT (1=YES)
U2.ESC='B'1000                  ;UNIT GENERATES ESCAPE SEQUENCES (1=YES)
U2.LOG='B'400                   ;USER LOGGED ON TERMINAL (0=YES)
U2.SLV='B'200                   ;UNIT IS A SLAVE TERMINAL (1=YES)
U2.DZ1='B'100                   ;UNIT IS A DZ11 (1=YES)
U2.HLD='B'40                    ;TERMINAL IS IN HOLD SCREEN MODE (1=YES)
U2.AT.='B'20                    ;MCR COMMAND AT. BEING PROCESSED (1=YES)
U2.PRV='B'10                    ;UNIT IS A PRIVILEGED TERMINAL (1=YES)
U2.L3S='B'4                     ;UNIT IS A LA30S TERMINAL (1=YES)
U2.VT5='B'2                     ;UNIT IS A VT05B TERMINAL (1=YES)
U2.LWC='B'1                     ;LOWER CASE TO UPPER CASE CONVERSION (0=YES)

;+
; BIT DEFINITIONS FOR U.MUP
;-

UM.OVR='B'1                     ;OVERRIDE CLI INDICATOR
UM.CLI='B'36                    ;CLI INDICATOR BITS
UM.DSB='B'200                   ;TERMINAL DISABLED SINCE CLI ELIMINATED
UM.NBR='B'400                   ;NO BROADCAST
UM.CNT='B'1000                  ;CONTINUATION LINE IN PROGRESS
UM.CMD='B'2000                  ;COMMAND IN PROGRESS
UM.SER='B'4000                  ;SERIAL COMMAND RECOGNITION ENABLED
UM.KIL='B'10000                 ;TTDRV SHOULD SEND KILL PKT ON CNTRL/C

;+
; RH11-RS03/RS04 CHARACTERISTICS WORD 2 (U.CW2) BIT DEFINITIONS
;-
U2.R04='B'100000                ;UNIT IS A RS04 (1=YES)

;+
; RH11-TU16 CHARACTERISTICS WORD 2 (U.CW2) BIT DEFINITIONS
;-

U2.7CH='B'10000                 ;UNIT IS A 7 CHANNEL DRIVE (1=YES)

;+
; TERMINAL DEPENDENT CHARACTERISTICS WORD 3 (U.CW3) BIT DEFINITIONS
;-

U3.UPC='B'20000                 ;UPCASE OUTPUT FLAG
U3.PAR='B'40000                 ;PARITY GENERATION AND CHECKING
U3.OPA='B'100000                ;PARITY SENSE (1=ODD PARITY)

;+
; VIRTUAL TERMINAL 3RD CHARACTERISTICS WORD DEFINITIONS
;-

U3.FDX='B'1                     ;FULL DUPLEX MODE (1=YES)
U3.DBF='B'2                     ;INTERMEDIATE BUFFERING DISABLED (1=YES)
U3.RPR='B'4                     ;READ W/PROMPT IN PROGRESS (1=YES)
```

UCBDF$

```
;+
; TERMINAL DEPENDENT CHARACTERISTICS WORD 4 (U.CW4) BIT DEFINITIONS
;-
U4.CR='B'100                    ;LOOK FOR CARRIAGE RETURN

;+
; UNIT CONTROL PROCESSING FLAG DEFINITIONS
;-
UC.ALG='B'200                   ;BYTE ALIGNMENT ALLOWED (1=NO)
UC.NPR='B'100                   ;DEVICE IS AN NPR DEVICE (1=YES)
UC.QUE='B'40                    ;CALL DRIVER BEFORE QUEUING (1=YES)
UC.PWF='B'20                    ;CALL DRIVER AT POWERFAIL ALWAYS (1=YES)
UC.ATT='B'10                    ;CALL DRIVER ON ATTACH/DETACH (1=YES)
UC.KIL='B'4                     ;CALL DRIVER AT I/O KILL ALWAYS (1=YES)
UC.LGH='B'3                     ;TRANSFER LENGTH MASK BITS

;+
; UNIT STATUS BIT DEFINTIONS
;-
US.BSY='B'200                   ;UNIT IS BUSY (1=YES)
US.MNT='B'100                   ;UNIT IS MOUNTED (0=YES)
US.FOR='B'40                    ;UNIT IS MOUNTED AS FOREIGN VOLUME (1=YES)
US.MDM='B'20                    ;UNIT IS MARKED FOR DISMOUNT (1=YES)
US.PWF='B'10                    ;POWERFAIL OCCURED (1=YES).

