IDENTIFICATION

Product Id: ZZ-CXDRIVER-6.0

Product Title: CXDRIVER - VAX CI Diagnostic Controller Class Driver

Department: Base Systems Diagnostic Engineering

Product Date: 1-July-1984

Table of Contents

1.0 Loading ..... page 1

1.0 Loading

To load the VAX diagnostic device driver for the CI780 or CI750 on a VAX/VMS operating system, use the following command sequence:

```
$ MCR SYSGEN
SYSGEN> LOAD SYS$MAINTENANCE:CXDRIVER
SYSGEN> CONNECT CX/NOADAP
SYSGEN> EXIT
```

Once this is done the state of the device can be checked by doing:

```
$SHO DEV CX
```

If the CI is not online (SHO DEV PA), then the driver will not be loaded.

Object Module Synopsis

Module Name	Ident	Bytes	File	Creation Date	Creator
CXTABLES	V6-000	553	CXTABLES.OBJ;15	7-JUL-1984 15:23	VAX-11 Macro V03-01
CXCMDFDT	V06-000	1120	CXCMDFDT.OBJ;17	7-JUL-1984 15:20	VAX-11 Macro V03-01
CXCMD	V6-000	1976	[SHULL.EVXCI.CXDRIVER]CXCMD.OBJ;36	7-JUL-1984 15:19	VAX-11 Macro V03-01
CXMAINT	V6-000	900	[SHULL.EVXCI.CXDRIVER]CXMAINT.OBJ;15	7-JUL-1984 15:22	VAX-11 Macro V03-01
CXINPUT	V6-000	533	[SHULL.EVXCI.CXDRIVER]CXINPUT.OBJ;17	7-JUL-1984 15:21	VAX-11 Macro V03-01
CXCANCEL	V6-000	567	CXCANCEL.OBJ;22	7-JUL-1984 15:18	VAX-11 Macro V03-01
CXINIT	V6-000	158	[SHULL.EVXCI.CXDRIVER]CXINIT.OBJ;16	7-JUL-1984 15:21	VAX-11 Macro V03-01
SYS	V03-047	0	SYS$SYSROOT:[SYSEXE]SYS.STB;1	3-JUN-1984 17:36	VAX-11 Linker V3A-18
GLOBALS	V03-003	0	SYS$SYSROOT:[SYSEXE]SYSDEF.STB;1	3-JUN-1984 15:02	VAX-11 Macro V03-01

Program Section Synopsis

Psect Name	Module Name	Base	End	Length	Align	Attributes
$$$105_PROLOGUE		00000000	00000051	00000052 (82.)	BYTE 0	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	CXTABLES	00000000	00000051	00000052 (82.)	BYTE 0	
$$$115_DRIVER		00000054	000016B5	00001662 (5730.)	LONG 2	NOPIC,USR,CON,REL,LCL,NOSHR, EXE, RD, WRT,NOVEC
	CXTABLES	00000054	0000022A	000001D7 (471.)	LONG 2	
	CXCMDFDT	0000022C	0000068B	00000460 (1120.)	LONG 2	
	CXCMD	0000068C	00000E43	000007B8 (1976.)	LONG 2	
	CXMAINT	00000E44	000011C7	00000384 (900.)	LONG 2	
	CXINPUT	000011C8	000013DC	00000215 (533.)	LONG 2	
	CXCANCEL	000013E0	00001616	00000237 (567.)	LONG 2	
	CXINIT	00001618	000016B5	0000009E (158.)	LONG 2	

Symbols By Name

Symbol	Value	Symbol	Value	Symbol	Value
ACP$ACCESS	80008A8E	ACP$GW_SYSACC	80003E80	BOO$A_BOOPARAM	800257E0
ACP$ACCESSNET	80008A93	ACP$GW_WORKSET	80003E76	BOO$C_BOOPARSZ	00000084
ACP$DEACCESS	80008ABF	ACP$MODIFY	80008B15	BOO$C_SYSPARSZ	000003E8
ACP$GB_BASEPRIO	80003E86	ACP$MOUNT	80008B1D	BOO$GB_NODENAME	80025850
ACP$GB_DATACHK	80003E85	ACP$READBLK	80008B3C	BOO$GB_SYSTEMID	8002582E
ACP$GB_MAXR_AD	80003E82	ACP$V_READCHK	00000000	BOO$GL_BOOTCB	80025820
ACP$GB_SWAPFLGS	80003E87	ACP$V_SWAPGRP	00000001	BOO$GL_CHKPRTLOA	80025848
ACP$GB_WINDOW	80003E83	ACP$V_SWAPMAG	00000003	BOO$GL_CLSLOA	80025840
ACP$GB_WRITBACK	80003E84	ACP$V_SWAPPRV	00000002	BOO$GL_DEVNAME	80025860
ACP$GW_DIRCACHE	80003E74	ACP$V_SWAPSYS	00000000	BOO$GL_DSKDRV	800257E0
ACP$GW_EXTCACHE	80003E7A	ACP$V_WRITECHK	00000001	BOO$GL_ERAPATLOA	80025844
ACP$GW_EXTLIMIT	80003E7C	ACP$WRITEBLK	80008B78	BOO$GL_FPEMUL	8002585C
ACP$GW_FIDCACHE	80003E78	ALLOC_DG_ERR	000007EF-R	BOO$GL_IRPCNT	800257FC
ACP$GW_HDRCACHE	80003E72	BDL$GL_DISK_LOG	80024400	BOO$GL_LRPCNT	8002580C
ACP$GW_MAPCACHE	80003E70	BDL$L_SYSDLOG	00000034	BOO$GL_LRPMIN	80025804
ACP$GW_QUOCACHE	80003E7E	BDL$S_CRELNM_ITMLST	00000074	BOO$GL_LRPSIZE	80025800

Symbol	Value	Symbol	Value	Symbol	Value
BOO$GL_LRPSPLIT	80025808	BUG$_BDPPURGERR	00000488	BUG$_INVLOCKID	00000648
BOO$GL_MTACCESSLOA	8002584C	BUG$_BRDMSGLOST	000004A0	BUG$_INVPTEFMT	000001D0
BOO$GL_NPAGEDYN	800257F4	BUG$_CEBREFNEG	00000498	BUG$_INVRSPID	000005C0
BOO$GL_PRTDRV	80025834	BUG$_CHMONIS	000000B8	BUG$_INVTQEFMT	000001D8
BOO$GL_SCSLOA	8002583C	BUG$_CHMVEC	000004F0	BUG$_IVBAKADIO	000001E0
BOO$GL_SPLITADR	800257F8	BUG$_CIPORT	000005E8	BUG$_IVBYTEALGN	00000690
BOO$GL_SPTFREH	80003F7C	BUG$_CJF	00000678	BUG$_IVGBLTYP	000001E8
BOO$GL_SPTFREL	80003F78	BUG$_CJFFAILOVR	00000680	BUG$_IVLISTK	000001F0
BOO$GL_SRPCNT	80025814	BUG$_CLUEXIT	00000668	BUG$_IVSSRVRQST	000001F8
BOO$GL_SRPSPLIT	80025810	BUG$_CNXMGRERR	00000638	BUG$_IVWSETLIST	00000200
BOO$GL_SYSLOA	800257E4	BUG$_CONSOLRX50	000006B8	BUG$_JNLACP	00000610
BOO$GL_TRMDRV	800257E8	BUG$_CONTRACT	000000C0	BUG$_JNLDRV	00000608
BOO$GL_UCODE	80025838	BUG$_CPUCEASED	00000660	BUG$_KRNLSTAKNV	00000208
BOO$GL_VAXEMUL	80025858	BUG$_DBLERR	000000C8	BUG$_KRPEMPTY	000006C0
BOO$GQ_FILCACHE	80025818	BUG$_DECPTREF	000000D0	BUG$_LKBGRANTED	000005B0
BOO$GQ_INILOA	800257EC	BUG$_DELCONPFN	000000D8	BUG$_LKBREFNEG	00000510
BOO$GT_TOPSYS	80025824	BUG$_DELGBLSEC	000000E0	BUG$_LOCKMGRERR	00000630
BUFF_AD	00000000	BUG$_DELGBLWCB	000000E8	BUG$_MACHINECHK	00000210
BUFF_LEN	00000004	BUG$_DELWSLEX	000000F8	BUG$_MAKEWSLE	00000218
BUG$A_PAGED	80025A00	BUG$_DEQSUBLCKS	00000508	BUG$_MAPCNTZER	000004E0
BUG$A_PAGEDEND	80028750	BUG$_DIRENTRY	00000100	BUG$_MBACBHUNG	000004A8
BUG$FATAL	800089E2	BUG$_DISKCLASS	000005E0	BUG$_MFYNULPGFL	00000228
BUG$T_MESSAGES	80026004	BUG$_DOUBLDALOC	00000108	BUG$_MODRELNBAK	00000220
BUG$_ACCVIOKSTK	000006A0	BUG$_DOUBLDEALO	00000110	BUG$_MPASYNCWRT	00000548
BUG$_ACCVIOMCHK	00000698	BUG$_ERRCACHFUL	000004D0	BUG$_MPBADMCK	00000538
BUG$_ACPMBFAIL	00000008	BUG$_ERRHALT	00000118	BUG$_MPCHMONIS	00000588
BUG$_ACPRECURS	000004B8	BUG$_EXHFUL	00000120	BUG$_MPCHMVEC	00000590
BUG$_ACPUNSTAK	000004C0	BUG$_EXPANDPHD	00000128	BUG$_MPDBLERR	00000560
BUG$_ACPVAFAIL	00000010	BUG$_EXTCACHIV	000004D8	BUG$_MPERRHALT	00000580
BUG$_ALCPHD	00000018	BUG$_FATALEXCPT	00000130	BUG$_MPHALT	00000568
BUG$_ALCSMBCLR	00000020	BUG$_FILCNTNONZ	000004F8	BUG$_MPILLVEC	00000570
BUG$_APTREFHIGH	00000028	BUG$_FREEPAGREF	00000138	BUG$_MPIVLISTK	00000558
BUG$_APTWRTERR	00000030	BUG$_FREWSLX	00000140	BUG$_MPKNLSTKNV	000005A0
BUG$_ASYNCWRTER	00000038	BUG$_GBLPAGSZRO	00000148	BUG$_MPMCHECK	00000540
BUG$_BADALORQSZ	00000040	BUG$_GBLWSLXERR	00000150	BUG$_MPNOUSRWCS	00000578
BUG$_BADBOOTCB	000000F0	BUG$_GPGNULPGFL	00000158	BUG$_MPSCBRDERR	00000598
BUG$_BADBUFADR	00000048	BUG$_HALT	00000160	BUG$_MPUNEXPINT	000005A8
BUG$_BADBUFTYP	00000050	BUG$_HDRNOTMAP	00000168	BUG$_MPUNKRSTRT	00000550
BUG$_BADDALRQSZ	00000058	BUG$_ICONCLUDAT	000006D0	BUG$_MPWALCIRP	00000230
BUG$_BADFID	00000060	BUG$_ICONPFNDAT	00000170	BUG$_MSCPCLASS	000006C8
BUG$_BADFORKIPL	00000068	BUG$_ICPAGELOC	00000178	BUG$_MSCPSERV	000006A8
BUG$_BADLCKWSLE	00000070	BUG$_IFREPAGCNT	00000180	BUG$_MTXCNTNONZ	00000238
BUG$_BADMCKCOD	00000078	BUG$_ILLEVTNUM	00000188	BUG$_NEGSHBREF	000004B0
BUG$_BADPAGFILA	00000080	BUG$_ILLVEC	00000190	BUG$_NETNOBUF	00000240
BUG$_BADPAGFILD	00000088	BUG$_INCONSTATE	00000198	BUG$_NETNOSTATE	00000248
BUG$_BADPAGTYPE	00000090	BUG$_INCPTREF	000001A0	BUG$_NETRCVPKT	00000250
BUG$_BADQHDR	00000478	BUG$_INSNFREPAG	000001A8	BUG$_NETSYSSRV	00000258
BUG$_BADRSEIPL	00000098	BUG$_INSSWPFIL	000001B0	BUG$_NETTRANCNT	00000260
BUG$_BADRVNWCB	000004C8	BUG$_INSWAPERR	000001B8	BUG$_NOACPCHAN	00000268
BUG$_BADSBMBLK	000000A0	BUG$_INVCHAN	000001C0	BUG$_NOACPMAIL	00000270
BUG$_BADSWPVBN	000000A8	BUG$_INVCJFIOD	000005F8	BUG$_NOAQBACP	00000278
BUG$_BADWCBPT	00000080	BUG$_INVEXCEPTN	000001C8	BUG$_NOBUFPCKT	00000280

Symbol	Value	Symbol	Value	Symbol	Value
BUG$_NOBVPVCB	00000288	BUG$_SCBRDERR	00000528	CDC$_REPEAT	00000002
BUG$_NOHDJMT	00000600	BUG$_SECREFNEG	000003B0	CDC$_SPECADR	00000001
BUG$_NOMULTBK	00000290	BUG$_SHRCNTNEG	000003B8	CDC$_UMSTATE	00000001
BUG$_NONEXSTACP	00000298	BUG$_SSRVEXCEPT	000003C0	CHECK_ACTV_CONN	00000617-R
BUG$_NORCP	00000618	BUG$_STATENTSVD	00000530	CHECK_CONN	000005DD-R
BUG$_NORLVBUF	000002A0	BUG$_STRNOTWCB	000003C8	CHECK_LINK_FQ	000015A3-R
BUG$_NOSHMGSD	00000490	BUG$_SWAPWSLE	000003D0	CHECK_LINK_PQ	000015B6-R
BUG$_NOTDDBDDB	000002A8	BUG$_SYSADJWSL	000003D8	CHECK_VC	00000C2C-R
BUG$_NOTFCBFCB	000002B0	BUG$_SYSTRMERR	000003E0	CHECK_VC_ERR	000007A0-R
BUG$_NOTFCBWCB	000002B8	BUG$_TAPECLASS	00000628	CIA$GL_MUTEX	80002CA8
BUG$_NOTFCPWCB	000002C0	BUG$_TIPCUFLOW	000003E8	CIA$GQ_INTRUDER	80002CAC
BUG$_NOTIRPAQB	000002C8	BUG$_UBMAPEXCED	000003F0	CJF$ASSJNL	7FFEE518
BUG$_NOTLKB	000005B8	BUG$_UDAPORT	000005D8	CJF$DEASJNL	7FFEE538
BUG$_NOTMTLMTL	000002D0	BUG$_UNABLCREVA	000003F8	CJF$DELJNL	7FFEE548
BUG$_NOTPCB	000002D8	BUG$_UNEXUBAINT	00000400	CLEAN_CDB	00001580-R
BUG$_NOTRULUCB	000005F0	BUG$_UNKNPRQ	00000480	CLEAN_CDRP	000015CD-R
BUG$_NOTRVTVCB	000002E0	BUG$_UNKRSTRT	00000408	CLEAR_BUFFER	0000065B-R
BUG$_NOTUCBIRP	000002E8	BUG$_UNSUPRTCPU	00000670	CLEAR_CLER_REG	00001179-R
BUG$_NOTUCBRVT	000002F0	BUG$_UNXINTEXC	00000410	CLEAR_GOU_BIT	0000116B-R
BUG$_NOTUCBUCB	000002F8	BUG$_UNXSIGNAL	00000418	CLEAR_GO_BIT	0000114C-R
BUG$_NOTUCBWCB	000004E8	BUG$_VBNMAPFAIL	00000420	CLEAR_STS_REG	00001108-R
BUG$_NOTVCBUCB	00000300	BUG$_WACKQEMPTY	00000428	CLS$AL_LOAVEC	800036B8
BUG$_NOTVVPVCB	00000308	BUG$_WCBFCBMNG	000005C8	CLU$AL_LOAVEC	800036B8
BUG$_NOTWCBIRP	00000310	BUG$_WCSCORR	00000658	CLU$GB_QDISK	80003EB0
BUG$_NOTWCBWCB	000005D0	BUG$_WRTINVBUF	00000430	CLU$GB_VAXCLUSTER	80003EA8
BUG$_NOUSRWCS	00000318	BUG$_WRTINVHDR	00000438	CLU$GL_ALLOCLS	80003EC4
BUG$_OPERATOR	00000470	BUG$_WRTPGSBAK	00000440	CLU$GL_CLUB	800036AC
BUG$_OPERCRASH	000006D8	BUG$_WSLENOVAL	00000448	CLU$GL_CLUSVEC	800036B0
BUG$_OUTOFSYNC	00000688	BUG$_WSLPAGCNT	00000450	CLU$GL_LOA_ADDR	800036B8
BUG$_OUTSWPERR	00000320	BUG$_WSLVANVAL	00000458	CLU$GW_LCKDIRWT	80003EC8
BUG$_PAGEREDERR	00000328	BUG$_WSLXVANMAT	00000460	CLU$GW_MAXINDEX	800036B4
BUG$_PAGEWRTERR	00000330	BUG$_WSSIZEERR	00000500	CLU$GW_QDSKINTERVAL	80003EC2
BUG$_PAGNTRNVAL	00000338	BUG$_XQPFRR	00000640	CLU$GW_QDSKVOTES	80003EC0
BUG$_PFNLISTCNT	00000340	BUG$_ZEROPAGE	00000463	CLU$GW_QUORUM	80003EAA
BUG$_PFNREFNZRO	00000348	CDC$_ALL	00000001	CLU$GW_RECNXINT	80003EAE
BUG$_PGFGBLBAD	00000350	CDC$_AUTOPATH	000000FF	CLU$GW_VOTES	80003EAC
BUG$_PGFIPLHI	00000358	CDC$_COND	00000001	CNX$ALLOC_CDRP	8000370A
BUG$_PGFLOCBAD	00000360	CDC$_DEFADR	00000000	CNX$ALLOC_CDRP_ONLY	80003710
BUG$_PROCGONE	00000368	CDC$_FORCE	00000000	CNX$ALLOC_WARMCDRP	80003716
BUG$_PTELENVIOL	00000370	CDC$_GARBAGE	00000001	CNX$ALLOC_WARMCDRP_CSB	8000371C
BUG$_PTRCNT	00000378	CDC$_MSTATE	00000000	CNX$BLOCK_READ	80003764
BUG$_PURGWSSCN	00000380	CDC$_NOACTOFF	00000000	CNX$BLOCK_READ_IRP	8000376A
BUG$_QUEUEMPTY	00000388	CDC$_NOACTON	00000001	CNX$BLOCK_WRITE	80003770
BUG$_RDSNONRES	00000390	CDC$_NOGARBAGE	00000000	CNX$BLOCK_WRITE_IRP	80003776
BUG$_REFCNTNEG	00000398	CDC$_OFF	00000000	CNX$BLOCK_XFER	80003752
BUG$_RESEXH	000006B0	CDC$_ON	00000001	CNX$BLOCK_XFER_IRP	80003758
BUG$_RMSBUG	000003A0	CDC$_PATH0	00000000	CNX$CHANGE_QUORUM	80003788
BUG$_RSBREFNEG	00000518	CDC$_PATH1	00000001	CNX$DEALL_MSG_BUF_CSB	80003722
BUG$_RSBREFNZRO	00000520	CDC$_POLLEROFF	00000001	CNX$DEALL_WARMCDRP_CSB	80003728
BUG$_RUF	00000620	CDC$_POLLERON	00000000	CNX$DISK_CHANGE	8000379A
BUG$_SBIAERROR	00000650	CDC$_PS512	00000001	CNX$INIT_CDRP	8000372E
BUG$_SCANDEADPT	000003A8	CDC$_PS576	00000002	CNX$PARTNER_FINISH	8000377C

