

TOPS-10 Stopcodes Specification

AA-BJ93C-RB

January 1989

This document describes stopcodes for the TOPS-10 monitor, GALAXY, and DECnet-10 software.

Operating System:	TOPS-10 Version 7.04 GALAXY Version 5.1
Software:	DECnet-10 Version 4.0

**digital equipment corporation
maynard, massachusetts**

First Printing, December 1980
Revised, April 1984
Revised, April 1986
Revised, January 1989

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

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

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

Copyright ©1980, 1984, 1986, 1989 Digital Equipment Corporation

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

The 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:

CI	DEctape	LA50	SITGO-10
DDCMP	DECUS	LN01	TOPS-10
DEC	DECwriter	LN03	TOPS-20
DECmail	DELNI	MASSBUS	TOPS-20AN
DECnet	DELUA	PDP	UNIBUS
DECnet-VAX	HSC	PDP-11/24	UETP
DECserver	HSC-50	PrintServer	VAX
DECserver 100	KA10	PrintServer 40	VAX/VMS
DECserver 200	KI	Q-bus	VT50
DECsystem-10	KL10	ReGIS	
DECSYSTEM-20	KS10	RSX	

digital™

CONTENTS

PREFACE

1	MONITOR STOPCODE DEFINITION	1
2	TYPES OF STOPCODES	1
2.1	DEBUG Stopcodes	2
2.2	JOB Stopcodes	2
2.3	STOP Stopcodes	3
2.4	CPU Stopcodes	3
2.5	HALT Stopcodes	4
2.6	INFO Stopcodes	4
3	CONTINUABLE STOPCODES	5
4	LIST OF MONITOR STOPCODES	7
5	LIST OF DECNET-10 STOPCODES	91
6	GALAXY-10 STOPCODES	104
7	LIST OF GALAXY STOPCODES	105

PREFACE

This document briefly describes the TOPS-10 monitor, GALAXY, and DECnet-10 stopcodes. This specification was designed to assist the user in analyzing stopcode messages, to investigate and undertake corrective action when the TOPS-10 monitor, GALAXY, or DECnet-10 software unexpectedly fails.

Audience Requirements

The following manuals provide additional information for analyzing and regenerating a TOPS-10 monitor, GALAXY, and DECnet-10 software.

- o TOPS-10 Software Installation Guide describes how to generate and install TOPS-10 and GALAXY software.
- o DECnet-10 Network Generation and Installation Procedures describes how to generate and install DECnet-10 software on DECSYSTEMS-1090/1091 and DECSYSTEM-2020.
- o DECnet-10 System Manager's and Operator's Guide describes the operational, control, and monitoring facilities of DECnet-10.
- o TOPS-10 Operator's Guide describes how to reload a TOPS-10 system.
- o TOPS-10 Crash Analysis Guide describes how to analyze a crash, and how to prevent future monitor errors.
- o TOPS-10/TOPS-20 Batch Reference Manual describes how to use the TOPS-10 and TOPS-20 batch systems.
- o TOPS-10/TOPS-20 SPEAR Manual describes the SPEAR program, which provides a dump to help analyze a hardware errors.

MONITOR STOPCODE DEFINITION

1 MONITOR STOPCODE DEFINITION

A stopcode message is generated when the monitor detects a serious error in its database. When the TOPS-10 monitor encounters an internal error it issues a one to six character name called a stopcode name. The stopcode is displayed on the console terminal (CTY) and alerts you of possible system failure, depending on the severity of the error. The stopcode message is generated by a stopcode macro when the monitor detects an error in the database. This macro prints the following message on the CTY:

```
?Stopcode xxxxyy, Type=type on CPUn at date/time
```

Where:

xxxxyy is the stopcode name.

type is the type of stopcode. Stopcode types are listed below.

n is the CPU number.

date/time is the date and time of day when the stopcode occurred.

2 TYPES OF STOPCODES

The stopcodes are divided into categories, depending on the severity of the error that cause the stopcode, and the portions of the system that might be affected by the error. The types of monitor stopcodes are:

1. DEBUG stopcodes indicate internal errors only. These are continuable stopcodes.
2. JOB stopcodes indicate that the error may endanger a user job. These are continuable stopcodes. Only the job that was running when the stopcode occurred will be stopped.
3. STOP stopcodes indicate errors that might endanger system operation. These are not continuable stopcodes.
4. CPU stopcodes indicate errors that prevent continued operation of a CPU. In a single-CPU configuration, or if the CPU that encountered the stopcode was the only CPU running at the time, the CPU stopcode is equivalent to a STOP stopcode. If there are other CPUs running in an SMP configuration at the time of the stopcode, a dump is taken and the affected CPU halts. Any job that was running on the affected CPU at the time of the stopcode is stopped.
5. HALT stopcodes indicate fatal errors that halt the system, and, by definition, are not continuable.
6. INFO stopcodes are continuable stopcodes that are generated to inform the system operator of a system event.

MONITOR STOPCODE DEFINITION

The monitor generates a stopcode using the STOPCD macro. There are several modules in the monitor, however, that use the BUG. macro to generate a stopcode. These modules are common to both TOPS-10 and TOPS-20. They generate three types of BUG. stopcodes, which are equivalent to the following types of stopcodes:

<u>BUG. Type</u>	<u>STOPCD Type</u>
HLT	STOP
CHK	DEBUG
INF	INFO

2.1 DEBUG Stopcodes

A DEBUG stopcode is a stopcode that is not immediately harmful to any job or to the system. When the monitor encounters an internal error at the interrupt level, a dump is performed and processing continues. The following message is printed on the CTY after the stopcode notification:

[Continuing system]

2.2 JOB Stopcodes

A JOB stopcode indicates that an internal error endangers the integrity of the job that is currently running. The monitor aborts the current job and continues processing.

The following message is printed on the CTY after the stopcode notification:

[Aborting job]

On the user's terminal, the following message is displayed:

```
?Monitor error at {user/exec} PC nnnnnn;  
UUO at {user/exec} PC mmmmmm
```

Where:

nnnnnn and mmmmmm are one of the following virtual memory locations:

- o User location nnnnnn;
- o Exec location nnnnnn; Exec called from exec location mmmmmmm
- o Exec location nnnnnn; Exec called from user location mmmmmmm

MONITOR STOPCODE DEFINITION

2.3 STOP Stopcodes

A STOP stopcode indicates an internal error that endangers the integrity of the entire system. All jobs are aborted and the system begins to dump and reload the monitor automatically. The monitor prints the following message on the CTY after the stopcode notification:

Reload monitor

If the monitor obtains the necessary information, it prints a supplementary message on the CTY of the form:

```
JOB jobn on TTYnnn running name  
UUO is octal representation at user PC address  
File filespec
```

Where:

<u>jobn</u>	is the number of the job causing the error.
<u>nnn</u>	is the number of the job controlling the terminal.
<u>name</u>	is the name of the program running for that job.
<u>octal representation</u>	is the octal representation of the monitor call failing for that job.
<u>address</u>	is the value of the program counter for that job.
<u>filespec</u>	is the file specification for the file being accessed.

2.4 CPU Stopcodes

A CPU stopcode is handled differently, depending on the number of CPUs in the configuration running at the time of the crash. For a single-CPU system, a CPU stopcode has the same effect as a STOP stopcode. When a CPU stopcode occurs on the only processor running in a multiprocessor (SMP) system, the effect is the same as a STOP stopcode. All user jobs are aborted and the system begins to dump and reload the monitor automatically. The following message is printed on the CTY of the CPU that experienced the failure, after the stopcode notification:

Reload monitor

However, in a SMP system where more than one CPU is running at the time of the stopcode, the CPU stopcode aborts the job currently running, a dump is taken, and the CPU that encountered the stopcode halts.

For SMP systems in which more than one CPU is running, the following message is displayed on the CTY for the CPU that encountered the stopcode:

[Stopping CPU]

MONITOR STOPCODE DEFINITION

If the monitor obtains the necessary information, it prints a supplementary message on the CTY of the form:

Job jobn on TTYnnn running name
UUO is octal representation at user PC address
File filespec

Where:

<u>jobn</u>	is the number of the job causing the error.
<u>nnn</u>	is the number of the job controlling the terminal.
<u>name</u>	is the name of the program running for that job.
<u>octal representation</u>	is the octal representation of the monitor call failing for that job.
<u>address</u>	is the value of the program counter for that job.
<u>filespec</u>	is the file specification for the file being accessed.

2.5 HALT Stopcodes

A HALT stopcode indicates a fatal error and affects the entire system. The monitor cannot automatically reload. The system halts, and you must manually dump and reload the monitor. (Refer to the TOPS-10 Operator's Guide.)

HALT stopcodes generate the following message:

KL HALTED

2.6 INFO Stopcodes

An INFO stopcode is a report on a system event that may be of interest in debugging crashes. These stopcodes are for informational purposes only, and do not interrupt system or job execution.

MONITOR STOPCODE DEFINITION

3 CONTINUABLE STOPCODES

Continuable stopcodes occur when the monitor executes a STOPCD macro, dumps the memory image, and continues the system automatically. HALT and STOP stopcodes are not continuable. A CPU stopcode is continuable on SMP systems where more than one CPU is running (see Section 2.4).

The following examples illustrate three continuable DEBUG stopcodes and a JOB stopcode, as displayed on the CTY:

Example 1:

?CPU0 monitor error. Stopcode name is ICN

CPU Status Block on 30-May-86 19:29:21

CONI APR, = 001060,,004102
CONI PI, = 000000,,000777
CONI PAG, = 000000,,020000
DATAI PAG, = 500100,,000002
[Dumping on DSK:CRASH.EXE[1,4]]
[Continuing system]

Example 2:

?CPU1 monitor error. Stopcode name is EUE
Job 5 on TTY1 running DDT User [1,2]
UUO is 0 at user PC 002472

CPU Status Block at 4-Oct-86 8:16:36

APRID = 000231,,342002
ERA = 600000,,040513
CONI APR, = 007760,,000003
CONI PI, = 000000,,000377
CONI PAG, = 000000,,620001
DATAI PAG, = 700100,,002255
AR ARX Data Word = 000000,,057000
IO Page Fail Word = 000000,,000000

SBUS Diags:

CNTRLR FNC 0 FNC 1
000004 007040,,040610 000200,,000000

[Dumping on DSK:CRASH.EXE[1,4]]
[Continuing system]

Example 3:

?CPU1 monitor error. Stopcode name is IEZ
File DSKE0:OPSER.LOG[1,2]
Job 1 on CTY running OPSER User [1,2]
[Dumping on DSK:CRASH.EXE[1,4]]
[Continuing system]

MONITOR STOPCODE DEFINITION

The following example illustrates a JOB stopcode message that is displayed on the CTY:

```
%DECsystem-10 not running
```

```
?CPU0 monitor error. Stopcode name is IME  
Job 1 on TTY5 running FH702 User [1,2]  
UO is 47240040770 at user PC 006477
```

```
CPU Status Block on 24-Jan-86 15:06:00
```

```
APRID = 640336,364654  
ERA = 024000,,006451  
CONI APR, = 007760,000001  
CONI PI, = 000000,,000377  
CONI PAG, = 000000,,660001  
DATAI PAG, = 700100,,001340  
AR ARX Data Word = 000000,,000000  
IO Page Fail Word = 000000,,000000
```

```
SBUS Diags:
```

```
CNTRLR FNC 0          FNC 1  
000004 001740,,017321 000200,,000000  
000010 006160,,006603 000500,,001000
```

```
[Dumping on BLUI:CRASH.EXE[1,4]]  
[Aborting Job]
```

```
[DECsystem-10 Continued]
```

The same JOB stopcode message would also appear on the job's terminal, in addition to the following message:

```
?Monitor error at user PC 006476
```

MONITOR STOPCODE DEFINITION

4 LIST OF MONITOR STOPCODES

A list of stopcodes for all systems that run the TOPS-10 monitor is presented on the following pages in alphabetical order. The list shows the name of each stopcode, the calling module, the type of stopcode, a phrase message (for which the name is a symbol), and a brief explanation of the containing routine, the error that caused the stopcode, and any data items that can be helpful in analyzing dumps.

<u>Name</u>	<u>Module</u>	<u>Type</u>	<u>Message and Explanation</u>
AAD	FILEND	DEBUG	Access table Already Dormant ATNLNK unlinks an access table from a name block (NMB) ring. This stopcode occurs when an attempt is made to make an access table dormant, but the table is already dormant. Data Items: T1 = location of access table T2 = location of predecessor T3 = location of next in ring
AAO	APRSER	JOB	Access Allowed is Off The monitor converts virtual IOWDs into physical IOWDs. This stopcode occurs while checking the access bits for a page pointed to by the IOWD, and access to that page is not allowed. Data Items: T1 = total number of words accumulated so far T2 = number of words for current page T3 = current page number within this segment T4 = next page number within this segment (T4 is the page that was not allowed access)
ABK	APRSER	EVENT	Address Break
AES	FILEND	JOB	Abnormal,End of Search list SLXAES is called from several places in FILEND. This stopcode occurs for many reasons, such as unexpectedly encountering the end of a search list.
ALW	FILUO	JOB	Access table Linked Wrong
ANFAIB	NETSER	STOP	No buffer set up when advancing input When attempting to finish processing the filling of the current user's input buffer, no input buffer (DEVAXI) is found to be set up. Data Items: F = address of DDB

MONITOR STOPCODE DEFINITION

ANFAOB NETSER STOP No buffer set up when calling NTDAOB

When attempting to finish processing the emptying of the current user's output buffer, no output buffer (DEVAXO) is found to be set up.

Data Items: F = address of DDB

ANFBLW NETSER STOP Buffer Length Wrong

On a call to MKNPCB/MKUPCB to allocate an ANF Protocol Control Block, the length of the data buffer associated with the PCB did not match the calculated length. (The PCBs are stored in free lists sorted by the PCB data buffer size.)

Data Items: U = address of PCB
T1 = length of PCB data buffer, in words
T2 = length PCB data buffer should be

ANFCGM NETSER STOP Cannot Get Message

After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT to get a free PCB failed.

ANFCIL NETSER STOP Connect Initiate, message too Long

An attempt was made to send a Connect Initiate (or possibly a Connect Confirm) message that exceeded the maximum "reasonable" size of a connect message.

Data Items: F = address of DDB
W = address of NDB
P3 = byte count for proposed connect message

ANFCLA NETSER STOP LAT still assigned in CLNNDB

CLNNET was called to "clean up" an ANF network Device Data Block, but the DDB claims to still be connected to a remote node. (CLNNET should be called only after the device has been disconnected.)

Data Items: F = address of network DDB

ANFCND NETSER STOP CLNNDB has No DDB

CLNNET was called to "clean up" an ANF network Device Data Block, but register F contained zero.

MONITOR STOPCODE DEFINITION

ANFDDQ NETSER STOP Data request count went negative
The count of outstanding data requests available for a network device went negative.
Data Items: F = address of DDB

| ANFDLA NETSER STOP No DLA on connect
When disconnecting an ANF network device, a call to NCS DSC discovered that the DDB's Destination Link Address was zero.
Data Items: F = address of DDB

ANFDMU NETSER STOP Data buffer Messed Up
On a call to MKNPCB/MKUPCB to allocate an ANF Protocol Control Block, a free PCB's data buffer check words were found to have been corrupted. The memory word immediately preceding the data buffer should contain SIXBIT/NET/ in its left half, and the memory word immediately following the data buffer should contain SIXBIT/NETMEM/.
Data Items: U = address of free PCB
T1 = contents of bottom-end check word
T2 = contents of top-end check word

ANFDRQ NETSER STOP Data ReQuest count negative
When processing a Data Request message for a device from a remote node, the current data request count was negative.
Data Items: F = address of DDB

ANFDRZ NETSER STOP Sending Data Requests to device 'Zero'
A call to NCS DRQ to send data requests to a remote node discovered that the network device's Destination Link Address field was zero.
Data Items: F = address of DDB

ANFDS1 NETSER STOP Cannot send Disconnect message (no PCB?)
After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT to get a free PCB failed.

MONITOR STOPCODE DEFINITION

ANFFCW NETSER STOP Free PCB Count Wrong

The PCBSEC routine encountered a discrepancy in a PCB free list: the free list counter claimed there are no free PCBs in this particular list, yet the free list chain contains at least one free PCB.

Data Items: T2 = index into NTFREC/NTFREF tables

ANFFEK NETSER STOP FEK bad, FEKOAD and FEKOPT in conflict

On a call to NETWRT/FRCWRT to queue an output network Protocol Control Block to an ANF network Front End Kontroller, the FEK output queue was found to be inconsistent: the count of output PCBs as specified by FEKOPT did not match the actual output queue in FEKOAD.

Data Items: U = address of PCB
J = address of FEK
T3 = actual count of PCBs in FEKOAD queue

ANFGFK NETSER STOP Garbage FEK pointer

Some routine attempted to convert an ANF network Front End Kontroller (data block) address into a logical line number, but the FEK address is not in the FEK chain (that is, the FEK does not exist).

Data Items: J = address for non-existent FEK

ANFIFC NETSER STOP Illegal FEKINT function Code

An ANF network Front End Kontroller called NETSER's FEK interrupt service (FEKINT) with an illegal function code in T1.

Data Items: J = address of FEK
T1 = FEK interrupt function code

ANFINP NETSER STOP INPUT done interrupt with no input buffer

An ANF network Front End Kontroller called NETSER's FEK interrupt service (FEKINT) to process an "input done" interrupt yet had no ANF network Protocol Control Block specified in either FEKIDN or FEKIAD (depending on the interrupt function code).

Data Items: J = address of FEK

MONITOR STOPCODE DEFINITION

ANFLAA NETSER STOP LAT Already Assigned

When attempting to establish a connection to a remote device's node, the Device Data Block was found to already have a Link Address Table entry.

Data Items: F = address of DDB
T1 = LAT address

ANFLAT NETSER STOP DDB and LAT don't agree

When freeing up an ANF network Device Data Block, a call to GIVSLA discovered that the DDB's link address did not match the network Link Address Table. (The LAT entry did not point back to the DDB which points to the LAT entry.)

Data Items: F = address of DDB
P2 = DDB's link address (index into LAT)

ANFLCC NETSER STOP Not in confirm wait -- LAT table messed up

While waiting for a Connect Initiate message to be answered, the Link Address Table state for the device transited into an illegal state (a state that was neither success, nor failure, nor still-waiting).

Data Items: F = address of DDB
T1 = LAT state

ANFLDD NETSER STOP LAT and DDB Disagree

When disconnecting an ANF network device, a call to NCSDSC discovered that the DDB's link address did not match the network Link Address Table. (The LAT entry did not point back to the DDB which points to the LAT entry.)

Data Items: F = address of DDB

ANFMBL NETSER STOP Buffer Length wrong

On a call to RMVPCB to free up an ANF network Protocol Control Block, the PCB data buffer length was found to be too large (larger than the maximum allowable data buffer size).

Data Items: U = address of PCB
T2 = PCB data buffer length, modulo MSGAGW

MONITOR STOPCODE DEFINITION

ANFMDL NETSER STOP Must have a DLA assigned

While trying to format an ANF network output message header, a device was found to have no Destination Link Address set.

Data Items: F = address of DDB
U = address of PCB

ANFMRL NETSER STOP Message Request too Large

A call was made to the MKNPCB/MKUPCB routine(s) to allocate an ANF network Protocol Control Block (data buffer) which exceeded the maximum configured message size.

Data Items: T1 = requested PCB size, in words

ANFMSQ NETSER STOP Message Queues are Screwed up

ANFNCT NETSER STOP NCT processors shouldn't skip

An internal-to-NETSER message processing routine took a "skip" return, which should never happen.

Data Items: U = probably an address of a PCB

ANFNFI NETSER STOP Sending Node ID to the null FEK

An attempt was made to send an ANF Node ID message to the null Front End Kontroller. The null FEK never goes offline, and so should never come online, and so should never need to be greeted with a Node ID message.

Data Items: J = address of FEK

ANFNFK NETSER STOP This is the Null FEK

The monitor received and processed a Node ID message from the null Front End Kontroller. Since NETSER never sends a Node ID over the null FEK, it should never receive one.

ANFNFP NETSER STOP No Free PCBs or no free messages

After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT to get a free PCB failed.

ANFNIL NETSER STOP Not at Interrupt Level

The main ANF network service loop NETSCN was called at a level other than interrupt level.

MONITOR STOPCODE DEFINITION

ANFNNT NETSER STOP No NDB for Terminal

In the process of disconnecting a network terminal, no Node Data Block can be found to match the remote node number as specified in the terminal's LDB (LDPRNN pointer).

Data Items: U = address of terminal LDB

ANFNPL NETSER STOP No PCBs on List

A call was made to CLNFEEK to "clean up" and reinitialize an ANF network Front End Kontroller (data block). The count of queued transmit Protocol Control Blocks (contained in FEKOCT) did not match the actual output PCB chain (starting with FEKOAD).

Data Items: J = address of FEK

ANFNUL NETSER STOP NULL FEK being "cleaned"

A call was made to CLNFEEK to "clean up" and reinitialize an ANF network Front End Kontroller (data block), but the specified FEK was the so-called null FEK, which should never go offline.

Data Items: J = address of FEK

ANFNWA NETSER STOP Node Went Away

When disconnecting an ANF network device, a call to NCSDSC discovered that the device's remote node "went away" (the node is no longer in communication with the local host). In this case, the device service routine should have simply destroyed the DDB, rather than trying to disconnect it first (for example, just call ZAPNET).

Data Items: F = address of DDB

ANFOBS NETSER STOP OBSolete feature

On a call to ZAPPCB to deallocate (and return to the monitor's general purpose free pool) an ANF network Protocol Control Block, a PCB was found to own a "secondary" data buffer (in addition to the primary data buffer), a feature no longer in use.

Data Items: U = address of PCB

MONITOR STOPCODE DEFINITION

ANFOUT NETSER STOP OUTput done interrupt with no PCB

An ANF network Front End Kontroller called NETSER's FEK interrupt service (FEKINT) to process an "output done" interrupt, yet had no ANF network Protocol Control Block address in FEKODN.

Data Items: J = address of FEK

ANFPCC NETSER STOP Count of PCBs on list is wrong

A call was made to CLNFEK to "clean up" and reinitialize an ANF network Front End Kontroller (data block). The count of queued transmit Protocol Control Blocks (contained in FEKOCT) did not match the actual output PCB chain (starting with FEKOAD).

Data Items: J = address of FEK

ANFPCL NETSER STOP PCB Lists screwed up

On a call to MKNPCB/MKUPCB to allocate an ANF Protocol Control Block, the PCB "free" list(s) were found to be inconsistent. The PCB free pool consists of linked lists (sorted by PCB size) of unused (and therefore available to be re-used) free PCBs. The count of free PCBs disagrees with the length of the linked list of free PCBs.

Data Items: U = address of (alleged) free PCB
 T2 = index into NTFREC/NTFREF linked list table

ANFPCM NETSER STOP PCB data buffer check words trashed

On a call to RMVPCB to free up an ANF network Protocol Control Block, the PCB data buffer check words were found to be corrupted. The memory word immediately preceding the data buffer should contain SIXBIT/NET/ in its left half, and the memory word immediately following the data buffer should contain SIXBIT/NETMEM/.

Data Items: U = address of PCB
 T1 = contents of bottom-end check word
 T2 = contents of top-end check word

MONITOR STOPCODE DEFINITION

ANFPCR NETSER STOP PCB tag word trashed

On a call to RMVPCB to free up an ANF network Protocol Control Block, the PCB was found to be corrupted, either the PCBTAG word did not contain SIXBIT/PCBTAG/, or the PCB data buffer length was not a proper size (in particular, was not a multiple of the MSGAGW allocation "granularity").

Data Items: U = address of PCB
T1 = contents of PCBTAG word
T2 = length of PCB data buffer

ANFPCT NETSER STOP PCB trashed

On a call to MKNPCB/MKUPCB to allocate an ANF Protocol Control Block, an unused PCB removed from the PCB free lists was found to contain junk in the PCBTAG check word (which should contain SIXBIT/PCBTAG/).

Data Items: U = address of unused PCB
T1 = contents of PCBTAG word

ANFPCV NETSER STOP PCB not Valid

On a call to NETWRT/FRCWRT to queue an output network Protocol Control Block to an ANF network Front End Kontroller, the PCB check word was found to be corrupted. The PCBTAG word of the PCB should contain SIXBIT/PCBTAG/.

Data Items: U = address of PCB
J = address of FEK

ANFRDN NETSER STOP Routing header has bad Destination Node

When processing a just-received ANF network message, the destination node number (as specified in the message) did not match the local host node number.

Data Items: U = address of PCB
W = address of NDB which sent the message
P1 = current byte pointer into PCB data buffer

ANFRDT NETSER STOP Releasing DDB Twice

ZAPNET was called to release an ANF network Device Data Block, but the DDB is already marked as having been cleared (and has been returned to the monitor free pool).