;+
; CARD READER DEPENDENT UNIT STATUS BIT DEFINITIONS
;-
US.ABO='B'1                     ;UNIT IS MARKED FOR ABORT IF NOT READY (1=YES)
US.MDE='B'2                     ;UNIT IS IN 029 TRANSLATION NODE (1=YES)

;+
; FILES-11 DEPENDENT UNIT STATUS BITS
;-
US.WCK='B'10                    ;WRITE CHECK ENABLED (1=YES)
US.SPU='B'2                     ;UNIT IS SPINNING UP (1=YES)
US.VV='B'1                      ;VOLUME VALID IS SET (1=YES)

;+
; TERMINAL DEPENDENT UNIT STATUS BIT DEFINITIONS
;-
US.CRW='B'4                     ;UNIT IS WAITING FOR CARRIER (1=YES)
US.DSB='B'2                     ;UNIT IS DISABLED (1=YES)
US.OIU='B'1                     ;OUTPUT INTERRUPT IS UNEXPECTED ON UNIT (1=YES)

;+
; LPS11 DEPENDENT UNIT STATUS BIT DEFINITIONS
;-
US.FRK='B'2                     ;FORK IN PROGRESS (1=YES)
US.SHR='B'1                     ;SHAREABLE FUNCTION IN PROGRESS (0='B'YES)

;+
; ANSI MAGTAPE DEPENDANT UNIT STATUS BITS
;-
US.LAB='B'4                     ; UNIT HAS LABELED TAPE ON IT (1=YES)
```

```
;+
; UNIT STATUS EXTENSION (U.ST2) BIT DEFINITIONS
;-

US.OFL='B'1                     ;UNIT OFFLINE (1=YES)
US.RED='B'2                     ;UNIT REDIRECTABLE (0=YES)
US.PUB='B'4                     ;UNIT IS PUBLIC DEVICE (1=YES)
US.UMD='B'10                    ;UNIT ATTACHED FOR DIAGNOSTICS (1=YES)
US.PDF='B'20                    ;PRIVILEGED DIAGNOSTIC FUNCTIONS ONLY (1=YES)
US.MUN='B'40                    ;MULTI-UNIT FLAG
US.TRN='B'100                   ;UNIT TRANSITION HAS OCCURRED (1=YES)
US.SIO='B'200                   ;STALL I/O TO UNIT (1=YES)
;+
; MAGTAPE DENSITY SUPPORT DEFINITION IN U.CW3
;-

UD.UNS='B'0                     ; UNSUPPORTED
UD.200='B'1                     ; 200BPI, 7 TRACK
UD.556='B'2                     ; 556BPI, 7 TRACK
UD.800='B'3                     ; 800BPI, 7 OR 9 TRACK
UD.160='B'4                     ;1600BPI, 9 TRACK
UD.625='B'5                     ;6250BPI, 9 TRACK
UD.8K='B'6                      ;8K BPI - SERIAL, SERPENTINE RECORDING.

        .MACRO  UCBDF$,X,Y,Z,A
        .ENDM
        .ENDM
```

Appendix D
Common Error Code Definitions

This appendix lists the following:

1. Facility-independent error code definitions
2. Standard Bugcheck formats for facility-defined error codes

```
------------------------------------------------
Common (facility independent) error code definitions
------------------------------------------------

        SST type errors - Major error code 1

BE.ODD = 000100         Odd address or other trap four
BE.SGF = 000102         Segment fault
BE.BPT = 000104         Breakpoint or T-bit trap
BE.IOT = 000106         IOT instrtuction
BE.ILI = 000110         Illegal instruction
BE.EMT = 000112         EMT instruction
BE.TRP = 000114         Trap instruction
BE.STK = 000116         Stack overflow

        Internal inconsistency errors - error code 2

BE.NPA = 000200         Task with no parent aborted
BE.SGN = 000201         Feature not included in system
BE.2FR = 000202         Double fork
BE.ISR = 000203         Int. service routine clobbered reg.
BE.FHW = 000204         Fatal hardware error
BE.CSR = 000205         Device CSR disappeared
BE.IDC = 000206         Internal database consistency error
BE.ACP = 000207         ACP task aborted
BE.HSP = 000210         Header subpacket problem
BE.NCT = 000211         No current task