Symbol	Value	Symbol	Value	Symbol	Value
CNX$PARTNER_INIT_CSB	8000375E	CTL$AQ_EXCVEC	7FFEFE34	CTL$GL_PRCALLCNT	7FFEFF9C
CNX$PARTNER_RESPOND	80003782	CTL$A_COMMON	7FFE1600	CTL$GL_RDIPTR	7FFEFFA0
CNX$POWER_FAIL	80003794	CTL$A_DISPVEC	7FFE6000	CTL$GL_RMSBASE	7FFEFF44
CNX$SEND_ANY_MSGS	8000373A	CTL$C_CLIDATASZ	000015A4	CTL$GL_RUF	7FFEFF5C
CNX$SEND_MSG	8000373A	CTL$C_KRP_COUNT	00000004	CTL$GL_SITESPEC	7FFEFF60
CNX$SEND_MSG_CSB	80003740	CTL$C_KRP_SIZE	00000200	CTL$GL_THEXEC	7FFEFE54
CNX$SEND_MSG_RESF	80003746	CTL$GB_DEFLANG	7FFEFF51	CTL$GL_THSUPR	7FFEFE58
CNX$SEND_MSG_RSPID	8000374C	CTL$GB_MSGMASK	7FFEFF50	CTL$GL_UAF_FLAGS	7FFE3018
CNX$SHUTDOWN	8000378E	CTL$GB_PWRMODE	7FFEFF24	CTL$GL_USRCHME	7FFEFF1C
COM$DELATTNAST	8000DE76	CTL$GB_SSFILTER	7FFEFF25	CTL$GL_USRCHMK	7FFEFF18
COM$DELATTNASTP	8000DE7E	CTL$GL_CCBBASE	7FFEFF38	CTL$GL_USRUNDWN	7FFEFF54
COM$DELCTRLAST	8000DFB8	CTL$GL_CLINTOWN	7FFE2204	CTL$GL_VECTORS	7FFEFE00
COM$DELCTRLASTP	8000DFC0	CTL$GL_CMCNTX	7FFEFF88	CTL$GL_VIRTPEAK	7FFEFEC0
COM$DRVDEALMEM	8000DF05	CTL$GL_CMHANDLR	7FFEFE30	CTL$GL_VOLUMES	7FFEFEC4
COM$FLUSHATTNS	8000DEC5	CTL$GL_CMSUPR	7FFEFE28	CTL$GL_WSPEAK	7FFEFEBC
COM$FLUSHCTRLS	8000E05F	CTL$GL_CMUSER	7FFEFE2C	CTL$GQ_ALLOCREG	7FFEFE8C
COM$POST	8000DEF7	CTL$GL_CREPRC_FLAGS	7FFEFFC0	CTL$GQ_COMMON	7FFEFE60
COM$SETATTNAST	8000DF4B	CTL$GL_CTLBASVA	7FFEFE7C	CTL$GQ_DBGAREA	7FFEFF3C
COM$SETCTRLAST	80C0E0AB	CTL$GL_DCLPRSOWN	7FFE2200	CTL$GQ_HELPFLAGS	7FFEFFA8
CON$ABORT	800034E4	CTL$GL_F11BXQP	7FFEFF90	CTL$GQ_ISTART	7FFEFEC8
CON$DISCONNECT	800034FC	CTL$GL_FINALSTS	7FFEFEB8	CTL$GQ_LOGIN	7FFEFEB0
CON$DS_SET	800034C6	CTL$GL_FIXUPLNK	7FFE2E08	CTL$GQ_MOUNTLST	7FFEFE94
CON$INITIAL	80003502	CTL$GL_GETMSG	7FFEFE68	CTL$GQ_POALLOC	7FFEFF94
CON$INITLINE	80003508	CTL$GL_IAFEXE	7FFE2E10	CTL$GQ_PROCPRIV	7FFEFF10
CON$INTDISI	80001354	CTL$GL_IAFLAST	7FFE2E04	CTL$GQ_TERMCHAR	7FFEFFB0
CON$INTDISO	80001378	CTL$GL_IAFLINK	7FFE2E00	CTL$GT_CLINAME	7FFE301C
CON$INTINP	8000350E	CTL$GL_IAFLNKPTR	7FFEFF8C	CTL$GT_SPAWNCLI	7FFE313C
CON$INTOUT	80003514	CTL$GL_IAFPERM	7FFE2E10	CTL$GT_SPAWNTABLE	7FFE315C
CON$NULL	800034F6	CTL$GL_IBIOCNT	7FFEFEE8	CTL$GT_TABLENAME	7FFE303C
CON$OWNCTY	80003526	CTL$GL_ICPUTIM	7FFEFED0	CTL$GW_CHINDX	7FFEFE02
CON$RELEASECTY	8000352C	CTL$GL_IDIOCNT	7FFEFEE4	CTL$GW_NMIOCH	7FFEFE00
CON$RESUME	800034EA	CTL$GL_IFAULTIO	7FFEFED8	CTL$GW_PPMSGCHN	7FFEFF52
CON$SENDCONSCMD	8C00351A	CTL$GL_IFAULTS	7FFEFED4	CTL$T_ACCOUNT	7FFEFEA8
CON$SET_LINE	800034C0	CTL$GL_IMGHDRBF	7FFEFE80	CTL$T_NODEADDR	7FFEFEF0
CON$SET_MODEM	800034F0	CTL$GL_IMGLSTPTR	7FFEFE84	CTL$T_NODENAME	7FFEFEF7
CON$STARTIO	800034BA	CTL$GL_IPAGEFL	7FFEFEE0	CTL$T_REMOTEID	7FFEFEFE
CON$STOP	800034D8	CTL$GL_IVOLUMES	7FFEFEEC	CTL$T_USERNAME	7FFEFE9C
CON$STOP2	800034DE	CTL$GL_IWSPEAK	7FFEFEDC	CTP$REVISION	00000000
CON$XOFF	800034D2	CTL$GL_KNOWNFIL	7FFEFF64	CTP$VERSION	00000003
CON$XON	8C0034CC	CTL$GL_KRP	7FFE6600	CX$B_CMDERRCNT	00000118-R
CONTROLLER_NAME	000000A8-R	CTL$GL_KRPBL	7FFEFFBC	CX$B_GARBAGE_FLAG	00000199-R
COPY_RSP	000013B9-R	CTL$GL_KRPFL	7FFEFFB8	CX$B_PWRFAIL	0000021A-R
CTL$AG_CLIDATA	7FFE325C	CTL$GL_KSPINI	7FFE7E00	CX$CONNECT_ERR	000012AE-R
CTL$AG_CLIMAGE	7FFE3008	CTL$GL_KSTKBAS	7FFE7800	CX$DDT	00000054-R
CTL$AG_CLITABLE	7FFE3010	CTL$GL_KSTKBASEXP	7FFE7000	CX$DG_IN	000011C8-R
CTL$AG_CMEDATA	7FFE1E00	CTL$GL_LNMDIRECT	7FFEFE08	CX$L_CECC	00000121-R
CTL$AL_CLICALBK	7FFE3000	CTL$GL_LNMHASH	7FFEFE04	CX$L_CETEMP	0000011D-R
CTL$AL_CMCNTX	7FFE1E00	CTL$GL_P1MERGE	7FFE2E0C	CX$L_CINT_PDT	0000021B-R
CTL$AL_FINALEXC	7FFEFF28	CTL$GL_PCB	7FFEFF58	CX$L_CMDBUFBGN	0000012D-R
CTL$AL_IPASTVEC	7FFEFF68	CTL$GL_PHD	7FFEFE88	CX$L_CMDBUFEND	00000131-R
CTL$AL_STACK	7FFEFE10	CTL$GL_POWERAST	7FFEFF20	CX$L_CUR_CMD	00000139-R
CTL$AL_STACKLIM	7FFEFE6C	CTL$GL_PPMSG	7FFEFF48	CX$L_LOCPRTLST	00000151-R

Symbol	Value	Symbol	Value	Symbol	Value
CX$L_NXT_RSP	0000013D-R	ERL$LOGMESSAGE	8000CA76	ERL$VEC264	800045C4
CX$L_PORT_STATE_PDT	00000227-R	ERL$LOGSTATUS	8000C9FB	ERL$VEC268	800045C8
CX$L_RSPBUFBGN	00000131-R	ERL$LOG_DMSCP	8000CAE0	ERL$VEC272	800045CC
CX$L_RSPBUFEND	00000135-R	ERL$LOG_TMSCP	8000CAD5	ERL$VEC276	800045D0
CX$L_RSPCNTREM	00000141-R	ERL$RELEASEMB	8000CC12	ERL$VEC28	80004674
CX$L_RSPSEQNUM	00000145-R	ERL$UNEXP	80004674	ERL$VEC280	800045D4
CX$L_RST_STR_CDRP	00000223-R	ERL$VEC0	80004674	ERL$VEC284	800045D8
CX$L_RST_STR_PDT	0000021F-R	ERL$VEC100	80004674	ERL$VEC288	800045DC
CX$L_SEQNUM	00000129-R	ERL$VEC104	80004674	ERL$VEC292	800045E0
CX$L_STATE	000000C8-R	ERL$VEC108	80004674	ERL$VEC296	800045E4
CX$L_STOPCMD	00000125-R	ERL$VEC112	80004674	ERL$VEC300	800045E8
CX$L_TEMP	00000119-R	ERL$VEC116	80004674	ERL$VEC304	800045EC
CX$MSG_IN	000011C8-R	ERL$VEC12	80004674	ERL$VEC308	800045F0
CX$Q_MAPLST	00000149-R	ERL$VEC120	80004674	ERL$VEC312	800045F4
CX$T_GARBAGE_BUF	0000019A-R	ERL$VEC124	80004674	ERL$VEC316	800045F8
CX$X_CNCLSAV	000000E4-R	ERL$VEC128	80004674	ERL$VEC32	80004674
CX$X_IRPSAV	000000CC-R	ERL$VEC132	80004674	ERL$VEC320	800045BC
CX$X_STOPSAV	00000100-R	ERL$VEC136	80004674	ERL$VEC324	800045C0
CX$_ALRDYSTOP	00000004	ERL$VEC140	80004674	ERL$VEC328	800045C4
CX$_CDRPNFND	00000001	ERL$VEC144	80004674	ERL$VEC332	800045C8
CX$_CDVERSION	00000006	ERL$VEC148	80004674	ERL$VEC336	800045CC
CX$_CONN_ACTV	00000003	ERL$VEC152	80004674	ERL$VEC340	800045D0
CX$_NOCMDBUF	00000005	ERL$VEC156	80004674	ERL$VEC344	800045D4
CX$_NOCONN	00000002	ERL$VEC16	80004674	ERL$VEC348	800045D8
CX_CANCELIO	000013E0-R	ERL$VEC160	80004674	ERL$VEC352	800045DC
CX_DG_RSP_SIZ	00000195-R	ERL$VEC164	80004674	ERL$VEC356	800045E0
CX_END	000016B6-R	ERL$VEC168	80004674	ERL$VEC36	80004674
CX_FUNCTABLE	0000008C-R	ERL$VEC172	80004674	ERL$VEC360	800045E4
CX_MSG_RSP_SIZ	00000191-R	ERL$VEC176	80004674	ERL$VEC364	800045E8
CX_STARTDATA	0000022C-R	ERL$VEC180	80004674	ERL$VEC368	800045EC
CX_STARTIO	0000068C-R	ERL$VEC184	80004674	ERL$VEC372	800045F0
CX_UNIT_INIT	00001619-R	ERL$VEC188	80004674	ERL$VEC376	800045F4
CX_UNSOLINT	00001695-R	ERL$VEC192	80004674	ERL$VEC380	800045F8
CX_VERSION	00000006	ERL$VEC196	80004674	ERL$VEC384	800045BC
DEALLOC_CDRP	00001600-R	ERL$VEC20	80004674	ERL$VEC388	800045C0
DEAL_MSG	000013C8-R	ERL$VEC200	80004674	ERL$VEC392	800045C4
DIAG_VERSION	00000008	ERL$VEC204	80004674	ERL$VEC396	800045C8
DTS_RUJNL	00000001	ERL$VEC208	80004674	ERL$VEC4	80004674
ECC$REENABLE	80003466	ERL$VEC212	80004674	ERL$VEC40	80004674
ERL$ALLOCEMB	8000CB3E	ERL$VEC216	80004674	ERL$VEC400	800045CC
ERL$AL_BUFADDR	800030B8	ERL$VEC220	80004674	ERL$VEC404	800045D0
ERL$COLDSTART	8000CB1C	ERL$VEC224	80004674	ERL$VEC408	800045D4
ERL$DEVICEATTN	8000C983	ERL$VEC228	80004674	ERL$VEC412	800045D8
ERL$DEVICERR	8000C8ED	ERL$VEC232	80004674	ERL$VEC416	800045DC
ERL$DEVICTMO	8000C8F1	ERL$VEC236	80004674	ERL$VEC420	800045E0
ERL$GB_BUFFLAG	800030C1	ERL$VEC24	80004674	ERL$VEC424	800045E4
ERL$GB_BUFIND	800030C0	ERL$VEC240	80004674	ERL$VEC428	800045E8
ERL$GB_BUFPTR	800030C2	ERL$VEC244	80004674	ERL$VEC432	800045EC
ERL$GB_BUFTIM	800030C3	ERL$VEC248	80004674	ERL$VEC436	800045F0
ERL$GETFULLNAME	8000CBDE	ERL$VEC252	80004674	ERL$VEC44	80004674
ERL$GL_ERLPID	800030C4	ERL$VEC256	800045BC	ERL$VEC440	800045F4
ERL$GL_SEQUENCE	800030C8	ERL$VEC260	800045C0	ERL$VEC444	800045F8

Symbol	Value	Symbol	Value	Symbol	Value
ERL$VEC448	800045BC	EXE$ALLOCATE	8000A8CE	EXE$CEBREFLCK	8000AF99
ERL$VEC452	800045C0	EXE$ALLOCBUF	8000A775	EXE$CHECKACL	80012000
ERL$VEC456	800045C4	EXE$ALLOCCEB	8000A714	EXE$CHECKACMODE	800121EE
ERL$VEC460	800045C8	EXE$ALLOCIRP	8000A726	EXE$CHECKCLASS	80012227
ERL$VEC464	800045CC	EXE$ALLOCJIB	8000A71D	EXE$CHECKPROT	80012165
ERL$VEC468	800045D0	EXE$ALLOCPCB	8000A72E	EXE$CHECKPROT_16	80012139
ERL$VEC472	800045D4	EXE$ALLOCTQE	8000A737	EXE$CHECK_BYPASS	8001229E
ERL$VEC476	800045D8	EXE$ALLOC_CSD	800057A0	EXE$CHKCREACCES	80010172
ERL$VEC48	80004674	EXE$ALONONPAGED	8000A801	EXE$CHKDELACCES	8001016D
ERL$VEC480	800045DC	EXE$ALONPAGVAR	8000A845	EXE$CHKEXEACCES	8001017C
ERL$VEC484	800045E0	EXE$ALONPAGWAIT	8000A7B0	EXE$CHKIMAGNAME	8001B9BA
ERL$VEC488	800045E4	EXE$ALONPAGWAITS	8000A7A3	EXE$CHKLOGACCES	8001016D
ERL$VEC492	800045E8	EXE$ALOP1IMAG	80010640	EXE$CHKPHYACCES	80010172
ERL$VEC496	800045EC	EXE$ALOP1PROC	80010622	EXE$CHKPRO	80011BAD
ERL$VEC500	800045F0	EXE$ALOPAGED	8000A894	EXE$CHKPRO_INT	80011DB5
ERL$VEC504	800045F4	EXE$ALOPAGWAIT	8000A7C3	EXE$CHKRDACCES	80010177
ERL$VEC508	800045F8	EXE$ALOPHYCNTG	8000AC34	EXE$CHKWAIT2	8000AEB8
ERL$VEC52	80004674	EXE$ALOSHARED	8000AA00	EXE$CHKWRTACCES	80010181
ERL$VEC56	80004674	EXE$ALTQUEPKT	80009B45	EXE$CLEANUP_ORB	800101CE
ERL$VEC60	80004674	EXE$AL_LOAVEC	80003410	EXE$CLI_UTILSRV	80020BF6
ERL$VEC64	80004674	EXE$AL_STACKS	80003F1C	EXE$CLOSE_MSG	8001848B
ERL$VEC68	80004674	EXE$AL_TQENOREPT	80002B20	EXE$CLOSE_RDB	80017B6A
ERL$VEC72	80004674	EXE$ARITH	800046AC	EXE$CLREF	8000B114
ERL$VEC76	80004674	EXE$ASCEFC	80010C02	EXE$CMEXEC	80009303
ERL$VEC8	80004674	EXE$ASCTIM	8001321F	EXE$CMKRNL	80009313
ERL$VEC80	80004674	EXE$ASSIGN	80011111	EXE$CMODEXEC	8000FA58
ERL$VEC84	80004674	EXE$ASTDEL	80009D63	EXE$CMODEXECX	8000FA30
ERL$VEC88	80004674	EXE$ASTFLT	800046B7	EXE$CMODKRNL	8000FC08
ERL$VEC92	80004674	EXE$ASTRET	80009D66	EXE$CMODKRNLX	8000FBE0
ERL$VEC96	80004674	EXE$A_BOOPARAM	800257E0	EXE$CMODSUPR	800046D4
ERL$VEC_RETURN	8000467C	EXE$A_SYSPARAM	80003C00	EXE$CMODUSER	800046EC
ERL$WAKE	8000CC3C	EXE$BINTIM	800132C0	EXE$CNTREG	8001250E
ERL$WARMSTART	8000CB21	EXE$BLDPKTGSR	80009A92	EXE$COMPAT	8000470C
EVT$_AST	00000000	EXE$BLDPKTGSW	80009A9A	EXE$CONTSIGNAL	80017E1A
EVT$_COLPGA	00000000	EXE$BLDPKTSWPR	80009AA2	EXE$CRELNM	800146F1
EVT$_EVENT	00000001	EXE$BLDPKTSWPW	80009AAA	EXE$CRELNT	800146EC
EVT$_FPGA	00000004	EXE$BOOTCB_CHK	80004403	EXE$CRELOG	800144D5
EVT$_PFCOM	00000007	EXE$BRDCST	80011ADC	EXE$CREMBX	80014761
EVT$_RESUME	00000006	EXE$BREAK	800046CC	EXE$CREPRC	800128F5
EVT$_SETPRI	00000008	EXE$BRKTHRU	80011485	EXE$CRETVA	8001244D
EVT$_SWPOUT	00000009	EXE$BUFFRQUOTA	8000A631	EXE$CRE_JGTABLE	80020E9D
EVT$_WAKE	00000005	EXE$BUFQUOPRC	8000A63D	EXE$CRMPSC	800131AA
EXE$$ASCTOID	800176D0	EXE$BUG_CHECK	8000427E	EXE$CSP_BRDCST	800037AC
EXE$$FINISH_RDB	80017795	EXE$BUILDPKTR	80009ABA	EXE$CSP_BRKTHRU	800037BE
EXE$$IDTOASC	800177F5	EXE$BUILDPKTW	80009AB2	EXE$CSP_CALL	800037B2
EXE$$IMGACT	8001C7D4	EXE$CANCEL	800091A4	EXE$CSP_COMMAND	800037B8
EXE$ABORTIO	80009B1E	EXE$CANCELN	80009195	EXE$C_ACCGRNMSK	0000000F
EXE$AB_HEXTAB	800045A9	EXE$CANEXH	80013649	EXE$C_CMSTKSZ	00000014
EXE$ACVIOLAT	80004684	EXE$CANTIM	800092CA	EXE$C_SYSEFN	0000001F
EXE$ADJSTK	80010B6D	EXE$CANWAK	800092E8	EXE$C_SYSPARSZ	000003E4
EXE$ADJWSL	80010A65	EXE$CARRIAGE	8000960E	EXE$DACEFC	80010BDB
EXE$ALLOC	80013BE7	EXE$CATCH_ALL	80020C0D	EXE$DALLOC	80013C8D

Symbol	Value	Symbol	Value	Symbol	Value
EXE$DASSGN	8000AFB9	EXE$FAOL	8001EB70	EXE$GL_GSDGRPBL	80002A8C
EXE$DCLAST	80011422	EXE$FINDACL	800120E7	EXE$GL_GSDGRPFL	80002A88
EXE$DCLCMH	80013606	EXE$FINISHIO	80009B2D	EXE$GL_GSDMTX	80002B4C
EXE$DCLEXH	80013678	EXE$FINISHIOC	80009B2B	EXE$GL_GSDSYSBL	80002A94
EXE$DEALLOCATE	8000A9AA	EXE$FORCEJNL	800037E6	EXE$GL_GSDSYSFL	80002A90
EXE$DEALLOC_CSD	800037A6	EXE$FORCEJNLW	800037EC	EXE$GL_INTSTK	80003F20
EXE$DEANONPAGED	8000A8F9	EXE$FORCEX	80009360	EXE$GL_INTSTKLM	80002C00
EXE$DEANONPGDSIZ	8000A905	EXE$FORK	80009F48	EXE$GL_KFIMTX	80002B88
EXE$DEAP1	800106EA	EXE$FORKDSPTH	80009F9C	EXE$GL_KNOWNFIL	80002B84
EXE$DEAPAGED	8000A95D	EXE$FORK_WAIT	80008A7B	EXE$GL_KNOWN_FILES	80002B64
EXE$DEAPGDSIZ	8000A961	EXE$FRKIPL10DSP	80009F7C	EXE$GL_LOCKRTRY	80003D20
EXE$DEASHARED	8000AA1D	EXE$FRKIPL11DSP	80009F88	EXE$GL_MCHKERPS	80002BDC
EXE$DEASJNL	800037E0	EXE$FRKIPL6DSP	80009F64	EXE$GL_MEMERRS	80002BE0
EXE$DELLNM	800146F6	EXE$FRKIPL8DSP	80009F94	EXE$GL_MP	80002BF8
EXE$DELLOG	80014557	EXE$FRKIPL9DSP	80009F70	EXE$GL_MSGFLAGS	80003D9C
EXE$DELMBX	80014C99	EXE$GB_CPUDATA	80003F88	EXE$GL_NONPAGED	80002BB0
EXE$DELPRC	8000B0C2	EXE$GB_CPUTYPE	80003F98	EXE$GL_NUMNEXUS	800022A0
EXE$DELTVA	8001259F	EXE$GETCHN	800142F9	EXE$GL_PAGED	80002BC0
EXE$DEQ	80014158	EXE$GETDEV	80014303	EXE$GL_PFAILTIM	80002AE8
EXE$DERLMB	80009334	EXE$GETDVI	8001430D	EXE$GL_PFATIM	80002AEC
EXE$DGBLSC	80013DBB	EXE$GETJPI	800166D8	EXE$GL_PGDYNMTX	80002B48
EXE$DLCEFC	80010DF0	EXE$GETLKI	800166DD	EXE$GL_PQBBL	80002A64
EXE$DUMPCPUREG	80003472	EXE$GETMSG	80C1F147	EXE$GL_PQBFL	80002A60
EXE$DVI_FREEBLOCKS	8001B05D	EXE$GETPTI	80014312	EXE$GL_PWRDONE	80002214
EXE$ENQ	80014058	EXE$GETQUI	80014F98	EXE$GL_PWRINTVL	80002218
EXE$EPID_TO_IPID	8000B386	EXE$GETSYI	800166E5	EXE$GL_RPB	80003F74
EXE$EPID_TO_PCB	8000B35D	EXE$GETTIM	800166EA	EXE$GL_RTBITMAP	80002BD0
EXE$ERAPAT	8000B5EE	EXE$GL_ABSTIM	80002AD0	EXE$GL_RTIMESPT	80003D28
EXE$ERAPAT_RTN	800162EC	EXE$GL_ACLMTX	80002B5C	EXE$GL_RUFBASE	800038C0
EXE$ERAPAT_VEC	80003804	EXE$GL_ACMFLAGS	80002C3C	EXE$GL_SAVEDUMP	800029E0
EXE$EXCEPTABLE	80016717	EXE$GL_ARCHFLAG	80003F84	EXE$GL_SCB	80003F80
EXE$EXCEPTION	800047CF	EXE$GL_BLAKHOLE	80002BE8	EXE$GL_SHBLIST	80002BCC
EXE$EXCMSG	80018160	EXE$GL_BOOTCB	800029DC	EXE$GL_SHMGSMTX	80002B50
EXE$EXCPTN	8000FBD2	EXE$GL_BUGCHECK	80000EC8	EXE$GL_SHMMBMTX	80002B54
EXE$EXCPTNE	8000FA0D	EXE$GL_CEBMTA	80002B44	EXE$GL_SITESPEC	80002BFC
EXE$EXIT	80016658	EXE$GL_CJFBASE	800037DC	EXE$GL_SPLITADR	80002BBC
EXE$EXIT_IMAGE	80020C00	EXE$GL_CLITABL	80003D2C	EXE$GL_STATE_FLAGS	80002944
EXE$EXPANDSTK	80017FEA	EXE$GL_CONFREG	80002298	EXE$GL_STATIC_FLAGS	80003D98
EXE$EXPREG	800122D4	EXE$GL_CONFREGL	80002294	EXE$GL_SVAPTE	80002C90
EXE$EXTENDPOOL	8000AA5C	EXE$GL_CPUNODSP	80002290	EXE$GL_SYSID_LOCK	80002B60
EXE$EXTRA1	80003440	EXE$GL_DEFFLAGS	80003D90	EXE$GL_SYSMSG	80002BA8
EXE$EXTRA10	80003592	EXE$GL_DYNAMIC_FLAGS	80003D94	EXE$GL_SYSUCB	800029B4
EXE$EXTRA2	80003448	EXE$GL_ENQMTX	80002B58	EXE$GL_SYSUIC	80003D00
EXE$EXTRA3	80003450	EXE$GL_ERASEPB	800029E4	EXE$GL_SYSWCBBL	80002AAC
EXE$EXTRA4	80003458	EXE$GL_ERASEPPT	800029E8	EXE$GL_SYSWCBFL	80002AA8
EXE$EXTRA5	80003460	EXE$GL_FKWAITBL	80002994	EXE$GL_TENUSEC	80002BF0
EXE$EXTRA6	8000357A	EXE$GL_FKWAITFL	80002990	EXE$GL_TODR	80003C08
EXE$EXTRA7	80003580	EXE$GL_FLAGS	80002940	EXE$GL_TQFL	80002AF0
EXE$EXTRA8	80003586	EXE$GL_FPEXCVEC	80002954	EXE$GL_UBDELAY	80002BF4
EXE$EXTRA9	8000358C	EXE$GL_GPT	80002B90	EXE$GL_USRCHMC	8000295C
EXE$FAILURE	8000FD1A	EXE$GL_GSDDELBL	80002A9C	EXE$GL_USRCHMK	80002958
EXE$FAO	8001EB63	EXE$GL_GSDDELFL	80002A98	EXE$GL_USRUNDWN	80002BAC