Data Items: F = address of network DDB

MONITOR STOPCODE DEFINITION

ANFRSN NETSER STOP Routing header has bad Source Node

When processing a just-received ANF network message, the source node number (as specified in the message) did not match the remote node number from the ANF network Node Data Block from which this message was de-queued.

Data Items: U = address of PCB
W = address of NDB which sent the message
P1 = current byte pointer into PCB data buffer

ANFSBA NETSER STOP Secondary Buffer Allocated ("old feature")

On a call to RMVPCB to free up an ANF network Protocol Control Block, a PCB was found to own a "secondary" data buffer (in addition to the primary data buffer), a feature no longer in use.

Data Items: U = address of PCB

ANFSLA NETSER STOP SLA on a connect

On a call to NCSCNT to send a Connect Initiate message (or possibly from NCSCNC to send a Connect Confirm message) to a remote node for a device, the network Device Data Block had no Source Link Address.

Data Items: F = address of DDB
W = address of NDB

ANFTLK NETSER STOP W points at NETNDB (TaLKing to itself)

An illegal operation was detected directed at the Node Data Block for the local host. For example, an attempt was made (in CLNNDB) to delete the NDB as no longer reachable on the network.

Data Items: W = address of NDB

ANFTMF NETSER STOP Too Many FEKs

When rebuilding the neighbor's table (for sending Neighbors messages to other nodes), there were more than NGHMAX Front End Kontrollers (neighbors).

Data Items: J = address of FEK
T1 = node number of FEK neighbor

MONITOR STOPCODE DEFINITION

ANFUBN NETSER STOP Unsent Byte count went Negative

When processing the segmentation of a user output buffer, the count of bytes remaining to be transmitted (DEVAXO+1) went negative.

Data Items: F = address of DDB

ANFUND NETSER STOP Unreasonable Network DDB length

The MAKDDB routine was called to create an ANF network Device Data Block with an "unreasonable" DDB size (either smaller than the minimum DDB length, or much (10 octal words) bigger).

Data Items: T2 = requested DDB size, in words

ANFWLA NETSER STOP Wrong LAT Assigned

A call to GETSLA to assign a free link address from the Link Address Table was made with an ANF network Device Data Block or a terminal Line Data Block that was already in the link address table.

Data Items: T1 = address of DDB or LDB

ANFWLS NETSER STOP Wrong LAT State

When processing a RELEASE monitor call for an ANF network device, the device was found to already be disconnected when its Link Address Table entry state claimed to be "OK".

Data Items: F = address of DDB
S = DEVIOS flags (IOSCON clear)
T2 = LAT state

ANFWMB NETSER STOP User Wrote in Memory before Block

On a call to GIVZWD to deallocate a block of NETSER memory and return it to the monitor's general purpose free pool, the top-end check word was found to be corrupted. The left half of the word immediately preceding the memory block should contain SIXBIT/NET/.

Data Items: T1 = LH contents of top-end check word
T2 = (adjusted) address of memory block
-1(P) = (adjusted) length of memory block

MONITOR STOPCODE DEFINITION

ANFWPE	NETSER	STOP	<p>User Wrote Past the End of the block</p> <p>On a call to GIVZWD to deallocate a block of NETSER memory and return it to the monitor's general purpose free pool, the bottom-end check word was found to be corrupted. The word immediately after the memory block should contain SIXBIT/NETMEM/.</p> <p>Data Items: T1 = LH contents of top-end check word T2 = contents of bottom-end check word 0(P) = (adjusted) address of memory block -1(P) = (adjusted) length of memory block</p>
ANFXDS	NETSER	STOP	<p>Cannot Send Disconnect message (no PCB?)</p> <p>After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT to get a free PCB failed.</p>
ANFXMT	NETSER	STOP	<p>No buffer set up when calling NTDXMT</p> <p>When processing an output buffer from a user program, NTDXMT found no output buffer pointer (DEVAXO) set up.</p> <p>Data Items: F = address of DDB</p>
ANFZFK	NETSER	STOP	<p>Zero FEK pointer</p> <p>An ANF network Front End Kontroller called NETSER's FEK interrupt service (FEKINT) with no FEK address in J.</p>
ANIUBN	NETSER	STOP	<p>Unsent Byte counts went Negative</p>
ANIXMT	NETSER	STOP	<p>No buffer set up when calling NTDXNT</p>
ANU	FILIO	DEBUG	<p>AU Not owned by Us</p> <p>UPAU obtains the Alter-UFD (AU) resource. This stopcode occurs when there is no UFD data block (UFB) for the given DDB or a job returns an Alter-UFD (AU) resource that is already owned by someone else.</p> <p>Data Items: P1 = 0 if no UFB, otherwise the UFB for DDB P2 = not applicable if no UFB, otherwise job number trying to release the AU resource UFBAUJ(P1) = job owning the AU resource</p>

MONITOR STOPCODE DEFINITION

AOC	FILEND	DEBUG	<p>Already Own CB</p> <p>GETCB gets the core block (CB) resource. This stopcode occurs when a job requests the CB resource, but already owns it.</p> <p>Data Items: J = job number</p>
AOWNSR	CLOCK1	STOP	<p>Already OWN Sharable Resource</p>
APE	APRSER	CPU	<p>Address Parity Error</p>
APF	VM SER	DEBUG	<p>Allocated Page Free</p> <p>This stopcode occurs when the monitor finds an unallocated page of memory in the list of pages allocated to a segment.</p> <p>Data Items: P2 = disk address-1 P3 = current physical page allocated P4 = number of pages T3 = byte pointer to the map slot</p>
ARF	CORE1	STOP	<p>Attempt to Return Free page</p> <p>GVPAGS returns pages to the free-core list. This stopcode occurs when the monitor checks its table of free pages before returning a page and finds that the page is already marked as being free.</p> <p>Data Items: T1 = first page on free-core list T2 = page being returned to the free-core list (this is the page that caused the stopcode) T3 = number of pages returned so far T4 = bit being tested in the page table (400000,,0) PAGTAB(T2) = page-status bits (status-bit definitions can be found in module S.MAC)</p>
ARM	FILEND	STOP	<p>Access Rings Messed up</p> <p>ATNLNK unlinks an access table from a name block ring. This stopcode occurs when an access ring is not linked to any access table.</p>
AUT18B	AUTCON	INFO	<p>18-Bit DF10/DF10C</p>
AUTTMK	AUTCON	STOP	<p>Too Many KDBs on channel</p>
AUTTTL	SYSINI	DEBUG	<p>AUTCON Took Too Long</p>

MONITOR STOPCODE DEFINITION

	AWN	FILIO	DEBUG	AU Waiters Negative
	BAC	CORE1	DEBUG	Bit Already Clear SETZRS sets zeros in a table. This stopcode occurs when the SETZRS routine attempts to zero bits that are already zero. Data Items: T2 = still on the stack = AOBJN pointer to tables T1 = number of bits to clear
	BAD	FILEND	JOB	Block Already Dormant ATSFR0 puts an access table in the free-core list. This stopcode occurs when an attempt is made to make the access table dormant, but the table is already dormant. Data Items: T1 = location of access table
	BAO	FILIO	DEBUG	Bit Already One TAKBLK allocates blocks from the disk. This stopcode occurs when the monitor attempts to allocate a block that is already allocated. Data Items: PDL = cluster address.
	BAZ	FILIO	DEBUG	Bit Already Zero GIVBLK returns disk blocks. This stopcode occurs when the monitor attempts to return blocks that are already free; this can occur when a damaged file is deleted. If this stopcode occurs with any frequency, it is suggest you run DSKRAT on the structure indicated in the stopcode printout on the CTY.
	BBS	D85INT	STOP	Bad Byte Size DLBP makes the DL10 the byte pointer for data. This stopcode occurs when the number of bytes per word supplied is illegal. The number of bytes per word must be from 2 to 6. Data Items: T4 = wrong byte size
	BDP	VMSER	DEBUG	BaD Page
	BEC	MSCCOM	DEBUG	Packet with bad End Code received
	BFC	D6DINT	DEBUG	Bad Function Code to FEK

MONITOR STOPCODE DEFINITION

BFO	TAPUO	DEBUG	<p>Better Find One</p> <p>INVERT generates a transfer list for a read backwards. This stopcode occurs when the end of the original transfer list for a read backwards is not found.</p> <p>Data Items: T2 = head of old transfer list T3 = item number to find</p>
BIN	FILIO	STOP	<p>Block number Is Negative</p> <p>MONRED reads a block or a series of blocks. This stopcode occurs when this routine is called with a negative block number.</p> <p>Data Items: T1 = IOWD for data T2 = block number</p>
BMR	FILUO	JOB	<p>Block Missing from RIB</p> <p>ALLPOA is used to work with a RIB whose blocks are allocated but not used. This stopcode occurs when the RIB for a file shows that the file has more blocks than actually exist.</p> <p>Data Items: T3 = missing block</p>
BNR	FILUO	JOB	<p>Block Not RIB</p> <p>NOTOLD creates a new name in a directory block. This stopcode occurs when a pointer to a block is not found in the RIB.</p> <p>Data Items: P2 = block that is being looked for</p>
BNT	FILEND	DEBUG	<p>Block Not There</p> <p>UFORSS gets a UFD or an SFD access block. This stopcode occurs when a core block (AT or UFB) that is known to exist is not found.</p>
BNU	SYSINI	HALT	<p>Bootstrap NXMTAB Unavailable</p>
BNZ	CORE1	DEBUG	<p>Bit Not Zero</p> <p>CSETOS sets bits in a table. This stopcode occurs when one of the bits to be set in a table is already set.</p> <p>Data Items: T3 = number of bits to set T4 = address,,position</p>

MONITOR STOPCODE DEFINITION

BPE CLOCK1 JOB Breakpoint PC Executive mode

The user PC on a control-D to a DDT unsolicited breakpoint trap did not have the USRMOD bit, although the user was supposed to be in user mode.

Data Items: T1 = PC word that the user is currently running

BPF CLOCK1 JOB Breakpoint PUTWRD Failed

The attempt to store the return PC (in processing an unsolicited ^D breakpoint) into the location pointed to by .JBBPT failed even though the address had been previously address checked.

BPT FILFND JOB Bad search list Pointer

SLXBPT is used only for this stopcode and is called from FILUOO. This stopcode occurs when an attempt is made to build the search list, but no search list can be found.

BRC COMCON DEBUG Bad Return from CMPBIT

SAVEXE saves a virtual memory system core image. This stopcode occurs when the routine that computes the flag bits for a directory entry gives an error return.

BSN SEGCON STOP Bad Segment Number

COMIT compares the job number with the right half of JBTSN(T1). This stopcode occurs when the right half of J and the right half of JBTSN(T1) are not the same.

Data Items: T1 = high-segment number of job
J = job number

BSY XTCSE DEBUG DA28 BuSY

The BUSY bit was (still) on when the DA28 interrupted.

BWA FILIO JOB Block Went Away

SETLST sets up an I/O list block. This stopcode occurs when NXTBLK, which is used to return the next block address, gives an error or a non-skip return that indicates end-of-file, writing in the middle of a file, or another similar error.

Data Items: J = job number

MONITOR STOPCODE DEFINITION

	CAC	VM SER	STOP	Cannot Allocate Clock level pages
	CAO	FILUOO	DEBUG	Cluster Address Odd ADJALC allocates the initial blocks for a file. This stopcode occurs when a block supercluster address is not an even multiple of a block cluster address. Data Items: T1 = number of blocks in group P2 = remainder
	CBB	COMCON	DEBUG	Command Block Bad In copying the program-to-run information from a user defined command block to the SAVGET locations, it was found that the command block was too big to fit.
	CCPNUL	SYSINI	STOP	Cannot create PDB for NUL1 job
	CCR	CLOCK1	STOP	Can't Continue with Resource
	CCW	CLOCK1	DEBUG	Control-C Count Wrong
	CDA	FILIO	DEBUG	In-core Copy Doesn't Agree DD2MN copies pointers from a DDB to a monitor buffer to perform monitor I/O. This stopcode occurs when the cluster pointer from the structure data block does not agree with the in-core copy. Data Items: T1 = in-core copy T4 = pointer from structure data block
	CDE	APR SER	CPU	Cache Directory parity Error
	CDRASC	NETDEV	STOP	NTDSIB failed in C.ASCI After calling NTDIBA to guarantee a user input buffer is available, the ANF network card reader service routine was unable to set up an input buffer to receive a network message. Data Items: F = address of DDB U = address of PCB

MONITOR STOPCODE DEFINITION

CDRBIN NETDEV STOP NTDSIB failed in C.BIN

After calling NTDIBA to guarantee a user input buffer is available, the ANF network card reader service routine was unable to set up an input buffer to receive a network message.

Data Items: F = address of DDB
U = address of PCB

CDRIMG NETDEV STOP NTDSIB failed in C.IMAG

After calling NTDIBA to guarantee a user input buffer is available, the ANF network card reader service routine was unable to set up an input buffer to receive a network message.

Data Items: F = address of DDB
U = address of PCB

CFP CLOCK1 JOB Can't Find PDB

SETRUN sets the job-status run bit. This stopcode occurs when there is no process data block (PDB) for this job.

Data Items: J = job number

CGB FILUOO DEBUG Can't Get Buffer to read bad block

CI7 UOCON DEBUG CI7 continuable snapshot dump

A user requested continuable snapshot dump was requested (CONFIG program SNAPSHOT command. See TOPS-10 Operator's Guide.)

CIB CLOCK1 CPU CPU Interlocks Broken

APRSUB services common APR interrupts. This stopcode occurs when the CPU interlock has been modified. Typically, the stopcode occurs while trying to continue a CPU that has stopped due to a fatal error.

Data Items: .CPNBI = CPU interlock that was modified

CIF FH XKON DEBUG RC10 Isn't Fancy

This stopcode occurs when the monitor attempts an RP04-only function on an RC10-controlled device.

MONITOR STOPCODE DEFINITION

CIO REFSTR DEBUG CPF Is Odd

COMCFP computes a CFP. This stopcode occurs when the number of blocks per supercluster is not a multiple of the number of blocks per cluster.

| CLO SCNSER DEBUG Chunk Links to 0

DELCHR deletes characters from the user's input buffers when he presses the RUBOUT or DELETE key. This stopcode occurs when a TTY chunk has a backward link to 0.

Data Items: T2 = current chunk

CME FILEND DEBUG CFP Modulo Error

SETCFP computes a CFP. This stopcode occurs when CFP does not start at an even supercluster boundary.

Data Items: T2 = supercluster address relative to block 0 of unit
T3 = remainder

CMP LOKCON STOP Can't Move Page

In attempting to move pages out of a block of memory that is being set off-line, routine PAGFRE discovered that the free page into which we were trying to move a page was not really free.

Data Items: P2 = target (free) page

CMS VMSER DEBUG CORE1 Must Skip

SEGCON returns core allocated to a nonsharable high segment. This stopcode occurs when CORE1 gives a nonskip return when asked for core in use.

| CMU SEGCON INFO Core Messed Up

CHKTAL compares CORTAL with a value obtained from various job tables. This stopcode occurs when the core usage tables are inconsistent.

Data Items: U = free+idle+dormant
CORTAL = bit table

MONITOR STOPCODE DEFINITION

CNA	SCHED1	STOP	Core Not Available
			SWAPI swaps in a job or high segment. This stopcode occurs when an error return is given by the core-allocation routine (CORGET), which indicates that no core is available, although it has already been verified that enough core is available.
		Data Items:	J = job number
CNE	FILUOO	DEBUG	Cluster Not Even
			ADJALC allocates the initial blocks for a file. This stopcode occurs when the block computed as the start of a supercluster does not begin at an even supercluster address.
		Data Items:	T2 = starting block number
CORCNA	CORE1	DEBUG	Chunk Not Allocated
CORNAH	CORE1	DEBUG	No Allocation Header block
CORNCB	CORE1	DEBUG	Not on a Chunk Boundary
CPNDDN	APRSER	EVENT	CPNDDT Non-zero
CQO	CLOCK1	STOP	Clock Queue Overflowed
CRU	MSCCOM	DEBUG	Command Reference number Unknown
CSA	COMCON	DEBUG	Couldn't Set Access allowed
			SETAA sets the access-allowed bit for a page. This stopcode occurs when the PAGE monitor call function to set access allowed fails.
CSB	SYSINI	STOP	Crosses Section Boundary
CSE	FILIO	STOP	Checksum Error
			CHKSUM computes a folded checksum from the first data word. This stopcode occurs when the pointer for checksumming points to a word that is not in the user's address space.
		Data Items:	RH(M) = address that caused the error; J = job number

MONITOR STOPCODE DEFINITION

CSP	SEGCON	JOB	Cannot Store Path
			STONAM is used during SAVE, GET, R, and RUN commands to search a path. This stopcode occurs when there is not enough free core to store the full path specification.
		Data Items:	T2 = number of words available
CTX	CTXSER	INFO	ConTeXt skew
			The context number and saved page counts are not correct for the function being performed.
CTXFWA	CTXSER	STOP	CTXMIG's First context Went Away
CTXMCT	CTXSER	STOP	CTXMIG called Twice
CTXNIP	CTXSER	STOP	Context migration Not In Progress
CU0	NETDEV	STOP	Can't use Zero dispatch
			This stopcode occurs when an attempt is made to use a zero dispatch in the SCNSER dispatch table. A zero dispatch is illegal.
CWN	NETSER	DEBUG	Core allocation Went Negative
			GIVZWD returns monitor free core. This stopcode occurs when GIVZWD attempts to return more free core than it has.
		Data Items:	core allocation count -1(P) = number of words returned
CWP	VM SER	DEBUG	Can't Write-enable Page
D78BI	D78INT	JOB	Bad IOWD
D78IN0	D78INT	JOB	Input character count Non-0
D78NC	D78INT	JOB	Not enough free monitor Core
D78ON	D78INT	JOB	Output character count is Not equal to 0
D78PI	D78INT	JOB	Positive IOWD
D78VI	D78INT	DEBUG	Version Incorrect

MONITOR STOPCODE DEFINITION

	D8EFNC	D8EINT	DEBUG	Slave FEK Not in FEK Chain
	D8EISF	D8EINT	STOP	Bad ISR Function code
	D8ERFU	D8EINT	DEBUG	RDD Request Fouled Up
	D8ESFI	D8EINT	DEBUG	Slave FEK has Input buffer
	D8ETME	D8EINT	DEBUG	Too Many Ethernets
	D8EWFU	D8EINT	DEBUG	WRT request Fouled Up
	DA28B	XTCSER	DEBUG	DA28 is Broken
	DAU	CPNSER	DEBUG	DIE Already Unlocked
	DBZ	FILIO	DEBUG	DEPLPC Bit Zero
				<p>USET00 does a USET0. This stopcode occurs when the last group of pointers for a file is not the last group in the RIB; there should be more file pointers.</p>
				<p>Data Items: W = last block allocated T3 = last pointer in core flag</p>
	DC75WE	D85INT	DEBUG	DC75 Wrong PDP11 code
	DC76MS	D76INT	DEBUG	DC76 Message is Short
	DC76QF	D76INT	DEBUG	DC76 Queue Full
	DCR	FILUUO	DEBUG	DELRIB CPOPJ Return
				<p>CLSDL1 deletes a file. This stopcode occurs when DELRIB does not skip, even though it should always give a skip return.</p>
	DDN	MSCCOM	DEBUG	Driver Dispatch address Not setup
	DDPAHB	NETDEV	DEBUG	Already Have receive Buffer
				<p>An ANF DDP "kontroller" dispatch function call was made to post a new receive buffer, but the DDP already had a buffer pending.</p>
				<p>Data Items: F = address of DDB</p>

MONITOR STOPCODE DEFINITION

DDPALA	NETDEV	STOP	<p>Destination ALlocated Address is zero</p> <p>When trying to process an input message from a DDP in "kontroller" mode, the DECnet message block format was invalid. (The Message Segment Descriptor allocated address was zero.)</p>
DDPBAU	NETDEV	STOP	<p>Being Awfully Uncooperative!</p> <p>After making an ANF DDP device into a "kontroller", DECnet subsequently rejected the new "kontroller".</p> <p>Data Items: F = address of DDB</p>
DDPBMM	NETDEV	STOP	<p>Bad output Message Block</p> <p>When trying to process an output message for a DDP in "kontroller" mode, the DECnet message block format was invalid. (There was no first Message Segment Descriptor block.)</p> <p>Data Items: F = address of DDB P1 = address of DECnet message block</p>
DDPBCD	NETDEV	STOP	<p>Byte Copy Didn't</p> <p>When trying to process an input message from a DDP in "kontroller" mode, the EXTEND/MOVSLJ instruction to copy the data from the ANF Protocol Control Block into the DECnet Message Segment Descriptor block failed.</p>
DDPFIX	NETDEV	STOP	<p>This needs to be FIXed</p> <p>When servicing an ANF DDP device output queue, no Node Data Block could be found for the node that owns the DDP device.</p> <p>Data Items: F = address of DDB</p>
DDPFNC	NETDEV	DEBUG	<p>Illegal controller FuNction call</p> <p>The ANF DDP "kontroller" dispatch routine was called with an illegal or unknown kontroller function code.</p> <p>Data Items: T1 = controller function code</p>
DDPIDV	NETDEV	STOP	<p>Illegal DriVer number</p> <p>The DDP service routine tried to dispatch to a unknown or illegal higher-level driver type.</p>

MONITOR STOPCODE DEFINITION

DDPIOT NETDEV DEBUG Illegal Owner Type

The DDP service routine tried to dispatch to a unused or illegal higher-level driver type.

DDPKON NETDEV STOP Can't make a Kontroller Out of DDP

The DDPCKN routine was unable to set an ANF DDP device into "kontroller" mode on a DDP-initiated connect cycle.

Data Items: F = address of DDB

DDPMTB NETDEV STOP DDP Message Too Big for driver

When running in "kontroller" mode, a message was received which was too big to give to the associated higher-level driver routine.

DDPNDA NETDEV STOP No allocated Data Address

When trying to process an output message for a DDP in "kontroller" mode, the DECnet message block format was invalid. (A Message Segment Descriptor had no data buffer address.)

Data Items: F = address of DDB
P1 = address of DECnet MSD

DDPNDT NETDEV STOP No NDT entry

The Network Device Table entry in the system NDTTAB table for the DDP-class device was zero.

DDPRBA NETDEV DEBUG Receive Buffer already Allocated

On an ANF DDP "kontroller" dispatch function call to "OPEN" (or initialize) a DDP kontroller or circuit, the kontroller (DDP device) already had a receive buffer outstanding.

Data Items: F = address of DDB

DDPRDQ NETDEV STOP NTDRDQ failed

DDPSIB NETDEV STOP NTDSIB failed after NTDIBA succeeded

After calling NTDIBA to guarantee a user input buffer is available, the ANF network DDP service routine was unable to set up an input buffer to receive a network message.

Data Items: F = address of DDB
U = address of PCB

MONITOR STOPCODE DEFINITION

DDPSLJ NETDEV STOP Couldn't move the Sludge

When trying to process an output message for a DDP in "kontroller" mode, an EXTEND/MOVSLJ instruction failed in copying bytes from the DECnet message block into an ANF network Protocol Control Block.

Data Items: F = address of DDB

DDPXMT NETDEV STOP NTDXMT failed, not running out of freecore

A call to NTDXMT to output a user buffer to an ANF DDP device failed for some reason other than a lack of available network Protocol Control Blocks to hold the message.

Data Items: F = address of DDB

DDS FILUOO DEBUG DELRIB Didn't Skip

BADUFD deletes a file. This stopcode occurs when DELRIB does not skip, even though it should always give a skip return.

DEFTTI SCNSER STOP DEFault Tty Type Invalid

DEL CBD SCNSER DEBUG DELCHR went BaD

DELMBD SCNSER DEBUG DELMID went BaD

DER FILUOO DEBUG DELRIB Error Return

CLSFUL is used when there is no space on a structure or all pointer slots are taken. This stopcode occurs when DELRIB does not skip, even though it should always give a skip return.

DFU NETSER DEBUG Device Unrecognized

DSPOBJ dispatches on the object type. This stopcode occurs when the specified device is not on the network.

Data Items: T4 = object type
DEVCHR(F) = device characteristics

DHA FILIO DEBUG Don't Have AU resource

DWNAU releases the Alter-UFD resource. This stopcode occurs when this file attempts to return the AU resource when it does not own it.