        System pool related errors - error code 3

BE.NPL = 000300         No pool for operation
BE.DDA = 000301         Double deallocation
BE.SIZ = 000302         Size of block invalid
BE.BAK = 000303         Deallocated block below pool
BE.POV = 000304         Deallocation overlaps end of pool
BE.FSI = 000305         Allocation - fragment size invalid
```

```
                Group global event flag errors - error code 4
BE.GGF = 000400          Task locked to non-existent flags
        ------------------------------------------------
        Standard bugcheck format facility code definitions
        ------------------------------------------------

        P/OS keyboard handler - Major facility code 1
BF.PKS = 000100          P/OS Keyboard handler

        I/O driver subsystem - facility code 2
BF.TTD = 000200          Terminal driver

        Executive components - facility code 3
BF.EXE = 000300          Exec - General and miscellaneous
BF.XDT = 000301          Exec - Executive debugging tool
BF.MP  = 000302          Exec - Multiprocessing

                mP specific type errors
        BE.NDS = 100100  Init failure - d-space not loaded
        BE.NCK = 100200  clock not available
        BE.URM = 100300  Fork to offline UNIBUS run
        BE.WTL = 100400  Attempt to walk through lock
        BE.UNO = 100500  Attempt to unlock not by owner
        BE.ILC = 100600  Illegal lock count value
        BE.LNS = 100700  Lock not locked
        BE.OCP = 101000  At entry another CPU showed ownership
        BE.MLK = 101100  Attempt to exit multiple lock
        BE.NIN = 101200  No reason for interprocessor int.
        BE.UNP = 101300  Some UNIBUS run not connected

BF.POL = 000303          Exec - Pool handling routines (CORAL)
BF.ERR = 000304          Exec - hardware error processing subsystem
BF.INT = 000305          Exec - Internal consistence checking routine
BF.INI = 000306          Exec - INITL - initilization module
BF.DVI = 000307          Exec - DVINT common interrupt handler
BF.PAR = 000310          Exec - Parity memory support
BF.XIT = 000311          Exec - task exit/abort procesing
BF.QIO = 000312          Exec - QIO directive
BF.OPT = 000313          Exec - Seek optimization
BF.ACC = 000314          Exec - System resource accounting
BF.KAS = 000315          Exec - Kernal AST support
BF.DIR = 000316          Exec - Miscellaneous directives
BF.SAN = 000317          Exec - Crash with sanity timer message

        P/OS startup task - facility code 4
BF.UP  = 000400          P/OS startup task

                UP specific error messages
        BE.IN1 = 100100  Can't install task CBOOT
        BE.SP1 = 100200  Can't Spawn task CBOOT
        BE.SP2 = 100300  Can't Spawn task CMAIN
        BE.FNF = 100400  Required file not found
        ------------------------------------------------
        Non-standard bugcheck format facility definitions
        ------------------------------------------------