Symbol	Value	Symbol	Value	Symbol	Value
EXE$GL_VAXFXCVEC	80002950	EXE$JBCRSP	80015441	EXE$OUTHEX	80025F6C
EXE$GL_WCBDELBL	80002AA4	EXE$KERSTKNV	8000474C	EXE$OUTZSTRING	80025FF5
EXE$GL_WCBDELFL	80002AA0	EXE$LCKPAG	80014363	EXE$PAGRDERR	80004792
EXE$GQ_BOOTCB_D	800029D8	EXE$LCLDSKVALID	800093E9	EXE$POWERAST	80014DBF
EXE$GQ_BOOTTIME	80002AE0	EXE$LDB_SYNCH	8000FDAE	EXE$POWERFAIL	80005064
EXE$GQ_ERLMBX	80002948	EXE$LINK_VEC	80024200	EXE$PRCDELMSG	8001E3A6
EXE$GQ_GBLHOOK1	80002240	EXE$LKWSET	8001435C	EXE$PRCPURMSG	8001E39D
EXE$GQ_GBLHOOK2	80002248	EXE$LNM_SYNTAX_DAT	8000F900	EXE$PROBER	8000A68A
EXE$GQ_GBLHOOK3	80002250	EXE$LOAD_EDISP	800035B7	EXE$PROBER_DSC	80010216
EXE$GQ_GBLHOOK4	80002258	EXE$LOAD_ERROR	8000359C	EXE$PROBEW	8000A6D3
EXE$GQ_GBLHOOK5	80002260	EXE$LOAD_ESPR1	800035B7	EXE$PROBEW_DSC	8001021A
EXE$GQ_GBLHOOK6	80002268	EXE$LOAD_ESPR2	800035BD	EXE$PROCIMGACT	80020B2E
EXE$GQ_GBLHOOK7	80002270	EXE$LOAD_KCJF	8000359E	EXE$PROCSTRT	8000AF3C
EXE$GQ_GBLHOOK8	80002278	EXE$LOAD_KDISP	8000359E	EXE$PURGWS	80014CC8
EXE$GQ_GBLHOOK9	80002280	EXE$LOAD_KRUF	800035A4	EXE$PUTMSG	8001676C
EXE$GQ_GBLHOOKA	80002288	EXE$LOAD_KSPR1	800035AA	EXE$PWRTIMCHK	80004256
EXE$GQ_KFE_LCKNAM	80002B7C	EXE$LOAD_KSPR2	800035B0	EXE$QIO	80009829
EXE$GQ_RIGHTSLIST	80002AB0	EXE$LOAD_NOP	8000359D	EXE$QIOACPPKT	80009B58
EXE$GQ_SYSDISK	8001FDE0	EXE$MAXACMODE	80010207	EXE$QIODRVPKT	80009B41
EXE$GQ_SYSTIME	80002AD8	EXE$MCHECK	80004750	EXE$QIORETURN	80009B77
EXE$GQ_TODCBASE	80003C00	EXE$MCHK	80003410	EXE$QXQPPKT	80009B88
EXE$GRANTID	80017577	EXE$MCHK_BUGCHK	8000485B	EXE$RADRMOD	8000479C
EXE$GT_STARTUP	80003FC4	EXE$MCHK_ERRCNT	80003598	EXE$READ	80009423
EXE$GW_PGFL_FID	80003F9C	EXE$MCHK_PRTCT	80004833	EXE$READCHK	800094E6
EXE$GW_SCANPIX	80002BA6	EXE$MCHK_TEST	8000489D	EXE$READCHKR	800094FA
EXE$HIBER	8000B24F	EXE$MGBLSC	800131B2	EXE$READEF	8000B173
EXE$HWCLKINT	80009FC8	EXE$MNTVERSHDOL	80003568	EXE$READLOCK	8000944F
EXE$IMGDELMSG	8001E29E	EXE$MNTVERSIO	80003562	EXE$READLOCKR	80009462
EXE$IMGDMP_EXEC	80020D4D	EXE$MNTVERSP1	8000356E	EXE$READP_TODR	80003550
EXE$IMGDMP_MERGE	80020D53	EXE$MNTVERSP2	80003574	EXE$READ_TODR	80003532
EXE$IMGFIX	8001DEDF	EXE$MODIFY	8000941D	EXE$REFLECT	80017C76
EXE$IMGPURMSG	8001E295	EXE$MODIFYLOCK	80009455	EXE$REGRESTOR	80003478
EXE$IMGSTA	8001E0CC	EXE$MODIFYLOCKR	80009458	EXE$REGSAVE	8000347E
EXE$INIBOOTADP	8000346C	EXE$MOUNTVER	8000355C	EXE$RESETVEC	80016C06
EXE$INIPROCREG	80003484	EXE$MTACCESS	8000B5F6	EXE$RESTART	80004000
EXE$INIT	8002467C	EXE$MTACCESS_RTN	8001631D	EXE$RESUME	8000B23A
EXE$INIT_DEVICE	800041B6	EXE$MTACCESS_VEC	8000380A	EXE$REVOKID	8001757E
EXE$INIT_TODR	8000353E	EXE$MULTIQUOTA	8000A656	EXE$RMVTIMQ	80008A09
EXE$INI_TIMWAIT	8000354A	EXE$NAMPID	8000B276	EXE$ROPRAND	800047A4
EXE$INSERTIRP	80009BC7	EXE$NETSNDERL	80015882	EXE$RUNDWN	80016987
EXE$INSIOQ	80009BA9	EXE$NULLPROC	80008A89	EXE$SCHDWK	80009656
EXE$INSTIMQ	800089E2	EXE$NUMTIM	800134E4	EXE$SEARCH_RIGHT	80C12072
EXE$INT54	80003418	EXE$ONEPARM	800093DD	EXE$SENDMSG	8001581B
EXE$INT58	80003420	EXE$OPCCUS	80004758	EXE$SENSEMODE	800095E6
EXE$INT5C	80003428	EXE$OPCDEC	80004760	EXE$SETAST	80011400
EXE$INT60	80003430	EXE$OPEN_MSG	800182F9	EXE$SETCHAR	800095A9
EXE$IOFORK	80009F44	EXE$OPEN_RDB	800179AC	EXE$SETEF	8000B197
EXE$IORSNWAIT	80009384	EXE$OPRSNDERL	8001587C	EXE$SETEXV	80014D12
EXE$IPAPBKAST	80009E46	EXE$OUTBLANK	80025F86	EXE$SETIME	8001730E
EXE$IPCONTROL	8000DC4C	EXE$OUTCHAR	80025F8B	EXE$SETIME_INT	80017304
EXE$IPID_TO_EPID	8000B389	EXE$OUTCRLF	80025FE6	EXE$SETIMR	800096B8
EXE$IPID_TO_PCB	8000B364	EXE$OUTCSTRING	80025FF0	EXE$SETMODE	800095C0

Symbol	Value	Symbol	Value	Symbol	Value
EXE$SETOPR	800158D8	EXE$V_EXPLICITS	00000015	FIL$GT_DDDEV	80002988
EXE$SETPFM	80016CA2	EXE$V_FATAL_BUG	0000000A	FIL$GT_DDSTRING	7FFE032C
EXE$SETPRA	80014D88	EXE$V_INIT	00000008	FIL$GT_TOPSYS	800029C6
EXE$SETPRI	80009BE5	EXE$V_JOBQUEUES	0000001B	FIL$INIWCB	8001FD33
EXE$SETPRN	8000B3E6	EXE$V_MOUNTMSG	00000000	FIL$MOUNT	8001F810
EXE$SETPRT	80014E30	EXE$V_MULTACP	0000000B	FIL$OPENFILE	8001F547
EXE$SETPRV	80016FF7	EXE$V_NOAUTOCNF	00000001	FIL$OPENFILE_1	8001F550
EXE$SETRWM	80014D55	EXE$V_NOCLOCK	00000005	FIL$RDCHKFILADR	8001FAF7
EXE$SETSFM	80014D5C	EXE$V_NOCLUSTER	0000000C	FIL$RDWRTLBN	8001FD02
EXE$SETSSF	800170D8	EXE$V_PAGFILDMP	00000019	FIL$READVBN	8001FBC2
EXE$SETSTK	800170F5	EXE$V_PGFLCRIT	00000017	FIL$STATBLK	8001FC55
EXE$SETSWM	80014D6A	EXE$V_PGFLFRAG	00000016	FIL$WRITEVBN	8001FBBB
EXE$SET_RDIPTR	80017BC2	EXE$V_POOLPGING	00000003	GARBAGE_SIZE	00000200
EXE$SHMCEBDEL	8000AF63	EXE$V_REINITQUE	0000001C	GARBCANSIZ	00000080
EXE$SHUTDWNADP	800034A2	EXE$V_RESALLOC	00000011	IAC$AL_IMGACTBUF	7FFE4800
EXE$SIGTORET	80017FC3	EXE$V_SAVEDUMP	0000001A	IAC$AL_VECADDR	7FFE2E58
EXE$SNDACC	80015533	EXE$V_SBIERR	00000007	IAC$AL_VECOPCOD	7FFE2E68
EXE$SNDERR	80015888	EXE$V_SETTIME	00000009	IAC$AW_VECRESET	7FFE2E6C
EXE$SNDEVMSG	800019C8	EXE$V_SHRF1'ACP	0000000F	IAC$AW_VECSET	7FFE2E74
EXE$SNDJBC	80014FA0	EXE$V_SIMULATOR	00000004	IAC$GL_FIRST_ICB	7FFE2E98
EXE$SNDOPR	80015559	EXE$V_SSINHIBIT	00000013	IAC$GL_ICBFL	7FFE2E8C
EXE$SNDSMB	80015546	EXE$V_SYSPAGING	00000002	IAC$GL_IMAGCTX	7FFE2E50
EXE$SNGLEQUOTA	8000A653	EXE$V_SYSUAFALT	0000000E	IAC$GL_IMAGE_LIST	7FFE2E7C
EXE$SRCHANDLER	80017DD1	EXE$V_SYSWRTABL	00000000	IAC$GL_MAIN_ICB	7FFE2E94
EXE$SSFAIL	800047C0	EXE$V_TBCHK	00000018	IAC$GL_PROCCTX	7FFE2E54
EXE$STARTUPADP	8000349C	EXE$V_WRITESYSPARAMS	00000001	IAC$GL_STACK_SIZE	7FFE2E9C
EXE$SUCCESS	8000FD22	EXE$V_XQP_RESIDENT	00000000	IAC$GL_WORK_LIST	7FFE2E84
EXE$SUSPND	8000B1C4	EXE$WAITFR	8000B523	IMG$ALLOCATE_ICB	8001D244
EXE$SWAPINIT	8001E995	EXE$WAKE	8000B265	IMG$DEALLOCATE_ICB	8001D28B
EXE$SWTIMINT	8000A028	EXE$WFLAND	8000B512	IMG$DECODE_IHD	8001D454
EXE$TBIT	C00047AC	EXE$WFLOR	8000B519	IMG$DO_WORK_LIST	8001D77C
EXE$TEST_CSR	8000348A	EXE$WRITE	8000942C	IMG$GET_HEADER	8001CCE2
EXE$TIMEOUT	8000A0FC	EXE$WRITECHK	800094EC	IMG$GET_NEXT_ISD	8001D5F1
EXE$TRNLNM	800146FB	EXE$WRITECHKR	80009558	IMG$IS_IT_MAPPED	8001DF52
EXE$TRNLOG	80014583	EXE$WRITEJNL	800037F2	IMG$OPEN_IMAGE	8001CAB8
EXE$T_ID_UPCASE	800175D0	EXE$WRITEJNLW	800037F8	IMG$PRVSHRIMG	8001E062
EXE$UBCLRINT	80009FD0	EXE$WRITELOCK	80009452	INI$ALLOC_CRB	800253AE
EXE$ULKPAG	8001434C	EXE$WRITELOCKR	80009469	INI$ALONONPAGED	800255C7
EXE$ULWSET	80014345	EXE$WRITEP_TODR	80003556	INI$BRK	8000E95C
EXE$UNWIND	80017147	EXE$WRITE_TODR	80003538	INI$CONSOLE	80003544
EXE$UPCASE_DAT	8000F800	EXE$WRTMAILBOX	80001A1A	INI$MASTERWAKE	800048B8
EXE$UPDSEC	8001594A	EXE$ZEROPARM	800093E3	INI$MPMADP	80003496
EXE$VAL_IDNAME	8001025F	FIL$CACHE_INIT	8001F6BD	INI$RDONLY	8000E965
EXE$V_BUGDUMP	00000010	FIL$CACHE_TRUNC	8001F731	INI$WRITABLE	8000E95E
EXE$V_BUGREBOOT	0000000D	FIL$CVTFIDNAM	8001F46A	IO$GL_SCB_INTO	80002BEC
EXE$V_CONLOAD	0000001D	FIL$CVT_DTB	800165E4	IO$GL_UBA_INTO	80002BE4
EXE$V_CONSYSROU	0000001E	FIL$CVT_HTB	800165F2	IOC$ALLOSPT	8000D27D
EXE$V_CLASS_PROT	00000000	FIL$CVT_OTB	800165EB	IOC$ALOUBAMAP	8000CF9B
EXE$V_CONCEALED	00000012	FIL$C_DIR_SIZE	00000024	IOC$ALOUBAMAPN	8000CF94
EXE$V_CRDENABL	0C000006	FIL$C_SIZE	00000218	IOC$ALOUBAMAPSP	8000D005
EXE$V_DISMOUMSG	00000001	FIL$FINDFILID	8001F895	IOC$ALOUBMAPRM	8000D0AB
EXE$V_EXPLICITP	00000014	FIL$GQ_CACHE	800029D0	IOC$ALOUBMAPRMN	8000D0A4

Symbol	Value	Symbol	Value	Symbol	Value
IOC$ALTREQCOM	8000CD6E	IOC$GL_SRPREM	80002A20	IOC$REQMAPREG	8000CF5F
IOC$APPLYECC	8000D7D8	IOC$GL_SRPSIZE	80002A14	IOC$REQMAPUDA	8000CF4A
IOC$BROADCAST	8000D37D	IOC$GL_SRPSPLIT	80002A1C	IOC$REQPCHANH	8000CD37
IOC$BUFPOST	800049E2	IOC$GL_TU_CDDB	80002A84	IOC$REQPCHANL	8000CD40
IOC$CANCELIO	8000CC5D	IOC$GQ_BRDCST	80002A74	IOC$REQSCHANH	8000CD23
IOC$CHKMBXQUOTA	80010574	IOC$GQ_MOUNTLST	80002A6C	IOC$REQSCHANL	8000CD2D
IOC$CHKUCBQUOTA	80010579	IOC$GW_LAMAPREG	80003D26	IOC$RETURN	8000D236
IOC$CLONE_UCB	8000D800	IOC$GW_MAXBUF	80003D06	IOC$SCAN_IODB	8000D3EE
IOC$COPY_UCB	8000DB26	IOC$GW_MBXBFQUO	80003D08	IOC$SCAN_IODB_2P	8000D433
IOC$CREATE_UCB	80010605	IOC$GW_MBXMXMSG	80003D0A	IOC$SEARCH	800102F6
IOC$CREDIT_UCB	800105C1	IOC$GW_MBXNMMSG	80003D0C	IOC$SEARCHALL	800102F0
IOC$CTRLINIT	8000D48D	IOC$GW_MVTIMEOUT	80003D04	IOC$SEARCHCONT	8000D702
IOC$CVTLOGPHY	8000D84C	IOC$GW_XFMXRATE	80003D24	IOC$SEARCHDEV	800102EA
IOC$CVTLOGPHYU	8000D855	IOC$INITBUFWIND	8000C88E	IOC$SEARCHINT	8000D60E
IOC$CVT_DEVNAM	8000D2A4	IOC$INITDRV	80025BA4	IOC$SENSEDISK	8000D96A
IOC$DALLOC_DEV	800104BE	IOC$INITIATE	8000CE31	IOC$SEVER_UCB	8000DBF4
IOC$DEBIT_UCB	800105A1	IOC$IOPOST	800048C4	IOC$TESTUNIT	8000D711
IOC$DELETE_UCB	8000DBE4	IOC$LAST_CHAN	8000CC7B	IOC$THREADCRB	80003652
IOC$DIAGBUFILL	8000CCB1	IOC$LAST_CHAN_AMBX	8000CC74	IOC$TRANDEVNAM	80010338
IOC$DIRPOST1	80004F66	IOC$LINK_UCB	8000DBB4	IOC$UNITINIT	8000D4C7
IOC$DISMOUNT	8000FE8E	IOC$LOADMBAMAP	8000D975	IOC$UNLOCK	8001052A
IOC$FFCHAN	800102A6	IOC$LOADUBAMAP	8000D9D6	IOC$UNLOCK_DEV	800104E6
IOC$FILSPT	8000C8C5	IOC$LOADUBAMAPA	8000D9B4	IOC$UPDATRANSP	8000D92E
IOC$FREE_UCB	8000DC16	IOC$LOADUBAMAPN	8000DA5D	IOC$VERIFYCHAN	8001053D
IOC$GETBYTE	8000C86C	IOC$LOCK_DEV	8001040D	IOC$WAKACP	80004DD8
IOC$GL_ADPLIST	80000ED0	IOC$LUBAUDAMAP	8000DA9C	IOC$WFIKPCH	8000D237
IOC$GL_AQBLIST	80002A68	IOC$MAPVBLK	8000D877	IOC$WFIRLCH	8000D259
IOC$GL_CRBTMOUT	80002A7C	IOC$MNTVER	8000CE28	IPL$_SCS	00000008
IOC$GL_DEVLIST	80000F48	IOC$MOVFRUSER	8000C895	IPL$_SYNCH	00000008
IOC$GL_DPTLIST	80000ED4	IOC$MOVFRUSER1	8000C8A6	KFI$GL_F11AACP	80002B8C
IOC$GL_DU_CDDB	80002A80	IOC$MOVFRUSER2	8000C899	KILL_CDB	00001558-R
IOC$GL_IRPBL	800029FC	IOC$MOVTOUSER	8000C8AD	LCK$BREAK_DEADLOCK	8000BA6F
IOC$GL_IRPCNT	80002A04	IOC$MOVTOUSER1	8000C8BE	LCK$CANCEL_CVT	8000C4F9
IOC$GL_IRPFL	800029F8	IOC$MOVTOUSER2	8000C8B1	LCK$CHECK_RSB	8000C729
IOC$GL_IRPMIN	80002A08	IOC$PARSDEVNAM	8000D4FA	LCK$COMPAT_TBL	8000BB24
IOC$GL_IRPREM	80002A00	IOC$PTETOPFN	8000DA49	LCK$COMP_GGMODE	8000C3CE
IOC$GL_LRPBL	80002A2C	IOC$PURGDATAP	80003490	LCK$CVTNOTQED	8000BCB8
IOC$GL_LRPCNT	80002A40	IOC$PUTBYTE	8000C87D	LCK$CVT_GRANTED	8000BC8F
IOC$GL_LRPFL	80002A28	IOC$QNXTSEG	80004D2F	LCK$CVT_ID_TO_LKB	80003704
IOC$GL_LRPMIN	80002A34	IOC$QNXTSEG1	80004D3B	LCK$DEALLOC_RSB	8000C742
IOC$GL_LRPREM	80002A3C	IOC$REINITDRV	80025BAA	LCK$DEQLOCK	8000C596
IOC$GL_LRPSIZE	80002A30	IOC$RELCHAN	8000CCE0	LCK$DLCKEXIT	8000BB14
IOC$GL_LRPSPLIT	80002A38	IOC$RELDATAP	8000CEE9	LCK$EXTEND_IDTBL	8000C7BB
IOC$GL_MUTEX	80002B40	IOC$RELDATAPUDA	8000CEDE	LCK$EXTEND_IDTBLW	8000C7C4
IOC$GL_PFKBINT	80002A5C	IOC$RELMAPREG	8000D170	LCK$GB_HTBLSHFT	80002C38
IOC$GL_POOLFKB	80002A44	IOC$RELMAPUDA	8000D155	LCK$GB_MAXDEPTH	80002C39
IOC$GL_PSBL	800029F4	IOC$RELOC_DDT	80025940	LCK$GB_REBLD_STATE	80002C3B
IOC$GL_PSFL	800029F0	IOC$RELSCHAN	8000CCD6	LCK$GB_STALLREQS	80002C3A
IOC$GL_SRPBL	80002A10	IOC$REQLOM	8000CD99	LCK$GL_DIRVEC	80002C20
IOC$GL_SRPCNT	80002A24	IOC$REQDATAP	8000CE5E	LCK$GL_EXTRASTK	80003C2C
IOC$GL_SRPFL	80002A0C	IOC$REQDATAPHW	8000CE70	LCK$GL_HASHTBL	80002C10
IOC$GL_SRPMIN	80002A18	IOC$REQDATAPUDA	8000CE7E	LCK$GL_HTBLCNT	80002C14