Data Items: S = should have had IOSAW(200000) set

MONITOR STOPCODE DEFINITION

	DHC	SEGCON	DEBUG	Don't Have CA
	DHD	FILIO	DEBUG	Don't Have DA
				DWNDA returns the DISK ALLOCATION queue. This stopcode occurs when this file does not own the DA resource.
			Data Items:	PJOBN = job number
	DHF	FILIO	DEBUG	Don't Have FA
	DIEBAD	ERRCON	STOP	BAD PC in DIE
	DN60DD	D6SINT	DEBUG	PDP11 tried to give us too much
	DN60DI	D6SINT	DEBUG	Not expecting a To-10 done interrupt
	DN60ID	D6SINT	DEBUG	PDP11 is trying to give
	DN60VI	D6DINT	DEBUG	DN60 wrong PDP11 code
	DND	DSXKON	DEBUG	Dispatch Not in DSXKON
	DNF	FILUOO	DEBUG	DDB Not Found
				CLRddb clears a disk data block (DDB). This stopcode occurs when an attempt is made to return a DDB, but no predecessor DDB is found.
			Data Items:	F = location of DDB
	DNM	AUTCON	STOP	Data Not Mapped
	DNR	FILUOO	DEBUG	DELRIB Non-skip Return
				SETEN5 is used when the RIB is set up to insert constant values and write them out. This stopcode occurs when DELRIB does not skip, even though it should always give a skip return.
	DNS	FILUOO	DEBUG	DELRIB Non-Skip return
				CLOSR2 is called by CLRSTS when a rename is in progress at the time of a delete. This stopcode occurs when DELRIB does not skip, even though it should always give a skip return.

MONITOR STOPCODE DEFINITION

DOC FILEFND DEBUG Don't Own CB

GVCBJ returns the CB resource for a job. This stopcode occurs if GVCBJ is called, but the job does not own the CB resource.

DOM CORE1 STOP Don't Own MM resource

On a multiprocessor KL, the processor that wishes to manipulate pages in memory must own the memory-management resource before it can do so. This stopcode occurs when a processor tries to manipulate memory pages and does not own the memory-management resource. This resource can also be owned by a job.

Data Items: If the CPU owns the MM resource, INTLMO is the APRID of the CPU. If a job owns the resource, .CPJOB contains the job number (in this case, MMUSER is the job that owns the MM resource).

| DOWNSR CLOCK1 STOP Don't OWN Sharable Resource

DPL COMCON DEBUG Directory Page Lost

GTSAPV reads in a page that was output earlier. This stopcode occurs either when the page already exists or when an I/O error occurs.

DPN COMCON DEBUG Directory Page Nonexistent

RELDIR gets rid of the directory page and restores any pages that were written out earlier, after it creates the directory. This stopcode occurs when a PAGE. monitor call with a function of 1 fails.

DQR DZINT DEBUG Illegal Queue Routine

The address of the routine to transfer to on dataset timeout for a DZ-11 line was zero.

| DRQNEG NETDEV STOP Data ReQuest went NEGative

| DSX3IF DSXKON STOP 3330 Isn't Fancy

DTEARD DTESER STOP Runaway Driver

DTEBAA DTESER STOP Buffer Already there

DTEBDN DTESER STOP Bad Device Number

MONITOR STOPCODE DEFINITION

DTEBMB	DTESER	DEBUG	Bad Message Block pointer
DTEBTC	DTESER	DEBUG	Bad Transfer Count(s)
DTECDF	DTESER	STOP	Copy Data Failed
DTECOW	DTESER	STOP	Called for Output on Wrong CPU
DTEDNE	DTESER	STOP	Count Not Even
DTEDNH	DTESER	STOP	Driver Not Hungry
DTEDWA	DTESER	DEBUG	DDB Went Away?
DTEWC	DTESER	DEBUG	DTECLR called on Wrong CPU
DTEEFI	DTESER	STOP	Illegal Function code
DTEFNG	DTESER	STOP	Illegal Function code
DTEI1S	DTESER	DEBUG	Illegal To-11 done State for QP2
DTEIBA	DTESER	DEBUG	Input Buffer already Allocated
DTEIDX	DTESER	STOP	Bad InDeX in byte pointer
DTEIKF	DTESER	DEBUG	Illegal Kontroller Function
DTEIOP	APRSER	INFO	DTE20 I/O Page failure
DTEIPA	DTESER	STOP	No Post Address
DTEITM	DTESER	DEBUG	Illegal To-10 transfer Mode in QP2
DTEITS	DTESER	DEBUG	Illegal To-10 Transfer State
DTEIUD	DTESER	DEBUG	Illegal User for DTE
DTEMDM	DTESER	STOP	No Master DTE
DTEMDS	DTESER	JOB	MOVSLJ Didn't Skip

MONITOR STOPCODE DEFINITION

DTENFC	DTESER	STOP	No Free Core
DTENFP	DTESER	STOP	No Forward Pointer in MSD
DTENIS	DTESER	STOP	DTE in wrong State
DTENOD	DTESER	STOP	NO Data
DTENOM	DTESER	STOP	NO Message block
DTEOBA	DTESER	DEBUG	Output Buffer already Allocated
DTEPCI	DTESER	STOP	Function Code Illegal
DTEPTL	DTESER	STOP	Packet Too Large
DTEQEF	DTESER	STOP	Queue Entry Full
DTERWC	DTESER	DEBUG	DTERLD called on Wrong CPU
DTETNI	DTESER	STOP	DTE Not Idle
DTETQP	DTESER	STOP	Found Queue Point
DTEWCN	DTESER	STOP	Wrong CPU Number
EMA	CORE1	STOP	Exceeded Memory Allocation
EMS	VMSEK	STOP	Exceeding Maximum Section
ENQATA	QUESER	DEBUG	Bad Access Table Address Code assumes that the access table addresses are under 400,000. FRECOR access tables must be within 400,000, or you must make appropriate code changes to QUESER.
ENQAVE	QUESER	DEBUG	Already have EQ AVESTP is a stopcode-only routine. This stopcode occurs when a job has the Enqueue/Dequeue Wait resource when it should not have it.

Data Items: J = job number

MONITOR STOPCODE DEFINITION

ENQCWD	QUESER	JOB	Can't Wake job/context after DEQ DEQ. process is completed, but WAKE. to requesting program or idle context failed, for some reason. Make sure the requesting program or context handle is valid.
ENQCWJ	QUESER	JOB	Can't Wake Job/context DEQ. process is completed, but lock cannot be granted. Check the waiting process for invalid requesting job or context handle.
ENQDNL	QUESER	DEBUG	DEQ Not interLocked DEQIT dequeues one Q-entry. This stopcode occurs when the job does not own the DQ interlock. Data Items: J = job number
ENQIJC	QUESER	JOB	Illegal Job/Context handle Code tried to obtain a q-chain header for a particular context, but the attempt failed because of internal discrepancies. An invalid job or context handle was requested.
ENQLNF	QUESER	DEBUG	Lock Not Found REDTB fills a user table with data from the LOCK-associated table. This stopcode occurs when the system cannot find the LOCK block.
ENQQFU	QUESER	JOB	Q-blocks Fouled Up TSTAAC tries to determine if we need to increment the read count in the access table to make it stay around. This stopcode occurs when a zero entry is found in the link to the next queue.
ENTMPB	ETHUOO	DEBUG	Missing user Portal Block Occurs when attempting to release a user portal which doesn't exist in the user portal table.
ENTPCN	ETHUOO	DEBUG	User Portal Count Non-zero Occurs when ETHUOO tries to release an EJB (ethernet job block) which hasn't closed all user portals.

MONITOR STOPCODE DEFINITION

ENTPTF	ETHUOO	DEBUG	<p>User Portal Table Full</p> <p>Occurs when the user portal count claims there is a free entry in the user portal table, but all the entries are used.</p>
ENTUEE	ETHUOO	DEBUG	<p>Unexpected Ethser Error code</p> <p>ETHUOO got an unexpected error code back from ETHSER.</p>
ERB	REFSTR	INFO	<p>Error Reading BAT block</p> <p>REDBAT reads in BAT blocks and returns a new unit pointer. This stopcode occurs when an error is encountered while reading.</p> <p>Data Items: U = address of current unit</p>
ERF	TAPSER	STOP	<p>Error Recovery procedure Fouled up</p> <p>ERPINT is used when an interrupt is received while error recovery is in progress. This stopcode occurs when a pointer that should be pointing at an I/O request block is pointing elsewhere.</p> <p>Data Items: T1 = bad pointer TKBERB(W) = good pointer</p>
ERM	ONCMOD	DEBUG	<p>Error Reading MFD</p> <p>SPTSSB creates the SPT table and the swapping SAT table for a disk unit. This stopcode occurs when an I/O error occurs while reading the MFD read-in block.</p> <p>Data Items: P1 = address of first word of MFD RIB (RH) T2 = logical block number to read U = unit data block address F = file data block address S = (RH) standard error bits</p>
ERP	REFSTR	STOP	<p>Extraneous Retrieval Pointer</p> <p>HOMRBS stores a retrieval pointer in the HOME.SYS read-in block. This stopcode occurs when the byte pointer is confused.</p> <p>Data Items: T1 = current byte pointer</p>
ESS	FILFND	JOB	<p>Empty System Search list</p> <p>SLXEES is a stopcode-only routine and is called throughout FILFND. Examine the stack for the location.</p>

MONITOR STOPCODE DEFINITION

ETHCCC	ETHSER	DEBUG	Can't Create Channel block
ETHCCM	ETHSER	DEBUG	Can't Create Multi-cast block
ETHCCP	ETHSER	DEBUG	Can't Create PORTAL block
ETHIFC	ETHSER	DEBUG	Invalid Function Code
ETHIPS	ETHSER	DEBUG	Invalid Protocol State
ETHUDS	ETHSER	DEBUG	Unexpected Disable protocol State
ETHUES	ETHSER	DEBUG	Unexpected Enable protocol State
EUE	ERRCON	JOB	Executive UO Error EMUERR is called when a monitor call is made illegally at executive level. This stopcode occurs when the monitor call occurs at non-interrupt level.
EWB	REFSTR	DEBUG	Error Writing Block BLKWRT writes out a block. This stopcode occurs when the subroutine to do the actual writing of the block, OWNWRT, gives an error return that indicates an I/O error. Data Items: DEVMBF(F) = IOWD T2 = logical block number U = address of unit
EWH	REFSTR	DEBUG	Error Writing Home block HOMUPD updates the home blocks. This stopcode occurs when the subroutine used to do the physical I/O (WRTRUN) gives an error return. Data Items: T2 = list of items to be written S = standard error bits
EXFCHK	SCNSER	STOP	Exhausted Free CHunKs
FAD	FILUO	DEBUG	File Already Dormant CLRSTG is used during a CLOSE monitor call to do general clean-up tasks. This stopcode occurs when the access-table entry for this file is mistakenly marked dormant. Data Items: ACCDOR(T1) = access-table entry for this file

MONITOR STOPCODE DEFINITION

FDL	VMSER	DEBUG	FRDCR Lied
FDP	FILIO	DEBUG	Fixed-head Device Position FREINT handles unsolicited interrupt from a device. This stopcode occurs when a position-done interrupt occurs for a fixed-head device. This may indicate a hardware problem.
		Data Items:	KONPOS(J) = unit positioning flag
FEM	ERRCON	HALT	Fatal Error in Monitor PARHALT halts a CPU when there is a serious error in the monitor.
FFU	NETSER	STOP	F Fouled Up NETHIB puts a network job in the HIBER state. This stopcode occurs when NETHIB is called with F = 0.
FIP	VMSER	DEBUG	Free-Page In use
FLE	SCNSER	STOP	Free List Empty GETCHK fetches chunks from the SCNSER free-chunk chain. This stopcode occurs when the pointer to the first chunk (TFTAK) is zero.
FNU	FILIO	DEBUG	FA Not owned by Us
FON	VMSER	STOP	Funny address Overlaps Next GVFWDS returns words acquired by GTFWDC or GTFWDU. This stopcode occurs when the size of the current chunk plus the address of the current chunk overlaps the first word of the next chunk.
		Data Items:	T1 = words to return T2 = address
FOP	VMSER	STOP	Funny address Overlaps Previous GVFWDS returns words acquired by GTFWDS or GTFWDU. This stopcode occurs when the first word of the funny address is in the previous chunk.
		Data Items:	T1 = words to return T2 = address

MONITOR STOPCODE DEFINITION

FPE	VMSER	DEBUG	<p>Funny Page must Exist</p> <p>GVPWDS returns words acquired by GTFWDS or GTFWDU. This stopcode occurs when the monitor tries to get a page map entry, but no funny page exists from which to get the page map entry.</p> <p>Data Items: T1 = words to return T2 = address</p>
FPI	CORE1	STOP	<p>GTPAGS Free Page In use</p> <p>This stopcode occurs when the monitor tries to get a page from the free-core list, but finds a page marked "in use" on the free list.</p> <p>Data Items: T1 = first page added to or taken from the free-core list; T2 = negative number of pages to add or take from the free-core list</p>
FPN	LOKCON	STOP	<p>SETMFL Free Page Not found</p> <p>SETMFL moves monitor pages when its current pages are being set off line. This stopcode occurs when the target page is not found on the free-core list.</p> <p>Data Items: T1 = current page (being set off line) P2 = target page (this is the page that caused the stopcode)</p>
GDS	COMCON	DEBUG	<p>GETSEG Didn't Skip</p>
GNA	DATMAN	STOP	<p>GETWRD Not Available</p> <p>GETWRD was called at clock level but the word requested was not available.</p>
HIF	FILIO	DEBUG	<p>Hole In File</p> <p>USET00 is used to do a USET0. This stopcode occurs when the last block of the file exists, but some preceding block does not.</p>
HSF	FILIO	JOB	<p>HSC50 not Fancy</p>
HWU	FILIO	JOB	<p>Hardware Wrong Unit</p> <p>POSERC is used during disk error recovery. This stopcode occurs when the wrong unit on a disk controller interrupts. This is a hardware problem.</p> <p>Data Items: P2 = error bits T1 = retry count</p>

MONITOR STOPCODE DEFINITION

IBI	CLOCK1	JOB	Intercept Block Illegal
			ESTOP stops the user on an error and flags it as an error stop. This stopcode occurs when the user-defined intercept block is illegal for some reason.
IBZ	FILIO	JOB	I/O to Block Zero
			UUOPWR performs CALLIs for a disk channel. This stopcode occurs when Block 0 is requested, but the file is not HOME.SYS[1,4] or the drive is not an RP04. An RP04 reads block 0 to check for format errors.
		Data Items:	T1 = PPN T2 = file name
ICI	MSCCOM	DEBUG	Invalid Connect-ID
ICL	UUOCON	JOB	Illegal Channel number
			JDAADR returns the address of a channel in USRJDA or the extended channel table in T1. This stopcode occurs when the channel number is greater than the maximum number allowed.
		Data Items:	T2 = channel number HIGHXC# = maximum
ICN	SEGCON	DEBUG	In-core Count Negative
			DECCNT decrements the high-segment in-core count for a job that has a very high segment, which must be in core. This stopcode occurs when the count of the number of jobs in core sharing a high segment becomes negative.
		Data Items:	J = job number JBTSTS(J) = in-core count, which should be zero now
IEZ	APRSER	DEBUG	IOWD Equals Zero
			The monitor attempted to convert virtual IOWDs to physical IOWDs. This stopcode occurs when this routine is called with an IOWD of zero.
		Data Items:	T2 = IOWD P3 = location of channel data block P4 = frame count, characters/word (if DX10 channel)

MONITOR STOPCODE DEFINITION

IFI	TAPSER	STOP	<p>Illegal Function at Interrupt</p> <p>TAPIFI is a general interrupt error halt. In one case, for example, the monitor found an illegal function in an I/O request block while at interrupt level. Examine the stack for the specific error address.</p>
IIP	FILIO	STOP	<p>I/O In Progress - error</p> <p>MONIO reads a block or series of blocks from the disk. This stopcode occurs when the monitor attempts to start I/O for a DDB that already has I/O active.</p> <p style="margin-left: 40px;">Data Items: S = status bits T1 = IOWD for data T2 = block number</p>
IME	APRSER	JOB	<p>Illegal Memory reference from Executive</p> <p>This stopcode occurs when there is a page fault while in executive mode that is not an address break.</p> <p style="margin-left: 40px;">Data Items: .CPAPC = page fault PC .CPPFW = page fault word</p>
INIBEC	MSCCOM	EVENT	Bad End Code during INItialization
IOP	APRSER	CPU	<p>I/O Page failure</p> <p>There is usually a serious hardware failure. Call your Field Service representative.</p>
IPC	APRSER	CPU	<p>Illegal Page failure trap Code</p> <p>SEILM processes page failure traps. This stopcode occurs when the trap code returned by the pager after getting a page fail trap is not in the range 0 through 23 or 25.</p> <p style="margin-left: 40px;">Data Items: T1 = page fail code .CPTCX = page trap context .CPTPI = PI state</p>
IPE	VMSER	DEBUG	In Progress queue Empty
IPF	VMSER	DEBUG	<p>In-use Page Free</p> <p>SCNPT scans the page table. This stopcode occurs when SCNPT finds a page on the free-core list that is listed in the page table as allocated.</p>

MONITOR STOPCODE DEFINITION

IPM	VMSER	DEBUG	<p>Illegal Pointer in MEMTAB</p> <p>This stopcode occurs when the monitor finds an inconsistency in the swapping database.</p>
IPN	VMSER	DEBUG	<p>IPCF Page Nonexistent</p> <p>GVIPCP returns IPCF pages to the free-core list. This stopcode occurs when GVIPCP swaps out IPCF pages that are not in the swap list.</p>
IPT	CPNSER	STOP	<p>Illegal Pointer Type</p>
IPU	LOKCON	STOP	<p>IPCF Page Unowned</p> <p>In attempting to swap out an IPCF page, LOKCON was unable to find a job that contained the IPCF page in its queue.</p>
IRO	DSXKON	DEBUG	<p>I/O to Record 0</p>
ISN	DSXKON	DEBUG	<p>Illegal Sector Number</p>
IUI	FILIO	DEBUG	<p>Illegal Use of UPPDC at Interrupt level</p>
IUN	FILUOO	DEBUG	<p>Invalid Unit Number</p> <p>ERRFIN finishes a CLOSE when an error occurred. This stopcode occurs when the unit number for the UFD is illegal.</p> <p>Data Items: UN1PTR = pointer to number in structure of the unit; T1 = unit of UFD</p>
JAC	UUOCON	DEBUG	<p>Job data Area Clobbered</p> <p>IOALL does I/O for all devices assigned to a job. This stopcode occurs when the highest channel number in use is greater than 17.</p> <p>Data Items: T2 = highest channel in use</p>
JDJ	ONCMOD	DEBUG	<p>JFFO Didn't Jump</p> <p>CMPLOG computes the SIXBIT logical unit number within a structure. This stopcode occurs after a call to subroutine MSKUNI, which sets up a search mask for a unit name, and it returns no mask for the unit name given.</p> <p>Data Items: P2 = unit number STRNAM(P2) = unit name T2 = should have been the complement of the search mask</p>

MONITOR STOPCODE DEFINITION

JNC	FILIO	DEBUG	<p>Job Not in Core</p> <p>ADRINT checks that a job is in core, possibly at interrupt level. This stopcode occurs when the job is not in core.</p> <p>Data Items: T1 = job number</p>
JNE	CLOCK1	STOP	<p>JBTADR Not Equal to CORTAL</p> <p>In cross-checking JBTADR and CORTAL a mismatch was found.</p> <p>Data Items: P3 = amount of free-core specified by the sum of JBTADR entries</p>
KAF	APRSER	CPU	<p>Keep-Alive Failure</p> <p>This routine/stopcode is executed because the console front end detected that the KL did not update the keep-alive counter. This stopcode occurs when the front end executes a XCT 71.</p>
KCP	RNKKON	DEBUG	<p>KDB Command in Progress</p> <p>CMDWAT waits for a command to complete and calls RNKINR to process it before starting a new command.</p>
KDS	DPKKON	DEBUG	<p>KONEC2 Didn't Skip</p> <p>POSINT handles positioning interrupts. This stopcode occurs when the subroutine KONEC2 does not skip, even though it should always give a skip return.</p>
KLPBIO	KLPSER	STOP	<p>Buffer descriptor table Index Out of range</p>
KLPBRC	KLPSER	INF	<p>Bad Read-Counters packet</p>
KLPCKE	KLPSER	CHK	<p>SET-CIRCUIT command Error</p>
KLPCLB	KLPSER	INF	<p>CLOSE Buffer function failed</p>
KLPCEPE	KLPSER	INFO	<p>KLIPA CRAM Parity Error</p> <p>A parity error occurred on the CI20, which may indicate a hardware condition.</p>
KLPICRD	KLPSER	INFO	<p>Can't Restock Datagram free queue</p>
KLPICRM	KLPSER	INFO	<p>Can't Restock Message free queue</p>

MONITOR STOPCODE DEFINITION

KLPCRR	KLP SER	CHK	READ-REGISTER Command failed
KLPDED	KLP SER	INF	CI-20 is DEaD
KLPDFQ	KLP SER	INF	Datagram Free Queue empty
KLPEPB	KLP SER	CHK	Received Bad Error logging Packet
KLPERR	KLP SER	INF	Received packet with ERRor
KLPHLT	KLP SER	INFO	KLIPA microprocessor HaLTed The CI20 microprocessor halted for some reason.
KLPHNG	KLP SER	INF	CI-20 is HuNG
KLPILP	KLP SER	INF	Software response terminated In Locally-generated Packet
KLPIPA	KLP SER	INF	Invalid Packet Arrived
KLPIRD	KLP SER	INF	Invalid Remotely-generated Data request
KLPIRP	KLP SER	INF	Software response bit on In Remotely-generated Packet
KLPKAF	KLP SER	INFO	KLIPA Keep Alive Failed The KLIPA will reload automatically.
KLPKRW	KLP SER	HLT	CI-20 read the Wrong Register
KLPLBF	KLP SER	INF	LoopBack Failed
KLPMBE	KLP SER	INF	MBUS Error
KLPMCE	KLP SER	INF	Received an MCNF or an MDATREC with an Error
KLPMCR	KLP SER	CHK	Received an unexpected MCNF or MDATREC
KLPMFQ	KLP SER	INF	Message Free Queue empty
KLPNBD	KLP SER	HLT	No Buffer Descriptor table

MONITOR STOPCODE DEFINITION

KLPNDB	KLPSER	INF	No Datagram Buffer
KLPNDE	KLPSER	INF	Packet with bad NoDE number
KLPNEN	KLPSER	INFO	CI-20 Not ENabled
KLPNMG	KLPSER	INF	No MessaGe buffer
KLPNOD	KLPSER	HLT	Can't get datagram buffers during initialization
KLPNOR	KLPSER	INF	Remote port is sick
KLPNPB	KLPSER	HLT	No Path Block at PPD0VC
KLPNRS	KLPSER	INF	Closing VC due to No ReSponse
KLPOHF	KLPSER	HLT	Internal software inconsistency
KLPONC	KLPSER	HLT	Trying to Open a virtual Circuit which isn't closed
KLPOPC	KLPSER	INF	Packet with bad OPCode
KLPPIA	KLPSER	INF	CI-20 has lost its PI Assignment
KLPPIC	KLPSER	HLT	PCB Is Corrupted
KLPPPD	KLPSER	INF	Packet with bad PPD byte
KLPTQ	KLPSER	INF	Queue interlock Timeout
KLPRCE	KLPSER	CHK	READ-COUNTERS Command failed
KLPRLF	KLPSER	INF	CI-20 microcode ReLoad Failed
KLPRMQ	KLPSER	INF	Queue interlock timeout
KLPRSF	KLPSER	INF	CI-20 Restart Failed
KLPRSH	KLPSER	INF	Received SHUTDOWN message
KLPSCR	KLPSER	CHK	SET-CIRCUIT Command Received

MONITOR STOPCODE DEFINITION

	KLPSFQ	KLP SER	INF	Spurious Free Queue error
	KLPSTR	KLP SER	INF	CI-20 STaRted
	KLP SWO	KLP SER	INF	Received a START When VC was Open
	KLPVIR	KLP SER	CHK	VIRtual address in packet is wrong
	KLPWAB	KLP SER	INF	CI Wire A has gone from good to Bad
	KLPWAG	KLP SER	INF	CI Wire A has gone from bad to Good
	KLPWBB	KLP SER	INF	CI wire B has gone from good to bad
	KLPWBG	KLP SER	INF	CI wire A has gone from bad to good
	KNF	XTC SER	STOP	Kontroller Not Free
				XTC SER received a remote interrupt request, the the controller was not free.
	KNIARD	KNISER	INFO	KLNI auto-reload disabled
	KNICAC	KNISER	DEBUG	Can't allocate counters data buffer
	KNICAD	KNISER	DEBUG	Can't Allocate counters Data buffer
	KNICAM	KNISER	DEBUG	Can't Allocate MCAT table
	KNICAP	KNISER	DEBUG	Can't Allocate PTT table
	KNICCK	KNISER	DEBUG	Can't Create Ethernet Kontroller block
	KNICPE	KNISER	INFO	KLNI CRAM Parity Error
				A parity error occurred on the NIA20, which may indicate a hardware condition.
	KNICWS	KNISER	DEBUG	KLNI Counters buffer is Wrong Size
	KNIDPE	KNISER	INFO	KLNI Data Path Error
	KNIEPE	KNISER	INFO	KLNI EBUS Parity Error

MONITOR STOPCODE DEFINITION

	KNIHLT	KNISER	INFO	KLINI microprocessor HaLTed The NIA20 microprocessor halted for some reason.
	KNIICO	KNISER	DEBUG	Invalid Command Opcode
	KNIIFC	KNISER	DEBUG	Illegal Function Code
	KNIKAF	KNISER	INFO	KLNI Keep Alive failed
	KNIMBE	KNISER	INFO	KLNI MBUS Error
	KNIMWS	KNISER	DEBUG	KLNI MCAT table is Wrong Size
	KNIPIT	KNISER	INFO	PUTQUE Interlock Timeout
	KNIPWS	KNISER	DEBUG	KLNI PTT table is Wrong Size
	KNIRIT	KNISER	INFO	REMQUE Interlock Timeout
	KNIWUV	KNISER	INFO	Wrong microcode Version
	KSHME	APRSER	INFO	KS Hard Memory Error
	KSSME	APRSER	EVENT	KS Soft Memory Error
	KSW	TAPSER	INFO	Kontroller Status Wrong TAPSIO is used when the upper level wants to start I/O. This stopcode occurs when the tape-controller status is wrong.
			Data Items:	TKBSTS(W) = status
	LAPRBF	LATSRV	CHK	Specify Receive Buffer Failure
	LATICB	LATSRV	CHK	LATCBR called from NISRV with illegal callback function code NISRV has called the LATSRV callback routine with an invalid function code.
	LATIMT	LATSRV	CHK	LAT Illegal Message Type LAT virtual circuit message received with message type out of range.