        P/OS terminal subsystem - Major facility code 1
BF.PTS = 100400          P/OS terminal subsystem
```

Index

A

ABODF$, C-3
ACNDF$, C-5
/ACT, 2-3, 2-5
Active task
  address of TCB, 3-15
  AST queue, 3-17
  attribute bits, 3-16
  blocking bits, 3-16
  name, 3-15
  partition, 3-15
  receive-by-reference queue, 3-18
  receive queue, 3-17
  state bits, 3-16
  task image, 3-15
Active task dump, 3-15
  AST queue, 3-15
  Offspring Control Block, 3-15
  receive-by-reference queue, 3-15
  receive queue, 3-15
/ADV, 2-3, 2-5
/ALL, 2-4, 2-5
All devices switch
  See /ADV
Analysis listings, 3-1 to 3-45
  interpreting, 4-1
Analysis routines switch
  See /ALL
Analysis switches, 2-3 to 2-7
ANALYZE/CRASH_DUMP command, 1-7
  error messages, A-1
  examples, 2-20
  format, 2-11
  input to, 2-12
  qualifiers, 2-12 to 2-20
  specification, 2-12
Assign table dump, 3-1, 3-14
Assign table dump (cont'd.)
  logical device names, 3-14
  physical device names, 3-14
/ATL, 2-3, 2-5

B

Binary output file size switch
  See /MEMSIZ
/BL, 2-8
Block number switch
  See /BL
Bugcheck facility, 1-4

C

/CACHE, 2-4, 2-5
Cache dump, 3-44
Cache memory, 3-44
Cache memory switch
  See /CACHE
CDA
  analysis listings, 3-1 to 3-45
  function, 1-1
  input to, 1-5
  messages, A-1
  running, 1-6
    as installed task, 1-7
    as uninstalled task, 1-7
  switches, 2-3 to 2-11
CDA command line, 2-1 to 2-11
  binary file, 2-2
  crash input file, 2-3
  default, 2-10
  format, 2-1
  list file, 2-1
  symbol file, 2-2
/CLI, 2-4, 2-5
CLI parser block dump, 3-24

CLKDF$, C-12
Clock queue dump, 3-41
Clock queue switch
  See /CLQ
/CLQ, 2-4, 2-6
Command line interpreter switch
  See /CLI, /CPB
Common Block Directory dump, 3-29
/CPB, 2-4, 2-5
Crash
  cause
    determining, 4-1
    HALT, 1-5
    HALT instruction, 1-4
    infinite loop, 1-4, 1-5
    processor trap, 1-4
    restart procedure, 1-4
Crash dump
  obtaining, 1-4
Crash Dump Analyzer
  See CDA
Crash dump binary file, 1-5, 1-6, 2-2
Crash dump device, 1-1, 1-5
  displaying current one, 1-3
  invalid, 1-5
  limitations, 1-2
  loadable, 1-2
  restrictions, 1-2
  selecting, 1-3
  specifying, 2-2
  valid for RSX-11M/M-PLUS system, 1-2
  valid for Micro/RSX system, 1-2
  valid for pregenerated RSX-11M-PLUS
    system, 1-2
Crash dump driver, 1-1
  procedure, 1-5
  unloading, 1-4
Crash dump listing
  See also Analysis listings
  interpreting, 4-1 to 4-3
    mapping data, 4-1
    stack depth, 4-2
Crash notification device, 1-1, 1-2, 1-5
$CRAVL, 3-12, 3-36
CTBDF$, C-14
/CTL, 2-4, 2-6

D

/DCB, 2-4, 2-6
DCBDF$, C-15
/DENS, 2-8
/DEV, 2-4, 2-6
Device
  fixed media, 1-5
  switches, 2-6
  valid crash dump, 1-2
Device controller dump, 3-42
  common interrupt address, 3-42
  Controller Request Block, 3-43
  controller status, 3-43
  Controller Table, 3-42
Device controller switch
  See /CTL
Device information dump, 3-30
  control processing flags, 3-31
  device characteristics word, 3-31
  Device Control Block, 3-34
  I/O request packet, 3-34
  Status Control Block, 3-34
  terminal status word, 3-30
  UCB extension, 3-34
Device names
  logical, 3-14
  physical, 3-14
/DMP, 2-4, 2-6
/DUMP, 2-4, 2-6

E

EPKDF$, C-17
Error log buffer
  address of, 3-14
Error log packet dump, 3-1, 3-14
Error messages, A-1
EVNDF$, C-25
Executive Debugging Tool
  See XDT
Executive symbol table, 2-2, 2-14
/EXIT, 2-8
Exit switch
  See /EXIT

F

F11DF$, C-28
Function switches, 2-7 to 2-10

G

Group Global Event Flag dump, 3-1
Group global event flag dump, 3-14

H

/HDR, 2-4, 2-6

HDRDF$, C-32
HWDDF$, C-34

I

ITBDF$, C-39

K

/KDS, 2-4, 2-6
Kernel data space
  dump, 3-44
  switch
    See /KDS
Kernel instruction space
  dump, 3-44
  switch
    See /KIS
Kernel Mapping Register switch
  See /KMR
Kernel stack dump, 3-1, 3-4
/KIS, 2-4, 2-6
/KMR, 2-8, 2-9
KRBDF$, C-40

L

LCBDF$, C-42
/LIMIT, 2-8, 2-9
/LINES, 2-8, 2-9
Lines per page switch
  See /LINES
LNMDF$, C-43
Logical device assignments, 3-14
Logical device names, 3-14
Low core memory
  dump, 3-4
  labels, 3-5
Low Core Memory Dump, 3-1

M

Memory
  cache, 3-44
/MEMSIZ, 2-8, 2-9
MTADF$, C-46

N

No spool switch
  See /-SP

O

Offset mode, 3-1
OLRDF$, C-49