Symbol	Value	Symbol	Value	Symbol	Value
LCK$GL_HTBLSIZ	80003D38	LKI$SEARCH_BLOCKING	800CE7B2	MMG$ALLOCPAGFIL2	80006EDA
LCK$GL_IDTBL	80002C04	LKI$SND_BLKBY	800037D0	MMG$ALLOCPFN	800067EC
LCK$GL_IDTBLMAX	80003D34	LKI$SND_BLKING	800037CA	MMG$ALLOCSWPAREA	80006D39
LCK$GL_IDTBLSIZ	80003D30	LKI$SND_LOCKS	800037D6	MMG$ALOSHMGSD	80015C46
LCK$GL_MAXID	80002C0C	LKI$SND_STDREQ	800037C4	MMG$ALOSHMPAG	80015B3A
LCK$GL_NXTID	80002C08	LNM$AL_DIRTBL	800029A4	MMG$AL_BEGDRIVE	80001950
LCK$GL_PRCMAP	80002C24	LNM$AL_HASHTBL	80002998	MMG$AL_ENDDRIVE	80001E68
LCK$GL_TIMOUTQ	80002C18	LNM$AL_MUTEX	800029B0	MMG$AL_FIXUPTBL	80025648
LCK$GL_WAITTIME	80003D3C	LNM$CHECK_PROT	80018A33	MMG$AL_PGDCODEN	80021EB0
LCK$GQ_BITMAP_EXP	80002C28	LNM$CONTSEARCH	80018DCC	MMG$AL_SYSPCB	80002B00
LCK$GQ_BITMAP_EXPLCL	80002C30	LNM$DELBLK	80018A94	MMG$A_ENDVEC	80000A00
LCK$GRANTCVTS	8000C3F1	LNM$DELETE_LNMB	80018AC1	MMG$A_SYSPARAM	80003C00
LCK$GRANTWTRS	8000C43A	LNM$DELETE_TAB	80018B01	MMG$A_SYS_END	80028B00
LCK$GRANT_LOCK	8000C12A	LNM$FIRSTTAB	80018D86	MMG$CALCSDAPSIZE	80007059
LCK$GRANT_LOCK_ALT	8000C12F	LNM$GL_HTBLSIZP	80003D8C	MMG$CEFTRNLOG	80015E21
LCK$GRANT_REM	8000C17C	LNM$GL_HTBLSIZS	80003D88	MMG$CLR_BITMAP	80015ADB
LCK$HASH_SEARCH	8000C0BB	LNM$HASH	80018E41	MMG$CREDEL	8001260D
LCK$LOCAL_CVT	8000BC33	LNM$INIT_PROT	80018B12	MMG$CREPAG	800126DA
LCK$LOCAL_LOCK	8000BEF6	LNM$INSLOGTAB	80018B5A	MMG$CRETVA	80012445
LCK$NORET_VALBLK	8000C01A	LNM$LOCKR	80018F75	MMG$DALCBAKSTORE	80006962
LCK$NOT_QUEUED	8000C02C	LNM$LOCKW	80018F7D	MMG$DALCPAGFIL	80006F65
LCK$QUEUECVT	8000C2CD	LNM$PRESEARCH	80018DA2	MMG$DALCSTX	80010916
LCK$QUEUED_EXIT	8000C017	LNM$PROBER	80018F52	MMG$DALCSTXSCN	80010893
LCK$QUEUEWAIT	8000C2E7	LNM$SEARCHLOG	80018C7E	MMG$DALCSTXSCN1	8001088C
LCK$QUEUE_BLKAST	8000C397	LNM$SEARCH_ONE	80018D0F	MMG$DALLOCPFN	800069C6
LCK$QUEUE_BLOCKAST	8000C34C	LNM$SETUP	80018EC1	MMG$DEALLOCPAGFIL	80006F6E
LCK$QUEUE_REM	8000C2EC	LNM$SYSTEM_DIRECTORY	800030E0	MMG$DECPHDREF	80006788
LCK$REGRANTLOCK	8000C142	LNM$TABLE	80018ED1	MMG$DECPHDREF1	8000678C
LCK$RET_VALBLK	8000BFF7	LNM$UNLOCK	80018F85	MMG$DECPTREF	80006721
LCK$SEARCHDLCK	8000B720	LNM_SYSTEM_DIR_LNMTH	8000310B	MMG$DECSECREF	8000709C
LCK$SND_BLKING	800036E0	MA$INITIAL	800034B4	MMG$DECSHMREF	80008674
LCK$SND_CVTREQ	800036BC	MA$RAVAIL	800034A8	MMG$DELCONPFN	80006801
LCK$SND_DEQCV	800036D4	MA$REQUEST	800034AE	MMG$DELGBLSEC	80007BF2
LCK$SND_DEQGR	800036CE	MARK_CMD_DONE	00000C18-R	MMG$DELGBLWCB	80013FE7
LCK$SND_DEQWT	800036DA	MAX_LOC_PORTS	00000002	MMG$DELPAG	80007868
LCK$SND_DLCKFND	800036F8	MAX_NODES	00000010	MMG$DELPFNLST	800068FA
LCK$SND_GRANTED	800036C8	MB$DDT	80001950	MMG$DELSHMGS	80016244
LCK$SND_LOCKREQ	800036C2	MB$DPT	8000183C	MMG$DELWSLEPPG	80006611
LCK$SND_REDO_SRCH	800036FE	MB$GL_DDB	800013AC	MMG$DELWSLEX	800065ED
LCK$SND_RMVDIR	800036E6	MB$GL_ORB1	80001564	MMG$DGBLSC1	80013DB4
LCK$SND_SRCHDLCK	800036F2	MB$GL_ORB2	80001648	MMG$EXPKSTK	80007FE7
LCK$SND_TIMESTAMP_RQST	800036EC	MB$GL_UCB1	800014D4	MMG$EXPREG	80012206
LCK$SRCH_HSHTBL	8000C103	MB$GL_UCB2	800015B8	MMG$EXTRADYNWS	800076D9
LCK$SRCH_RESDLCK	8000B882	MB$ORB0	800013F0	MMG$FAST_CREATE	800123A3
LCK$SYNCCVT_TBL	8000BB2A	MB$UCB0	80001444	MMG$FINDTSTGSD	80015D3D
LCK$SYNC_EXIT	8000BFE2	MCHK$GL_MASK	80002BD4	MMG$FINDGSDPFN	800085FE
LIB$GB_OPINFO	00000000	MCHK$GL_SP	80002BD8	MMG$FINDGSNOTRN	80016143
LIB$GB_OPINFO1	80023190	MMG$ALCPHD	80010956	MMG$FINDSHB	80015D65
LIB$GB_OPINFO2	80023990	MMG$ALCSTX	8001022D	MMG$FINDSHD	80006711
LIB$INS_DECODE	80022AF6	MMG$ALC_PGFLVBN	800165A7	MMG$FREEGSD	80015CD1
LIB$_ATTCONSTO	0015827C	MMG$ALLOCONTIG	80006A4C	MMG$FREWSLE	80006467
LKI$SEARCH_BLOCKEDBY	8000E853	MMG$ALLOCPAGFIL1	80006D71	MMG$FREWSLX	800064FD

Symbol	Value	Symbol	Value	Symbol	Value
MMG$FRE_TRYSKIP	8000652A	MMG$INSPFNH	8000698A	MMG$WRITE_GSD	80015F7A
MMG$FRSTRONLY	80004000	MMG$INSPFNT	800069C8	MMG$WRTMFYPAG	80007316
MMG$GB_FREWFLGS	8000219A	MMG$IN_REGION	800124A5	MMG$WRTPGSBAK	80008306
MMG$GETGSNAM	80015D9A	MMG$IOLOCK	80006AAB	MMG$WSLEPFN	8000639A
MMG$GETNXTGSD	8000869A	MMG$IOLOCKPAG	80006BB0	MMG$WSPEAKCHK	80010B42
MMG$GETPTIPAG	80007D02	MMG$LCKULKPAG	80007DCD	MNT$CATCH_RST	00000007
MMG$GL_CTLBASVA	80003F18	MMG$LOCKPGTB	800066DD	MNT$CATCH_STR	00000008
MMG$GL_FPEMUL_BASE	800022A8	MMG$MAKEWSLE	80006679	MNT$MAINT_READ	00000005
MMG$GL_FRESVA	80003F30	MMG$MBXTRNLOG	80015E2A	MNT$MAINT_STATE	00000002
MMG$GL_GBLPAGFIL	800022B8	MMG$MOVPTLOCK	80007728	MNT$MAINT_WRITE	00000006
MMG$GL_GBLSECFND	800022B4	MMG$MOVPTLOCK1	80007727	MNT$NORMAL_STATE	00000004
MMG$GL_GPTBASE	80003F24	MMG$MPWCHECK	800089BF	MNT$POLLER_OFF	00000000
MMG$GL_GPTE	80003F28	MMG$M_NOLASTUPD	00000001	MNT$POLLER_ON	00000001
MMG$GL_IRPNEXT	80003F58	MMG$M_NOWAIT	00000002	MNT$UNINIT_STATE	00000003
MMG$GL_LRPNEXT	80003F5C	MMG$PAGEFAULT	80005C58	MPH$ASTDELCONT	80009CB2
MMG$GL_MAXGPTE	80003F2C	MMG$PAGETYPE	80007B70	MPH$ASTDELHK	80009CAC
MMG$GL_MAXMEM	80003F70	MMG$PGFLTWAIT	80005F9F	MPH$BUGCHKHK	80004362
MMG$GL_MAXPFIDX	8000220C	MMG$PTEADRCHK	80007197	MPH$INVALIDHK	80006522
MMG$GL_MAXPFN	80003F68	MMG$PTEINDX	800071CB	MPH$NEWLVLHK	80009F2C
MMG$GL_MAXSYSVA	80003F30	MMG$PTEINDXCHK	800071C5	MPH$QAST	80009E6B
MMG$GL_MINPFN	80003F6C	MMG$PTEREF	800071A0	MPH$QEMPTYCONT	80009CC6
MMG$GL_NPAGEDYN	80003F50	MMG$PURGWSSCN	800080A9	MPH$RESCHED	8000A5A4
MMG$GL_NPAGNEXT	80003F54	MMG$READ_GSD	80015F83	MPH$SCHED	8000A5CA
MMG$GL_NULLPFL	800021E4	MMG$REFCNTNEG	80006D31	MPW$AL_PTE	80001EF0
MMG$GL_PAGEDYN	80003F64	MMG$RELPFN	80006902	MPW$AW_INITVAL	80003C84
MMG$GL_PAGSWPVC	80002208	MMG$REMPFN	8000688E	MPW$AW_PHVINDEX	80001EF4
MMG$GL_PFNLOCK	80001EFC	MMG$REMPFNH	80006883	MPW$GB_PRIO	80003C8A
MMG$GL_PGDCOD	80003FE4	MMG$RESRCWAIT	80005F84	MPW$GL_BADPAGTOTAL	80001EF8
MMG$GL_PHYPGCNT	80003C98	MMG$RETADRINI	8001256C	MPW$GL_THRESH	80003C8C
MMG$GL_RMSBASE	800022A4	MMG$RETRANGE	80012588	MPW$GL_WAITLIM	80003C90
MMG$GL_SBICONF	8000229C	MMG$RET_BYT_QUOTA	8001A7F4	MPW$GW_HILIM	80003C86
MMG$GL_SBR	80003F4C	MMG$RLPFNSAVPTE	800068E5	MPW$GW_LOLIM	80003C88
MMG$GL_SPTBASE	80003F34	MMG$SCNWSLX	80007FAE	MPW$GW_MPWPFC	80003C84
MMG$GL_SPTLEN	80003F38	MMG$SETPRTPAG	80008188	MSG$_TRMBRDCST	00000053
MMG$GL_SRPNEXT	80003F60	MMG$SET_BITMAP	80015AD2	MT$CHECK_ACCESS	80009C84
MMG$GL_SYSLOA_BASE	800022AC	MMG$SHMTXLK	800161E6	NET$WCB	800017C4
MMG$GL_SYSPHD	80003F3C	MMG$SHMTXULK	80016223	NL$DDT	80001DF4
MMG$GL_SYSPHDLN	80003F40	MMG$SHRCNTNEG	80006D35	NL$DPT	80001875
MMG$GL_VAXEMUL_BASE	800022B0	MMG$SHRINKWS	80007650	NL$GL_DDB	8000169C
MMG$GSDMTXULK	80013D9B	MMG$SUBSECREF	8000708F	NL$GL_ORB0	800016E0
MMG$GSDSCN	80013ECD	MMG$SVAPTECHK	8000717E	NL$GL_UCB0	80001734
MMG$GSDTRNLOG	80015E33	MMG$SVPCTX	80005F56	NO$GL_DPT	80000EE0
MMG$GW_BIGPFN	80003F9A	MMG$SWAPWSLE	80007F72	NSA$ARGLST_IMGNAM	8001658B
MMG$GW_MINPFIDX	80002210	MMG$TRY_ALL	80012353	NSA$EVENT_AUDIT	800163BA
MMG$IMGACTBUF	7FFE4800	MMG$ULKGBLWSLE	80007DBC	NSA$GR_ALARMVEC	80002C68
MMG$IMGHDRBUF	7FFE6400	MMG$UNIQUEGSD	80016183	NSA$GR_JOURNVEC	80002C40
MMG$IMGRESET	80010746	MMG$UNLOCK	80006CDE	NSA$T_IDT	7FFE2800
MMG$INADRINI	8001255D	MMG$UPDSECAST	80015A5F	OP$DPT	800018AE
MMG$INCPTREF	800066E3	MMG$VALIDATEGSD	80008696	OPA$CRB	80001330
MMG$INCSHMREF	80008677	MMG$VFYSECFLG	80013D27	OPA$GL_DDB	80001124
MMG$INIBLDPKT	80006797	MMG$V_NOLASTUPD	00000000	OPA$ORB0	80001168
MMG$ININEWPFN	80006637	MMG$V_NOWAIT	00000001	OPA$UCB0	800011BC

Symbol	Value	Symbol	Value	Symbol	Value
OPA$VECTOR	80001913	PA_CQ0_M_CQC	00000001	PA_PMC_M_RSVD	00000020
P1SYSVECTORS	7FFEDE00	PA_CQ0_V_CQC	00000000	PA_PMC_M_UNIN	00000080
PAT$A_NONPGD_CODE	8000E9B2	PA_CQ1	0000090C	PA_PMC_M_WP	00000010
PAT$A_NONPGD_CODE_END	8000EC00	PA_CQ1_M_CQC	00000001	PA_PMC_M_XBPE	00000100
PAT$A_NONPGD_DATA	80003810	PA_CQ1_V_CQC	00000000	PA_PMC_M_XMPE	00000800
PAT$A_PAGED_CODE	8001851C	PA_CQ2	00000910	PA_PMC_V_CSPE	0000000E
PAT$A_PFN_FIXUP	800257D8	PA_CQ2_M_CQC	00000001	PA_PMC_V_IPE	0000000A
PAT$GL_EXP_NPG1	8000E9B2	PA_CQ2_V_CQC	00000000	PA_PMC_V_LSPE	0000000D
PAT$GL_EXP_NPG2	80003FE4	PA_CQ3	00000914	PA_PMC_V_MIE	00000002
PA_CNF	00000000	PA_CQ3_M_CQC	00000001	PA_PMC_V_MIF	00000003
PA_CNF_M_ADPTYP	000000FF	PA_CQ3_V_CQC	00000000	PA_PMC_V_MIN	00000000
PA_CNF_M_CIBPE	00008000	PA_C_DCACHESZ	00000003	PA_PMC_V_MTD	00000001
PA_CNF_M_CRD	00010000	PA_C_LSINDX	00000800	PA_PMC_V_OPE	00000009
PA_CNF_M_CTO	00002000	PA_C_LSLENGTH	00000800	PA_PMC_V_PE	0000000F
PA_CNF_M_CXTER	00040000	PA_C_MCACHESZ	00000003	PA_PMC_V_PSA	00000006
PA_CNF_M_CXTMO	00100000	PA_C_UCODEST	00000400	PA_PMC_V_RBPE	0000000C
PA_CNF_M_MAINT	00004000	PA_C_WCSSIZ	00000C00	PA_PMC_V_RSVD	00000005
PA_CNF_M_MXTFLT	08000000	PA_DFQ	00000928	PA_PMC_V_UNIN	00000007
PA_CNF_M_NOCI	00001000	PA_DFQ_M_DFQC	00000001	PA_PMC_V_WP	00000004
PA_CNF_M_PARFLT	80000000	PA_DFQ_V_DFQC	00000000	PA_PMC_V_XBPE	00000008
PA_CNF_M_PDN	00800000	PA_MADR	00000014	PA_PMC_V_XMPE	0000000B
PA_CNF_M_PFD	00000100	PA_MDATR	00000018	PA_PPR	00000940
PA_CNF_M_PUP	00400000	PA_MFQ	0000092C	PA_PPR_M_BUFLEN	0FFF0000
PA_CNF_M_RDS	00020000	PA_MFQ_M_MFQC	00000001	PA_PPR_M_MNODE	10000000
PA_CNF_M_RDTO	00080000	PA_MFQ_V_MFQC	00000000	PA_PPR_M_PN	000000FF
PA_CNF_M_TDEAD	00000200	PA_MTC	00000930	PA_PPR_S_BUFLEN	0000000C
PA_CNF_M_TFAIL	00000400	PA_MTC_M_MTC	00000001	PA_PPR_S_PN	00000008
PA_CNF_M_URDFLT	20000000	PA_MTC_V_MTC	00000000	PA_PPR_V_BUFLEN	00000010
PA_CNF_M_WSQFLT	40000000	PA_MTEC	00000934	PA_PPR_V_MNODE	0000001C
PA_CNF_M_XMTFLT	04000000	PA_PDC	00000920	PA_PPR_V_PN	00000000
PA_CNF_S_ADPTYP	00000008	PA_PDC_M_PDC	00000001	PA_PQBBR	000009C4
PA_CNF_V_ADPTYP	00000000	PA_PDC_V_PDC	00000000	PA_PS	00000900
PA_CNF_V_CIBPE	0000000F	PA_PEC	0000091C	PA_PSR	00000918
PA_CNF_V_CRD	00000010	PA_PEC_M_PEC	00000001	PA_PSR_M_PSC	00000001
PA_CNF_V_CTO	0000000D	PA_PEC_V_PEC	00000000	PA_PSR_V_PSC	00000000
PA_CNF_V_CXTER	00000012	PA_PESR	0000093C	PA_PS_M_DSE	00000010
PA_CNF_V_CXTMO	00000014	PA_PFAR	00000938	PA_PS_M_MFQE	00000002
PA_CNF_V_MAINT	0000000E	PA_PIC	00000924	PA_PS_M_MSE	00000020
PA_CNF_V_MXTFLT	0000001B	PA_PIC_M_PIC	00000001	PA_PS_M_MTE	80000000
PA_CNF_V_NOCI	0000000C	PA_PIC_V_PIC	00000000	PA_PS_M_PDC	00000004
PA_CNF_V_PARFLT	0000001F	PA_PMC	00000004	PA_PS_M_PIC	00000008
PA_CNF_V_PDN	00000017	PA_PMC_M_CSPE	00004000	PA_PS_M_RQA	00000001
PA_CNF_V_PFD	00000008	PA_PMC_M_IPE	00000400	PA_PS_M_SE	00000040
PA_CNF_V_PUP	00000016	PA_PMC_M_LSPE	00002000	PA_PS_V_DSE	00000004
PA_CNF_V_RDS	00000011	PA_PMC_M_MIE	00000004	PA_PS_V_MFQE	00000001
PA_CNF_V_RDTO	00000013	PA_PMC_M_MIF	00000008	PA_PS_V_MSE	00000005
PA_CNF_V_TDEAD	00000009	PA_PMC_M_MIN	00000001	PA_PS_V_MTE	0000001F
PA_CNF_V_TFAIL	0000000A	PA_PMC_M_MTD	00000002	PA_PS_V_PDC	00000002
PA_CNF_V_URDFLT	0000001D	PA_PMC_M_OPE	00000200	PA_PS_V_PIC	00000003
PA_CNF_V_WSQFLT	0000001E	PA_PMC_M_PE	00008000	PA_PS_V_RQA	00000000
PA_CNF_V_XMTFLT	0000001A	PA_PMC_M_PSA	00000040	PA_PS_V_SE	00000006
PA_CQ0	00000908	PA_PMC_M_RBPE	00001000	PB$B_CBC_STS	00000028