MONITOR STOPCODE DEFINITION

LATINE	LATSRV	CHK	LATINI failed to initialize
LATIST	LATSRV	INF	LAT Illegal Slot Type
LATMEM	LATSRV	CHK	LAT buffer overwritten While trying to return a buffer, LATSRV discovered it had been overwritten.
LATNSC	LATSRV	INF	LAT Host node stopped circuit LAT Host node stopped the circuit. Look at the Reason Code in T1 and the PC in T2. This error, if relatively infrequent is nothing to be concerned about. If it occurs frequently, use the CODE and PC to determine further action.
LDBNIU	SCNSER	DEBUG	LDB Not In Use A call to the FRELDB routine was made, though the terminal's LDB does not have the LTLUSE bit set in the LDBTTW word. Normally, this bit would be set on a return from GETLDB to indicate that that LDB is in use.
LLMIL1	LLMOP	INF	Received Invalid Loopback Message Received a Loopback message that was too short or was improperly formatted. This is a MOP protocol violation by a remote node. Data items: T1 contains the received message length. T2-T3 contains the Ethernet address of the transmitting node.
LLMILF	LLMOP	INF	Invalid Loopback Function Code Received a Loopback message that was neither a Loopback reply message or a forward data message. This is a MOP protocol violation by a remote node. Data items: T1 contains the function code. T2-T3 contains the Ethernet address of the transmitting node.
LLMIR1	LLMOP	INF	LLMOP Received Invalid Remote Console Message Received a Remote Console message that was too short, too long or was improperly formatted. This is a MOP protocol violation by a remote node.

MONITOR STOPCODE DEFINITION

LLMLXF LLMOP INF Loopback Transmit Failed
LLMOP was unable to transmit a forward data message.
Data items: T1 contains the error code returned from the DLL
T2 contains the channel status returned from the DLL
T3 contains the channel on which the failure occurred

LLMMCF LLMOP CHK Declare Multicast Address Failed
Attempt to declare the Assistant Multi-Cast Address failed when the Data Link Layer was called.
Data items: T1 contains the error code returned from the DLL

LLMOPF LLMOP CHK Open Portal Failed
Failed to open an NI portal with the Data Link Layer.
Data items: T1 contains the error code returned from the DLL.

LLMRQC LLMOP CHK RB Queue Corrupted
Attempted to remove an RB queue entry from an empty queue or the RB was not on the queue.

LLMRRE LLMOP INF Response Transmit Failed
LLMOP was unable to transmit a MOP request message.
Data items: T1 contains the error code returned from the DLL.
T3 contains the channel on which the failure occurred.

LLMSB2 LLMOP CHK Specify Receive Buffer Failure
LLMOP could not post a receive buffer to the Data Link Layer.
Data items: T1 contains the error code returned from the DLL.

LLMSCA LLMOP INF Ethernet Channel Address Change
LLMOP was called by NIDDL on change of state.

MONITOR STOPCODE DEFINITION

LLMSTC	LLMOP	INF	Data Link State Change
			LLMOP was called by NIDDL on change of state. This is for information only. No corrective action required.
LN1	ERRCON	STOP	Line Not Found
			EXCALP prints a monitor call PC message for a job. This stopcode occurs when no terminals a log line can be found for the job that is causing the error.
LND	FILUO	DEBUG	Logical Name Not Found
			LN MSTP consists of only the stopcode and its recovery. LN MSTP is called when the monitor could not set up the definition of LIB that was present before an ENTER UO that could not find a file.
LNP	FILIO	DEBUG	Last Pointer Not a Pointer
			OUTGRP allocates more space for an output file. This stopcode occurs when an allocation is made, but a RIB error occurred; or when the monitor tried to deallocate the space, but the RIB pointer was invalid.
			Data Items: T2 = pointer
LNS	SCNSER	STOP	Line Not Set Up
			TSETBI clears the input and output buffers for a line. This stopcode occurs when this routine is called before the line is set up.
LNT	ERRCON	STOP	Line Not There
			HALTI prints the "Halt at ..." message and stops the job. This stopcode occurs when there is no controlling terminal line associated with the job.
LPRIXC	LLMOP	HLT	Invalid Xmit Complete
			NIDDL called back to LLMOP with a transmit complete event for an RB which is not in Transmit Initiated state. This is a software bug. Call your DIGITAL Software Specialist.
			Data Items: T1 contains the current RB state. T3 contains the status in the UN block.

MONITOR STOPCODE DEFINITION

LPRLXF	LLMOP	INF	<p>Loop Request Transmit Failed</p> <p>LLMOP was unable to transmit a forward data message.</p> <p>Data Items: T1 contains the error code returned from the DLL. T3 contains the channel on which the failure occurred.</p>
LPSIFC	LLMOP	CHK	<p>LPSCBR called with invalid function code</p> <p>The LLMOP Loopback Protocol Server Call Back Routine was called by the Data Link Layer with an invalid callback function code. This is a software bug. Call your DIGITAL Software Specialist.</p>
LPU	FILUOO	JOB	<p>Last Pointer Unit-Change</p> <p>ALLPOB writes the redundant RIB in the last block of the RIB. This stopcode occurs when the RIB pointer is decoded as a unit-change pointer.</p> <p>Data Items: T2 = pointer</p>
MBW	VMSER	DEBUG	<p>Must Be in Working set</p>
MCCWNE	SCNSER	STOP	<p>Meta character called when not expected</p>
MCI	MSCCOM	DEBUG	<p>Missing connect id</p>
MCM	METCON	DEBUG	<p>Meter Channel Data Block Missing</p> <p>RELCHN releases a channel. This stopcode occurs when an attempt is made to release a meter channel data block (MCDB) that is not there.</p> <p>Data Items: T2 = predecessor MCDB (if any)</p>
MCN	FILFND	DEBUG	<p>Mount Count Negative</p> <p>SLSR6 documents the mount count when the search list is modified. This stopcode occurs when the mount count for a structure becomes negative.</p> <p>Data Items: STRMNT(T3) = mount count</p>

MONITOR STOPCODE DEFINITION

MCRBRN NETDEV STOP Bad remote node number

When processing an ANF network "node down" condition, NETMCR (MCRNWD) was called to deal with a terminal belonging to the no-longer-accessible node, but the terminal claimed to belong to a different node.

Data Items: U = address of LDB
T1 = node number from LDB
P1 = node number that "went down"

MCRDSF NETDEV STOP Disconnect failed

When processing an ANF network terminal Disconnect message, a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, then a subsequent call to PCBEGT to get a free PCB failed.

Data Items: U = address of LDB

MCRILS NETDEV STOP Illegal state

While waiting for an ANF network terminal request to be honored, the terminal's Link Address Table state transited into an illegal state (neither accepted, nor rejected, nor still waiting).

Data Items: U = address of LDB
T2 = illegal LAT state

MCRJIZ NETDEV STOP Job number Is Zero

On a program-initiated, or user command initiated connect request for a remote ANF network terminal, .CPJOB was zero.

Data Items: U = address of LDB
W = address of NDB

MCRNCO NETDEV DEBUG No character for output

On a call to MCXDAT to build a Data Message for an ANF network terminal, the LRLTTO flag (valid character in LDPCHR) was not set in the terminal's Line Data Block. MCXDAT is only called when there is at least one valid data character to be transmitted, and the first data character is always in LDPCHR.

Data Items: U = address of LDB

MONITOR STOPCODE DEFINITION

MCRNSN NETDEV STOP No source node for terminal

When processing a "Disconnect with reconnect" for an ANF network terminal, no Node Data Block can be found for the connected terminal (SRCNDB failed).

Data Items: U = address of LDB

MCRNWA NETDEV STOP MCRNWD should have disconnected us

When processing a Disconnect request for an ANF network terminal, no Node Data Block can be found for the connected terminal (SRCNDB failed).

Data Items: U = address of LDB

MCRSLZ NETDEV STOP SLA is zero

On a call to MCRXCN to send a Connect Initiate message (or possibly a Connect Confirm message) to a remote ANF network node for a network terminal, the terminal Line Data Block had no Source Link Address.

Data Items: U = address of LDB
W = address of NDB

| MCW LOKCON STOP MOFPLG Count Wrong

| MIC VMSEK JOB Monitor Is Confused

MIF MSCCOM STOP MSCP driver initialization failed

MIZ VMSEK DEBUG MEMTAB Is Zero

This stopcode occurs when the monitor finds an inconsistency in the swapping database.

MMR LOKCON STOP Moving Monitor Page Not Required

SETMFL sets memory pages off line. This stopcode occurs when the memory location labeled MOFLPG indicates that monitor pages must be moved, but none of the page's PAGTAB entries has the monitor code bit set.

Data Items: MOFLPG = number of monitor pages that were to be moved
LOKREL = number of pages, first page number to set off line

MONITOR STOPCODE DEFINITION

MNA FILIO JOB Monitor Buffer Not Available
GTMNBF gets a monitor buffer. This stopcode occurs when there is no buffer space available.

Data Items: T2 = number of words requested

MNM SYSINI STOP Monitor in Nonexistent Memory
KIINI initializes a KI10, KL10, or KS10. This stopcode occurs when a page is found to be nonexistent and the page is not free. Therefore, the monitor already has the page in question.

Data Items: T3 = page number

MNR ERRCON HALT Master -11 Not Running
DIE recovers/reloads after an internal system error. This stopcode occurs when the -11 with a DTE in privilege mode is not running.

MOIFC LLMOP INF LLMOP Received an invalid MOP message
The LLMOP Remote Console Protocol Server received a MOP message with an invalid function code. This is a MOP protocol violation by a remote node.

MPF MSCCOM DEBUG MSCCOM packet send failed

MPN LOKCON STOP Monitor Page Not Found
SETMFL sets monitor pages off line. This stopcode occurs when the source page cannot be found in the monitor.

Data Items: P3 = monitor source page

N8C CPNSER JOB Not 8 Cached Pages
FIXOTB fixes up OUCHTB when turning off the cache for some page so that OUCHE references through four cached pages. This stopcode occurs when four cached pages cannot be found.

Data Items: P3 = page for which cache is being turned off

MONITOR STOPCODE DEFINITION

NAP	FILUJO	JOB	Not Address Pointer
			UFDNXT initializes the next block for the directory. This stopcode occurs when the new pointer is decoded as other than an address pointer.
		Data Items:	P1 = location in monitor buffer T2 = bad pointer
NCA	CLOCK1	STOP	No Core Assigned
			NULADR restores the software state and then the hardware state of the new job to be run. This stopcode occurs when the job to be run has no core assigned to it.
		Data Items:	J = job number
NCC	LOKCON	STOP	Not Enough Contiguous Free Core
			Certain types of monitor pages must be kept contiguous. This stopcode occurs when LOKCON does not have enough contiguous space to lock jobs after memory has been set off-line.
NCE	UUOCON	DEBUG	No Command Slot Available
			This stopcode occurs when attempting to insert a user-defined command. The condition should have been caught earlier.
NCM	IPCSE	JOB	No Core For Message
			SETQSR sets up IPCF packets to send to QUASAR. This stopcode occurs when no core is available to build the message.
NCS	SYSINI	STOP	No Core at SYSINI time
NCT	SCNSER	STOP	No Core for Terminal chunks
NDJ	SCNSER	DEBUG	No DDB For Job
			TTYFND finds a terminal number for the job in AC J. This stopcode occurs when no device data block can be found for this job's terminal.
		Data Items:	J = job number
NDL	COMDEV	STOP	No DECnet Loaded
			This stopcode occurs if any DECnet-only routines are called, but DECnet is not assembled into the monitor.

MONITOR STOPCODE DEFINITION

NDP CLOCK1 JOB Not DDB Pointer
WSYNC waits until the current buffer activity is complete. This stopcode occurs when this routine is called with other than a DDB pointer in F.
Data Items: F = the supposed DDB pointer

NDS CLOCK1 STOP Null Job Did SAVGET
MONSTR sets up ACs for a monitor job that starts at monitor call level. This stopcode occurs when the job number is 0.

NEM LP2SER JOB No Exec Virtual Memory
DVL RAM loads the RAM or VFU with data from the user. This stopcode occurs when DVL RAM tries to map the user virtual address into exec virtual memory, but there is none.
Data Items: F = DDB
T1 = function

NER FILUOO DEBUG No Extended RIB
CLS02A looks for the last written block in the next RIB. This stopcode occurs when the pointer for the last block of a file is not in the RIB, and there is no extended RIB; the pointer is lost.

NEV UUOCON STOP No Executive Virtual Memory
NEWBUF sets up a byte pointer and item count for I/O. This stopcode occurs when the DDB does not have executive virtual memory.
Data Items: T1 = input buffer header address

NFB FEDSER STOP No Front-End Device Block
FDIGET gets the front-end device (FED) address in F. This stopcode occurs when no device data block is found for this front end.
Data Items: T1 = FED unit number (also on this stack)

NFD RPXKON DEBUG No Front-End Drive
DAVIN1 starts an operation on a drive that is busy on the other port. This stopcode occurs when DAVIN1 cannot find the drive number.

MONITOR STOPCODE DEFINITION

NIF	RNKKON	DEBUG	<p>RNKKON Isn't Fancy</p> <p>This stopcode occurs if the monitor tried to unload an RP20 or read/write 10/11 compatibility mode on an RP20.</p>
NIJ	IPCSER	STOP	No IPCF database for job/context
NIV	TAPUOO	STOP	<p>Null Interrupt Vector</p> <p>TPMDON is called by TAPSER when I/O is complete to dispatch to the correct routine for processing. This stopcode occurs when the routine address for this function is null.</p> <p>Data Items: P1 = function (T1) = should be dispatch address</p>
NJT	ERRCON	STOP	<p>Null Job Has TTY</p> <p>EXCALP prints the PC of a monitor call that caused an error. This stopcode occurs when NULJOB has control of the terminal.</p>
NLB	FILUOO	JOB	<p>No Last Block</p> <p>This stopcode occurs during UFD compression if we cannot find the pointer for the last block of the UFD.</p>
NMC	ONCMOD	STOP	<p>No More Core</p> <p>OK22B sets up controllers during system startup. This stopcode occurs when the routine used to create a device data block for ONCE-only I/O (SETDDO) gives an error return, which indicates that no core is available.</p> <p>Data Items: T2 = size of chunk needed</p>
NMCPUM	UOCON	DEBUG	Need missing CPU mask
NMU	REFSTR	DEBUG	<p>No More Units</p> <p>HOMZR2 writes zeros in unused blocks in HOME.SYS. This stopcode occurs when the count of units is greater than the number that can be accessed.</p> <p>Data Items: T1 = cluster count T2 = next retrieval pointer</p>

MONITOR STOPCODE DEFINITION

	NNE	FILUOO	STOP	<p>NMB Not Found</p> <p>GETNMB gets the location of the name block (NMB) from the DDB. This stopcode occurs when there is no access table entry for the user channel.</p>
	NNR	FILUOO	JOB	<p>No Next RIB</p> <p>DELGRP returns blocks on an update ENTER. This stopcode occurs when the last block pointer cannot be found in the current RIB, so an attempt is made to scan the next RIB, but there is no other RIB.</p>
	NNS	CORE1	DEBUG	<p>Not in Non-Zero Section</p> <p>In attempting to clear/set bits in a bit table, SETR was called requesting the usage of a relative AOBJN pointer. This is only relevant for bit tables in non-zero sections and the code was not executing in a non-zero section.</p>
	.NNU	ONCMOD	DEBUG	<p>Not a New Unit</p> <p>FILMAN finds and sets up all structures on the system. This stopcode occurs when the monitor expects to find a new unit pointer as the next item read, but does not.</p> <p>Data Items: T1 = AOBJN pointer for scanning retrieval information T2 = supposed new unit pointer (Bit 18 must be set to be a new unit pointer) T3 = logical unit number in this structure P2 = address of structure</p>
	NOB	COMDEV	DEBUG	<p>"Nobody" got Obsolete Buffer</p> <p>This stopcode occurs if someone tries to pass a message to a Front End which is not owned by anyone. This is probably caused by the line driver trying to return stale data to a previous line user and getting confused.</p>
	NOCSAT	ONCMOD	STOP	<p>NO Core for SATs</p>
	NODISK	ONCMOD	STOP	<p>NODISK units configured</p>
	NODRB	FILIO	STOP	<p>Can't allocate Disk IORB during ONCE</p>
	NOR	MSCCOM	STOP	<p>Notification code Out of Range</p>
	NOTSSL	FILFND	DEBUG	<p>Not Saved Search List</p>

MONITOR STOPCODE DEFINITION

NOW	FILIO	DEBUG	No opr wait (obsolete)
NPD	FILIO	DEBUG	No Pointers in DDB EXTRIB creates an extended RIB. This stopcode occurs when an extended RIB is needed, but no pointers exist in the DDB. Data Items: DEVMBF(F) = IOWD to monitor buffer
NPI	APRSER	STOP	Not Parity Instruction SWPTRP sweeps a memory for parity. This stopcode occurs when a page fail trap occurs that is not caused by the sweep routine and, in fact, is caused by the instruction at label CPLMPI. Data Items: T1 = PC of the instruction that caused the trap (also in .UPMP + .LMPFP) .UPMP + .LMPFW = page fail word
NPJ	DATMAN	STOP	No PDB for Job This stopcode occurs when no process data block can be found for this job. Data Items: J = job number
NPN	ERRCON	STOP	Nonexistent Page Not free CPINXF fixes up the core-allocation tables after pages have been marked out in NXMTAB because of parity errors or NXMs. This stopcode occurs when the page being marked off line is in use not free). Data Items: PAGTAB(T1) = page entry of page causing the stopcode
NPU	ERRCON	STOP	Null Pushdown list Underflow This stopcode occurs when there are more POPs on the null pushdown list than matching PUSHs.
NRF	VM SER	DEBUG	SWPLST Not Really Fragmented This stopcode occurs when there is a pointer to a fragmented SWPLST entry, but the entry is not really fragmented.

MONITOR STOPCODE DEFINITION

NRM	FILUJO	JOB	<p>Next RIB Missing</p> <p>RENTRIB is used when allocation or deallocation is done and set up to do the close. This stopcode occurs when the last block pointer is not found in the current RIB and there are no other RIBs.</p>
NRS	ONCMOD	DEBUG	<p>No RIB in SAT</p> <p>FILMAN finds and sets up all structures on the system. This stopcode occurs when the monitor expects to see a read-in block, but does not.</p> <p>Data Items: T1 = the supposed RIB</p>
NSA	SYSINI	STOP	No section allocated
NSE	VMSER	DEBUG	<p>No SWPLST Entry</p> <p>This stopcode occurs when the monitor attempts to compute the unit and block numbers corresponding to a SWPLST entry, but the pointer to SWPLST points to a zero word.</p>
NSH	SEGCON	JOB	No Such High segment
NSR	REFSTR	STOP	<p>No Second RIB</p> <p>FILSET creates a file of contiguous disk space and write zeros in data blocks. This stopcode occurs when the subroutine used to scan a block of retrieval pointers to find the group pointer, SCNPTR, gives an error return because it cannot find it.</p> <p>Data Items: P1 = pointer to cluster count T2 = number of clusters in this pointer</p>
NSS	REFSTR	DEBUG	<p>No Space for SAT</p> <p>ENDSAT allocates blocks in the HOME.SYS file for SATs. This stopcode occurs when there are no free clusters left.</p>
NSU	FILIO	DEBUG	<p>No Such Unit</p> <p>USTRIB reads in the RIB and scans it from the beginning if the pointers do not encompass the desired block. This stopcode occurs when the subroutine that finds a unit (NEWUN) gives an error return indicating that the desired unit is greater than the last unit in the structure.</p> <p>Data Items: S = error bits IOBKTL SET</p>

MONITOR STOPCODE DEFINITION

NTE	SCHED1	STOP	Not Processor Queue Error
			QLNKZ is used in the requeuing of a job. This stopcode occurs when this routine is called for a job that is not in a processor queue.
			Data Items: J = job number
NUB	FILEND	JOB	No UFB Block
			STRDN4 creates an access table entry. This stopcode occurs when there is no UFD for a file even though the file exists.
			Data Items: P2 = Structure data block (LH)
NUE	FILUOO	JOB	No UFB Error
			SETUFR sets the RIBUFD word in the RIB. This stopcode occurs when an error return is given by the subroutine used to compute the RIBUFD word, but actually there is no UFD or SFD, so there can be no UFB error.
			Data Items: DEVUFB(F) = pointer to UFD DEVVSFD(F) = pointer to SFD, if any
NUI	XTCSER	DEBUG	Non-existent Unit Interrupting
			XTCSER could not find the UDB for a unit from which it received an interrupt request.
NUL	VM SER	DEBUG	Not at UOO Level
NULCCR	NULFEK	STOP	Conversion Code out of Range
NULFNC	NULFEK	STOP	FuNction code out of Range
NUP	FILUOO	DEBUG	No Unit-Change Pointer
			LSTUNI finds the last unit-change pointer in a RIB. This stopcode occurs when no change pointer is found or when the pointer is not a unit-change pointer.
			Data Items: T2 = pointer
NUT	FILIO	CPU	No unit for transfer

MONITOR STOPCODE DEFINITION

NWA	NETDEV	STOP	<p>No-one Wrote Anything</p> <p>At the end of building an ANF network message for a network terminal, TWRPCB was called to send the accumulated message, but the byte count was zero (or negative).</p> <p>Data Items: U = address of LDB P3 = byte count for NCL message</p>
NXS	VMSEB	DEBUG	<p>Non-existent Section</p> <p>DNZSPG is called to return a non-zero section page to free core. This stopcode occurs if the section of the specified page does not exist.</p>
NXU	FILIO	DEBUG	<p>Non-existent Unit</p> <p>WRTRIB writes a RIB. This stopcode occurs when a unit-change pointer points to a unit that does not exist in the structure.</p> <p>Data Items: S = error bits U = 0 if not in any F/S</p>
O1F	VMSEB	DEBUG	<p>Only 1 Fragment</p> <p>This stopcode occurs when swapping space is fragmented, but there is only one entry in the fragment table.</p>
OMR	APRSER	JOB	<p>Out of Mapping Registers</p> <p>MAPIO sets up the UNIBUS adapter mapping registers for a given IOWD following the paging of a job. This stopcode occurs when an attempt is made to point to the next mapping register, but there is none.</p> <p>Data Items: P1 = address of next paging register to be used P2 = address of first paging register used</p>
ONC	FILUOO	DEBUG	<p>Odd-Numbered Cluster</p> <p>UPDGIV deallocates or truncates blocks from a file. This stopcode occurs when the number of blocks allocated to a file is not an even multiple of the number of clusters allocated.</p> <p>Data Items: T1 = number of clusters T2 = remainder</p>
ONCMBM	ONCMOD	DEBUG	<p>Monitor Buffer Missing</p>
ONCPUI	CPNSER	STOP	<p>ONCPU called at Interrupt level</p>

MONITOR STOPCODE DEFINITION

ONCPUX	CPNSER	JOB	<p>X out of range in ONCPU<X></p> <p>Caused by a call to ONCPU, with a bad CPU number. Correct the CPU number supplied by the calling routine.</p>
OOC	VMSER	STOP	<p>Out of core blocks</p> <p>MAPBAK called SAVCTS to set up for running at UO level, but SAVCTS returned at the non-skip return. This could occur if there was insufficient low core to save the job's current UO-level context.</p>
OVA	SYSINI	STOP	<p>Out of Virtual Address Space</p> <p>ONCMAP selects the physical and virtual address for space in the high segment. This stopcode occurs when the number of virtual pages is greater than 256K.</p> <p style="padding-left: 40px;">Data Items: R1 = virtual page number</p>
PAO	COMCON	STOP	<p>Page Already Out</p> <p>PAGRE creates a directory page. This stopcode occurs when an attempt is made to page out a page that is already out. The page is being paged out because the job's physical limit has been exceeded.</p> <p style="padding-left: 40px;">Data Items: J = job number</p>
PAW	VMSER	DEBUG	<p>Page Already Write-enabled</p>
PBO	NETSER	STOP	<p>PCB Buffer Overflow</p> <p>An ANF network Front End Kontroller interrupt service routine called NETSER (FEKINT) to process an "input done" interrupt. The network Protocol Control Block returned by the FEK contained a data byte count (PCBCTR) that was greater than the allocated length of the PCB data buffer (PCBALN), indicating that the FEK wrote past the end of the data buffer.</p> <p style="padding-left: 40px;">Data Items: J = address of FEK U = address of PCB</p>
PCN	IPCSER	DEBUG	<p>Packet Count Negative</p> <p>UIPCFR is used on an IPC receive monitor call. This stopcode occurs when the count of unreceived IPCF packets goes negative.</p> <p style="padding-left: 40px;">Data Items: P1 = PID</p>

MONITOR STOPCODE DEFINITION

PDA	FILIO	DEBUG	Pointers With Different Addresses
			DD2MN copies pointers from the DDB to the monitor buffer during monitor mode I/O. This stopcode occurs when the RIB pointers and those now in the monitor buffer differ.
		Data Items:	T3 = XORed RIB and monitor buffer pointers T4 = cluster pointer
PDLOVF	ERRCON	JOB	Exec PDL OverFlow
PEZ	CORE1	STOP	PAGPTR Equals Zero
			GTPAGS adds to or takes pages from the free-core list. This stopcode occurs when the location PAGPTR, which points to the first free page, is zero.
PFC	VMSEr	STOP	Page on Free Core List
			SETHMT prepared for high-segment swap. This stopcode occurs when the monitor finds a page that is in the free-core list while scanning pages allocated to a segment.
		Data Items:	T1 = first disk address T2 = first page number T3 = number of pages
PFHJOB	MONPFH	JOB	Wrong JOB owns device
PFHUUO	MONPFH	JOB	PAGE. UUO failed
		Data items:	T1 = error code Arglst stored at .JDAT+JOBUAL
PFHZER	MONPFH	JOB	Page fault on Page ZERo
PFL	VMSEr	STOP	Piece on Free List
			GVEFWS returns words acquired by GTFWDC or GTFWDU. This stopcode occurs when an attempt is made to return a chunk of funny space that is already on the free list.
PFN	APRSER	CPU	Page Fault in Null job
			A page fault occurred while the null job was running.
PFNOIO	MONPFH	JOB	PFH.has No IO to do (I'm stuck!)