Output listing limit switch
  See /LIMIT

P

/PAR, 2-4, 2-6
Partition
  base address, 3-15
  name, 3-15
  PCB address, 3-15
  status flags, 3-26
Partition Control Block, 3-26
  address, 3-24
  pointer to, 3-26
  switch
    See /PCB, /PAR
Partition information dump
  attachment descriptor, 3-27
  individual, 3-26
  system, 3-24
  wait queue, 3-27
/PCB, 2-4, 2-6
PCBDF$, C-57
Physical device names, 3-14
Physical memory switch
  See /DUMP
$PKAVL, 3-36
PKTDF$, C-62
/POOL, 2-4, 2-7
Pool
  bit map, 3-12
  dump, 3-36
  free, 3-12
  largest fragment, 3-12
  number of unallocated fragments, 3-12
  secondary pool dump, 3-36
  size, 3-12
  smallest possible block, 3-12
  total free bytes, 3-12
Pool statistics dump, 3-1, 3-12
Processor Status Word, 3-2
Program Counter, 3-2

Q

Qualifiers
  command, 2-12 to 2-14
  crash-input, 2-15 to 2-20

R

Register

Register (cont'd.)
cache control, 3-2
error, 3-2
general, 3-2
memory management, 3-2
memory system error, 3-2
page
address, 3-2
description, 3-2
UNIBUS mapping, 3-2
volatile, 3-1, 3-3
See also Volatile registers
Relative addresses, 3-1

S

/SCB, 2-4, 2-6
SCBDF$, C-69
Secondary pool
dump, 3-36
switch
See /SECPOL
/SECPOL, 2-4, 2-7
SHDDF$, C-72
/-SP, 2-8, 2-9
Stack pointer
kernel, 3-2
Stack pointer
kernel, 3-4
user, 3-2
/STB, 2-2, 2-8, 2-10
/STD, 2-4, 2-7
$STKDP
stack depth indicator, 4-2
Switches
analysis, 2-3 to 2-7
function, 2-7 to 2-10
/-SYS, 2-4, 2-7
System
device
switches, 2-6
information, 3-1
switch
See /-SYS
pool
dump, 3-36
switch
See /POOL
System common, 3-5
active task, 3-6
alphabetized dump, 3-1, 3-10
boot device, 3-6
System common (cont'd.)
dump
See System common dump
labels, 3-10 to 3-11
network UIC, 3-6
partition address, 3-6
stack depth, 3-6
stack depth indicator, 4-2
system size, 3-6
TCB address, 3-6
UIC, 3-6
System common dump, 3-1, 3-10 to 3-11
System Task Directory
dump, 3-36
switch
See /STD

T

/TAL, 2-4, 2-7
Tape density switch
See /DENS
/TAS, 2-4, 2-7
/TASK, 2-4, 2-7
Task Control Block switch
See /ACT, /ATL
Task data space
switch
See /TDS
Task dump, 3-39
window blocks, 3-39
Task header
switch
See /HDR
Task header dump, 3-21
Directive Status Word, 3-22
File Control Block, 3-22
Logical Unit Table, 3-22
Page Description Register, 3-23
swapping priority, 3-22
window blocks, 3-22
Task instruction space
switch
See /TIS
Task virtual address space switch, 2-7
/TCB, 2-4, 2-7
TCB
address of, 3-15, 3-18
pointer to current, 3-5
TCBDF$, C-74
/TDS, 2-4, 2-7
/TIS, 2-5, 2-7

/TSK, 2-4, 2-7

U

/UCB, 2-4, 2-6
UCBDF$, C-78

V

Volatile registers, 3-1
  dump, 3-2 to 3-3

X

XDT, 1-4, 1-5, 1-6

RSX–11M–PLUS and Micro/RSX
Crash Dump Analyzer
Reference Manual
AA–JS13A–TC

READER'S COMMENTS

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

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

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

What kind of user are you? ___ Programmer ___ Nonprogrammer

Years of experience as a computer programmer/user: _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

-- Do Not Tear - Fold Here and Tape ----------

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- Do Not Tear - Fold Here ----------

RSX–11M–PLUS and Micro/RSX
Crash Dump Analyzer
Reference Manual
AA–JS13A–TC

READER'S COMMENTS

Your comments and suggestions are welcome and will help us in our continuous effort to improve the quality and usefulness of our documentation and software.

Remember, the system includes information that you read on your terminal: help files, error messages, prompts, and so on. Please let us know if you have comments about this information, too.

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

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

What kind of user are you? ____ Programmer ____ Nonprogrammer

Years of experience as a computer programmer/user: _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

-- Do Not Tear - Fold Here and Tape ----------

digital™

No Postage
Necessary
if Mailed
in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
Corporate User Publications—Spit Brook
ZK01-3/J35
110 SPIT BROOK ROAD
NASHUA, NH 03062-9987

-- D Not Tear - Fold Here ----------