Symbol	Value	Symbol	Value	Symbol	Value
PB$B_PO_STS	00000029	PB$V_MAINT	00000000	PDT$L_DGOVRHD	000000B8
PB$B_P1_STS	0000002A	PB$V_PORT_TYP	00000000	PDT$L_DQELOGOUT	000002E0
PB$B_PROTOCOL	00000048	PB$V_STATE	00000001	PDT$L_FLINK	00000000
PB$B_RSTATE	00000021	PB$V_TIM	00000C00	PDT$L_GPTBASE	0000022C
PB$B_RSTATION	0000000C	PB$W_RETRY	00000022	PDT$L_GPTLEN	00000230
PB$B_RST_PORT	00000020	PB$W_SIZE	00000008	PDT$L_LBDG	00000184
PB$B_SUBTYP	0C00000B	PB$W_STATE	00000012	PDT$L_MAINTFCN	00000078
PB$B_TYPE	0000000A	PB$W_STS	00000044	PDT$L_MAP	0000002C
PB$C_CI750	00000002	PB$W_VCFAIL_RSN	00000046	PDT$L_MAPBYPASS	00000030
PB$C_CI780	00000002	PDT$B_DQIMAP	00000154	PDT$L_MAPIRP	00000034
PB$C_CINT	00000007	PDT$B_HSHUT_DG	000001B0	PDT$L_MAPIRPBYP	00000038
PB$C_CLOSED	00000000	PDT$B_MAX_PORT	0000017C	PDT$L_MAXBCNT	000000BC
PB$C_DISAB	00000001	PDT$B_NXT_PORT	0000017E	PDT$L_MFQ	00000100
PB$C_ENAB	00000002	PDT$B_PO_LBSTS	00000180	PDT$L_MFQHDR	0000020C
PB$C_HSC	00000004	PDT$B_P1_LBSTS	00000181	PDT$L_MQELOGOUT	00000320
PB$C_KL10	00000006	PDT$B_PDT_TYPE	00000007	PDT$L_MRESET	00000070
PB$C_LENGTH	00000054	PDT$B_PLOGMAP	00000134	PDT$L_MSGHDRSZ	000000B4
PB$C_NI	00000008	PDT$B_PORTMAP	00000114	PDT$L_MSTART	00000074
PB$C_OPEN	00000003	PDT$B_PORT_NUM	0000017D	PDT$L_MTC	00000104
PB$C_PS	00000009	PDT$B_REQIDPS	0000017F	PDT$L_PFAR	00000108
PB$C_PWR_FAIL	00004000	PDT$B_SUBTYP	0000000B	PDT$L_PMC	000000E8
PB$C_ST_REC	00000002	PDT$B_TYPE	0000000A	PDT$L_POLLERDUE	0000018C
PB$C_ST_SENT	00000001	PDT$C_HSHUT_SIZ	00000014	PDT$L_POLLSWEEP	000000D8
PB$C_UNINIT	00000000	PDT$C_LENGTH	000000E4	PDT$L_POOLDUE	00000188
PB$C_VC_FAIL	00008000	PDT$C_PA	00000001	PDT$L_PPR	0000010C
PB$K_LENGTH	00000054	PDT$C_PAREGBASE	000000E4	PDT$L_PS	000000EC
PB$L_BLINK	00000004	PDT$C_PAREGEND	00000110	PDT$L_PSR	000000F8
PB$L_CDTLST	00000034	PDT$C_PE	00000003	PDT$L_QUEUEDG	0000003C
PB$L_DUETIME	0000003C	PDT$C_PS	00000004	PDT$L_QUEUEMDGS	00000040
PB$L_FLINK	00000000	PDT$C_PU	00000002	PDT$L_RCHMSGBUF	00000044
PB$L_PDT	0000002C	PDT$C_SCSBASE	0000000C	PDT$L_RCLMSGBUF	00000048
PB$L_RPORT_FCN	0000001C	PDT$C_SCSEND	00000084	PDT$L_READCOUNT	00000068
PB$L_RPORT_REV	00000018	PDT$K_LENGTH	000000E4	PDT$L_REJECT	0000004C
PB$L_RPORT_TYP	00000014	PDT$K_SCSBASE	0000000C	PDT$L_REQDATA	00000050
PB$L_SBLINK	00000030	PDT$K_SCSEND	00000084	PDT$L_RLSCOUNT	0000006C
PB$L_SCSMSG	00000040	PDT$L_ACCEPT	0000000C	PDT$L_SENDDATA	00000054
PB$L_WAITQBL	0000003C	PDT$L_ADP	000000E0	PDT$L_SENDDG	00000058
PB$L_WAITQFL	00000038	PDT$L_ALLOCDG	00000010	PDT$L_SENDMSG	0000005C
PB$M_CUR_CBL	00000001	PDT$L_ALLOCMSG	00000014	PDT$L_SENDRGDG	0000007C
PB$M_CUR_PS	00000001	PDT$L_CNF	000000E4	PDT$L_SNDCNTMSG	00000060
PB$M_DUALPATH	80000000	PDT$L_CNTCDRP	000000D4	PDT$L_SPTBASE	00000224
PB$M_MAINT	00000C01	PDT$L_CONNECT	00000018	PDT$L_SPTLEN	00000228
PB$M_TIM	00000001	PDT$L_CQ0	000000F0	PDT$L_STOP_VCS	00000080
PB$S_LPORT_NAME	00000004	PDT$L_CQ1	000000F4	PDT$L_UCBO	000000DC
PB$S_PBDEF	00000054	PDT$L_DCONNECT	00000028	PDT$L_UNMAP	00000064
PB$S_PORT_TYP	0000001F	PDT$L_DEALLOCDG	0000001C	PDT$L_VBDT	0000021C
PB$S_RSTATION	00000006	PDT$L_DEALLOMSG	00000C20	PDT$L_VPQB	00000218
PB$S_STATE	00000002	PDT$L_DEALRGMSG	00000024	PDT$L_WAITQBL	000000B0
PB$T_LPORT_NAME	00000024	PDT$L_DFQ	000000FC	PDT$L_WAITQFL	000000AC
PB$V_CUR_CBL	00000000	PDT$L_DFQHDR	00000208	PDT$M_CNTBSY	00000001
PB$V_CUR_PS	00000000	PDT$L_DGHDRSZ	00000190	PDT$M_CNTRLS	00000002
PB$V_DUALPATH	0000001F	PDT$L_DGNETHD	00000194	PDT$M_CUR_LBS	00000001

Symbol	Value	Symbol	Value	Symbol	Value
PDT$M_LBDG	00000004	PFN$AX_WSLX	80003FB4	PMS$GL_DEPLOCPK	800032B8
PDT$M_PRV_LBS	00000002	PFN$A_BASE	80003FA4	PMS$GL_DEQ_IN	800032E4
PDT$M_PUP	00000002	PFN$C_LONG_LEN	00000016	PMS$GL_DEQ_LOC	800032E0
PDT$M_PWF_CLNUP	00000001	PFN$C_WORD_LEN	00000012	PMS$GL_DEQ_OUT	800032E8
PDT$M_SNGLHOST	00000001	PFN$GB_LENGTH	80003F99	PMS$GL_DIRDATA_HIT	80000B4C
PDT$M_X_LBS	00000004	PFN$GL_PHYPGCNT	80001E8C	PMS$GL_DIRDATA_MISS	80000B50
PDT$Q_COMQ2	000001F0	PHV$GL_PIXBAS	80002238	PMS$GL_DIRHIT	80000B24
PDT$Q_COMQ3	000001F8	PHV$GL_REFCBAS	8000223C	PMS$GL_DIRIO	80003290
PDT$Q_COMQBASE	000001E0	PIO$AL_RMSEXH	7FFE0314	PMS$GL_DIRMISS	80000B28
PDT$Q_COMQH	000001E8	PIO$A_DIRCACHE	7FFE0800	PMS$GL_DIR_IN	80003300
PDT$Q_COMQL	000001E0	PIO$A_TRACE	7FFE0600	PMS$GL_DIR_OUT	80003304
PDT$Q_DFREEQ	000001D0	PIO$GB_DFMBC	7FFE0224	PMS$GL_DLCKFND	80003314
PDT$Q_FORMPB	00000174	PIO$GB_DFMBFHSH	7FFE022A	PMS$GL_DLCKMSGS_IN	80003308
PDT$Q_MFREEQ	000001D8	PIO$GB_DFMBFIDX	7FFE0229	PMS$GL_DLCKMSGS_OUT	8000330C
PDT$Q_RSPQ	00000200	PIO$GB_DFMBFREL	7FFE0228	PMS$GL_DLCKSRCH	80003310
PDT$Q_TEMP_RSPQ	00000198	PIO$GB_DFMBFSDK	7FFE0225	PMS$GL_DOSTATS	8000340C
PDT$S_CNTOWNER	00000010	PIO$GB_DFMBFSMT	7FFE0226	PMS$GL_DPTSCN	80001EE8
PDT$S_PDTDEF	000000E4	PIO$GB_DFMBFSUR	7FFE0227	PMS$GL_DZROFLTS	80001EC4
PDT$T_CNTOWNER	000000C4	PIO$GB_DFNBC	7FFE022B	PMS$GL_ENQCVT_IN	800032D8
PDT$V_CNTBSY	00000000	PIO$GB_RMSPROLOG	7FFE022C	PMS$GL_ENQCVT_LOC	800032D4
PDT$V_CNTRLS	00000001	PIO$GL_DIRCACHE	7FFE0230	PMS$GL_ENQCVT_OUT	800032DC
PDT$V_CUR_LBS	00000000	PIO$GL_DIRCFRLH	7FFE0238	PMS$GL_ENQNEW_IN	800032CC
PDT$V_LBDG	00000002	PIO$GL_FMLH	7FFE0200	PMS$GL_ENQNEW_LOC	800032C8
PDT$V_PRV_LBS	00000001	PIO$GL_IIOFSPLH	7FFE0208	PMS$GL_ENQNEW_OUT	800032D0
PDT$V_PUP	00000001	PIO$GL_NXTIRBSEQ	7FFE0240	PMS$GL_ENQNOTQD	800032F0
PDT$V_PWF_CLNUP	00000000	PIO$GL_RULOCK	7FFE023C	PMS$GL_ENQWAIT	800032EC
PDT$V_SNGLHOST	00000000	PIO$GQ_IIODEFAULT	7FFE0324	PMS$GL_ERASEIO	80000B64
PDT$V_X_LBS	00000002	PIO$GT_DDSTRING	7FFE032C	PMS$GL_EXTHIT	80000B3C
PDT$W_BDTLEN	00000220	PIO$GT_ENDSTR	7FFE0212	PMS$GL_EXTMISS	80000B40
PDT$W_DQELEN	00000210	PIO$GW_DFPROT	7FFE0222	PMS$GL_FAULTS	80001EB0
PDT$W_FLAGS	000000C0	PIO$GW_IIOIMPA	7FFE0270	PMS$GL_FCP	80000A00
PDT$W_LPORT_STS	00000110	PIO$GW_PIOIMPA	7FFE0248	PMS$GL_FCP2	80000A00
PDT$W_MQELEN	00000214	PIO$GW_RMSEXTEND	7FFE022D	PMS$GL_FIDHIT	80000B34
PDT$W_PBCOUNT	00000112	PIO$GW_STATUS	7FFE0210	PMS$GL_FIDMISS	80000B38
PDT$W_PORTCHAR	00000004	PIO$S_EODSTR	00000010	PMS$GL_FILHDR_HIT	80000B44
PDT$W_SIZE	00000008	PMS$ABORT_RQ	8000442A	PMS$GL_FILHDR_MISS	80000B48
PFM$GETBUF	80016F24	PMS$AL_READTBL	80003380	PMS$GL_GVALID	80001EEC
PFM$MON	8000B480	PMS$AL_TRANSFLT	80001EC8	PMS$GL_HIT	80000B20
PFM$PURGE	80016EA2	PMS$AL_WRITETBL	800033AC	PMS$GL_IOPFMPDB	800032AC
PFN$AB_STATE	80003FBC	PMS$END_IO	80004445	PMS$GL_IOPFMSEQ	800032B0
PFN$AB_TYPE	80003FC0	PMS$END_RQ	80004465	PMS$GL_JNLBUFIO	80003334
PFN$AL_BAK	80003FA8	PMS$GB_PROMPT	80003408	PMS$GL_JNLBUFWR	80003344
PFN$AL_HEAD	80001E68	PMS$GL_ACCLCK	80000B78	PMS$GL_JNLCHNLS	8000331C
PFN$AL_HILIMIT	80001E90	PMS$GL_ARRLOCPK	800032B4	PMS$GL_JNLDIRIO	80003330
PFN$AL_LOLIMIT	80001E9C	PMS$GL_ARRTRAPK	800032BC	PMS$GL_JNLFORFL	80003340
PFN$AL_MFYLSTHD	80001E6C	PMS$GL_BLK_IN	800032F8	PMS$GL_JNLFORNL	8000333C
PFN$AL_PTE	80003FA4	PMS$GL_BLK_LOC	800032F4	PMS$GL_JNLJRNLS	80003318
PFN$AW_REFCNT	80003FAC	PMS$GL_BLK_OUT	800032FC	PMS$GL_JNLWRTAI	80003320
PFN$AW_SWPVBN	80003FB8	PMS$GL_BUFIO	80003294	PMS$GL_JNLWRTAT	80003328
PFN$AX_BLINK	80003FB4	PMS$GL_CHME	80003374	PMS$GL_JNLWRTBI	80003324
PFN$AX_FLINK	80003FB0	PMS$GL_CHMK	80003370	PMS$GL_JNLWRTFM	80003348
PFN$AX_SHRCNT	80003FB0	PMS$GL_COMPAT	80002ACC	PMS$GL_JNLWRTRU	8000332C

Symbol	Value	Symbol	Value	Symbol	Value
PMS$GL_JNLWRTSS	80003338	PMS$START_RQ	800044B7	PPD$C_RDCNT	0000001A
PMS$GL_KERNEL	80002AB8	PPD$B_DEF_ST	0000001C	PPD$C_REQDAT	00000008
PMS$GL_LDPCTX	80003400	PPD$B_FLAGS	0000000F	PPD$C_REQDAT0	00000008
PMS$GL_LOGNAM	80003298	PPD$B_HWVERS	00000034	PPD$C_REQDAT1	0000C009
PMS$GL_LRGRWP	800033E8	PPD$B_LBDATA	00000012	PPD$C_REQDAT2	0000000A
PMS$GL_MBREADS	8000329C	PPD$B_LCB_0	00000012	PPD$C_REQID	00000005
PMS$GL_MBWRITES	800032A0	PPD$B_LCB_LPORT	00000010	PPD$C_REQMDAT	0000000E
PMS$GL_NOSTDTRM	800033F0	PPD$B_LCB_NPORT	0000000F	PPD$C_RETCNF	00000003
PMS$GL_OPEN	80000B5C	PPD$B_LCB_OPC	00000011	PPD$C_RETDAT	00000011
PMS$GL_OPENS	80000B60	PPD$B_LCB_PORT	0000000E	PPD$C_SCS_DG	00000003
PMS$GL_PAGES	80003378	PPD$B_OPC	0000000E	PPD$C_SCS_MSG	00000004
PMS$GL_PASSALL	800033E0	PPD$B_PORT	0000000C	PPD$C_SETCKT	00000019
PMS$GL_PREADIO	80001EB8	PPD$B_PROTOCOL	0000001A	PPD$C_SNDDAT	00000010
PMS$GL_PREADS	80001EB4	PPD$B_RSTATE	00000025	PPD$C_SNDDG	00000001
PMS$GL_PWRITES	80001EBC	PPD$B_RST_PORT	00000024	PPD$C_SNDLB	0000000D
PMS$GL_PWRITIO	80001EC0	PPD$B_STATUS	0000000D	PPD$C_SNDMDAT	00000012
PMS$GL_QUOHIT	80000B2C	PPD$B_SWFLAG	0000000B	PPD$C_SNDMSG	00000002
PMS$GL_QUOMISS	80000B30	PPD$B_SYSTEMID	00000014	PPD$C_SNDRST	00000006
PMS$GL_RCVBUFFL	800032C4	PPD$B_TYPE	0000000A	PPD$C_SNDSTRT	00000007
PMS$GL_RDFLTS	80001EB4	PPD$C_ACK	00000002	PPD$C_STABO	00000004
PMS$GL_READCNT	800033D8	PPD$C_ACK_LEN	00000004	PPD$C_STACK	00000001
PMS$GL_RUFABORT	8000336C	PPD$C_CACHECLR	00008000	PPD$C_STACK_LEN	0000003E
PMS$GL_RUFACTIV	8000334C	PPD$C_CACHE_LEN	00000002	PPD$C_START	00000000
PMS$GL_RUFCHNLS	80003354	PPD$C_CNFREC	00000023	PPD$C_START_LEN	0000003E
PMS$GL_RUFJNLS	80003350	PPD$C_DATREC	00000031	PPD$C_STINVDP	00000002
PMS$GL_RUFMARK	80003364	PPD$C_DGREC	00000021	PPD$C_STPSV	00000000
PMS$GL_RUFMRKRB	80003368	PPD$C_DISAB	00000001	PPD$C_STURC	00000003
PMS$GL_RUFREADS	8000335C	PPD$C_ELOG	00000005	PPD$C_STURP	00000001
PMS$GL_RUFWRTS	80003358	PPD$C_ENAB	00000002	PPD$C_TYPACCV	00000004
PMS$GL_RUFXTNDS	80003360	PPD$C_HOSTSHUT	00000006	PPD$C_TYPBLV	00000003
PMS$GL_RWP	800033E4	PPD$C_HSHUT_LEN	00000002	PPD$C_TYPBMSE	00000006
PMS$GL_RWPNOSTD	800033F4	PPD$C_IDREC	0000002B	PPD$C_TYPINVBN	00000002
PMS$GL_RWPSUM	800033EC	PPD$C_INVTC	00000018	PPD$C_TYPNP	00000005
PMS$GL_SPLIT	80000B1C	PPD$C_LBDAT_LEN	00000030	PPD$C_TYPOK	00000000
PMS$GL_STORAGMAP_HIT	80000B54	PPD$C_LBREC	0000002D	PPD$C_TYPOTHER	00000007
PMS$GL_STORAGMAP_MISS	80000B58	PPD$C_LB_LENGTH	00000046	PPD$C_TYPVCC	00000001
PMS$GL_SWITCH	80003404	PPD$C_LCB_DATA	00000013	PPD$C_UNINIT	00000000
PMS$GL_SYNCHLCK	80000B70	PPD$C_LENGTH	00000012	PPD$C_VCDCL	00000006
PMS$GL_SYNCHWAIT	80000B74	PPD$C_MCNFREC	00000029	PPD$K_LB_LENGTH	00000046
PMS$GL_TRCNGLOS	800032C0	PPD$C_MDATREC	00000033	PPD$K_LENGTH	00000012
PMS$GL_TREADS	800032A4	PPD$C_MIN_DGSIZ	00000050	PPD$L_BLINK	00000004
PMS$GL_TTY_CODE1	800033F8	PPD$C_MSGREC	00000022	PPD$L_DG_DISC	00000028
PMS$GL_TTY_CODE2	800033FC	PPD$C_OSEQ	00000005	PPD$L_FLINK	00000000
PMS$GL_TURN	80000B18	PPD$C_PRT_BASE	00000000	PPD$L_IN_VCD	00000018
PMS$GL_TWRITES	800032A8	PPD$C_PRT_ELOG	00000001	PPD$L_LBCRC	00000042
PMS$GL_VOLLCK	80000B68	PPD$C_PSAUTO	00000000	PPD$L_PO_ACK	00000010
PMS$GL_VOLWAIT	80000B6C	PPD$C_PSP0	00000001	PPD$L_PO_NAK	00C00014
PMS$GL_WRTCNT	800033DC	PPD$C_PSP1	00000002	PPD$L_PO_NRSP	00000018
PMS$GL_XQPCACHEWAIT	80000B7C	PPD$C_PTHACK	00000000	PPD$L_P1_ACK	0000001C
PMS$GW_BATCH	8000337C	PPD$C_PTHARB	000C0003	PPD$L_P1_NAK	00000020
PMS$GW_INTJOBS	8000337E	PPD$C_PTHNAK	00000001	PPD$L_P1_NRSP	00000024
PMS$START_IO	80004494	PPD$C_PTHNO_RSP	00000002	PPD$L_REC_BOFF	00000028

Symbol	Value	Symbol	Value	Symbol	Value
PPD$L_REC_NAME	00000024	PPD$V_NS	0000000D	PQL$GMWSEXTENT	80003E54
PPD$L_RPORT_FCN	00000020	PPD$V_P	00000007	PQL$GMWSQUOTA	80003E48
PPD$L_RPORT_REV	0000001C	PPD$V_PORT_TYP	00000000	PR$_IPL	00000012
PPD$L_RPORT_TYP	00000018	PPD$V_PS	00000001	PR$_TBIA	00000039
PPD$L_SND_BOFF	00000020	PPD$V_PSTS	00000009	PRT$C_UREW	0000000D
PPD$L_SND_NAME	0000001C	PPD$V_PTH0	00000001	PRV$V_DIAGNOSE	00000006
PPD$L_ST_ADDR	00000018	PPD$V_PTH1	00000003	RESPONDER_NAME	000000B8-R
PPD$L_XCT_LEN	00000018	PPD$V_RP	00000001	RE_QUEUE_DG	000013D0-R
PPD$M_CST	00008000	PPD$V_RSP	00000000	RM$DIRCACHE_BLKAST	8000AF4D
PPD$M_DISPOSE	00000001	PPD$V_SP	00000004	RM$RESET	800109F0
PPD$M_DQI	00001000	PPD$V_STATE	00000001	RM$SET	80010A31
PPD$M_DS	00000001	PPD$V_STSST	00000001	RMS$GL_SFDBASE	80002BC8
PPD$M_DSTART	00000080	PPD$V_STSTYP	00000005	RMS$GW_GBLBUFQUO	80000B80
PPD$M_DUALPATH	80000000	PPD$W_LCB_LEN7	0000000C	RMS$_STALL	00018001
PPD$M_ERR	00000001	PPD$W_LENGTH	00000010	RMS$_STR	000187BC
PPD$M_FORCE	00000080	PPD$W_MASK	00000010	RND$V_IACLOCK	00000000
PPD$M_MAINT	00000001	PPD$W_MAXDG	0000001C	SB$B_ENBMSK	0000005A
PPD$M_NR	00004000	PPD$W_MAXMSG	0000001E	SB$B_HWVERS	00000038
PPD$M_NS	00002000	PPD$W_MTYPE	00000012	SB$B_SUBTYP	0000000B
PPD$M_P	00000080	PPD$W_M_VAL	00000014	SB$B_SYSTEMID	00000018
PPD$M_POSTS	00000006	PPD$W_SIZE	00000008	SB$B_TYPE	0000000A
PPD$M_PISTS	00000018	PQL$AB_FLAG	80003E5F	SB$C_LENGTH	00000060
PPD$M_RSP	00000001	PQL$AB_SYSPQL	8001E94E	SB$K_LENGTH	00000060
PPD$Q_CURTIME	00000048	PQL$AL_DEFAULT	80003DE4	SB$L_BLINK	00000004
PPD$Q_NODENAME	00000040	PQL$AL_MIN	80003E20	SB$L_CSB	0000005C
PPD$Q_SWINCARN	00000028	PQL$C_SYSPQLLEN	00000046	SB$L_DDB	00000054
PPD$Q_XCT_ID	00000010	PQL$GDASTLM	80003DE8	SB$L_FLINK	00000000
PPD$S_M	00000003	PQL$GDBIOLM	80003DEC	SB$L_PBBL	00000010
PPD$S_PORT_TYP	0000001F	PQL$GDBYTLM	80003DF0	SB$L_PBCONNX	00000014
PPD$S_PS	00000002	PQL$GDCPULM	80003DF4	SB$L_PBFL	0000000C
PPD$S_PSTS	00000002	PQL$GDDIOLM	80003DF8	SB$Q_SWINCARN	0000002C
PPD$S_PTH0	00000002	PQL$GDENQLM	80003E14	SB$S_ENBMSK	00000002
PPD$S_PTH1	00000002	PQL$GDFILLM	80003DFC	SB$S_HWTYPE	00000004
PPD$S_RP	00000002	PQL$GDJTQUOTA	80003E1C	SB$S_HWVERS	0000000C
PPD$S_SP	00000002	PQL$GDPGFLQUOTA	80003E00	SB$S_NODENAME	00000010
PPD$S_STATE	00000002	PQL$GDPRCLM	80003E04	SB$S_SBDEF	00000060
PPD$S_STSST	00000004	PQL$GDTQELM	80003E08	SB$S_SWINCARN	00000008
PPD$S_STSTYP	00000003	PQL$GDWSDEFAULT	80003E10	SB$S_SWTYPE	00000004
PPD$T_HWTYPE	00000030	PQL$GDWSEXTENT	80003E18	SB$S_SWVERS	00000004
PPD$T_SWTYPE	00000020	PQL$GDWSQUOTA	80003E0C	SB$S_SYSTEMID	00000006
PPD$T_SWVERS	00000024	PQL$GMASTLM	80003E24	SB$T_HWTYPE	00000034
PPD$V_CST	0000000F	PQL$GMBIOLM	80003E28	SB$T_NODENAME	00000044
PPD$V_DISPOSE	00000000	PQL$GMBYTLM	80003E2C	SB$T_SWTYPE	00000024
PPD$V_DQI	0000000C	PQL$GMCPULM	80003E30	SB$T_SWVERS	00000028
PPD$V_DS	00000000	PQL$GMDIOLM	80003E34	SB$W_MAXDG	00000020
PPD$V_DSTART	00000007	PQL$GMENQLM	80003E50	SB$W_MAXMSG	00000022
PPD$V_DUALPATH	0000001F	PQL$GMFILLM	80003E38	SB$W_SIZE	00000008
PPD$V_ERR	00000000	PQL$GMJTQUOTA	80003E58	SB$W_TIMEOUT	00000058
PPD$V_FORCE	00000007	PQL$GMPGFLQUOTA	80003E3C	SCH$AQ_COMH	80001F08
PPD$V_M	00000004	PQL$GMPRCLM	80003E40	SCH$AQ_COMOH	80002008
PPD$V_MAINT	00000000	PQL$GMTQELM	80003E44	SCH$AQ_COMOT	8000200C
PPD$V_NR	0000000E	PQL$GMWSDEFAULT	80003E4C	SCH$AQ_COMT	80001F0C