MONITOR STOPCODE DEFINITION

PFR	VM SER	DEBUG	Piece out of Free Range GVFWDS returns words acquired by GTFWDC or GTFWDU. This stopcode occurs when GVFWD S is called with an address that is not in funny space.
PGL	COMCON	STOP	Pages Got Lost PAGFRE creates a directory page. This stopcode occurs when the page cannot be paged out.
PGTPAR	APR SER	CPU	PaGe Table PARity
PIE	ERRCON	CPU	Priority Interrupt Error This stopcode occurs when a device interrupts to the wrong location. A jump occurred to an even address between 42 and 66 (octal).
PIF	VM SER	DEBUG	Page Is Free This stopcode occurs when the monitor finds a page that is in the free-core list while it is scanning pages allocated to a segment.
PIN	VM SER	STOP	Page IN Working set WSBIT gets bit and index for WSBTB and AABTAB. This stopcode occurs when the monitor finds a page in the working set that has been verified as not in the working set.
PIW	VM SER	DEBUG	Page Isn't in Working set PAGOMT sets up MEMTAB for paging out. This stopcode occurs when the monitor decides that a page must be in the working set, but it is not.
PLP	FILIO	DEBUG	Past Last Pointer USETO1 is used to do a USETO when the requested block is higher than the highest allocated block. This stopcode occurs when the SCNPTR routine, which scans pointers, cannot find a block that should be in the file.
Data Items:			P1 = top block to allocate P2 = first block to allocate DEVLPC(F) = RIB pointer

MONITOR STOPCODE DEFINITION

PMU	CORE1	STOP	<p>PAGTAB Messed Up</p> <p>This stopcode occurs when a zero is encountered as the link to the next page in the segment while setting up the user's page-map page to reflect the location of the pages in physical memory.</p> <p>Data Items: T1 = byte pointer to the map T2 = page attributes T4 = number of pages -1 left in this segment</p>
PMW	VM SER	DEBUG	<p>Page Map Wrong</p>
PNA	DATMAN	STOP	<p>PUTWRD Not Available</p> <p>PUTWRD was called at clock level but the requested word was not available.</p>
PNE	FILIO	DEBUG	<p>Pointers Not Equal</p> <p>PTRTST reads the pointers into core, compares the old pointers in the RIB with the new pointers in the DDB, and rewrites the RIB if they differ. This stopcode occurs when an error is found in the cluster pointer after the pointers in the RIB have been updated.</p> <p>Data Items: (T1) = pointer in the monitor buffer T3 = XORED RIB and monitor buffer pointers T4 = cluster pointer</p>
PNP	LOKCON	STOP	<p>Page Not Present</p> <p>PAGMOV finds the target page on the free-core list or within the current segment and exchanges it with the source page. This stopcode occurs when the source page cannot be found in the current segment.</p> <p>Data Items: T2 = successor to current page</p>
PNW	VM SER	DEBUG	<p>Page Not in Working set</p> <p>DLTMPG returns a funny page to the free-core list.</p>
POR	SEGCON	STOP	<p>Process Out of Range</p> <p>COMIT is used to right half of J with JBTS GN(T1). This stopcode occurs when the job number is out of range.</p> <p>Data Items: J = job number</p>

MONITOR STOPCODE DEFINITION

PQW	VMSER	DEBUG	Paging Queue Wrong
			A page that was supposed to be on one of the in-core queues cannot be found, because either the count for the queue indicates there are no pages in the queue, and/or the first page pointer for the queue lists zero for the first page.
		Data Items:	T3 points to queue header first word of header contains count, and second word contains first page on queue T2 contains page of page that should be on queue.
PRF	APRSER	CPU	Page Refill Failure
			This stopcode occurs when a page-fail code of 22 is returned by the pager.
		Data Items:	T1 = page-fail code .CPTPI = PI state .CPTCX = trap context
PSC	LOKCON	DEBUG	Page Should be in Core
			This stopcode occurs on a LOCK request for SPECIFIED PAGES. PSC occurs if pages to be LOCKed in core are not tagged as being in the working set.
PSF	CORE1	STOP	Page in Segment Free
			This stopcode occurs when a page is found that is marked in PAGTAB as being free while scanning a job's pages looking for page n or the last page.
		Data Items:	T1 = current page within this segment T2 = number of pages left to scan T3 = PAGTAB entry for next page in segment, that is, PAGTAB(T1)
PTT	CORE1	DEBUG	Past Top of Table
			SETZRS sets zeros in a table. This stopcode occurs when the SETZRS routine attempts to zero more bits than exist.
		Data Items:	(T2) = top of table T4 = final address to clear bits
PUF	SEGCON	JOB	PATH. UUO Failed
			PTHFIL looks up a file and returns the path for it. This stopcode occurs when the PATH. monitor call fails.

MONITOR STOPCODE DEFINITION

RAXCRM	RAXKON	EVENT	Command Reference number Missing
RAXCSF	RAXKON	DEBUG	Connect to MSCP Server Failed
RAXKIM	RAXKON	DEBUG	Kontroller Id Mismatch
RAXKN5	RAXKON	EVENT	Kontroller Not in 512-byte sector mode
RAXRAX	RAXKON	STOP	RAXKON is miserable
RAXUGA	RAXKON	DEBUG	UDB Gone Away
RAXUN5	RAXKON	EVENT	Unit Not in 512-byte sector mode
RAXUOF	RAXKON	DEBUG	Unit Online Failed
RAXUWA	RAXKON	DEBUG	UDB Went Away?!
RAXXIF	RAXKON	STOP	RAXKON Isn't Fancy
RBQ	SCHED1	STOP	Requeueing to Beginning of Queue QFIX is used in the requeueing of jobs. This stopcode occurs when an attempt is made to requeue a job to the beginning of the same queue.
RCC	SCNSER	DEBUG	Range-Checked Chunk This stopcode is called by several places, each doing a range check on a character address. This stopcode occurs when the character address is not within the TTY buffer pool. Data Items: T2 = character address character address (in T2) that was not in the TTY buffer pool. This can be caused by attempting TTY output without first setting up U to point to an LDB.
RCD	SCNSER	DEBUG	Random Chunk Discrepancy This stopcode is called from a number of places in SCNSER where it is noted that chunk pointers and counts are inconsistent.

MONITOR STOPCODE DEFINITION

RCS3XF LLMOP INF LLMOP Transmit Failed
LLMOP was unable to transmit a forward data message.
Data items: T1 contains the error code returned from the DLL
T2 contains the channel on which the failure occurred

RCSIFC LLMOP CHK RCSCBR called with Invalid Function Code
The LLMOP Remote Console Protocol Server Call Back Routine was called by the Data Link Layer with an invalid callback function code. This is a software bug. Call your DIGITAL Software Specialist.

RCSPIS LLMOP INF Ethernet Periodic Identify-Self
This is a temporary debugging BUGINF. It is here to provide an indication that the periodic Identify-Self transmission is being performed.

RDN TAPUO DEBUG Regular DDB Not Found
SETODN sets the density in the other DDB. This stopcode occurs when there is no regular DDB.
Data Items: R3 = UDB

RDP FSXKON DEBUG RS04 Doesn't Position
FSXPOS is a stopcode-only routine that is used when the FILIO module tries to position an RS04.

RDS SEGCON STOP REMAP Didn't Skip
GETFIN remaps the save file after it has been read in its entirety. This stopcode occurs when the remap fails because the arguments are wrong, pages do not all exist in the page specified, or moving the pages to the virtual address specified would cause the high and low segments to overlap.

RDXDAT NETDEV STOP NTDSIB failed in R.DATA
After calling NTDIBA to guarantee a user input buffer is available, the ANF network remote data entry device service routine was unable to set up an input buffer to receive a network message.
Data Items: F = address of DDB
U = address of PCB

MONITOR STOPCODE DEFINITION

REFMBM	REFSTR	DEBUG	Monitor Buffer Missing
REH	ERRCON	HALT	<p>Recursion in Error Handler</p> <p>DIE recovers/reloads after an internal system error. This stopcode occurs when another stopcode occurs before the previous one is done.</p>
RFU	TAPSER	STOP	<p>Recovery Fouled Up</p> <p>ERPINT handles interrupts while error recovery is in progress. This stopcode occurs when the function code for dispatching is greater than 6.</p> <p style="margin-left: 40px;">Data Items: T2 = function code T1 = pointer</p>
RHN	FILIO	DEBUG	<p>Reread Home block-count Negative</p> <p>SETMDL sets the file to idle when monitor I/O is done. This stopcode occurs when the flag DEVRHB(F) indicates that the HOME blocks are being reread, but the flag that tells the number of units that are rereading HOME blocks (HOMFG) is negative.</p>
RID	ERRCON	HALT	Recursion In DIE
RIE	XTCSER	DEBUG	<p>Remote Interrupt Error</p> <p>This stopcode occurs if there is any error bits are lit on an interrupt from a remote system on the DA28.</p>
RIF	DPXKON	DEBUG	<p>RP10 Isn't Fancy</p> <p>These are stopcode-only routines. This stopcode occurs when the monitor attempts an RP04-only function, such as an UNLOAD, on an RP10-controlled device.</p>
RJO	CLOCK1	DEBUG	<p>Requeue Job 0</p> <p>REQUE requeues a job to run. This stopcode occurs when an attempt is made to requeue job 0 (the null job) or a job number greater than JOBMAX, to run.</p>

MONITOR STOPCODE DEFINITION

RJZ	SCHED1	STOP	<p>Requeue Job Zero</p> <p>QXFER is used in the requeuing of a job. This stopcode occurs when a call is made to this routine with a job number less than or equal to zero or greater than JOBMAX.</p> <p>Data Items: J = job number</p>
RLD	UUOCON	STOP	<p>ReLoaD monitor</p> <p>This is a result of the RECON. UUO function .RCRLD, which is callable by a privileged user or the CONFIG command SHUTDOWN. Refer to the <u>TOPS-10 Operator's Guide</u> for more information.</p>
RNP	VMSER	DEBUG	<p>Returning Non-existent Page</p> <p>DNZSPG returns non-zero section pages to free core. This stopcode occurs if an attempt is made to return a page that does not exist.</p>
ROU	ONCMOD	STOP	<p>Ran Out of Units</p> <p>NXTSAT reads all the SATs on a unit and computes the number of free clusters left in each SAT. This stopcode occurs when the SAT pointers indicate that there is another unit, when in fact there is no other unit.</p> <p>Data Items: P4 = number of units remaining</p>
RPM	ONCMOD	DEBUG	<p>Retrieval Pointer Mismatch</p> <p>FILMAN finds and sets up all structures on a system. This stopcode occurs when the unit-change pointer in the file SAT.SYS did not point to the next unit in the file structure.</p> <p>Data Items: T2 = expected unit-change pointer T3 = logical unit number expected</p>
RQD	SCNSER	DEBUG	<p>RECINT Queue Discrepancy</p> <p>This stopcode occurs if we just emptied the RECINT queue but the taker and putter pointers do not match.</p>
RQF	SCNSER	DEBUG	<p>RECINT Queue Full</p> <p>This stopcode occurs the RECINT character queue wraps around.</p>
RS04IF	FSXKON	DEBUG	<p>RS04 Isn't Fancy</p>

MONITOR STOPCODE DEFINITION

RSJ	CLOCK1	DEBUG	<p>Requeue Same Job</p> <p>REQUE requeues a job to run. This stopcode occurs when an attempt is made to queue the same job again.</p> <p>Data Items: J = job number</p>
RTM	NETDEV	STOP	<p>Requested Too Much</p> <p>A call to TRQPCB to get an ANF network Protocol Control Block (for a network terminal) requested a message size which was either negative or exceeded the maximum configured terminal PCB buffer size.</p> <p>Data Items: U = address of LDB T1 = requested PCB size (in bytes)</p>
RTTIME	COMMON	STOP	<p>Illegal Real-Time Memory reference in Exec</p>
RWD	FILIO	DEBUG	<p>Returning Wrong unit's DA</p> <p>DWNDA gives up a disk allocation request. This stopcode occurs when the unit DA being dequeued is not correct for this job.</p> <p>Data Items: PJOBN = job number</p>
RWS	VMSER	DEBUG	<p>Returning Space to Wrong Section</p> <p>GVFWDS returns funny space pages. This stopcode occurs if the monitor attempts to return funny space from a section from which it cannot be allocated.</p>
RX2	RX2SER	STOP	<p>RX2SER fouled up</p>
RXX	RX2SER	DEBUG	<p>Unimplemented error recovery</p>
SAC	ERRCON	DEBUG	<p>Strange APR Condition</p> <p>This stopcode occurs when an APR interrupt occurs with no known error bits set.</p> <p>Data Items: S = APR error condition</p>
SAH	DSXKON	DEBUG	<p>SA10 Hung</p>

MONITOR STOPCODE DEFINITION

SAU	CPNSER	DEBUG	Scheduler Already Unlocked
			ULKSCD unlocks the scheduler interlock. This stopcode occurs when the interlock is already free.
		Data Items:	SCKLOK = Interlock
SAXBAI	SAXSER	DEBUG	SA10 Base Address In use
SAXISR	SAXSER	INFO	Invalid Status Request
SAXNSI	SAXSER	DEBUG	No Status on Interrupt
SBE	APRSER	CPU	SBUS Error
SBT	FILUO	DEBUG	Shouldn't Be Truncating
			CLSRIB closes a file. This stopcode occurs when an attempt is made to truncate unwritten blocks, but the highest block number in the file is too small.
		Data Items:	P2 = current block of RIB P3 = DEVREL
SBW	VMSEr	DEBUG	SWPLST Bits Wrong
			This stopcode occurs when an entry in SWPLST shows both that I/O is in progress and that I/O is complete.
SBZ	VMSEr	STOP	Swap Block Zero
			This stopcode occurs if, in picking the next swap list entry, we find that it specifies an invalid disk address.
SCABMT	SCASER	CHK	Bad Tessage Type from remote node
SCACCI	SCASER	HLT	Cannot Complete Initialization
SCACFO	SCASER	HLT	SC.CON received Failure from SC.OUT
SCACLB	SCASER	HLT	Incoming connection request CLosed on VC
SCACRB	SCASER	CHK	Can't Reclaim Buffers
SCACSC	SCASER	CHK	Can't Send Credit request

MONITOR STOPCODE DEFINITION

SCACVC	SCASER	INF	Virtual Circuit Closure requested
SCADLL	SCASER	HLT	Don't care Listener Linked to CB
SCAEED	SCASER	CHK	Block state is zero when trying to send connection management request
SCAFN2	SCASER	HLT	Can't complete deferred call to SC.DIS
SCAFN3	SCASER	HLT	Can't complete deferred call to SC.DRQ
SCAFOO	SCASER	HLT	Oh, FOO
			A general BUG. for errors which "should never happen."
SCAILC	SCASER	HLT	Illegal Lock Count in connection block
SCALCC	SCASER	HLT	Connection block Lock Count has Changed
SCALFO	SCASER	HLT	SC.LIS received Failure from SC.OUT
SCANBN	SCASER	HLT	No Buffer for Notification table
SCANLF	SCASER	CHK	Notice table Full
SCANMB	SCASER	CHK	CanNot return SCS control Message Buffer
SCANOC	SCASER	CHK	Received packet and Connection block doesn't exist
SCANP1	SCASER	HLT	.CBNPO has gone Negative
SCANP2	SCASER	HLT	.CBNPO has gone Negative
SCANP3	SCASER	HLT	.CBNPO has gone Negative
SCANPC	SCASER	HLT	No Page for CID table
SCANSC	SCASER	CHK	Negative path Count
SCAOF2	SCASER	CHK	Duplicate OFFline for a node
SCAPER	SCASER	CHK	Protocol ERror

MONITOR STOPCODE DEFINITION

SCAQQQ	SCASER	CHK	Unexpected credit field in creditrequest
SCARBS	SCASER	CHK	Reap Bit is Set when block state is non-zero
SCASBN	SCASER	CHK	Block State already Non-zero
SCASCQ	SCASER	HLT	SCA Credit Queue failed
SCASSS	SCASER	CHK	Connect block already linked
SCATMO	SCASER	INF	SCA TiMed Out remote node
SCAUXR	SCASER	CHK	UnexPected Response
SCB	XTCSER	DEBUG	Spurious CONI Bit This stopcode occurs if certain random error bits are lit on the CONI status read on an XTC interrupt.
SCNIU	COMMON	CPU	SCNSER Interlock Unowned
SCNRIA	COMMON	STOP	Recursive Interlock Attempt
SCR	DEBUG	SEGCON	Segment Couldn't be Read INPSEG is called to read in a high segment. This stopcode occurs if INPSEG returns non-skip.
SCSCIS	SCSUUO	DEBUG	Can't Initialize SCS. UUO interface
SCSNOR	SCSUUO	STOP	Notification code Out of Range
SCSBEB	SCSUUO	DEBUG	Bad Event Block
SCSUET	SCSUUO	DEBUG	Unknown Event Type
SDE	FILIO	DEBUG	SAT Doesn't Exist GIVBLK returns disk blocks. This stopcode occurs if GIVBLK cannot find the SAT in which the blocks are supposed to exist.

MONITOR STOPCODE DEFINITION

SDS	UUOCON	DEBUG	SWPADR Didn't Skip
			SWPADR converts a swap space address to a unit/disk address. This stopcode occurs when the JOBPEK. UUO determines it needs to read/write the swapping space and calls SWPADR to convert a swap address to a unit/disk address. SWPADR does not have a non-skip return.
SER	FILUUO	JOB	SETDDO Error Return
			FAKDDB sets a DDB. This stopcode occurs when subroutine SETDDO gives an error return indicating no core is available to build a device data block, although space had been found just before the call.
			Data Items: T1 = address in memory found previous to call
SFU	FILIO	DEBUG	Swapper Fouled Up
			SWAPIO puts a swap request into the queue. This stopcode occurs when this routine is called with no request.
			Data Items: SQREQ = 0, should have been the request
SHU	SCHED1	DEBUG	Swapper Hung Up
			NOFORC times out devices that are active to a job waiting to be swapped out. This stopcode occurs when the device-hung timer times out while the job was in FORCEF.
SIE	VM SER	DEBUG	SWPLST Is Empty
			DLTSLE was called to delete a SWPLST entry, but there were no entries in SWPLST.
SIN	VM SER	DEBUG	SWPCNT Is Negative
			This stopcode occurs when the count of the number of outstanding swapping requests becomes negative while an entry from SWPLST is being deleted.
SIU	SCASER	CPU	SCA Interlock Unowned
SLF	VM SER	DEBUG	SWPLST Full
			This stopcode occurs when there is no room for an entry in the swap list table.

MONITOR STOPCODE DEFINITION

SLM	FILUO	DEBUG	Search List Missing
			FNDFRA is used when the PPB and/or the UFB are deleted. This stopcode occurs when the SETSRC routine cannot set up a search list, even though it seemed possible when the call started.
SLO	FILEND	JOB	Search List Overflow
			SLXSLO is a stopcode-only routine. Examine the stack for the location of the error.
SLZ	VMSE	DEBUG	SLECNT Is Zero
			This stopcode occurs when the subroutine to find an entry in the SWPLST table is called when there are no entries in the table.
SMP	LOKCON	STOP	Shouldn't Move Page
SMP	COMMON	STOP	Shouldn't Move Page
SMU	SCHED1	DEBUG	SWPCNT Messed Up
			SWAP is used to swap jobs.
			Data Items: J = job number SWPCNT = count of completed swapping operations
SNASHR	SEGCON	STOP	SNA/JBTSHR discrepancy
SNF	LOKCON	STOP	Segment Not Found
			LOCK0 locks a segment in core. This stopcode occurs when the monitor cannot find a segment that contains a certain page.
			Data Items: T3 = absolute page address being looked for
SNI	SWPSER	DEBUG	Swapping Not In Progress
			SWPINT is used when paging or swapping I/O is done for a monitor that includes virtual memory. This stopcode occurs when the swap-in progress count goes negative.
			Data Items: SPRCNT = Swap-in progress count

MONITOR STOPCODE DEFINITION

SNS	NETDEV	STOP	NTRPCB Not Set up
			TWRPCB writes back the count field and updates the pointer in the PCB. It also removes garbage from the stack.
		Data Items:	T1 = minimum number of bytes
SOD	SCHED1	STOP	Space On Disk
			SWAPI swaps in either a job or high segment. This stopcode occurs when the core-allocation routine (CORGET) assigns space on the disk, but the assignment is illegal.
		Data Items:	J = job number
SOR	ERRCON	STOP	Segment Out of Range
			ERRPNT prints common error messages. This stopcode occurs when the job or segment number is too large.
		Data Items:	J = job number
SPM	FILUOO	JOB	Second Pointer Missing
			UFDNXT initializes the next block for a directory. This stopcode occurs when the pointer to the second RIB is missing from the first RIB.
		Data Items:	T3 = Supposed location of second RIB pointer
SRE	ONCMOD	DEBUG	SAT Read Error
			NXTSAT reads all SATs on a unit and computes the number of free clusters left in each SAT. This stopcode occurs when a read error occurs while reading the SAT.
SRO	SWPSER	STOP	Space Ran Out
			NXUN is used when we have filled the current unit and we need more swapping space. This stopcode occurs when there are no more units for swapping.
SSCNEG	SEGCON	DEBUG	Segment Share Count NEGative

MONITOR STOPCODE DEFINITION

SSD	SWPSER	STOP	Swap Space Disappeared
			FOUND is used when contiguous space has been found on a unit for swapping. This stopcode occurs when an attempt is made to allocate that space, which for some reason is no longer available.
			Data Items: U = address
SSO	LOKCON	STOP	Segment Swapped Out
			LOCK0 locks a segment in core. This stopcode occurs when a high segment that is neither dormant nor idle has no low segment in core.
SWN	SWPSER	DEBUG	SQREQ Went Negative
			SWPINT is used when paging or swapping I/O is done. This stopcode occurs when the count of paging or swapping requests goes negative.
SYVCTS	SYSINI	DEBUG	SYmbol VeCtor Too Short
SWZ	SEGCON	DEBUG	Segment Wait count Zero
TC0	XTCSER	DEBUG	XTCSER Stopcode Zero
			XTCSER has found the controller free and the unit unlocked, but there are requests in the queue waiting to be processed.
TC1	XTCSER	STOP	XTCSER Stopcode One
			XTCSER should have already set a "Waiting for Input" message, but has not.
TC2	XTCSER	DEBUG	XTCSER Stopcode Two
			XTCSER expected the DAS28 to be idle, but it was not.
TC3	XTCSER	DEBUG	XTCSER Stopcode Three
			The number of pseudo active tasks in the XTC UDB went negative.
TC4	XTCSER	DEBUG	XTCSER Stopcode Four
			The number of pseudo active tasks in the XCT KDB went negative.