Symbol	Value	Symbol	Value	Symbol	Value
SCH$AQ_WQHDR	800020FC	SCH$GQ_SUSPO	80002174	SCS$C_ACCP_REQ	00000002
SCH$ASTDEL	80009C98	SCH$GW_AWSMIN	80003CB0	SCS$C_ACCP_REQL	00000042
SCH$CHSE	8000A305	SCH$GW_CEBCNT	800021AC	SCS$C_ACCP_RSP	00000003
SCH$CHSEP	8000A321	SCH$GW_DELPHDCT	800021AE	SCS$C_ACCP_RSPL	00000012
SCH$CLREF	8000B165	SCH$GW_DORMANTWAIT	80003CC6	SCS$C_APPL_BASE	00000000
SCH$CLREFR	8000B15F	SCH$GW_IOTA	80003CC2	SCS$C_APPL_DG	0000000B
SCH$C_MAXPIX	0000003F	SCH$GW_LOCALNODE	80002934	SCS$C_APPL_MSG	0000000A
SCH$C_SWPPIX	00000001	SCH$GW_LONGWAIT	80003CC4	SCS$C_CON_BASE	00000004
SCH$FORCEDEXIT	8000A40A	SCH$GW_PROCCNT	8000219C	SCS$C_CON_REQ	00000000
SCH$GB_PRI	800021D8	SCH$GW_PROCLIM	8000219E	SCS$C_CON_REQL	00000042
SCH$GB_RESCAN	80002199	SCH$GW_QUAN	80003C82	SCS$C_CON_RSP	00000001
SCH$GB_SIP	80002198	SCH$GW_SWPFAIL	80003CC8	SCS$C_CON_RSPL	00000012
SCH$GETEFC	8000B124	SCH$GW_SWPFCNT	80002236	SCS$C_CR_REQ	00000008
SCH$GL_AWSTIME	80003CB4	SCH$IOLOCKR	8000ACE1	SCS$C_CR_REQL	00000012
SCH$GL_BORROWLIM	80003D1C	SCH$IOLOCKW	8000ACC7	SCS$C_CR_RSP	00000009
SCH$GL_COMQQS	80002194	SCH$IOUNLOCK	8000AD55	SCS$C_CR_RSPL	0000000E
SCH$GL_COMQS	80002190	SCH$LOCKR	8000ACE8	SCS$C_DISC_REQ	00000006
SCH$GL_CURPCB	8000218C	SCH$LOCKW	8000ACCE	SCS$C_DISC_REQL	00000012
SCH$GL_FREECNT	80001E80	SCH$LOCKWNOWAIT	8000ACAF	SCS$C_DISC_RSP	00000007
SCH$GL_FREELIM	80001E9C	SCH$NEWLVL	80009F09	SCS$C_DISC_RSPL	0000000E
SCH$GL_FREEREQ	80001E90	SCH$OSWPSCHED	8000878A	SCS$C_OVHD	0000000E
SCH$GL_GROWLIM	80003D18	SCH$POSTEF	8000ADCF	SCS$C_REJ_REQ	00000004
SCH$GL_MAXPIX	80002928	SCH$QAST	80009E6B	SCS$C_REJ_REQL	00000012
SCH$GL_MFYCNT	80001E84	SCH$QEND	8000A3A7	SCS$C_REJ_RSP	00000005
SCH$GL_MFYLIM	80001E94	SCH$RAVAIL	8000AD45	SCS$C_REJ_RSPL	0000000E
SCH$GL_MFYLIMSV	80001EA8	SCH$REMOVACB	80009F40	SCS$C_STDISC	00000019
SCH$GL_MFYLOLIM	80001EA0	SCH$RESCHED	8000A5A4	SCS$C_STINSFCR	00000021
SCH$GL_MFYLOSV	80001EAC	SCH$RSE	8000A27D	SCS$C_STNOMAT	0000000A
SCH$GL_NULLPCB	80002440	SCH$RWAIT	8000ACA5	SCS$C_STNORMAL	00000001
SCH$GL_PCBVEC	80002920	SCH$SCHED	8000A5CA	SCS$C_STNORS	00000012
SCH$GL_PFRATH	80003CA0	SCH$SWAPACBS	80009F3D	SCS$DEALL_CDT	80003628
SCH$GL_PFRATL	80003C9C	SCH$SWPWAKE	8000A532	SCS$DEALL_RSPID	8000362E
SCH$GL_PFRATS	80003CA4	SCH$UNLOCK	8000AD5C	SCS$DIR_LOOKUP	80003682
SCH$GL_PIXLAST	8000292C	SCH$UNWAIT	8000A2CA	SCS$DISCONNECT	80003634
SCH$GL_PIXWIDTH	80002930	SCH$V_MPW	00000002	SCS$ENTER	8000363A
SCH$GL_RESMASK	800021D4	SCH$V_REORD	00000000	SCS$FIND_RDTE	80003676
SCH$GL_SEQVEC	80002924	SCH$V_SIP	00000000	SCS$GA_DFLTMSK	800035F8
SCH$GL_SWPPCB	800026E0	SCH$WAIT	8000B594	SCS$GA_EXISTS	800035FC
SCH$GL_SWPPID	80002740	SCH$WAITK	8000B59E	SCS$GA_LOCALSB	80000EF4
SCH$GL_SWPRATE	80003CB8	SCH$WAITL	8000B5A9	SCS$GB_NODENAME	80003D54
SCH$GL_WSDEC	80003CAC	SCH$WAITM	8000B5AA	SCS$GB_PAMXPORT	80003D63
SCH$GL_WSINC	80003CA8	SCH$WAKE	8000A50D	SCS$GB_PANOPOLL	80003D69
SCH$GQ_CEBHD	800021A4	SCS$ACCEPT	80003604	SCS$GB_PANPOLL	80003D62
SCH$GQ_COLPGWQ	80002108	SCS$ALLOC_CDT	8000360A	SCS$GB_PASANITY	80003D68
SCH$GQ_FPGWQ	80002180	SCS$ALLOC_RSPID	80003610	SCS$GB_SYSTEMID	80003D4C
SCH$GQ_HIBOWQ	8000215C	SCS$AL_CDAVEC	80003604	SCS$GB_SYSTEMIDH	80003D50
SCH$GQ_HIBWQ	80002150	SCS$B_CON_DAT	00000024	SCS$GB_UDABURST	80003D86
SCH$GQ_LEFOWQ	80002144	SCS$B_PPD	FFFFFFE0	SCS$GL_BDT	800035DC
SCH$GQ_LEFWQ	80002138	SCS$CANCEL_MBX	800036A0	SCS$GL_CDL	800035E0
SCH$GQ_MWAIT	80002114	SCS$CONFIG_PTH	80003616	SCS$GL_MCADR	800035EC
SCH$GQ_PFWQ	8000212C	SCS$CONFIG_SYS	8000361C	SCS$GL_MCLEN	800035E8
SCH$GQ_SUSP	80002168	SCS$CONNECT	80003622	SCS$GL_MSCP	800035F0

Symbol	Value	Symbol	Value	Symbol	Value
SCS$GL_PDT	800035F4	SCS$T_DST_PROC	00000004	SGN$GL_LRPCNTV	80003C5C
SCS$GL_RDT	800035E4	SCS$T_SRC_PROC	00000014	SGN$GL_LRPMIN	80003C64
SCS$GQ_CONFIG	800035C4	SCS$UNSTALLUCB	8000365E	SGN$GL_LRPSIZE	80003C60
SCS$GQ_DIRECT	800035CC	SCS$W_CREDIT	FFFFFFF6	SGN$GL_MAXGPGCT	80003C14
SCS$GQ_POLL	800035D4	SCS$W_LENGTH	FFFFFFF0	SGN$GL_MAXVPGCT	80003C4C
SCS$GW_BDTCNT	80003D40	SCS$W_MIN_CR	00000000	SGN$GL_MAXWSCNT	80003C3C
SCS$GW_CDTCNT	80003D42	SCS$W_MTYPE	FFFFFFF4	SGN$GL_NPAGEDYN	80003C40
SCS$GW_FLOWCUSH	80003D4A	SCS$W_STATUS	00000002	SGN$GL_NPAGEVIR	80003C44
SCS$GW_MAXDG	80003D46	SEC$V_RESIDENT	0000000D	SGN$GL_P1LWCNT	80003F0C
SCS$GW_MAXMSG	80003D48	SEC$W_FLAGS	00000014	SGN$GL_PAGEDYN	80003C48
SCS$GW_NEXTBIT	800035FA	SET_SEQ_NUM	000013AE-R	SGN$GL_PE1	80003D6C
SCS$GW_PAPOLINT	80003D64	SGN$C_BALSETCNT	00000018	SGN$GL_PE2	80003D70
SCS$GW_PAPOOLIN	80003D66	SGN$C_DFWSCNT	00000064	SGN$GL_PE3	80003D74
SCS$GW_PAPPDDG	80003D60	SGN$C_DFWSQUOTA	00000078	SGN$GL_PE4	80003D78
SCS$GW_PASTMOUT	80003D5E	SGN$C_GBLSECCNT	00000028	SGN$GL_PE5	80003D7C
SCS$GW_PRCPOLINT	80003D5C	SGN$C_MAXGPGCNT	00000800	SGN$GL_PE6	80003D80
SCS$GW_RDTCNT	80003D44	SGN$C_MAXPAGCNT	00004000	SGN$GL_PHDAPCNT	80003F04
SCS$K_APPL_BASE	00000000	SGN$C_MAXPGFL	00001000	SGN$GL_PHDLWCNT	80003F08
SCS$K_CON_BASE	00000004	SGN$C_MAXPSTCNT	00000005	SGN$GL_PHDPAGCT	80003F10
SCS$K_STDISC	00000019	SGN$C_MAXVPGCNT	00002000	SGN$GL_PTPAGCNT	80003F14
SCS$K_STINSFCR	00000021	SGN$C_MAXWSCNT	00000400	SGN$GL_SPTREQ	80003C5C
SCS$K_STNOMAT	0000000A	SGN$C_MINWSCNT	0000000A	SGN$GL_SRPCNT	80003C68
SCS$K_STNORMAL	00000001	SGN$C_NPAGEDYN	00006800	SGN$GL_SRPCNTV	80003C6C
SCS$K_STNORS	00000012	SGN$C_NPROCS	00000040	SGN$GL_SRPMIN	80003C74
SCS$LISTEN	80003640	SGN$C_PAGFDYN	00004000	SGN$GL_SRPSIZE	80003C70
SCS$LKP_MSGWAIT	8000367C	SGN$C_PFNPATSIZ	00000010	SGN$GL_USER3	80003CF4
SCS$LKP_RDTCDRP	80003664	SGN$C_PHYPAGCNT	00001000	SGN$GL_USER4	80003CF8
SCS$LKP_RDTWAIT	8000366A	SGN$C_SYSDWSCNT	00000028	SGN$GL_USERD1	80003CEC
SCS$LOCLOOKUP	80003646	SGN$C_SYSVECPGS	00000005	SGN$GL_USERD2	80003CF0
SCS$L_DST_CONID	FFFFFFF8	SGN$C_SYSWSCNT	00000060	SGN$GL_VMS5	80003CDC
SCS$L_LCONID	FFFFFFF0	SGN$GB_KFILSTCT	80003C10	SGN$GL_VMS6	80003CE0
SCS$L_REC_BOFF	00000008	SGN$GB_PGTBPFC	80003C0E	SGN$GL_VMS7	80003CE4
SCS$L_REC_NAME	00000004	SGN$GB_STARTUP_P1	80003ECC	SGN$GL_VMS8	80003CE8
SCS$L_RSPID	FFFFFFF4	SGN$GB_STARTUP_P2	80003ED0	SGN$GL_VMSD1	80003CCC
SCS$L_SND_BOFF	00000000	SGN$GB_STARTUP_P3	80003ED4	SGN$GL_VMSD2	80003CD0
SCS$L_SND_NAME	FFFFFFFC	SGN$GB_STARTUP_P4	80003ED8	SGN$GL_VMSD3	80003CD4
SCS$L_SRC_CONID	FFFFFFFC	SGN$GB_STARTUP_P5	80003EDC	SGN$GL_VMSD4	80003CD8
SCS$L_XCT_LEN	FFFFFFF8	SGN$GB_STARTUP_P6	80003EE0	SGN$GW_CTLIMGLIM	80003C7E
SCS$NEW_SB	80003688	SGN$GB_STARTUP_P7	80003EE4	SGN$GW_CTLPAGES	80003C7C
SCS$POLL_MBX	8000369A	SGN$GB_STARTUP_P8	80003EE8	SGN$GW_DFPFC	80003C0C
SCS$POLL_MODE	80003694	SGN$GB_SYSPFC	80003C0F	SGN$GW_DFWSCNT	80003E10
SCS$POLL_PROC	8000368E	SGN$GB_TAILORED	80003ECA	SGN$GW_GBLSECNT	80003C12
SCS$RECY_RSPID	80003670	SGN$GL_BALSETCT	80003C30	SGN$GW_IMGIOCNT	80003C80
SCS$REMOVE	8000364C	SGN$GL_EXTRACPU	80003CFC	SGN$GW_ISPPGCT	80003C2A
SCS$RESUMEWAITR	80003658	SGN$GL_EXUSRSTK	80003C54	SGN$GW_MAXPRCCT	80003C1C
SCS$SHUTDOWN	800036A6	SGN$GL_FREEGOAL	80003D14	SGN$GW_MAXPSTCT	80003C20
SCS$S_CON_DAT	00000010	SGN$GL_FREELIM	80003D10	SGN$GW_MINWSCNT	80003C22
SCS$S_DST_PROC	00000010	SGN$GL_GBLPAGFIL	80003C18	SGN$GW_PAGFILCT	80003C24
SCS$S_PPD	00000010	SGN$GL_IRPCNT	80003C34	SGN$GW_PCHANCNT	80003C78
SCS$S_SCSDEF	00000054	SGN$GL_IRPCNTV	80003C38	SGN$GW_PIOPAGES	80003C7A
SCS$S_SCSDEF1	0000001C	SGN$GL_LOADFLAGS	80003DA0	SGN$GW_PIXSCAN	80003C1E
SCS$S_SRC_PROC	00000010	SGN$GL_LRPCNT	80003C58	SGN$GW_SWPFILCT	80002210

Symbol	Value	Symbol	Value	Symbol	Value
SGN$GW_SWPFILES	80003C26	SWP$GB_PRIO	80003C8B	SYS$DACEFC	7FFEDED0
SGN$GW_SYSDWSCT	80003C28	SWP$GB_SHLP1PT	80003EF6	SYS$DALLOC	7FFEDED8
SGN$GW_TPWAIT	80003D84	SWP$GL_BALBASE	80003F44	SYS$DASSGN	7FFEDEE0
SGN$GW_WSLMXSKP	80003C94	SWP$GL_BALSPT	80003F48	SYS$DCLAST	7FFEDEE8
SGN$V_LOADCHKPRT	00000001	SWP$GL_BSLOTSZ	80003EF8	SYS$DELLNM	7FFEE488
SGN$V_LOADERAPAT	00000000	SWP$GL_HISWPCNT	800021D0	SYS$DELMBX	7FFEDF00
SGN$V_LOADMTACCESS	00000002	SWP$GL_HOSWPCNT	800021CC	SYS$DELPRC	7FFEDF08
SIO_ARMPATH	00000E8B-R	SWP$GL_INPCB	800021B4	SYS$DELTVA	7FFEDF10
SIO_CARRIER	0000109A-R	SWP$GL_ISPAGCNT	800021B8	SYS$DEQ	7FFEE3C8
SIO_COMPLETE	00000786-R	SWP$GL_ISWPCNT	800021C4	SYS$DISCONNECT	7FFEE1D0
SIO_COMPLETE_ERR	0000078B-R	SWP$GL_ISWPPAGES	800021C0	SYS$ENQ	7FFEE3C0
SIO_INITCINT	00000E70-R	SWP$GL_MAP	80003EFC	SYS$ENQW	7FFEE3D0
SIO_LRGPACKET	00001094-R	SWP$GL_OSWPCNT	800021C8	SYS$EXIT	7FFEDF40
SIO_MAINT_STATE	000010DD-R	SWP$GL_PHDBASVA	80003F00	SYS$EXPREG	7FFEDF48
SIO_NAK	0000102D-R	SWP$GL_SHELIO	8000A62D	SYS$FAO	7FFEDF50
SIO_NOACK	00001056-R	SWP$GL_SHELL	800021B0	SYS$FAOL	7FFEDF58
SIO_NORSP	00001004-R	SWP$GL_SHELLBAS	80021000	SYS$FORCEX	7FFEDF60
SIO_NXT_CMD	00000709-R	SWP$GL_SHELLSIZ	80003EEC	SYS$GB_BRK_LIM	80003E9F
SIO_OVERSIZEPKT	00001071-R	SWP$GL_SLOTCNT	800021A0	SYS$GB_DEFPRI	80003E88
SIO_POLLER	0000CE44-R	SWP$GL_SWPPGCNT	80003CBC	SYS$GB_DEFQUEPRI	80003E94
SIO_READPATH	00000EA6-R	SWP$GL_SWTIME	800021DC	SYS$GB_DFMBC	80003DD8
SIO_SELADR	00000F27-R	SWP$GW_BAKPTE	80003EF0	SYS$GB_DFMBFHSH	80003DDE
SIO_SETUP_SCS	00000C06-R	SWP$GW_BALCNT	80002234	SYS$GB_DFMBFIDX	80003DDD
SS$_ACCVIO	0000000C	SWP$GW_EMPTPTE	80003EF2	SYS$GB_DFMBFREL	80003DDC
SS$_BADPARAM	00000014	SWP$GW_IBALSETX	800021AC	SYS$GB_DFMBFSDK	80003DD9
SS$_DUPLNAM	00000094	SWP$GW_SWPINC	80003CC0	SYS$GB_DFMBFSMT	80003DDA
SS$_ENDOFFILE	00000870	SWP$GW_WSLPTE	80003EF4	SYS$GB_DFMBFSUR	80003DDB
SS$_EXPRCLM	00002A34	SWP$K_KSTKSZ	000000A0	SYS$GB_DFNBC	80003DE6
SS$_EXQUOTA	0000001C	SWP$SHELINIT	00000A00	SYS$GB_KMASK	8000FE37
SS$_INSSWAPSPACE	00002264	SYS$ADJSTK	7FFEDE20	SYS$GB_KRNLNARG	8000FDE0
SS$_IVLOGNAM	00000154	SYS$ADJWSL	7FFEDE28	SYS$GB_MAXQUEPRI	80003E98
SS$_IVQUOTAL	00000164	SYS$ASCTIM	7FFEDE48	SYS$GB_PWD_TMO	80003E9C
SS$_IVSTSFLG	0000017C	SYS$ASCTOID	7FFEE4B0	SYS$GB_RETRY_LIM	80003E9D
SS$_NOLOGNAM	000001BC	SYS$ASSIGN	7FFEDE50	SYS$GB_RETRY_TMO	80003E9E
SS$_NOPRIV	00000024	SYS$BINTIM	7FFEDE58	SYS$GB_RMSPROLOG	80003DDF
SS$_NORMAL	00000001	SYS$BRKTHRUW	7FFEE4E8	SYS$GET	7FFEE180
SS$_NOSHRIMG	000021BC	SYS$CALL_HANDL	80000010	SYS$GETCHN	7FFEE0C8
SS$_NOSLOT	0000039C	SYS$CANCEL	7FFEDE60	SYS$GETDVIW	7FFEE418
SWI$GL_FQBL	80002964	SYS$CANTIM	7FFEDE68	SYS$GETJPI	7FFEE0D8
SWI$GL_FQFL	80002960	SYS$CANWAK	7FFEDE70	SYS$GETLKIW	7FFEE4A0
SWP$AL_PTRPAG	00000600	SYS$CLOSE	7FFEE1B8	SYS$GETMSG	7FFEE0B0
SWP$A_KSTK	80000EC0	SYS$CMKRNL	7FFEDE90	SYS$GL_BOOTDDB	80000F5C
SWP$C_DBGPTCNT	00000001	SYS$CONNECT	7FFEE1C0	SYS$GL_BOOTORB	80000FA0
SWP$C_KSTACK	00000003	SYS$CREATE	7FFEE1C8	SYS$GL_BOOTUCB	80000FF4
SWP$C_KSTACK_EX	00000004	SYS$CRELNM	7FFEE480	SYS$GL_BRK_TMO	80003EA0
SWP$C_KSTACK_EX_WSL	0000005F	SYS$CREMBX	7FFEDEB8	SYS$GL_FALLBACK	80000EF0
SWP$C_KSTACK_WSL	00000063	SYS$CREPRC	7FFEDEC0	SYS$GL_HIB_TIM	80003EA4
SWP$C_NDYN	00000001	SYS$CRETVA	7FFEDEC8	SYS$GL_JOBCTLMB	800014D4
SWP$C_SHELLPFIL	00000068	SYS$CRMPSC	7FFEDE78	SYS$GL_OPRMBX	800015B8
SWP$C_SHLFPTE	00000081	SYS$C_JOBCTLMB	3141424D	SYS$GL_UIS	80000EE8
SWP$C_SHLP1PT	00000002	SYS$C_MBXUCBSIZ	00000090	SYS$GQ_PWD	80002C40
SWP$GB_ISWPRI	800021BE	SYS$C_OPRMBX	3241424D	SYS$GQ_VERSION	80002B98