MONITOR STOPCODE DEFINITION

TC5	XTCSER	DEBUG	XTCSER Stopcode Five
			The number of pseudo active tasks in the XTC UDB went negative.
TC6	XTCSER	DEBUG	XTCSER Stopcode Six
			The number of pseudo active tasks in the XTC KDB went negative.
TC7	XTCSER	STOP	XTCSER Stopcode Seven
			This stopcode occurs when XTCSER expected to have the controller interlocked but found it did not.
TCI	FILUO	DEBUG	Truncation Check Inconsistent
			RENDEL deallocates or truncates on a RENAME. This stopcode occurs when an attempt is made to truncate too many blocks and a check on the same had already succeeded.
			Data Items: P1 = AOBJN pointer; P3 = number of blocks
TIC	SCNSER	DEBUG	LDBTIC wrong
TIO	TAPSER	STOP	Tape I/O to wrong CPU
TIU	TAPSER	CPU	Tape Interlock Unowned
TMDELE	SCNSER	INFO	Too Many DEletions from Echo
TMDELI	SCNSER	INFO	Too many s in Input
TME	SYSINI	STOP	Too Many pages reserved for EVM
TMP	FILIO	DEBUG	Too Many Pointers
			PTRWRT copies RIB pointers into a monitor buffer and writes it. This stopcode occurs when there are more retrieval pointers than can fit in a RIB. The counter DEVRSU should prevent this from happening.
			Data Items: T2 = remaining pointers (IOWD)

MONITOR STOPCODE DEFINITION

TMR REFSTR STOP Too Many Retrieval pointers

SATRBS stores retrieval pointers in the SAT.SYS read-in block. This stopcode occurs when the SAT byte pointer is messed up.

Data Items: T1 = SAT byte pointer

TMU ONCMOD STOP Too Many Units

NXTSAT reads all SATs on a unit and computes the number of free clusters left in each SAT. This stopcode occurs when there are pointers to more units after the last has been retrieved.

Data Items: U = pointer to more units

TSKAND NETDEV STOP Already got an NPD

On a user-program "LOOKUP" to an ANF network TSK device, the TSK Device Data Block already had one (or both) of the Network Process Descriptor blocks assigned. The NPDs contain the local and remote LOOKUP and/or ENTER "names", and as such should not yet be set for a TSK DDB entering passive connect wait.

Data Items: F = address of DDB

TSKIOS NETDEV STOP IOSCON is on

When attempting to put an ANF network TSK device into either "active" or "passive" connect wait, the TSK device was found not to be in the "idle" state. Either the Link Address Table state was not "idle" or the TSK Device Data Block IOSCON (device is connected) flag was set.

Data Items: F = address of DDB
S = DEVIOS word
T1 = LAT state

TSKLE2 NETDEV STOP NPD already assigned in LOOKUP/ENTER

On a user-program "LOOKUP" or "ENTER" to an ANF network TSK device, the TSK device was found to have one or both Network Process Descriptor blocks already assigned.

Data Items: F = address of DDB

TSKNIC NETDEV STOP Not in "CI" state

On a user-program "CLOSE" to an ANF network TSK device, the TSK device Link Address Table state is inconsistent with TSK device.

Data Items: F = address of DDB

MONITOR STOPCODE DEFINITION

TSKNID NETDEV STOP Not In Disconnect confirm

While waiting for an ANF network TSK device Disconnect request to be honored, the TSK device Link Address Table state transited into an illegal state (neither waiting for Disconnect Confirm, nor Disconnected).

Data Items: F = address of DDB
T1 = LAT state

TSKNIP NETDEV STOP Not In Passive state

On a user-program "ENTER" to an ANF network TSK device, the TSK device state was illegal (neither "idle", nor "OK", nor in passive connect wait).

Data Items: F = address of DDB
T1 = LAT state

TSKNPD NETDEV STOP NPD already assigned

On a user-program-generated request to implicitly (with a LOOKUP or ENTER monitor call) or explicitly (with a TSK. monitor call) set the Network Process Descriptor information for an ANF network TSK device, the TSK Device Data Block was found to have one or both of the NPDs already assigned.

Data Items: F = address of DDB

TSKSCC NETDEV STOP Send Connect Confirm failed

After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT failed to get a free PCB to acknowledge an incoming TSK device Connect Initiate request.

Data Items: F = address of DDB

TSKSIB NETDEV STOP No Input Buffer

After calling NTDIBA to guarantee a user input buffer is available, the ANF network TSK device service routine was unable to set up an input buffer to receive a network message.

Data Items: F = address of DDB
U = address of PCB

MONITOR STOPCODE DEFINITION

TSKSOR	NETDEV	STOP	<p>LAT State is Out of Range</p> <p>The ANF network Link Address Table state for a TSK device was found to be out of the known range of LAT states. The ANF Link Address Table is very probably corrupted.</p> <p>Data Items: F = address of DDB T1 = LAT state</p>
UAF	APRSER	STOP	<p>UNIBUS Addressing Failure</p> <p>SEILM processes page-failure traps. This stopcode occurs when what appears to be a page fault turns out to be a UNIBUS addressing failure. (KS10 only)</p>
UCR	MSCCOM	DEBUG	<p>Unexpected Connect Response</p>
UDBAIZ	FILIO	DEBUG	<p>UDB Address Is Zero</p> <p>DEVUNI contains zero when a USETI is done.</p>
UDE	FILIO	DEBUG	<p>Unit Doesn't Exist</p> <p>RIBCUR reads the current RIB. This stopcode occurs when a requested unit is not in any file structure.</p> <p>Data Items: DEYRBU = current RIB logical unit number pointer</p>
UDM	FILUOO	JOB	<p>UFD Data is Missing</p> <p>UFDALB allocates a block for a UFD. This stopcode occurs when the core tables show that the UFD is longer than it actually is.</p> <p>Data Items: T3 = supposed number of blocks of this UFD.</p>
UFI	FILUOO	STOP	<p>Unit Free-Count Inconsistent</p> <p>CLSOU5 is used during a CLOSE after finding a unit with space on it. This stopcode occurs when an attempt is made to allocate the space, but no space is available.</p> <p>Data Items: T2 = Number of blocks needed</p>
UID	D8SINT	DEBUG	<p>Unexpected Input Done</p> <p>T10DON handles To-10 Done interrupts from the -11. This stopcode occurs when no input is expected.</p> <p>Data Items: T1 = CPU number</p>

MONITOR STOPCODE DEFINITION

UIL	ERRCON	STOP	UUO at Interrupt Level EMUERR is called when an illegal monitor call occurs at exec level. This stopcode occurs when the monitor call occurs at interrupt level.
UIO2BB	USRONC	STOP	User I/O Too Big for Buffer
UIOCCS	USRONC	STOP	Can't Create Scratch file
UIONSU	USRONC	STOP	No Such Unit
UIP	XTCSER	DEBUG	Not a Unique Interrupt XTCSER decided to call routine DDBINT (for DDB doing data I/O) instead of UNIINT, but XKBIUN (pointer to UDB requesting interrupt) was non-zero, implying XTCSER should have called UNIINT. Only one of DDBINT or UNIINT should be called.
ULE	LP2SER	JOB	Unexpected LP20 Error LPTERR handles VFU errors for LP20 controllers. Data Items: F = DDB T1 = function
ULP	APRSER	DEBUG	UBA Lost its PI assignment KSSEC performs once-a-second tasks for the KS10.
UNA	APRSER	STOP	UPT Not Addressable
UNAAOR	UNASER	DEBUG	Address Out of Range
UNABOW	UNASER	DEBUG	Buffer Ownership Wrong
UNACQF	UNASER	DEBUG	Command Queue Full or Fouled
UNANIC	UNASER	DEBUG	No Interrupt Condition
UNAXQF	UNASER	DEBUG	Transmit (Xmit) Queue Fouled up

MONITOR STOPCODE DEFINITION

UNF	FILUOO	DEBUG	UFB Not Found	<p>NAMNW updates RIBNAM, RIBEXT, and RIBPPN when there is a CLOSE for RENAME. This stopcode occurs when a RENAME is done across UFDs and the UFB is not found.</p> <p>Data Items: T1 = Structure number T2 = Start of UFB chain</p>
UNJ	COMMON	STOP	UUO from Null Job	<p>This stopcode occurs when the null job executes a monitor call other than the doorbell call.</p>
UNL	VMSER	STOP	UPMP Not Last	<p>This stopcode occurs when the UPMP is not the last page swapped out.</p>
UPC	FILUOO	JOB	Unit-Change Pointer Clobbered	<p>SETENC enters a file. This stopcode occurs when the pointer to a unit of a RIB is lost during RIB definition.</p> <p>Data Items: S = status bits T3 = location of the access table</p>
UPF	APRSER	STOP	Unexpected Page Fail	<p>This stopcode occurs when there is a page fail trap during a recovery attempt of an AR/ARX trap, which is not caused by a test reference.</p> <p>Data Items: .UPMP+.LMPFW = page fail code .UPMP+.LMPFP = page fail PC</p>
UPI	FILIO	DEBUG	Unit Pointer Illegal	<p>EXTRIB creates an extended RIB. This stopcode occurs when an attempt is made to create an extended RIB on a nonexistent unit.</p> <p>Data Items: T2 = change unit pointer (should have been a real unit pointer)</p>
USW	TAPSER	INFO	Unit Status Wrong	<p>TAPSIO is used when the UUO level wants to start I/O on a unit. This stopcode occurs when the unit status is not as expected. For example, the unit was started (possibly on another controller) when it should have been stopped.</p> <p>Data Items: RUBSTS(U) = unit number</p>

MONITOR STOPCODE DEFINITION

VTMECC NETDEV STOP Echo counts messed up

Preparatory to sending characters for an ANF VTM terminal to a remote MCR/host, the count of characters about to be sent was greater than the terminal's count of echoed characters available.

Data Items: U = address of LDB

VTMILS NETDEV STOP Illegal State

An ANF network Disconnect was received for a VTM terminal that was neither connected to, in connect wait, or in disconnect wait state for the node which sent the disconnect message.

Data Items: U = address of LDB
T2 = LAT state

VTMLAL NETDEV STOP LDB And LAT do not agree

On a call to VTMCLR to "clean up" and reinitialize an ANF network VTM terminal Line Data Block, the LDB's Link Address Table entry does not point back to the LDB.

Data Items: U = address of LDB
T1 = LAT address (from LDB)

VTMLAT NETDEV STOP LAT address not set up

On a call to VTMXCN to send a Connect Initiate message (or possibly a Connect Confirm message) to a remote ANF network node for a VTM terminal, the terminal Line Data Block had no Source Link Address.

Data Items: U = address of LDB
W = address of NDB

VTMLDB NETDEV STOP No LDB in VTMEHQ

VTMEHQ was called to "queue" an ANF network VTM terminal Line Data Block for VTMSCN processing, but U contained 0.

VTMNDA NETDEV STOP Node number Doesn't Agree

When processing an ANF network "node down" condition, NETVTM (VTMNWD) was called to deal with a VTM terminal connected to the no-longer-accessible node, but the terminal claimed to be connected to a different node.

Data Items: U = address of LDB
T1 = node number from LDB
P1 = node number that "went down"

MONITOR STOPCODE DEFINITION

VTMNDB NETDEV STOP No NDB for LDB's node.
In the once-a-jiffy processing loop in NETVTM, a connected ANF VTM terminal's Node Data Block could not be found (SRCNDB failed).
Data Items: U = address of LDB

VTMNLA NETDEV STOP No LAT Address for virtual terminal?
In the once-a-jiffy processing loop in NETVTM, a connected ANF VTM terminal's Source Link Address was zero.
Data Items: U = address of LDB

VTMNNN NETDEV STOP But VTMNWD should have caught this
In the once-a-jiffy processing loop in NETVTM, a connected ANF VTM terminal's Node Data Block could not be found (SRCNDB failed).
Data Items: U = address of LDB

VTMQED NETDEV STOP Line not queued though LRLQED is set
VTMDEQ was called to "dequeue" an ANF network VTM terminal, but the VTM terminal Line Data Block was not queued (even though the LDB LRLQED (VTM terminal is queued) flag is set for the terminal).
Data Items: U = address of LDB

VTMSDF NETDEV STOP Send Disconnect Failed
After a call to PCBECK returned "guaranteeing" the availability of an ANF network Protocol Control Block, a subsequent call to PCBEGT failed to get a free PCB to send a Disconnect Confirm message for an ANF VTM terminal.
Data Items: U = address of LDB

WAD VMSER DEBUG WSBTBL and AABTBL Discrepancy
This stopcode occurs when there is an access page fault for a page that should have the access allowed bit on in the page map.

MONITOR STOPCODE DEFINITION

	WCT	SYSINI	HALT	Wrong CPU Type
				This stopcode occurs when ONCE is running on a different type of processor than the monitor was built for.
				Use the correct monitor or rebuild the monitor.
	WEM	NETSER	STOP	Generic ANF network crash
				This is a catch-all stopcode for the ANF network service. Examine the stack for the location of the error.
	WFC	D6SINT	DEBUG	Bad Function Code to FEK
	WNGUCV	COMMON	HALT	WroNG UCode Version
	WNP	VM SER	STOP	Wrong Number of Pages
	WNS	D85INT	DEBUG	Window was Not Setup?
	WPT	APRSER	STOP	Wrong Parity Trap
				This stopcode occurs when there is a page fail while trying to recover from an AR/ARX trap, which occurred because of a test reference, but the page fail code is not 36.
				Data Items: T1 = page fail code .UPMP + .LMPFW = page fail code .UPMP + .LMPFP = page fail PC
	WRF	COMMON	CPU	Warm Restart Failed
				A condition such as a DEX has occurred and the monitor has attempted to warm restart, but cannot due to various conditions being in effect at the time of the original failure.
	WRJ	COMMON	JOB	Warm Restart got Job
				A condition such as a DEX has occurred and the monitor has attempted a warm restart. The condition occurred while some job other than the null job was running in user mode.
	WSM	FILIO	STOP	Wrong Size Moved
				Routine CSSETL is called to set the size of a BLT to/from the disk cache. This stopcode occurs if the size is greater than one block worth of data.

MONITOR STOPCODE DEFINITION

WTP	CLOCK1	JOB	Wrong Type of PDL
			WSCHED is entered at monitor call level when a job goes into I/O wait or sharable-device wait. This stopcode occurs when the address of the pushdown list is too low to be a monitor call pushdown list.
			Data Items: P = pushdown list
XPW	LOKCON	STOP	EXchanged Page Went away
			FIXMAP finds a page with which a page was exchanged and fix the map slot for that page. This stopcode occurs when the monitor cannot find the page that was exchanged.
XTH	SCHED1	DEBUG	XJOB Too High
			FNDXPN finds the expanding job. This stopcode occurs when the count of the number of jobs that must be swapped out and back in to satisfy a core expansion request is positive, but no expanding job is found.
ZPS	VMSE	STOP	Zero Page Swap
			A request for swapping specified 0 as the number of pages to transfer.

LIST OF DECnet-10 STOPCODES

5 LIST OF DECNET-10 STOPCODES

<u>Name</u>	<u>Module</u>	<u>Type</u>	<u>Message and Explanation</u>
COM911	D36COM	HLT	<p>The date is past 9 November 2021</p> <p>The two-byte Julian half-day field in an event message is limited to 9 November 2021. The routine above calculated the Julian half-day, and found that it overflowed. It is unlikely that the date itself really went past 2021. An AC was probably destroyed, or the routine to get the time from the monitor is returning invalid information.</p>
COMAFB	D36COM	CHK	<p>A Free Block pointer is bad</p> <p>There is a block on a free list, most likely just added to the list, whose address is not in the expected range. The offending pointer is in P1. A subroutine whose address is on the stack is probably returning a block to the wrong free list, or is returning an invalid pointer.</p>
COMATB	D36COM	CHK	<p>A-Block request Too Big</p> <p>For the time being, until we get a real memory-manager for non-message-block requests, we only support requests for memory up to the size of a VBL block, see DEFBLK macro.</p> <p>Wait for the real memory manager or make the size of VBLs bigger than they are now.</p>
COMBBP	D36COM	HLT	DNSBP called with OWGBP
COMBNN	D36COM	CHK	<p>Bad local Node Number</p> <p>The node number that was set with the NODE command in the CONFIG file was higher than the DECNET MAXIMUM-ADDRESS value set in the same file. As a consequence DECnet cannot initialize.</p> <p>Change the startup file to be consistent.</p>
COMCAW	D36COM	CHK	Core Allocation Wrong
COMCHA	D36COM	CHK	<p>Number of available FB blocks too large</p> <p>When checking the CH begstr for a type of block, the code determined that more blocks were available than there were originally. DNCHFB is supposed to defend against this. CHNUM was probably trashed.</p>

LIST OF DECnet-10 STOPCODES

COMCHB	D36COM	CHK	<p>CH pointer off by a few</p> <p>A pointer internal to the core management routines is off by a few words. You have probably trashed an AC by adding to it or XORing some bits.</p>
COMCHO	D36COM	CHK	<p>CH pointer Out of range</p> <p>In the core block checking routines, the internal pointer to the CH begstr applying to this type of block is bad. Your executable code was probably trashed.</p>
COMCID	D36COM	CHK	<p>Couldn't Initialize DECNET</p> <p>SCTINI found some reason to object to the DECnet environment. See SCTINI for reasons why it takes a non-skip return.</p>
COMCWT	D36COM	CHK	<p>Check Word Trashed</p>
COMDNP	D36COM	CHK	<p>DNGPOS called with bad MS</p> <p>In range checking the ac MS, its contents were outside the range of addresses used for the MS block. Trace back to the caller and find out why it has a junk pointer.</p>
COMEBT	D36COM	CHK	<p>End of memory Block Trashed</p>
COMFBA	D36COM	CHK	<p>FB Available count is wrong</p> <p>DNCHFB walked through a free list and found a different number of blocks on the list than the header indicated. A forward pointer was probably destroyed in a previously returned block.</p>
COMFBB	D36COM	CHK	<p>FB in database is off by a few</p> <p>DNCHFB found a block on a free list, most likely just returned, whose address is not on a block boundary for blocks on this free list. The offending pointer is in P1. A caller on the stack is probably returning a junk pointer, either a real pointer to a block that has been incremented or decremented, or a completely junk pointer.</p>
COMFBF	D36COM	CHK	<p>FB is already on Free list</p> <p>The block that P1 points to is already on the free list and is being returned again. A caller on the stack is returning a block that is already free.</p>

LIST OF DECnet-10 STOPCODES

COMFBO	D36COM	HLT	<p>FB pointer is Out of range</p> <p>When checking a free block pointer, the code found that the pointer is not pointing to the free core allocated for this type of block. Identify the routine that supplied this pointer.</p>
COMFBT	D36COM	HLT	<p>FB pointer is off by a few</p> <p>A free block pointer is off by a few words. The user of this pointer probably added a constant, and forgot to restore it when returning the block. Trace the user of this pointer, and make sure the pointer is valid when given to the memory manager.</p>
COMFWZ	D36COM	CHK	<p>Tried to Free Words at Zero</p> <p>DNFWDS was called with a pointer of zero.</p>
COMIEL	D36COM	CHK	<p>Illegal End of List pointer</p> <p>CHAVL, the available count, said there was at least one block on the free list, but the first pointer was zero. A forward pointer was probably destroyed in a previously returned block.</p>
COMMMI	D36COM	CHK	<p>Memory Manager must be Initialized</p> <p>The field CHBOT, which indicates where a free core pool starts, is zero. This field gets set when the core manager is initialized. If DNINIM has already been called, check to make sure it is initializing all CH blocks.</p>
COMMMS	D36COM	HLT	<p>Bad pointer passed to Memory Manager</p> <p>When DNGWDS gives out a block of core, it leaves a check word immediately before the first word of core given to the user. This word contains the length of the block, and a "check" quantity to verify that this block contains what is expected. This bug means that the check word has been trashed, or the pointer that was passed to the memory manager is bad.</p>
COMMPR	D36COM	HLT	<p>Message Pointer check</p> <p>DNFMSG caller tried to return a piece of memory that is not in the range of message blocks. See stack for caller and find why it is trying to return a bad message block.</p>

LIST OF DECnet-10 STOPCODES

COMMS1	D36COM	HLT	Bad pointer passed to memory manager Header word trashed or bad pointer.
COMMS2	D36COM	HLT	Bad pointer passed to memory manager Header word trashed or bad pointer.
COMMS3	D36COM	HLT	Bad pointer passed to memory manager Header word trashed or bad pointer.
COMMTS	D36COM	CHK	New Message block Too Short
COMMZP	D36COM	CHK	DNMINI was passed a Zero Pointer A caller probably meant to ask for zero bytes of user data in T2 and mistakenly put the count in T1. T1 is supposed to contain the pointer to the message block being refreshed. Find caller on the stack and fix it.
COMODP	D36COM	CHK	DNGOPS called with bad MS In range checking the ac MS, its contents were found to be outside the range of addresses used for the MS block. Trace back to the caller and find out why he has an invalid pointer.
COMSTB	D36COM	CHK	Smear request Too Big The caller requested that a very large block be smeared. Find out what the caller really wanted to smear and fix the call.
CTHBCM	NRTSER	EVENT	Bad Configuration Message
CTHIBO	NRTSER	STOP	CTERM Host Input Buffer Overflow
CTHOCE	NRTSER	STOP	Output Count Exceeded
CTHPED	NRTSER	INFO	Protocol Error Detected
CTHPER	NRTSER	INFO	Protocol Error Reported
D36UBT	D36COM	CHK	End of UBL Trashed
DDIIFD	DNADLL	CHK	Illegal Function from DDP Driver

LIST OF DECnet-10 STOPCODES

	DMIIFD	DNADLL	CHK	Illegal Function from DMR Driver
	DNAWEM	DNADLL	STOP	Something confused DNBBP D36COM CHK Bad Byte-Pointer CIDLL is copying a DECnet message to a SCA buffer, and came across a bytepointer in a MSD where the first three bits are 5,6 or 7.
	DNDCGV	DNADLL	CHK	Couldn't Get memory for eVent argument block
	DNDIKF	DNADLL	CHK	Illegal Function code from DLL Kontroller
	DNDINF	DNADLL	CHK	Illegal Function code from NTMAN DNADLL was called with a bad function by NTMAN
	DNDIRF	DNADLL	CHK	Illegal Function code from ROUTER
	DNSLJ	D36COM	CHK	MOVSLJ Failed The MOVSLJ instruction did not skip
	DTIIFK	DNADLL	CHK	Illegal Function code from DTE Kontroller
	KDIIFD	DNADLL	CHK	Illegal Function from KDP Driver
	LLIBWK	LLINKS	CHK	SCTNSF call from sched without lock The DECnet entry point SCTNSF has been called from scheduler level when the Session Control interlock was locked. All scheduler level routines which call SCTNSF should first check SCTLOK. If SCTLOK is not -1, then the caller should wait for the next scheduler cycle before calling SCTNSF.
	LLIDIR	LLINKS	CHK	Duplicate Interrupt message Received The code found a duplicate interrupt message on the unacked interrupt receive queue. One should never get this message because the code is not allowed out of the NSP interlock with anything in this receive queue. Identify the problem. Either the interrupt flow control malfunctioned and sent more than one data request, or the remote node sent an interrupt message without a data request.
	LLIFNS	LLINKS	CHK	SCTL passed bad NSP PID

LIST OF DECnet-10 STOPCODES

LLIFZM	LLINKS	CHK	Tried to Free Zero Msg
LLIHTG	LLINKS	HLT	INIHSB can't get a hash table
LLIHTS	LLINKS	HLT	NSPHTS not set up
LLIIFC	LLINKS	CHK	Illegal Flow Control type
LLIILMA	LLINKS	CHK	RETBUF left LAR # LMA
LLIORC	LLINKS	CHK	ORC should never be negative
LLIORQ	LLINKS	CHK	ORQ is non-empty at port close This BUG only appears in DEBUG monitors.
LLIPIM	LLINKS	CHK	PROCXQ found Illegal Message type
LLIQIN	LLINKS	CHK	Queued INterrupt message illegal
LLIS2S	LLINKS	CHK	Illegal flow control at PRCRQS
LLITNE	LLINKS	CHK	Unknown Event Type at NSPEVT T1 contains an illegal NSP event type. Note that NSPEVT is called by SCLINK as well as LLINKS. Caller address is on the stack.
NISEC6	D36COM	CHK	Not in SECTion 6
NMXTBG	D36COM	CHK	NMXTIM Table Out Create a new table. I really doubt that this table has become obsolete. Look for a different bug.
NRTBPM	NRTSER	CHK	Bad Pointer passed to memory Manager
NRTFW0	NRTSER	CHK	Tried to Free Words at zero
NRTHBC	NRTSER	CHK	NRTHBR should never Be Called
NRTILS	NRTSER	CHK	NRT Link in unexpected state
NRTINP	NRTSER	CHK	NRT INput to DECnet failed

LIST OF DECnet-10 STOPCODES

NRTOUD	NRTSER	CHK	NRT OUTput to DECnet failed
NRTSAB	NRTSER	HLT	No memory for NRT's SAB
NRTSET	NRTSER	CHK	SCTPSQ returned wrong channel info
NRTSJB	NRTSER	HLT	No memory for NRT's SJB
NRTSJM	NRTSER	HLT	No memory for NRT's SJB
NTBSUP	D36COM	CHK	Buffer SUPplied The routine NTPARM was called to handle a network management parameter. The routine can only handle returns of a single value, but NTMAN had supplied a multi-word buffer.
NTBTSM	D36COM	CHK	Buffer Too Small NTMAN requested a show counter operation, but did not supply a buffer large enough to store all the counters.
NTMBCF	NTMAN	CHK	Bad Coded Field on output While formatting output for a SHOW, the program was requested to generate a Coded field of more than one byte. The program is not coded for this function. Look at the descriptor block pointed to by NT. Check to see if this item is supposed to be a multiple byte Coded. If not, fix the item's entry. If it is correct, write the code to handle multiple-byte Coded fields.
NTMBCL	NTMAN	CHK	Bad Counter byte Length
NTMBDL	NTMAN	CHK	Bad multiple byte Length The code generates output for a numeric field, and was asked to generate an illegal number of bytes.
NTMBFP	NTMAN	CHK	Bad Format Type encountered While the program was in the process of reading a value from the user string, the descriptor tables returned an invalid format for this item. The AC NT points to the descriptor for this item, and field NTSEQ should tell which item is being referenced. Fix the entry for this item so it contains a valid format type.

LIST OF DECnet-10 STOPCODES

	NTMCBL	NTMAN	CHK	Bad Counter Block Length
	NTMCNO	NTMAN	HLT	Circuit Name Overrun More than 16 bytes of data were returned to a 16 byte field. The data beyond the buffer was trashed. Examine the algorithm at NMXC2N to determine why the code returned more bytes than were expected. To avoid this halt, fix the above code to check for overrun while it is producing the bytes.
	NTMDVI	NTMAN	CHK	NMXDSP Value Illegal The code called a layer to obtain a value or set a value for an item. The routine value in the descriptor block pointed to by NT was illegal. Examine the data structure pointed to by NT. Probably this was caused by a trashed NT, since the descriptor block generation macros are supposed to range check this value.
	NTMEFO	NTMAN	CHK	Event Function Out of range The event function supplied by a DECnet layer to NMXEVT was out of range. Make callers of NMXEVT supply the correct function code
	NTMEOR	NTMAN	CHK	Entity type Out of Range While double-checking the entity ID, before dispatching on it, the code found the type value was illegal. Since the value supplied by the user is checked at GETBLK, this means that field NXENT was trashed.
	NTMFOR	NTMAN	CHK	Format Out of Range In formatting output for a SHOW, the format block for this item had an illegal format type. See NTMBFP.
	NTMFUR	NTMAN	CHK	FUnction code out of Range The code is going to dispatch by function code, and found that the function code is out of range. Since the function code supplied by the user is checked in GETBLK, this means that field NXFNC was trashed.
	NTMICF	NTMAN	CHK	Non-counter function in PRSCOU

LIST OF DECnet-10 STOPCODES

NTMILN NTMAN CHK Illegal Number size

The code is going to read a numeric value from the user's string. The format descriptor block for this item specified read of an illegal number of bytes.

NTMINT NTMAN CHK Invalid Numeric Type

The code is generating output for a numeric field, and was asked to generate something other than decimal, hexadecimal or octal.

NTMKOR NTMAN CHK Kontroller Out of Range in circuit-id

The Kontroller field in a line-id is out of range. The value LD.MAX defines the number of Kontrollers known by D36PAR, and thus by NTMAN. The most likely cause of this error is a trashed AC.

NOTE

A Kontroller is any device driver with which Router will interface. It is used to define the name of a circuit/line, under the assumption that each Kontroller will control only a single line type.

| NTMLTR NTMAN CHK Line Type is out of Range

NTMNEC NTMAN CHK No Error Code, with error return

Some routine took the non-skip return, but did not give an error code by calling NTExxx. This means that the program returned to top level and field NXERR was zero. Determine which routine is failing, and make the error return give an error code.

NTMNTR NTMAN CHK Node Type is out of Range

The code was going to select entries to return (for function .NTSHO) and needs to know the node type (executor, remote, or loop) in order to choose the correct one. For other entities (circuit, lines) this field should contain zero. This field is set by ENTCVT.

NTMORE NTMAN CHK Unrecognized Entity type

An event was received from a DECnet layer, and the entity type is not legal.

Find the routine that logged the event, and convince it to supply a legal entity type.

LIST OF DECnet-10 STOPCODES

	NTMSOR	NTMAN	CHK	<p>Selection criteria is Out of Range</p> <p>The code is going to select items to return (for .NTSHO) depending on the selection criteria, and found the criteria to be out of range. Fix the check in GETBLK or find out why field NXSEL is being trashed.</p>
	NTMSQF	NTMAN	INF	<p>Signal Queue Full</p> <p>The signal queue was full when a new signal was logged. This might be caused by a malfunctioning NMLT20 that does not read the signals from the signal queue, or it may be caused by a DECnet device driver going rogue. A signal is used to tell NMLT20 that a device needs attention/reload.</p> <p>Restart NMLT20, or turn off malfunctioning DECnet device. If necessary, reload any devices by hand.</p>
	NTMSRF	NTMAN	CHK	<p>Skipness of Return Fouled up</p> <p>The code returns from NTMAN with a skip return, but there is an error code stored in field NXERR. Identify the caller that is giving the error code (or trashing NXERR) and make it give a non-skip return.</p>
	NTNBFS	D36COM	CHK	<p>No BuFfer Supplied</p>
	NTNBUF	D36COM	CHK	<p>No BUffer Supplied</p> <p>NTMAN requested a show counter operation, but did not supply a buffer to store the counters in.</p>
	ROUATL	ROUTER	CHK	<p>A routing message contains a start ID greater than we can handle</p>
	ROUAWS	ROUTER	CHK	<p>Adjacency block in queue When State is unused</p> <p>An adjacency block has been left in the queue of active adjacencies but its state is unused.</p>
	ROUBCD	ROUTER	INF	<p>Bad Checksum Detected when building routing message</p> <p>This stopcode indicates that something got trashed. Look at P1; it points to the end of the normal routing vector [RTRNRV]+[RTRMXN]. Check the vector itself (pointed to by RTRNRV) and see if the topology appears reasonable. Make sure RTRCKS is less than 16 bits.</p>

LIST OF DECnet-10 STOPCODES

ROUBMB	ROUTER	CHK	Bad message block pointer A Kontroller called RTRDSP with a function requiring a message block, and the pointer supplied (in T3) is either 0 or out of range. Determine why the Kontroller gave an invalid pointer. The pointer should originally have been obtained from this module.
ROUBMC	ROUTER	CHK	Normal routing vector BitMap Corrupted
ROUBMT	ROUTER	CHK	Bad Message Type received from the DLL
ROUBSN	ROUTER	CHK	Bad Source Node in message from NSP
ROUBSZ	ROUTER	CHK	Router circuit Block size was Zero on a running circuit
ROUBTF	ROUTER	INF	Bad Test message Format
ROUBTM	ROUTER	INF	Bad hello or Test Message
ROUCGV	ROUTER	INF	Couldn't Get memory for event argument block
ROUEHB	ROUTER	CHK	No Message Block for Event data
ROUEHM	ROUTER	CHK	No Message Block for Event data
ROUIFD	ROUTER	CHK	Illegal function code from the DLL
ROUIFS	ROUTER	CHK	Router got through the forward routine without picking a route For some reason RTRFWD got through its Forward process and either did not pick up a router or failed to flag a message which was for the local node as such or an unreachable message as such.
ROUIKF	ROUTER	CHK	Illegal Kontroller function CALCON was called with an illegal function code. The only allowed values are KF.QOB, KF.INI, and KF.HLT.
ROUILS	ROUTER	CHK	Illegal Circuit Specified in NSP msg
ROUNAV	ROUTER	CHK	An Adjacency has No routing Vector

LIST OF DECnet-10 STOPCODES

ROUNLN	ROUTER	CHK	Trying to return msg to Non-Local NSP
ROUNMR	ROUTER	CHK	NMX out of range
ROUNSO	ROUTER	CHK	NSP sent out-of-range packet
ROURCE	ROUTER	INF	Bad NI Router list message format
ROURFN	ROUTER	INF	Routing message Received From Non-routing node
ROURML	ROUTER	CHK	Stored routing message format error in RTRRCR
ROUER	ROUTER	CHK	Unexpected end of routing message
ROUJET	ROUTER	CHK	Unknown event type in RTNEVT
ROUJOC	ROUTER	CHK	Unable to obtain count of nodes in Phase IV message
ROUXNZ	ROUTER	CHK	R2NCAL called with MB=0
ROUZXT	ROUTER	CHK	Tried to free msg with MB=0
RTRIFN	ROUTER	CHK	Illegal Function code from NTMAN
SCLA2N	SCLINK	CHK	Node database inconsistent The node database SCLA2N failed an internal consistency check. SCLBWK SCLINK CHK SCTNSF call from sched Without lock
SCLCBN	SCLINK	INF	Phase-II Buffering Not implemented
SCLNZE	SCLINK	CHK	Passing zero error code to SCMUUO The code arrived at a routine that is supposed to store an error code for the user, but found that the error code is zero. This is an illegal value. Find which routine called SCTNIE with T1 containing zero and correct the caller's behavior.
SCLPMI	SCLINK	CHK	Node database Pseudo page Map Inconsistent

LIST OF DECnet-10 STOPCODES

SCLRIB	SCLINK	CHK	<p>Bad SCTRIB call from LLINKS</p> <p>LLINKS has called SCTRIB for permission to send a message to SCLINK and has passed an invalid SLB address in T1.</p> <p>Find out what is in LLINK's ELSCB and why it is not an SLB pointer.</p>
SCLSLB	SCLINK	CHK	<p>SLB bad at FRESLB</p> <p>There is no Session Control Job Block (SJB) for this Session Control Link Block (SLB). This error could have happened at any time during the life of the link, after it actively transferred data.</p>
SCLSMS	SCLINK	CHK	<p>STRMAT Messed up the Stack pointer</p>
SCLSPF	SCLINK	CHK	<p>SLB self Pointers messed up in FNDSLB</p>
SCLTFJ	SCLINK	CHK	<p>Freeing SJB with SLB entries existing</p>
SCLTFS	SCLINK	CHK	<p>Tried to Free wrong SLB</p>
SCLVAS	SCLINK	CHK	<p>Couldn't get memory</p> <p>SCLINK called ASGVAS to assign virtual address space for the node name/address database. Since the requested memory is non-resident, this should always succeed. However, ASGVAS gave a fail return.</p>
SCTBWK	SCLINK	CHK	<p>SCTNSF call from schedule without lock</p> <p>The DECnet entry point SCTNSF has been called from scheduler level when the Session Control interlock was locked.</p> <p>All scheduler level routines which call SCTNSF should first check SCTLOK. If SCTLOK is not -1, then the caller should wait for the next scheduler cycle before calling SCTNSF.</p>

GALAXY-10 STOPCODES

6 GALAXY-10 STOPCODES

An alphabetical list of the GALAXY-10 stopcodes is provided in the following section. The list shows the name of each stopcode, the module in which it is found, the stopcode message (for which the name is a mnemonic), and an explanation.

Each GALAXY component is made up of one or more modules, thus a stopcode can be generated by a module with a name other than that of the component producing the stopcode.

When GALAXY encounters an internal error, a stopcode is generated. A message containing the stopcode, the stopcode message, the location of the error, the module name where the error occurred, and an explanation of the error (if available) will appear in the operator log file. The following example shows the form of the message sent to the operator log file:

```
? Stopcode - XXXxxx - message at PC <pc> in location  
Program PN n(nnnn) + GLXLIB n(nnn) error at PC <pc> in module MN  
Last GLXLIB error at PC <pc> was ##; No IPCF message is available  
Crash block begins at address  
[Stopping program]
```

Where:

XXXxxx is the stopcode mnemonic.

message is the stopcode message.

location is the PC of the next instruction to be executed.

PN is the program running for that job.

n(nnnn) is the version number.

MN is module in which the stopcode occurred.

is the number of the last GLXLIB error that occurred

address is the location of the crash block.

Example:

```
? Stopcode - ILM - Illegal memory reference at PC in .JBTPC  
Program OPR 5(1023) + GLXLIB 5(1564) error at PC 647737 in module GLXINT  
Last GLXLIB error at PC 004142 was 15; No IPCF message is available  
Crash block begins at 674000  
[Stopping program]
```

LIST OF GALAXY STOPCODES

7 LIST OF GALAXY STOPCODES

<u>Name</u>	<u>Module</u>	<u>Type</u>	<u>Message and Explanation</u>
ABS	GLXSCN	HALT	Atom Buffer too Small The command from OPR is too long to fit in the Atom buffer for parsing.
AIC	PLRLBP	HALT	Abort labeler request from Illegal Context PULSAR cannot safely unwind the current state of the TCB when it is told to abort at some stage during tape label processing.
AIE	QSRSCH	HALT	Attempt to add Invalid event queue Entry S\$EVENT detected that the entry to be added to EVENT QUEUE is not the correct size.
AMT	QSRMDA	HALT	Allocated is More than Total (VOL .VLVSL BLOCKS) SCNVOL detected that the number of words allocated for VOL block pointer is greater than the total number of VOL blocks.
APT	GLXINT	HALT	Unknown APR Trap at PC <pc> APR CONI = <octal CONIword>
ASE	GLXMEM	HALT	Addressing Space Exhausted GLXMEM cannot allocate the requested memory. All section zero memory for program is in use.
AZA	GLXCOM	HALT	Attempt to Zero the ACs Bad argument(s) passed to routine .ZCHNK.
BBF	PLRLBP	HALT	Bad Backspace File Incorrect TCB status detected backspacing a file on tape.
BBR	PLRLBP	HALT	Bad Backspace Record Incorrect TCB status detected backspacing a record on tape.

LIST OF GALAXY STOPCODES

BCN	PLRLBP	HALT	Bad Call to NXTFIL Routine NXTFIL was called to skip to the next file's HDR1 label, but the TCB status indicates that the tape is not positioned in user data.
BCP	PLRLBP	HALT	Bad Call to POSTAP PULSAR has determined that the tape needs to be positioned to the next file but the tape is not currently at a tape mark or header label.
BDS	GLXSCN	HALT	Bad Default String The first character in the default string (\$DEFAULT) is a null.
BFC	GLXSCN	HALT	Bad Function Code An invalid parse function code was detected in routine S%CMND.
BLI	QSRMDA	HALT	<text> The BLISS routines called by QUASAR detected an error that warranted a stopcode; "text" is the reason returned by the BLISS routine.
BME	QSRMDA	HALT	'B' Matrix Entry is missing RETBMA called D\$BMTX to find a user's 'B' matrix entry, but there is no corresponding 'B' matrix entry.
BPN	GLXMEM	HALT	Bad Page Number <page number> VALPAG determined that a page is not part of the initial core image or is not marked in use.
BRS	QSRFSS	HALT	Bad Request Size The argument passed to routine GETDPA is not in the range 1 to 1000 (octal).
BTA	GLXTXT	HALT	Bad \$TEXT Argument given at address <address>
BTF	GLXSCN	HALT	Bad Table Format TABLK detected two identical entries in a table. Table entries must be unique.

LIST OF GALAXY STOPCODES

BTT	GLXKBD	HALT	<p>Backing up Terminal Twice</p> <p>GLXKBD only stores the current character. The previous character cannot be retrieved.</p>
CAC	GLXMEM	HALT	<p>Count of Available pages Confused</p> <p>M%ACQP detected that PAGSTA points at or past the top of memory.</p>
CAD	CDRIVE	HALT	<p>Cannot ADD/DELETE reader to/from interrupt system</p> <p>The PISYS. UWO failed in routine INTCNL.</p>
CAS	IBMSPL	HALT	<p>Cannot Accomplish SIGNON</p> <p>After signing on, CTSGON cannot activate all the required tasks. S1 has the error from L%CENT which failed in ACTTSK.</p>
CAS	PLRDSK	HALT	<p>Can't Append to SPT list</p> <p>PULSAR cannot add an entry to the Sat Pointer Table list for a structure.</p>
CAT	IBMSPL	HALT	<p>Cannot Activate task</p> <p>Call to ACTTSK failed because L%CENT returned FALSE (could not create a list entry for some reason - error code in S1).</p>
CBD	CDRIVE	HALT	<p>CDRIVE can't Be a %DEMND spooler</p> <p>Because a card reader is a "free running" device, CDRIVE must always be running if a card reader is present and is to be used. CDRIVE can be either a %STCMD or %ONCE spooler.</p>
CCE	NEBULA	HALT	<p>Can't Create list Entry</p> <p>G\$SEND was unable to create a list entry in NEBULA's IPCF resend queue. S1 contains error from L%CENT.</p>
CCE	ORION	HALT	<p>Can't Create list Entry</p> <p>The call to L%CENT in ADDNOD failed to create a list entry.</p>
CCE	QSRQUE	HALT	<p>Can't Create list Entry</p> <p>C\$SEND was unable to create a list entry in the RESEND queue list of IPCF messages to be re-sent.</p>

LIST OF GALAXY STOPCODES

CCI	SPRINT	HALT	Can't clear UFD Interlock The SETUOO UOO failed to clear the UFD (User File Directory) interlock in routine CLRUEFL.
CCP	GLXMEM	HALT	Cannot Create Page The PAGE. UOO failed in routine CREPAG.
CCR	PLRTAP	HALT	Can't Check Ring status The TAPOP. UOO in T\$WRCK failed when checking for write ring status.
CCS	CDRIVE	HALT	Cannot Close Spool file F%REL failed to close the spooled reader file in CREATE.
CCT	PLRTAP	HALT	Can't Connect Tape to PSI system In T\$OPEN, the call to I\$PICD failed to connect the tape drive to the PSI interrupt system in order to trap off-line, resulting in hung device conditions.
CCW	PLRT10	HALT	Can't Clear Watch bits The SETUOO UOO took the error return in routine I\$INIT.
CDC	PLRT10	HALT	Can't Determine density Capabilities The TAPOP. UOO took the error return while performing the .TFPDN function in routine I\$PDEN.
CDC	QSRT10	HALT	Can't get Disk Characteristics for unit <unit name> The DSKCHR. UOO failed in routine I\$GATR.
CDD	QSRT10	HALT	Can't Determine tape Densities The TAPOP. UOO failed while performing the .TFPDN function in routine I\$GATR.
CDF	IBMSPL	HALT	Can't Delete hold File Call to F%DEL failed 3 times to delete the same file over a time span of 6 minutes. Each time delete fails, a WTO message is issued. S1 has error code from call to F%DEL.

LIST OF GALAXY STOPCODES

CDM	GALGEN	HALT	Can't Determine Monitor type The GETTAB. UUO failed trying to determine monitor type.
CDT	QSRT10	HALT	Can't Determine tape Track status The TAPOP. UUO failed while performing function .TFTRK in routine I\$GATR.
CEI	BATCON	HALT	Can't Enable Interrupts The PISYS. UUO failed in routine SYSINI.
CEM	QSRCAT	HALT	Catalog Entry is Missing for resource (see RSNNUM@RSNNAM)
CFC	GLXMEM	HALT	Count of Free pages Confused GLXMEM's database is corrupt.
CFE	IBMSPL	HALT	Couldn't Find File entry A .QCFIL block could not be found in the queue create message page.
CFE	SPRINT	HALT	Can't Find Files to load In EXECUTE, SPRINT could not position to the head of the list of files necessary to generate an "execute" command.
CFO	ORION	HALT	Can't GETTAB Operator PPN
CFU	QSRT10	HALT	Can't Find UCB for Unit <unit name> In I\$ISTR, QUASAR was unable to find the UCB (Unit Control Block) corresponding to the system structure identified.
CFV	QSRMDA	HALT	Can't Find VSL address in VOL entry In DELBSL, there is no link from a VOL (volume) block back to the VSL (Volume Set List). A link should have existed because the VOL block was using a link in the VSL pointing to the VOL block.
CGC	CDRIVE	HALT	Can't Get reader hardware Characteristics DEVOP. UUO failed in INPGET while trying to obtain reader characteristics.

LIST OF GALAXY STOPCODES

CGC	QSRT10	HALT	Can't Get Controller type for tape drive <tape drive> TAPOP. UUO failed while performing function .TFKTP in routine I\$GATR.
CGD	PLRT10	HALT	Can't Get Density TAPOP. UUO took the error return while performing the .TFDEN function in I\$GDEN.
CGD	QSRT10	HALT	Can't Get Disk physical unit SYSPHY. UUO failed in routine I\$INIT.
CGF	ORION	HALT	Can't GETTAB FRCLIN line number
CGP	GLXIPC	HALT	Can't Get a PID C%INIT called C%CPID to create a PID (Process ID) and C%PID was unsuccessful.
CGS	CDRIVE	HALT	Cannot Get Spool file PPN GETTAB. UUO failed in routine RDINIT.
CGS	PLRT10	HALT	Can't GETTAB States word The GETTAB. UUO took the error return in routine I%OPRP while trying to get the %CNSTS word from the monitor.
CGS	QSRT10	HALT	Can't Get Status of tape drive <tape drive> The TAPOP. UUO failed while performing function .TFSTS in routine I\$GATR.
CGT	GLXKBD	HALT	Cannot GETJFN Terminal
CGV	ORION	HALT	Cannot GETTAB montior Version
CIF	GALGEN	HALT	Command Initialization Failed The call to SCMND returned FALSE in GETANS.
CLS	GLXKBD	HALT	Can't Lookup Status of terminal JFN The FILOP. UUO in routine K%OPEN failed while performing the .FOGET function for a terminal. Location CHNJFN contains the channel number.

LIST OF GALAXY STOPCODES

CME	QSRMDA	HALT	'C' Matrix Entry is Missing D\$DLCK called D\$CMTX to find a user's 'C' matrix entry but the entry does not exist.
CMU	PLROPR	HALT	Can't Make TCB The call to G\$MTCB returned FALSE in I\$CREC.
CMV	PLROPR	HALT	Can't Make TCB The call to G\$MTCB returned FALSE in I\$CUNL.
CNE	ORION	HALT	Central site Node not present The call to FNDNOD in W\$NODE returned a failure while using G\$HOST as an argument.
CNL	BATCON	HALT	Could Not Logout. Call to I%KJOB failed.
CNL	CDRIVE	HALT	Could Not Logout. Call to I%KJOB failed.
CNL	LPTSPL	HALT	Could Not Logout. Call to I%KJOB failed.
CNL	SPRINT	HALT	Could Not Logout. Call to I%KJOB failed.
COP	QSRT10	HALT	Cannot Open Prime queue The FILOP. UUO failed in I\$OQUE when QUASAR was trying to open the master queue file.
COR	QSRT10	HALT	Cannot Open Redundant queue FILOP. UUO failed in I\$OQUE when QUASAR was trying to open the secondary queue file.
COS	CDRIVE	HALT	Cannot Open Spool file F%OOPN returned an error that was not 'file already exists' in GETFIL.
COT	GLXKBD	HALT	Cannot OPENF Terminal
CPE	GLXFIL	HALT	Can't Position to EOF
CPF	PLRT10	HALT	Clear label Parameters Failed TAPOP. UUO took the error return in I\$CLLP while performing the .TFLPR+.TFSET function.

LIST OF GALAXY STOPCODES

CRB	PLRT10	HALT	Can't Read Buffer size TAPOP. UUU took the error return in I\$RDLP performing the .TFBSZ function.
CRD	QSRQUE	HALT	Create Rejected defer Data In routine Q\$DEFER, the call to Q\$CREATE detected errors.
CRL	GLXFIL	HALT	Can't Read Last byte of file
CRL	QSRQUE	HALT	Create Rejected Logout data The call to Q\$CREATE in Q\$LOGOUT detected errors.
CRM	PLRT10	HALT	Can't Read user's Mode The TAPOP. UUU took the error return in routine I\$RDLP while performing the .TFMOD function.
CRM	QSRQUE	HALT	Create Rejected Modify The call to Q\$CREATE in Q\$MODIFY detected errors.
CRS	QSRQUE	HALT	Create Rejected Spooling data The call to Q\$CREATE in Q\$SPOOL detected errors.
CRS	SPRINT	HALT	Can't Read Searchlist This stopcode indicates one of two conditions. Either the JOBSTR. UUU failed in GETSRC while trying to read SPRINT's current search list, or the PATH. UUU failed in GETSRC while trying to read SPRINT's current path.
CSB	GLXKBD	HALT	Can't Set terminal Break mask
CSB	PLRTAP	HALT	Can't Set Blocksize
CSD	PLRLBP	HALT	Can't Set Density The call to I\$SDEN returned FALSE, meaning the TAPOP. UUU to set the density of a tape failed. TAPOP. error code is in P1 (or CRSHAC+P1).