Symbol	Value	Symbol	Value	Symbol	Value
SYS$GT_ANNOUNCE	80025864	TTY$GL_DEFPORT	80003DD4		
SYS$GW_BJOBCNT	80002BA4	TTY$GL_DELTA	80003DA4		
SYS$GW_BJOBLIM	80003E8C	TTY$GL_DPT	80000EDC		
SYS$GW_FILEPROT	80003DE2	TTY$GL_JOBCTLMB	80000EE4		
SYS$GW_GBLBUFQUO	80003DE4	TTY$GL_OWNUIC	80003DC4		
SYS$GW_IJOBCNT	80002BA0	TTY$GL_TIMEOUT	80003DCC		
SYS$GW_IJOBLIM	80003E8A	TTY$GW_ALTALARM	80003DBC		
SYS$GW_NJOBCNT	80002BA2	TTY$GW_ALTYPAHD	80003DBA		
SYS$GW_NJOBLIM	80003E8E	TTY$GW_CLASSNAM	80003DC8		
SYS$GW_RJOBLIM	80003E90	TTY$GW_DEFBUF	80003DAC		
SYS$GW_RMSEXTEND	80003DE0	TTY$GW_DMASIZE	80003DBE		
SYS$HIBER	7FFEDF88	TTY$GW_PROT	80003DC0		
SYS$IDTOASC	7FFEE4C0	TTY$GW_TYPAHDSZ	80003DB8		
SYS$IMGACT	7FFEDF90	UBA$UNEXINT	80003438		
SYS$IMGFIX	7FFEE400	UIS$GL_USB	80000EEC		
SYS$K_VERSION	394D3258	XDELBPT	80022988		
SYS$MGBLSC	7FFEDFA8	XDELIBRK	800220D8		
SYS$NUMTIM	7FFEDFB8	XDELTBIT	80022A20		
SYS$OPEN	7FFEE208	XDEL_LOADBASE	80022151		
SYS$PARSE	7FFEE230	XDS$GL_XESTRING	8002217D		
SYS$PUT	7FFEE188	XDS$GL_XFSTRING	80022181		
SYS$PUTMSG	7FFEE0E0	XDS$GT_LONG_PFN	80021EF8		
SYS$QIO	7FFEDFC8	XDS$GT_WORD_PFN	80021E00		
SYS$QIOW	7FFEDE00	XQP$BLOCK_ROUTINE	80009093		
SYS$RMSRUNDWN	7FFEE268	XQP$DEQBLOCKER	800090EF		
SYS$RUNDWN	7FFEDFE0	XQP$FCBSTALE	80009169		
SYS$SEARCH	7FFEE248	XQP$GL_DZRO	80002C98		
SYS$SETAST	7FFEDFF8	XQP$GL_FILESERVER	80002C9C		
SYS$SETEF	7FFEE000	XQP$GL_SECTIONS	80002C94		
SYS$SETEXV	7FFEE008	XQP$REL_QUOTA	80009103		
SYS$SETIMR	7FFEE020	XQP$UNLOCK_QUOTA	8000911A		
SYS$SETPFM	7FFEE0A8				
SYS$SETPRI	7FFEE028				
SYS$SETPRT	7FFEE030				
SYS$SETPRV	7FFEE100				
SYS$SETRWM	7FFEE038				
SYS$SETSFM	7FFEE040				
SYS$SYNCH	7FFEE440				
SYS$TRNLNM	7FFEE490				
SYS$TRNLOG	7FFEE058				
SYS$UNWIND	7FFEE070				
SYS$WAIT	7FFEE1A8				
SYS$WAITFR	7FFEE078				
SYSL$CLRSBIA	80003520				
TTY$GB_AUTOCHAR	80003DD0				
TTY$GB_DEFSPEED	80003DA9				
TTY$GB_DIALTYP	80003DA8				
TTY$GB_PARITY	80003DAB				
TTY$GB_RSPEED	80003DAA				
TTY$GB_SILOTIME	80003DCA				
TTY$GL_DEFCHAR	80003DB0				
TTY$GL_DEFCHAR2	80003DB4				

Symbol	Value	Symbol	Value	Symbol	Value

Key for special characters above:

*	- Undefined
U	- Universal
R	- Relocatable
X	- External

Image Synopsis

```
Virtual memory allocated:                00000000 000017FF 00001800 (6144. bytes, 12. pages)
Stack size:                                    0. pages
Image header virtual block limits:             1.        1. (    1. block)
Image binary virtual block limits:             2.       13. (   12. blocks)
Image name and identification:           CXDRIVER V6-000
Number of files:                              11.
Number of modules:                            10.
Number of program sections:                    5.
Number of global symbols:                   3095.
Number of image sections:                      1.
Image type:                              NOPIC, SHAREABLE. Global Section Match=EQUAL, Ident, Major=91, Minor=7631888
Map format:                              DEFAULT in file DRB2:[SHULL.EVXCI.CXDRIVER]CXDRIVER.MAP;28
Estimated map length:                    207. blocks
```

Link Run Statistics

Performance Indicators	Page Faults	CPU Time	Elapsed Time
Command processing:	135	00:00:01.21	00:00:02.13
Pass 1:	751	00:00:08.35	00:00:15.56
Allocation/Relocation:	13	00:00:00.20	00:00:00.97
Pass 2:	23	00:00:02.10	00:00:04.84
Map data after object module synopsis:	36	00:00:07.79	00:00:16.73
Symbol table output:	1	00:00:00.18	00:00:00.78
Total run values:	959	00:00:19.83	00:00:41.01

```
Using a working set limited to 2048 pages and 903 pages of data storage (excluding image)

Total number object records read (both passes):    828
    of which 46 were in libraries and 16 were DEBUG data records containing 378 bytes

Number of modules extracted explicitly              = 0
    with 1 extracted to resolve undefined symbols

0 library searches were for symbols not in the library searched

A total of 8 global symbol table records was written

LINK/MAP=CXDRIVER/SHARE=CXDRIVER/CONTIGUOUS/SYMBOL_TABLE=CXDRIVER CXTABLES,CXCMDFDT,CXCMD,CXMAINT,CXINPUT,CXCANCEL,CXINIT,SYS$SYSTEM
:SYS.STB,SYS$SYSTEM:SYSDEF.STB/SELECTIVE_SEARCH,SYS$INPUT/OPTION
```

Table of contents

(2) 43 $CXQIODEF, CXDRIVER QIO interface offsets
(3) 64 $CXSTATEDEF, CXDRIVER internal state definitions
(4) 93 $CXCDRPDEF, CXDRIVER CDRP extension
(5) 125 $CXCDBDEF, Class driver data block definition
(6) 177 $CINTREGDEF, CINT register definitions

```
0000     1          .TITLE  CXDRIVER MACRO DEFINITIONS
0000     2          .IDENT  'V05-000'
0000     3
0000     4  ;******************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1983 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;******************************************************************************
0000    23  ;
0000    24  ;++
0000    25  ;
0000    26  ;
0000    27  ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
0000    28  ;
0000    29  ;
0000    30  ; ABSTRACT:      Class Driver Definition File
0000    31  ;
0000    32  ; AUTHOR:        Richard Hennessy  27-Jul-1982
0000    33  ;                Base Systems Diagnostic Engineering
0000    34  ;
0000    35  ; MODIFIED BY:   Jim Klumpp   6-May-83
0000    36  ;                Base Systems Diagnostic Engineering
0000    37  ;
0000    38  ;--
0000    39
0000    40
0000    41
```

```
                        0000     43          .SBTTL  $CXQIODEF,      CXDRIVER QIO interface offsets
                        0000     44 ;+
                        0000     45 ;
                        0000     46 ;  $CXQIODEF
                        0000     47 ;
                        0000     48 ;  This macro defines the QIO parameters passed between the control
                        0000     49 ;  process and CXDRIVER. All offsets are from the AP.
                        0000     50 ;
                        0000     51 ;-
                        0000     52
                        0000     53          .MACRO  $CXQIODEF,      CXDRIVER QIO interface offsets
                        0000     54
                        0000     55                  BUFF_AD         ==      0       ; User buffer address
                        0000     56                  BUFF_LEN        ==      4       ; User buffer length
                        0000     57                  DIAG_VERSION    ==      8       ; Diagnostic version
                        0000     58
                        0000     59          .ENDM   $CXQIODEF
                        0000     60
                        0000     61
                        0000     62
```

```
                                        .SBTTL  $CXSTATEDEF,    CXDRIVER internal state definitions
                                ;+
                                ;
                                ;  $CXSTATEDEF
                                ;
                                ;  This macro defines the CXDRIVER internal state word.
                                ;
                                ;-

                                        .MACRO  $CXSTATEDEF

                                        _VIELD  -
                                                STATE,0,-
                                                <-
                                                <RMAP,,M>,-             ; Response Buffer is mapped
                                                <CMAP,,M>,-             ; Command Buffer is mapped
                                                <ALTGRBG,,M>,-          ; ALT GARBAGE dump has some garbage
                                                <CANCEL,,M>,-           ; CANCEL in progress
                                                <,5>,-
                                                <STOP,,M>,-             ; STOP function has been serviced
                                                <FDT,,M>,-              ; STARTDATA function is in FDT code
                                                <SIO,,M>,-              ;                               SIO
                                                <NOTPROC,,M>,-          ; NOT PROCESSING cmds, awaiting input
                                                >

                                        .ENDM   $CXSTATEDEF
```

```
                        93          .SBTTL  $CXCDRPDEF,     CXDRIVER CDRP extension
                        94 ;+
                        95 ;  CXDRIVER CDRP EXTENSION DEFINITIONS
                        96 ;
                        97 ;  The following CXDRIVER extensions have been made to the class driver
                        98 ;  request packet.
                        99 ;-
                       100
                       101          .MACRO  $CXCDRPDEF
                       102          $DEFINI CDRP                                ; CXDRIVER CDRP extension
                       103
                       104          ASSUME CDRP$C_BT_LEN EQ 64
                       105
                       106 . = 64
                       107
                       108 $DEF     CDRP$B_FLAGS            .BLKB               ; Flag byte
                       109                                  .BLKB   3
                       110 $DEF     CDRP$L_PENDQFL          .BLKL               ; Pending Queue forward link
                       111 $DEF     CDRP$L_PENDQBL          .BLKL               ; Pending Queue backward link
                       112 $DEF     CDRP$L_PDT              .BLKL               ; PDT address
                       113 $DEF     CDRP$K_CTPLEN                               ; CDRP length for DG/MSG
                       114
                       115 $DEF     CDRP$T_CX_LBUFHNDL      .BLKB   12          ; Local buff handle
                       116 $DEF     CDRP$T_CX_RBUFHNDL      .BLKB   12          ; Remote buff handle
                       117 $DEF     CDRP$K_XFRLEN                               ; CDRP length for DATA XFRs
                       118 $DEF     CDRP$K_CX_LEN
                       119
                       120          $DEFEND CDRP                        ; End of CDRP additions
                       121          .ENDM
                       122
                       123
```

```
0000    125          .SBTTL  $CXCDBDEF,      Class driver data block definition
0000    126 ;+
0000    127 ;
0000    128 ;  CLASS DRIVER DATA BLOCK DEFINITION
0000    129 ;
0000    130 ;  These blocks provide a mechanism for the class driver to maintain
0000    131 ;  connection-specific information in an orderly fashion. Each one also
0000    132 ;  provides a pending CDRP queue. This list will contain the outstanding
0000    133 ;  CDRP's (commands) that SCS is currently processing on the connection
0000    134 ;  associated with this CDB. These blocks are allocated at CONNECT time and
0000    135 ;  are only returned when a DISCONNECT has been completed, or a Virtual
0000    136 ;  Circuit crashes, or if the I/O is $CANCELed by the control process or VMS.
0000    137 ;
0000    138 ;-
0000    139
0000    140          .MACRO  $CXCDBDEF
0000    141          $DEFINI CDB
0000    142
0000    143 .=0
0000    144
0000    145 $DEF     CDB$L_QFL        .BLKL                    ; Forward link
0000    146 $DEF     CDB$L_QBL        .BLKL                    ; Backward link
0000    147 $DEF     CDB$W_SIZE       .BLKW                    ; Structure size
0000    148 $DEF     CDB$B_TYPE       .BLKB                    ; Structure type
0000    149                          .BLKB                    ; Unused
0000    150 $DEF     CDB$W_STATUS     .BLKW                    ; CDB status
0000    151 $DEF     CDB$B_RSTNADR    .BLKB   6                ; Remote station addr
0000    152                          .BLKW                    ; Unused
0000    153 $DEF     CDB$T_LPRTNAM    .BLKB   6                ; Local port name
0000    154 $DEF     CDB$L_CDT        .BLKL                    ; CDT address
0000    155 $DEF     CDB$L_PDT        .BLKL                    ; PDT address
0000    156 $DEF     CDB$L_CDBSLOT    .BLKL                    ; Addr of CDB SLOT in CONLST
0000    157 $DEF     CDB$B_CDRPCNT    .BLKW                    ; Outstanding CDRP count
0000    158 $DEF     CDB$L_CDRPQFL    .BLKL                    ; Pending CDRP queue flink
0000    159 $DEF     CDB$L_CDRPQBL    .BLKL                    ; Pending CDRP queue blink
0000    160 $DEF     CDB$K_LEN                                 ; Structure length
0000    161
0000    162 ; Define the Class driver data block status bits
0000    163
0000    164          $VIELD  -
0000    165                  CDB,0,-
0000    166                  <-
0000    167                  <CONN,,M>,-                       ; 00  Connect in progress
0000    168                  <DISC,,M>,-                       ; 01  Disconnect in progress
0000    169                  <CNCL,,M>,-                       ; 02  Cancel while suspended
0000    170                  >
0000    171
0000    172          $DEFEND CDB                               ; END of CDB definition
0000    173          .ENDM
0000    174
0000    175
```

```
0000    177          .SBTTL  $CINTREGDEF,    CINT register definitions
0000    178 ;+
0000    179 ;
0000    180 ;  $CINTREGDEF
0000    181 ;
0000    182 ;  CINT register definitions. The following macro defines the offsets
0000    183 ;  from the configuration register to the specified regsiter. The
0000    184 ;  configuration register address can be obtained from the PDT. Also
0000    185 ;  defined are the fields within each of the CINT registers.
0000    186 ;
0000    187 ;-
0000    188
0000    189          .MACRO  $CINTREGDEF
0000    190          $DEFINI $CINTREGDEF
0000    191
0000    192          $EQU    PA_CLSR <^X1000>        ; Control logic status register
0000    193          _VIELD  -
0000    194                  CLSR,0,-
0000    195                  <-
0000    196                  <CRC,,M>,-              ; 00 CRC error detected
0000    197                  <SRC,,M>,-              ; 01 Source field match
0000    198                  <DEST,,M>,-             ; 02 Distination field NOT match
0000    199                  <CDB,,M>,-              ; 03 Carrier detect B
0000    200                  <CDA,,M>,-              ; 04 Carrier detect A
0000    201                  <PATHB,,M>,-            ; 05 Packet received on path B
0000    202                  <VRD,,M>,-              ; 06 Valid receive data
0000    203                  <,8,>,-                 ; 7-14 Unused
0000    204                  <STSAVL,,M>,-           ; 15 Status available
0000    205                  <XMITDE,,M>,-           ; 16 Transmit data available
0000    206                  <,6,>,-                 ; 17-22 Unused
0000    207                  <BUSY,,M>,-             ; 23 Busy, CINT test in progress
0000    208                  >
0000    209
0000    210          $EQU    PA_GOR  <^X1008>        ; Go, Large packet register
0000    211          _VIELD  -
0000    212                  GOR,0,-
0000    213                  <-
0000    214                  <GO,,M>,-               ; 00 GO bit
0000    215                  <,14,>,-                ; 1-14 Unused
0000    216                  <XLP,,M>,-              ; 15 Transmit large packet
0000    217                  <RLP,,M>,-              ; 16 Receive large packet
0000    218                  >
0000    219
0000    220          $EQU    PA_GOUR <^X1010>        ; Go unconditional, UUT dead register
0000    221          _VIELD  -
0000    222                  GOUR,0,-
0000    223                  <-
0000    224                  <GOU,,M>,-              ; 00 Go unconditional
0000    225                  <,14,>,-                ; 1-14 Unused
0000    226                  <UUTD,,M>,-             ; 15 Unit under test dead
0000    227                  <RSVD,,M>,-             ; 16 Reserved
0000    228                  >
0000    229
0000    230          $EQU    PA_CLER <^X1018>        ; Control logic enable register
0000    231          _VIELD  -
0000    232                  CLER,0,-
0000    233                  <-
```

```
0000   234                  <CARRB,,M>,-          ; 00 Assert carrier on path B
0000   235                  <CARRA,,M>,-          ; 01 Assert carrier on path A
0000   236                  <ALL,,M>,-            ; 02 Accept all incoming packets
0000   237                  <NAK,,M>,-            ; 03 Arm NAK logic
0000   238                  <NO_RSP,,M>,-         ; 04 Arm NO_RSP logic
0000   239                  <IGACK,,M>,-          ; 05 Arm ignore ACK logic
0000   240                  <OVPAC,,M>,-          ; 06 Arm oversized packet logic
0000   241                  <LDSTA,,M>,-          ; 07 Load status register
0000   242                  <,7,>,-               ; 8-14 Unused
0000   243                  <STTST,,M>,-          ; 15 Status test
0000   244                  <LDALL,,M>,-          ; 16 Load all packets
0000   245                  <,6,>,-               ; 17-22 Unused
0000   246                  <DIAGI,,M>,-          ; 23 Diagnostic test in progress
0000   247                  >
0000   248
0000   249          $EQU    PA_NNCR <^X1020>      ; NAK, NORSP, LPC count register
0000   250          _VIELD  -
0000   251                  NNCR,0,-
0000   252                  <-
0000   253                  <C,8,M>,-             ; 0-7 NAK, NO_RSP count
0000   254                  <,8,>,-               ; 8-15 Unused
0000   255                  <LPC,3,M>,-           ; 16-18 Large packet count
0000   256                  >
0000   257
0000   258          $EQU    PA_CAR  <^X1028>      ; CINT address register
0000   259          _VIELD  -
0000   260                  CAR,0,-
0000   261                  <-
0000   262                  <A,8,M>,-             ; 0-7 New CINT address
0000   263                  >
0000   264
0000   265          $EQU    PA_SR   <^X1030>      ; Source register
0000   266          _VIELD  -
0000   267                  SR,0,-
0000   268                  <-
0000   269                  <S,8,M>,-             ; 0-7 Source field
0000   270                  >
0000   271
0000   272          $EQU    PA_NNCO <^X1038>      ; NAK, NO_RSP counter output register
0000   273          _VIELD  -
0000   274                  NNCO,0,-
0000   275                  <-
0000   276                  <CO,8,M>,-            ; 0-7 NAK, NO_RSP counter output
0000   277                  >
0000   278
0000   279          $DEFEND CINTREGDEF
0000   280          .ENDM   $CINTREGDEF
0000   281
0000   282
0000   283 ;+
0000   284 ; SET_DRVR_STATE
0000   285 ;
0000   286 ; Set CXDRIVER state bit(s).
0000   287 ;
0000   288 ; INPUTS:
0000   289 ;
0000   290 ;       ARG     - State bit identifier
```

```
0000    291 ;-
0000    292
0000    293         .MACRO  SET_DRVR_STATE  ARG
0000    294
0000    295                 BISL2   #STATE_M_'ARG,CX$L_STATE
0000    296
0000    297         .ENDM   SET_DRVR_STATE
0000    298
0000    299
0000    300 ;+
0000    301 ;  CLR_DRVR_STATE
0000    302 ;
0000    303 ;  Clear CXDRIVER state bit(s).
0000    304 ;
0000    305 ;  INPUTS:
0000    306 ;
0000    307 ;       ARG     - State bit identifier
0000    308 ;-
0000    309
0000    310         .MACRO  CLR_DRVR_STATE  ARG
0000    311
0000    312                 BICL2   #STATE_M_'ARG,CX$L_STATE
0000    313
0000    314         .ENDM   CLR_DRVR_STATE
0000    315
0000    316
0000    317
0000    318 ;+
0000    319 ;  SET_CMD_FLG
0000    320 ;
0000    321 ;  Set class driver command state flag(s).
0000    322 ;
0000    323 ;  INPUTS:
0000    324 ;
0000    325 ;       FLAG    - Flag to set
0000    326 ;       REG     - Pointer to current command block (default R3)
0000    327 ;-
0000    328
0000    329         .MACRO  SET_CMD_FLAG    FLAG,REG=R3
0000    330
0000    331                 BISW2   #FLAG_M_'FLAG, CDC$W_FLAG(REG)
0000    332
0000    333         .ENDM   SET_CMD_FLAG
0000    334
0000    335
0000    336 ;+
0000    337 ;  CLR_CMD_FLAG
0000    338 ;
0000    339 ;  Clear class driver command flag(s).
0000    340 ;
0000    341 ;  INPUTS:
0000    342 ;
0000    343 ;       FLAG    - Command flag specifier
0000    344 ;       REG     - Pointer to current command block (default R3)
0000    345 ;-
0000    346
0000    347         .MACRO  CLR_CMD_FLAG    FLAG,REG=R3
```

```
0000    348
0000    349                 BICW2   #FLAG_M_'FLAG, CDC$W_FLAG(REG)
0000    350
0000    351         .ENDM   CLR_CMD_FLAG
0000    352
0000    353
0000    354 ;+
0000    355 ;  CALC_RSP
0000    356 ;
0000    357 ;  Calculate the address of the start of the response area for the current
0000    358 ;  command.
0000    359 ;
0000    360 ;  INPUTS:
0000    361 ;
0000    362 ;       REG     - Pointer to current command (default R3)
0000    363 ;
0000    364 ;  OUTPUTS:
0000    365 ;
0000    366 ;       DST     - Address of response area (default R1)
0000    367 ;-
0000    368
0000    369         .MACRO  CALC_RSP        DST=R1,REG=R3
0000    370
0000    371                 MOVZWL  CDC$W_RSPOFSET(REG),DST
0000    372                 ADDL2   CX$L_RSPBUF3GN,DST
0000    373
0000    374         .ENDM   CALC_RSP
0000    375
0000    376
0000    377
0000    378 ;+
0000    379 ;  CALC_CDB_LIST
0000    380 ;
0000    381 ;  Macro to calculate the address of the connection list of a given
0000    382 ;  local port name.
0000    383 ;
0000    384 ;  INPUTS:
0000    385 ;
0000    386 ;       REG     - Pointer to current command (default R3)
0000    387 ;
0000    388 ;  OUTPUTS:
0000    389 ;
0000    390 ;       DST     - Address of connection list (default R2)
0000    391 ;-
0000    392
0000    393         .MACRO  CALC_CDB_LIST   DST=R2,REG=R3
0000    394
0000    395                 MOVZBL  CDC$L_LPORT+2(REG),DST
0000    396                 SUBB2   #^X41,DST
0000    397                 MOVL    CX$L_LOCPRTLST[DST],DST
0000    398
0000    399         .ENDM   CALC_CDB_LIST
0000    400
0000    401
0000    402 ;+
0000    403 ;  CALC_CDB_AD
0000    404 ;
```

```
                        0000    405 ;  Macro to calculate the address of the CDB associated with the given
                        0000    406 ;  local port name and remote port number.
                        0000    407 ;
                        0000    408 ;  INPUTS:
                        0000    409 ;
                        0000    410 ;       REG     - Pointer to current command (default R3)
                        0000    411 ;
                        0000    412 ;  OUTPUTS:
                        0000    413 ;
                        0000    414 ;       DST             address (default R2)
                        0000    415 ;-
                        0000    416
                        0000    417         .MACRO  CALC_CDB_AD     DST=R2,REG=R3
                        0000    418
                        0000    419                 PUSHL   R0
                        0000    420                 MOVZBL  CDC$L_LPORT+2(REG),DST
                        0000    421                 SUBB2   #^X41,DST
                        0000    422                 MOVL    CX$L_LOCPRTLST[DST],DST
                        0000    423                 MOVZBL  CDC$B_DSTPORT(REG),R0
                        0000    424                 MOVL    (DST)[R0],DST
                        0000    425                 POPL    R0
                        0000    426                 TSTL    DST
                        0000    427
                        0000    428         .ENDM   CALC_CDB_AD
                        0000    429
                        0000    430
                        0000    431
                        0000    432 ;+
                        0000    433 ;  CONDITIONAL BRANCH INSTRUCTIONS MACROS
                        0000    434 ;
                        0000    435 ;  The following macros are used when a conditional branch instruction needs
                        0000    436 ;  a word, rather than byte, displacement. They don't save any code; they just
                        0000    437 ;  make it easier to code.
                        0000    438 ;-
                        0000    439
                        0000    440         .MACRO  BNEQW,DISPLACEMENT,?SKIP
                        0000    441
                        0000    442                 BEQL    SKIP
                        0000    443                 BRW     DISPLACEMENT
                        0000    444 SKIP:
                        0000    445
                        0000    446         .ENDM
                        0000    447
                        0000    448         .MACRO  BNEQUW,DISPLACEMENT,?SKIP
                        0000    449
                        0000    450                 BEQLU   SKIP
                        0000    451                 BRW     DISPLACEMENT
                        0000    452 SKIP:
                        0000    453
                        0000    454         .ENDM
                        0000    455
                        0000    456         .MACRO  BEQLW,DISPLACEMENT,?SKIP
                        0000    457
                        0000    458                 BNEQ    SKIP
                        0000    459                 BRW     DISPLACEMENT
                        0000    460 SKIP:
                        0000    461
```

```
0000    462          .ENDM
0000    463
0000    464          .MACRO  BEQLUW,DISPLACEMENT,?SKIP
0000    465
0000    466                  BNEQU   SKIP
0000    467                  BRW     DISPLACEMENT
0000    468 SKIP:
0000    469
0000    470          .ENDM
0000    471
0000    472          .MACRO  BGTRW,DISPLACEMENT,?SKIP
0000    473
0000    474                  BLEQ    SKIP
0000    475                  BRW     DISPLACEMENT
0000    476 SKIP:
0000    477
0000    478          .ENDM
0000    479
0000    480          .MACRO  BLEQW,DISPLACEMENT,?SKIP
0000    481
0000    482                  BGTR    SKIP
0000    483                  BRW     DISPLACEMENT
0000    484 SKIP:
0000    485
0000    486          .ENDM
0000    487
0000    488          .MACRO  BGEQW,DISPLACEMENT,?SKIP
0000    489
0000    490                  BLSS    SKIP
0000    491                  BRW     DISPLACEMENT
0000    492 SKIP.
0000    493
0000    494          .ENDM
0000    495
0000    496          .MACRO  BLSSW,DISPLACEMENT,?SKIP
0000    497
0000    498                  BGEQ    SKIP
0000    499                  BRW     DISPLACEMENT
0000    500 SKIP:
0000    501
0000    502          .ENDM
0000    503
0000    504          .MACRO  BGTRUW,DISPLACEMENT,?SKIP
0000    505
0000    506                  BLEQU   SKIP
0000    507                  BRW     DISPLACEMENT
0000    508 SKIP:
0000    509
0000    510          .ENDM
0000    511
0000    512          .MACRO  BLEQUW,DISPLACEMENT,?SKIP
0000    513
0000    514                  BGTRU   SKIP
0000    515                  BRW     DISPLACEMENT
0000    516 SKIP:
0000    517
0000    518          .ENDM
```

```
0000    519
0000    520          .MACRO  BVCW,DISPLACEMENT,?SKIP
0000    521
0000    522                  BVS     SKIP
0000    523                  BRW     DISPLACEMENT
0000    524 SKIP:
0000    525
0000    526          .ENDM
0000    527
0000    528          .MACRO  BVSW,DISPLACEMENT,?SKIP
0000    529
0000    530                  BVC     SKIP
0000    531                  BRW     DISPLACEMENT
0000    532 SKIP:
0000    533
0000    534          .ENDM
0000    535
0000    536          .MACRO  BGEQUW,DISPLACEMENT,?SKIP
0000    537
0000    538                  BLSSU   SKIP
0000    539                  BRW     DISPLACEMENT
0000    540 SKIP:
0000    541
0000    542          .ENDM
0000    543
0000    544          .MACRO  BCCW,DISPLACEMENT,?SKIP
0000    545
0000    546                  BCS     SKIP
0000    547                  BRW     DISPLACEMENT
0000    548 SKIP:
0000    549
0000    550          .ENDM
0000    551
0000    552          .MACRO  BCSW,DISPLACEMENT,?SKIP
0000    553
0000    554                  BCC     SKIP
0000    555                  BRW     DISPLACEMENT
0000    556 SKIP:
0000    557
0000    558          .ENDM
0000    559
0000    560          .MACRO  BLSSUW,DISPLACEMENT,?SKIP
0000    561
0000    562                  BGEQU   SKIP
0000    563                  BRW     DISPLACEMENT
0000    564 SKIP:
0000    565
0000    566          .ENDM
0000    567
0000    568          .MACRO  BLBSW,REGISTER,DISPLACEMENT,?SKIP
0000    569
0000    570                  BLBC    REGISTER,SKIP
0000    571                  BRW     DISPLACEMENT
0000    572 SKIP:
0000    573
0000    574          .ENDM
0000    575
```

```
                                0000    576          .MACRO  BLBCW,REGISTER,DISPLACEMENT,?SKIP
                                0000    577
                                0000    578                  BLBS    REGISTER,SKIP
                                0000    579                  BRW     DISPLACEMENT
                                0000    580 SKIP:
                                0000    581
                                0000    582          .ENDM
                                0000    583
                                0000    584          .MACRO BBCW,POS,BASE,DISPLACEMENT,?SKIP
                                0000    585
                                0000    586                  BBS     POS,BASE,SKIP
                                0000    587                  BRW     DISPLACEMENT
                                0000    588 SKIP:
                                0000    589
                                0000    590          .ENDM   BBCW
                                0000    591
                                0000    592          .MACRO BBSW,POS,BASE,DISPLACEMENT,?SKIP
                                0000    593
                                0000    594                  BBC     POS,BASE,SKIP
                                0000    595                  BRW     DISPLACEMENT
                                0000    596 SKIP:
                                0000    597
                                0000    598          .ENDM   BBSW
                                0000    599
                                0000    600          .MACRO BBSSW,POS,BASE,DISPLACEMENT,?SKIP
                                0000    601
                                0000    602                  BBCS    POS,BASE,SKIP
                                0000    603                  BRW     DISPLACEMENT
                                0000    604 SKIP:
                                0000    605
                                0000    606          .ENDM   BBSSW
                                0000    607
                                0000    608
%MACRO-w-MISSINGEND, Missing .END statement
                                0000
                                0000      1 .END
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	121	00:00:00.47	00:00:01.90
Command processing	69	00:00:00.19	00:00:00.36
Pass 1	160	00:00:01.10	00:00:01.50
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	117	00:00:01.03	00:00:01.49
Symbol table output	0	00:00:00.00	00:00:00.00
Psect synopsis output	3	00:00:00.01	00:00:00.01
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	473	00:00:02.80	00:00:05.27

The working set limit was 900 pages.
3706 bytes (8 pages) of virtual memory were used to buffer the intermediate code.
There were 0 pages of symbol table space allocated to hold 0 non-local and 0 local symbols.
609 source lines were read in Pass 1, producing 0 object records in Pass 2.
40 pages of virtual memory were used to define 33 macros.

Macro library statistics

Macro library name	Macros defined
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	0

0 GETS were required to define 0 macros.

There were 0 errors, 1 warning and 0 information messages, on lines:
608 (6)

MACRO/LIST/NOOBJ CXMAC.MAR

Table of contents

(1) 57 DEFINITIONS
(1) 92 CX_CANCELIO, Cancel I/O Routine
(4) 185 CAN$INITCINT, Initialize any possible CINT logic
(5) 223 CLEAN_RST_STR, Clean possible reset/start thread
(6) 262 CLEAN_PORT_STATE, Return port to enabled state
(7) 293 CANCEL_CMD_BUF, Cancel all commands in command buffer
(8) 373 FINISH_IO, Complete the original I/O
(9) 397 CANCEL_MAPPED_BUFFERS, Unmap all mapped buffers
(10) 421 CANCEL_CONNECTIONS, Clean up all remaining connections
(11) 455 START_POLLER, Restart the configuration poller
(12) 492 KILL_CDB, Clean Up A Connection
(13) 543 CLEAN_CDB, Clean pending CDRP's off CDB
(14) 580 CHECK_LINK_FQ, Check for CDRP on resource queue
(15) 615 CHECK_LINK_PQ, Check for CDRP on pending queue
(16) 651 CLEAN_CDRP, Deallocate CDRP resources
(17) 699 DEALLOC_CDRP, Deallocate a CDRP appropriately