LIST OF GALAXY STOPCODES

CSE	GLXKBD	HALT	Cannot Set Echo on terminal
CSF	QSRCAT	HALT	CATLOG Startup Failed
CSI	GLXINT	HALT	Cannot Set up Interrupt system The PIINI. UWO failed in routine SETINT.
CSI	PLRTAP	HALT	Can't Set Industry compatible mode
CSM	PLRTAP	HALT	Can't Set DIGITAL compatible Mode
CSP	GLXINT	HALT	Cannot Activate Panic Channels
CSS	QSRT10	HALT	Can't get System Structure List The SYSSTR. UWO failed in routine I\$ISTR.
CSS	SPRINT	HALT	Can't Set Searchlist This stopcode indicates one of two conditions. Either the STRUWO. UWO failed in SETSRC trying to set SPRINT's search list, or PATH. UWO failed in SETSRC trying to set SPRINT's path.
CSU	PLRTAP	HALT	Can't Switch Units
CTL	GLXFIL	HALT	Cannot Trim LSN in buffered mode F%IBUF does not handle Line Sequenced Numbered files. F%IBYT must be used.
CUD	QSRFSS	HALT	Clearing Unused DPA QUASAR tried to release unused space in the failsoft file.
CUF	PLRT10	HALT	CHKACC. UWO Failed The CHKACC. UWO failed in routine I\$CKAC.
CWT	PLRTAP	HALT	Can't Write Tape-mark
DBC	ORION	HALT	Debug Crash - Keep this crash Execution continued at the location following a \$DEBRK macro in an interrupt service routine.

LIST OF GALAXY STOPCODES

DDC	OPRPAR	HALT	Device Designator Conversion error
DDF	ORION	HALT	Delete DN60 node Failed A call to L%DENT returned FALSE in DELNOD.
DSP	ORION	HALT	Delete Send failure PID table entry inconsistency The call to CHKFSL in DELSPL failed to find an entry in the "send failure PID table."
DTL	QSRFSS	HALT	DPA Too Large VALDPA detected a Disk Page Address for the failsoft file that is too large.
DTS	QSRFSS	HALT	DPA Too Small VALDPA detected a Disk Page Address for the failsoft file that is too small.
DTU	GLXINT	HALT	Date/Time Unavailable The GETTAB. UWO to get Universal Date/Time failed in routine I%NOW.
DUF	GLXINT	HALT	DEBRK. UWO Failed
EEP	QSRT10	HALT	Error Expanding Prime queue The FILOP. UWO failed in routine I\$WRIT.
EER	QSRT10	HALT	Error Expanding Redundant queue The FILOP. UWO failed in routine I\$WRIT.
EMF	QSRQUE	HALT	Event Modify Failed
ERT	IBMSPL	HALT	Unexpected Error in RELTKB A call to one of M%RMEM, M%RLNP, M%CLNC, M%RPAG, L%FIRST, L%NEXT, or L%DENT failed. Error code in S1, return PC of failing call in T1.
EWS	CDRIVE	HALT	Error Writing Spool file The call to F%OBUF took the error return in OUTCRD.

LIST OF GALAXY STOPCODES

FCE	GLXMEM	HALT	Free Count Exceeds FREINI The current count of free pages exceeds the initial count of free pages.
FCN	GLXMEM	HALT	Free Count Negative Routine REDUCE detected that the count of free pages went negative.
FFT	GLXKBD	HALT	Action FILOP. Failed to Terminal The FILOP. UWO failed in K%OPEN while trying to perform either the .FOSET or the .FOWRT function.
FIT	GLXFIL	HALT	FD location requested with Illegal Type Routine F%FD was called with illegal arguments.
FOF	GLXFIL	HALT	File Operation Failed unexpectedly
FSE	GLXKBD	HALT	File System Error TXTINP detected an error returned from F%IBYT that was not an EOF error.
FUD	QSRFSS	HALT	Found Unused DPA An unused DPA (Disk Page Address) indicates that the failsoft file system database is corrupt.
GNF	PLRT10	HALT	GETTAB for user's Name Failed
GSF	PLRT10	HALT	GETTAB for Serial number Failed
IAC	OPR	HALT	Argument count <count> not valid in display message An argument block of zero was found in a message from ORION.
IBN	GLXSCN	HALT	Illegal Base for Number The base for a number to be parsed was not in the range 2-10 (decimal).
IBO	GLXKBD	HALT	Input Buffer Overflow on escape sequence processing

LIST OF GALAXY STOPCODES

IBP	GLXKBD	HALT	<p>Illegal Byte Pointer in K%TXTI</p> <p>The byte pointer that CONVBP was going to convert is zero. This invalid byte pointer was found at RD+.RDDBP.</p>
IBS	GLXFIL	HALT	<p>Illegal Byte Size given</p> <p>An invalid byte size, out of the range 1-36 (decimal), was given in a call to open a file.</p>
IBU	BATCON	HALT	<p>Illegal BATCON. UUU</p> <p>In BATCON's LUUU handler, UUOCON, an opcode was detected that was out of range.</p>
IDC	IBMSPL	HALT	<p>Illegal Task/Device type code</p>
IDM	OPR	HALT	<p>Message argument type <argument type> not valid for Display Message</p> <p>The argument type was something other than the argument type constant, .CMTXT.</p>
IDM	OPRLOG	HALT	<p>Invalid Display Message type <msg type></p>
IDM	PLRLBP	HALT	<p>Invalid Date from Monitor</p> <p>A call to I\$DATE returned a string that STRNUM could not convert to a number.</p>
IEC	OPR	HALT	<p>Invalid Error Code for failure</p> <p>An OPR failure error code is not in the range expected in routine SETFAL.</p>
IEI	GLXKBD	HALT	<p>Illegal Escape sequence Instruction</p>
IFC	OPRPAR	HALT	<p>Invalid Function Code from command</p>
IFM	GLXFIL	HALT	<p>Illegal File Mode in subroutine call</p> <p>An operation was attempted on a file, but the file was opened in a mode that prevents the requested operation from succeeding.</p>
IFN	GLXFIL	HALT	<p>Illegal IFN provided in call</p> <p>The IFN passed to CHKIFN was not in IFNTAB.</p>

LIST OF GALAXY STOPCODES

IIF	GLXIPC	HALT	IPCF to Interrupt system connect Failed In CPIDI, the PISYS. UWO failed while trying to connect a job to the interrupt system.
IIP	GLXKBD	HALT	Illegal Input Pointer CONVBP detected a byte pointer of zero at RD+.RDIOJ.
IJM	QSRADM	HALT	Interlocked Job Missing Inconsistency in QUASAR's queue database was detected in KILPSB.
IJW	QSRADM	HALT	Interlocked Job Wrong Inconsistency in QUASAR's queue database was detected in KILPSB.
ILM	GLXINT	HALT	ILlegal Memory reference at PC <pc>
ILW	IBMSPL	HALT	Illegal Wakeup An internal task was awakened and with no wake-up conditions present.
IMR	GLXINT	HALT	Illegal Memory Read at PC in INTRPC Stack address is in SAVAC1+17.
IMV	QSRMDA	HALT	Invalid MDR/VSL forward/backchain pointers NSTUSR detected that a VSL does not contain a pointer to an MDR. Every VSL should point to an MDR.
IMW	GLXINT	HALT	Illegal Memory Write at PC in INTRPC, stack in SAVAC1+17
INlv1	GLXINT	HALT	Level LVL Interrupts Not supported
IOS	QSRMDA	HALT	Invalid Owner Specified in reassign message DEASSIGN detected that the job number in the DEASSIGN message does not match the job number in the MDR pointed to by the UCB of the device being deassigned.
IPE	PLEASE	HALT	Internal Parser Error
IPF	PLRTAP	HALT	Illegal Positioning Function

LIST OF GALAXY STOPCODES

IPH	OPRNET	HALT	Invalid Process Handle to kill
IPP	OPRPAR	HALT	Invalid PDB header in Parse block
IQN	GLXTXT	HALT	Illegal Qualifier Number <number> at <address> An illegal argument qualifier was used in a \$TEXT macro.
IRF	GLXIPC	HALT	IPCF Reception Failure In RCVMSG, the IPCFR. UO took the error return while trying to receive an IPCF message.
IST	GLXINT	HALT	Illegal instruction Trap at PC in INTRPC, Stack in SAVAC1+17
IT2	IBMSPL	HALT	Illegal Task type for 2780/3780 BLDTSK was asked to start a task for a device that is only valid using HASP protocol and 2780/3780 was currently being used.
ITD	QSRMDA	HALT	Invalid Tape Density specified for <tape drive name> The density for a tape drive returned in the .STSTS does not match any of the legal densities contained in the UCB for that tape drive.
IVU	QSRMDA	HALT	Invalid VOL/UCB forward/backchain pointers DSMACK detected that there is not a pointer to a UCB in a VOL block of a volume that is being dismounted from a device. When a volume is mounted, there should be a pointer in the VOL block to the UCB of the device, and a pointer in the UCB to the VOL block of the volume mounted.
IVV	QSRMDA	HALT	Invalid VSL/VOL forward/backchain pointers D\$FOWN did not detect a pointer in a VOL block back to a VSL when the VOL block was found by a pointer in a VSL. VSL and VOL blocks must be doubly linked.
LEM	CDRIVE	HALT	Lousy Error Message from D60SIN An unidentified error code was returned by D60SIN. S1 = error code.

LIST OF GALAXY STOPCODES

LGF	PLRTAP	HALT	Label Get Failed
LNA	IBMSPL	HALT	Logging Illegally LOGCHR was called and either the task was not active or didn't have the job's pages set up yet. S contains status bits.
LNA	QSRMDA	HALT	Logical Name Assignment failed The DEVLNM. UVO failed in REASSI while trying to assign a logical name to a device.
LNI	SPRINT	HALT	LOG Not Initialized LOGTXT was called to put a character in the log, there is not a LOG page set up for usage.
LPO	QSRDSP	HALT	List request message Page Overflowed
LRF	PLRTAP	HALT	Label Release Failed
MCF	PLRT10	HALT	MTAID. UVO Failed
MDS	QSRMEM	HALT	Moving Different Sizes QUASAR's queue database is corrupt.
MDV	QSRMEM	HALT	Moving Different Variabilities
MQE	QSRMDA	HALT	Missing QE for a pseudo process D\$CHKB detected that there was no QE page address in the MDR for a batch job in the input queue.
MRN	QSRMDA	HALT	Missing Resource Number
MRR	IBMSPL	HALT	Request Received while another active QUASAR sent IBMSPL a next job message and IBMSPL is still processing the previous request. S contains task status bits.
MST	OPR	HALT	Missing Syntax Table
MST	ORION	HALT	Missing Syntax Table
MUN	QSRT10	HALT	Missing Unit Name in .IPCST message

LIST OF GALAXY STOPCODES

MVP	QSRMDA	HALT	Missing VOL block Pointer
MWL	QSRMDA	HALT	Missing or Wrong VOL/UCB link
NAM	QSRMDA	HALT	Negative 'A' Matrix entry computed The count of an entry in the 'A' matrix went negative. There cannot be a negative number of any physical resource.
NBM	QSRMDA	HALT	Negative 'B' Matrix entry computed The count of an entry in the 'B' matrix went negative. There cannot be a negative number of allocations (claims) for a resource.
NBR	QSRSCH	HALT	Next-job'ing Bad Request In preparing a "next job" message for an object, NEXTJB called F\$RDRQ to find the address of the EQ (external queue) page on disk and the address returned was zero.
NCM	QSRMDA	HALT	Negative 'C' Matrix entry computed The count in a 'C' matrix went negative. There cannot be a negative number of owners (sharers) of a device.
NCS	OPRNET	HALT	No Current Server database in skew
NDE	ORION	HALT	Node Database Dmpty In DELNOD, it was discovered that the OPR node database is empty. There must be at least a central host node.
NEB	PLRTAP	HALT	No Error Bit Routine RETERR was called, but no error bits were lit in P1.
NFB	CDRIVE	HALT	First Block in message Not the object block
NFP	GLXMEM	HALT	No Free Pages M%IPRM was unable to get a free page for an IPCF receive.
NFV	PLRT10	HALT	No Free PS Vectors I\$PICD detected that there are no free interrupt vectors available.

LIST OF GALAXY STOPCODES

NGF	QSRT10	HALT	Necessary Gettab Failed The GETTAB. UWO in DOGTAB failed.
NID	NEBULA	HALT	No I/O Drivers included NEBULA was not linked with NEBCDN.
NIP	GLXINT	HALT	No Interrupt is in Progress The DEBRK. UWO took the skip return.
NLB	IBMSPL	HALT	Error finding Line Block Line block list is incorrect. T1 contains correct line block address, LB contains line block address returned by FNDLB.
NMF	QSRESS	HALT	No More Filespace The failsoft file is full.
NOP	ORION	HALT	No Page for OPRPAR ORION is out of dynamic memory. Won't occur unless GLXMEM loses the ASE stopcode.
NPB	IBMSPL	HALT	No Port Block on releasing line block T1 contains port,,line.
NQC	QSRSCH	HALT	Unimplemented Network Queue Control function QUASAR's scheduler dispatched to an NQC object specific function entry point which is not implemented.
NSF	ORION	CONT	NML Startup Failed See last GLXLIB error.
NSH	D60JSY	HALT	Can't find SIGNON device Handle after creating it
NUE	QSRMDA	HALT	Null UCB chain Encountered In D\$INIT, the call to L%FIRST failed to return the first UCB block in the UCB chain.
NUV	QSRDSP	HALT	No UCB PTR and no VSL PTR from VOL MDA's database is corrupt.

LIST OF GALAXY STOPCODES

NVC	QSRMDA	HALT	NEW-VOLUME Code not working
NVD	PLRT10	HALT	No Valid Density I\$PDEN was unable to determine a valid density for reading a tape drive.
NXM	GLXINT	HALT	Non-existent Memory at PC <pc>
OBR	GLXOTS	HALT	Obsolete Routine executed
ODE	ORION	HALT	OPR Delete Entry error DELOPR detected that the list of "operators" is empty.
ODI	ORION	HALT	OPR Database Inconsistent The call to VALOPR in SPDOPR failed.
ONV	QSRMDA	HALT	Offset of New Volume is invalid In D\$VSR, the calculated offset into the block of VOL block pointers in the VSL is negative.
OOR	GLXOTS	HALT	OTS Only Routine executed
OQT	QSRSCH	HALT	NEXTJB Object to Queue header Translation failed
OSF	OPR	HALT	ORION Send Failed
OSF	QSRT10	HALT	ORION Startup Failed
OTS	GLXFIL	HALT	File Open block is Too small
PAF	GLXMEM	HALT	Page Access check Failed While performing function .PAGCA, the PAGE. UO failed in routine M\$I\$PRC.
PBI	OPRQSR	HALT	P\$DEV Blew It
PDL	GLXINT	HALT	PushDown List overflow at PC <pc>

LIST OF GALAXY STOPCODES

PEF	GLXMEM	HALT	Page Existence check Failed The PAGE. UWO failed while performing function .PAGCA in PAGFRE.
PIR	GLXIPC	HALT	PID Index out of Range The system PID index passed to SPID is invalid.
PKF	GLXMEM	HALT	Page Kill Failed The PAGE. UWO failed in KILPAG.
PLM	PULSAR	HALT	Previous List TCB has been Meddled
PNR	PULSAR	HALT	PULSAR Not Restartable
PQI	QSRT10	HALT	Prime Queue is Interlocked In I\$OQUE, the FILOP. UWO error return indicates that the master queue is being modified.
PRF	PLRTAP	HALT	Positioning Request Failed
PSF	QSRT10	HALT	PULSAR Startup Failed
PWE	QSRT10	HALT	Prime Write Error The OUT. UWO in I\$WRIT took the error return. IO.BKT was not one of the error bits returned using the GETSTS. UWO.
QNR	QUASAR	HALT	QUASAR Not Restartable
QSF	CDRIVE	HALT	Send to QUASAR Failed
QSF	LPTSPL	HALT	Send to QUASAR Failed
QSF	SPRINT	HALT	QUASAR Send Failed
RAR	GLXIPC	HALT	Releasing Already Released IPCF message In C%REL, RCVMDB+MDB.MS contains zero.
RAT	PULSAR	HALT	Requesting work for Active TCB

LIST OF GALAXY STOPCODES

RCN	ORION	HALT	G\$RSDC is Negative, database Confused RSDMSG detected that the resend "retry count" is negative.
RCN	QSRFSS	HALT	Request Count Negative
RCO	ORION	HALT	G\$RSDC Off .. does not match list data In RSDMSG, the resend "retry count" indicates that there are more messages to resend, but the list is empty.
RCW	QSRFSS	HALT	Rebuild Count Wrong This stopcode indicates problems rebuilding part of the in-core queues from the current section of the failsoft file.
REF	QSRT10	HALT	Reading End of File A second EOF error return was generated because there is no more data to be read.
REI	ORION	HALT	Remembered Entry <entry #> in list <list #> Invalid NXTMSG detected an error in its IPCF message database.
RIE	QSRT10	HALT	Read I/O Error In I\$READ, the IN. UO took the error return, and the error was not EOF.
RJM	QSRADM	HALT	Requeue Job Missing QUASAR's object database is corrupt.
RKD	PLRDSK	HALT	Running a Killed Disk TDB
RKM	PLRTAP	HALT	Running a Killed Magtape TDB
RLT	PLRTAP	HALT	Failed Reading Label Type
RMB	QSRMDA	HALT	Resource number Missing in 'B' matrix The unique resource identifier is missing in a 'B' matrix entry.

LIST OF GALAXY STOPCODES

RMC	QSRMDA	HALT	Resource number Missing in 'C' matrix The unique resource identifier is missing in a 'C' matrix entry.
RNF	GLXMEM	HALT	Received Non-existent page M%IPRC detected that a page created by IPCF does not exist.
RNR	QSRMDA	HALT	Returning Non-existent resource
RNW	GLXMEM	HALT	Ridiculous Number of Words requested The number of words requested is greater than number of words available in M%GMEM.
RPF	PLRT10	HALT	Read label Parameters Failed The TAPOP. UVO failed in I\$RDLP.
RRF	QSRFSS	HALT	Rebuild Routine Failed One of the queue rebuild routines in REBTBL failed.
RSE	PULSAR	HALT	ReSchedule from Exec level
RSF	PLRT10	HALT	TAPOP. to Read Statistics Failed
RTS	GLXFIL	HALT	Rename block Too Small
RTT	IBMSPL	HALT	Couldn't find task to be released Accumulator TK contains bad task block address
RUJ	QSRSCH	HALT	Releasing Uninterlocked Job In JOBDUN, the ITN of the job and object do not match when trying to release the job-object interlock.
RWE	QSRT10	HALT	Redundant Write Error In I\$WRIT, the OUT. UVO took the error return. IO.BKT is not the error when writing redundant queue.
RZP	GLXMEM	HALT	Request for Zero Pages

LIST OF GALAXY STOPCODES

SCE	QSRMDA	HALT	Structure Catalog Entry is missing A known structure in the 'A' matrix was not found in the structure catalog.
SDF	OPR	HALT	Setup Dialog Failed
SFI	ORION	HALT	Send Failure table Inconsistent
SFO	OPR	HALT	Setup Failure by OPR
SFP	GLXSCN	HALT	Scanning Floating Point not implemented XCMFLT was called.
SIO	PLRTAP	HALT	Switch units with OPEN Label DDB
SLT	PLRT10	HALT	Set Label Type failed
SND	PLRTAP	HALT	Switch units with Non-existent Device <device>
SPF	PLRT10	HALT	Set label Params Failed
SQF	BATCON	HALT	Send Failure to QUASAR
SQF	IBMSPL	HALT	Send to QUASAR Failed S1 contains error code from C%SEND.
SSR	PLRLBP	HALT	Strange Skip Record PULSAR wanted to skip a tape record but the tape's position was unknown.
STS	OPRPAR	HALT	Shared switch table Size (in TEMTSZ) Too Small for table of size in T2.
TBI	PLEASE	HALT	S%TXTI Block Incorrect
TDE	OPRPAR	HALT	Table Delete Error
TFF	GLXKBD	HALT	FILOP. OUT failed to terminal
TML	GLXTXT	HALT	Too Many Levels of call SAVLVL detected that it was called more than once to save T%TEXT context.

LIST OF GALAXY STOPCODES

TML	LPTSPL	HALT	Too Many Log buffers required LOGBUF detected that more than ten pages are being used to build LPTSPL's RUN LOG.
TMS	CDRIVE	HALT	Too Many Setups CDRIVE was told to start more readers than it can handle.
TMS	LPTSPL	HALT	Too Many Setups LPTSPL was told to start more printers than it can handle.
TMS	NEBULA	HALT	Too Many Setup messages NEBULA was told to start more streams than it can handle.
TMS	QSRFSS	HALT	Too Many Sections Corrupt failsoft queue.
TMT	GLXSCN	HALT	Too Much Text The buffer for the command being parsed cannot hold any more text.
TNE	IBMSPL	HALT	Task Not active Active task list is corrupt.
TNO	GLXKBD	HALT	Terminal Never Opened
TSB	SPRINT	HALT	Tried Stacking Binary cards
TSQ	IBMSPL	HALT	Tasks Still Queued to line block on release S1 contains address of task list queued to line block.
TUF	PLRT10	HALT	TAPOP. UVO Failed The TAPOP. UVO failed while trying to get REELID in I\$RDEV.
UDL	QSRMDA	HALT	
UFI	GLXFIL	HALT	Unknown File Information descriptor F%INFO was called with an invalid argument.

LIST OF GALAXY STOPCODES

ULS	PLRDSK	HALT	Unit parameter List is Short
UMS	SPRINT	HALT	Unsupported recording Mode specified <mode>
UMT	GALGEN	HALT	Unrecognized Monitor Type
UNR	GLXOTS	HALT	UNimplemented Routine executed
URM	SPRINT	HALT	Unknown Recording Mode <mode> error in NEXTJOB message
USM	QSRT10	HALT	Unique Stream Missing Calls to either L%FIRST or L%NEXT in UNIFST returned FALSE.
VAM	QSRMDA	HALT	VSL Address is Missing in a MDR
VPF	QSRMDA	HALT	Volume Pointer not Found SCNVOL detected that VSL's VOL block(s) links are inconsistent.
VSA	QSRMDA	HALT	VSL Address is missing in a VOL There is no pointer to a VSL in a VOL block found by a pointer from a VSL. VSL and VOL blocks should be doubly linked.
WBL	QSRT10	HALT	Writing Bad Length The block length to be written in I\$WRIT is greater than one page (512 words).
WFO	GLXINT	HALT	WTO Function <function> Out of range at address <address>
WLT	OPR	HALT	Wrong Length Table entry block Command syntax tables are corrupt.
WLT	ORION	HALT	Wrong Length Table entry block Command syntax tables are corrupt.
WNF	PULSAR	HALT	Waiting TCB Not Found
WQV	QSRFSS	HALT	Wrong Version of master Queue file

LIST OF GALAXY STOPCODES

ZTE	OPR	HALT	Zero entry in syntax Table Entry block Command syntax tables are corrupt.
ZTE	ORION	HALT	Zero entry in syntax Table Entry block Command syntax tables are corrupt.
ZTS	OPR	HALT	Zero Tables Setup for OPR No commands tables were found.
ZTS	ORION	HALT	Zero Tables Setup for OPR No commands tables were found.
ZWR	GLXMEM	HALT	Zero Words of memory Returned

READER'S COMMENTS

Your comments and suggestions help us to improve the quality of our publications.

For which tasks did you use this manual? (Circle your responses.)

- (a) Installation (c) Maintenance (e) Training
 (b) Operation/use (d) Programming (f) Other (Please specify.) _____

Did the manual meet your needs? Yes No Why? _____

Please rate the manual in the following categories. (Circle your responses.)

	Excellent	Good	Fair	Poor	Unacceptable
Accuracy (product works as described)	5	4	3	2	1
Clarity (easy to understand)	5	4	3	2	1
Completeness (enough information)	5	4	3	2	1
Organization (structure of subject matter)	5	4	3	2	1
Table of Contents, Index (ability to find topic)	5	4	3	2	1
Illustrations, examples (useful)	5	4	3	2	1
Overall ease of use	5	4	3	2	1
Page Layout (easy to find information)	5	4	3	2	1
Print Quality (easy to read)	5	4	3	2	1

What things did you like *most* about this manual? _____

What things did you like *least* about this manual? _____

Please list and describe any errors you found in the manual.

Page	Description/Location of Error
_____	_____
_____	_____
_____	_____

Additional comments or suggestions for improving this manual: _____

Name _____	Job Title _____
Street _____	Company _____
City _____	Department _____
State/Country _____	Telephone Number _____
Postal (ZIP) Code _____	Date _____

Fold Here and Tape

Affix
Stamp
Here

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
200 FOREST STREET MRO1-3/L12
MARLBOROUGH, MA 01752-9101

Fold Here