```
0000     1          .TITLE  CXCANCEL, Cancel I/O Module
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                 MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;*****************************************************************************
0000    23  ;
0000    24  ;++
0000    25  ;
0000    26  ;
0000    27  ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
0000    28  ;
0000    29  ;
0000    30  ; ABSTRACT:      This module contains the CANCEL code, the UNIT_INIT code,
0000    31  ;                the UNSOLINT code, and the END tag.
0000    32  ;
0000    33  ; AUTHOR:        Richard Hennessy  27-Jul-1982
0000    34  ;                Base Systems Diagnostic Engineering
0000    35  ;
0000    36  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    37  ;                Base Systems Diagnostic Engineering
0000    38  ;
0000    39  ;        6-000   Dave Shull      07-July-1984
0000    40  ;                VMS V4 Modifications/Release
0000    41  ;
0000    42  ;        5-003   Dave Shull      14-June-1984
0000    43  ;                Modified routine START_POLLER to not set the 0 on PAA0 to 0.
0000    44  ;                This should be passed as an Ascii '0' for VMS V4.
0000    45  ;        5-002   Dave Shull      7-March-1984
0000    46  ;                Changed routine CLEAN_RST_STR to use CX$L_RST_STR_CDRP versus
0000    47  ;                PDT$L_RSTCDRP(R4) for finding CDRP.
0000    48  ;
0000    49  ;        5-001   Dave Shull      9-Feb-1984
0000    50  ;                Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SYS$B_PPD
0000    51  ;                as in VMS V4 all SCS$ symbols reference from the Application
0000    52  ;                Data area versus Packet Pointer.
0000    53  ;--
0000    54
0000    55          .DEFAULT DISPLACEMENT,WORD
0000    56          .ENABLE  SUPPRESSION
0000    57          .SBTTL   DEFINITIONS
```

```
                        58
                        59 ; Set PSECT to driver code
                        60
00000000                61          .PSECT  $$$115_DRIVER,LONG
                        62
                        63 ;  System definitions (LIB.MLB):
                        64
                        65          $CDRPDEF                        ; Define CDRP offsets
                        66          $CDTDEF                         ; Define CDT offsets
                        67          $DYNDEF                         ; Define DYN offsets
                        68          $IODEF                          ; Define I/O FUNCTION codes
                        69          $IRPDEF                         ; Define IRP offsets
                        70          $PBDEF                          ; Define path block offsets
                        71          $PDTDEF                         ; Define PDT offsets
                        72          $SCSDEF                         ; Define SCS packet offsets
                        73          $SSDEF                          ; Define system status symbols
                        74
                        75
                        76 ;  PADRIVER definitions (PALIB.MLB):
                        77
                        78          $CINTREGDEF                     ; Define CINT register offsets
                        79          $PAPDTDEF                       ; Define PAPDT offsets
                        80          $PPDDEF                         ; Define PPD offsets
                        81
                        82
                        83 ;  CXDRIVER definitions (CXLIB.MLB):
                        84
                        85          $CDCDEF                         ; Define CDC interface offsets
                        86          $CXCDBDEF                       ; Define CXCDB offsets
                        87          $CXCDRPDEF                      ; Define CX CDRP extension
                        88          $CXSTATEDEF                     ; Define CXDRIVER states
                        89
                        90
                        91
                        92          .SBTTL  CX_CANCELIO,    Cancel I/O Routine
```

```
0000     94
0000     95 ;+
0000     96 ;
0000     97 ;  CX_CANCELIO
0000     98 ;
0000     99 ;  WARNING!! - THIS MODULE IS EXTREMELY COMPLEX AND SHOULD BE UNDERSTOOD
0000    100 ;                THOROUGHLY BEFORE BEING MODIFIED.
0000    101 ;
0000    102 ;  This routine services a $CANCEL I/O request performing the appropriate steps
0000    103 ;  to clean up the I/O operation that may be in progress, as well as any other
0000    104 ;  driver state needs cleaning up.
0000    105 ;
0000    106 ;  This routine performs several tricks that should be explained here:
0000    107 ;
0000    108 ;       - In cleaning up connections, a BSBW KILL_CDB call is used. The
0000    109 ;         KILL_CDB routine is where the DISCONNECT call is actually issued
0000    110 ;         to SCS. SCS suspends the disconnect call, and RSB's to the next
0000    111 ;         level PC, which happens to be the return address pushed on the
0000    112 ;         stack from the BSBW KILL_CDB instruction. This allows the cancel
0000    113 ;         routine to maintain control until disconnects have been issued
0000    114 ;         for all connections.
0000    115 ;
0000    116 ;         When each suspended disconnect thread is resumed, the RSB from the
0000    117 ;         KILL_CDB routine returns into SCS rather than the cancel routine.
0000    118 ;
0000    119 ;       - The BSBW SIO_COMPLETE at the end of the CANCEL_CMDS section allows
0000    120 ;         us to complete the I/O, then finish cleaning up the remaining state
0000    121 ;         in the driver.
0000    122 ;
0000    123 ;  The following steps are taken in servicing the cancel:
0000    124 ;
0000    125 ;  CANCEL CURRENT I/O
0000    126 ;
0000    127 ;       - Check to see if a command buffer is present.
0000    128 ;       - Find each command that may have an associated CDRP
0000    129 ;       - Remove CDRP from possible wait queues (SCS and CXDRIVER)
0000    130 ;       - Return all resources held by CDRP
0000    131 ;       - Deallocate the CDRP itself
0000    132 ;       - Complete the I/O request
0000    133 ;
0000    134 ;  CLEAN UP MAPPED BUFFERS
0000    135 ;
0000    136 ;       - Find each IRP/CDRP on mapped buffer queue
0000    137 ;       - Issue a UNMAP call for each IRP/CDRP
0000    138 ;       - Deallocate the IRP/CDRP
0000    139 ;
0000    140 ;  CLEAN UP CONNECTIONS
0000    141 ;
0000    142 ;       - Find each CDB (connection block)
0000    143 ;       - Terminate all outstanding operations queued to CDB
0000    144 ;       - Kill each connection
0000    145 ;       - Deallocate the CDB
0000    146 ;       - Return
0000    147 ;
0000    148 ;  INPUTS:
0000    149 ;
0000    150 ;       IPL             - IPL$_SCS
```

```
                                0000   151 ;       R2              - Channel number
                                0000   152 ;       R3              - IRP address (IRPSAV if I/O in progress)
                                0000   153 ;       R4              - PCB address
                                0000   154 ;       R5              - UCB address
                                0000   155 ;
                                0000   156 ;  OUTPUTS:
                                0000   157 ;
                                0000   158 ;       R0-R2           - Destroyed
                                0000   159 ;       All other registers preserved
                                0000   160 ;
                                0000   161 ;-
                                0000   162
                                0000   163 CX_CANCELIO::
                                0000   164
                      03   E2   0000   165         BBSS    #STATE_V_CANCEL,-       ; Don't allow multiple threads through
                2B 0000'CF         0002   166                 CX$L_STATE,20$          ; this routine
                   03F0 8F   BB    0006   167         PUSHR   #^M<R4,R5,R6,R7,R8,R9>  ; Save regs
                      0025   30    000A   168         BSBW    CAN$INITCINT            ; Initialize possible CINT logic
                      003E   30    000D   169         BSBW    CLEAN_RST_STR           ; Clean up possible reset/start thread
                      005C   30    0010   170         BSBW    CLEAN_PORT_STATE        ; Put port in enabled state if appropriate
                        01   E1    0013   171         BBC     #STATE_V_CMAP,-         ; Branch of no command buffer is
                06 0000'CF         0015   172                 CX$L_STATE,10$          ; present
                      006A   30    0019   173         BSBW    CANCEL_CMD_BUF          ; Cancel the current command buffer
                      00D6   30    001C   174         BSBW    FINISH_IO               ; Finish the original I/O
                      00DE   30    001F   175 10$:    BSBW    CANCEL_MAPPED_BUFFERS   ; Unmap all mapped buffers
                      00F3   30    0022   176         BSBW    CANCEL_CONNECTIONS      ; Cancel all existing connections
                      0114   30    0025   177         BSBW    START_POLLER            ; Start the configuration poller
                   03F0 8F   BA    0028   178         POPR    #^M<R4,R5,R6,R7,R8,R9>  ; Restore regs
                        08   CA    002C   179         BICL2   #STATE_M_CANCEL,-       ; Allow more cancels to occur
                0000'CF         002E   180                 CX$L_STATE
                             05   0031   181 20$:     RSB                             ; Return to EXE$CANCEL
                                  0032   182
                                  0032   183
```

```
                     0032   185          .SBTTL  CAN$INITCINT,  Initialize any possible CINT logic
                     0032   186 ;+
                     0032   187 ;
                     0032   188 ;  CAN$INITCINT
                     0032   189 ;
                     0032   190 ;  Routine to initialize the CINT logic if appropriate. The longword
                     0032   191 ;  CX$L_CINT_PDT is nonzero if any CINT commands were issued since the
                     0032   192 ;  last CANCEL occurred. If this is the case, the CINT logic is initialized.
                     0032   193 ;
                     0032   194 ;  INPUTS:
                     0032   195 ;
                     0032   196 ;       CX$L_CINT_PDT                   - PDT address of CINT if CINT needs
                     0032   197 ;                                           to be initialized
                     0032   198 ;                                       - 0 otherwise
                     0032   199 ;  OUTPUTS:
                     0032   200 ;
                     0032   201 ;       CX$L_CINT_PDT                   - 0
                     0032   202 ;       All registers                   - Preserved
                     0032   203 ;
                     0032   204 ;-
                     0032   205
                     0032   206 CAN$INITCINT:
                     0032   207
            1F   BB  0032   208          PUSHR   #^M<R0,R1,R2,R3,R4>     ; Save regs
 54  0000'CF   D0  0034   209          MOVL    CX$L_CINT_PDT,R4        ; Get PDT address for CINT port
            10   13  0039   210          BEQL    10$                     ; No CINT port, branch
                     003B   211
          FFC2'  30  003B   212          BSBW    CLEAR_GO_BIT            ; Clear the GO bit
          FFBF'  30  003E   213          BSBW    CLEAR_GOU_BIT           ; Clear the GOU bit
          FFBC'  30  0041   214          BSBW    CLEAR_CLER_REG          ; Clear the CLER register
          FFB9'  30  0044   215          BSBW    CLEAR_STS_REG           ; Clear the status register
     0000'CF   D4  0047   216          CLRL    CX$L_CINT_PDT           ; Show CINT has been inited
                     004B   217
            1F   BA  004B   218 10$:     POPR    #^M<R0,R1,R2,R3,R4>     ; Restore regs
                 05  004D   219          RSB
                     004E   220
                     004E   221
```

```
                                004E   223          .SBTTL  CLEAN_RST_STR,  Clean possible reset/start thread
                                004E   224 ;+
                                004E   225 ;
                                004E   226 ;  CLEAN_RST_STR
                                004E   227 ;
                                004E   228 ;  If a reset or start packet was expected but never arrived, clean up
                                004E   229 ;  the suspended thread that was waiting for the packet to arrive and
                                004E   230 ;  deallocate the CDRP.
                                004E   231 ;
                                004E   232 ;  INPUTS:
                                004E   233 ;
                                004E   234 ;       CX$L_RST_STR_PDT                - PDT address if expected reset or
                                004E   235 ;                                         start packet never arrived
                                004E   236 ;                                       - 0 otherwise
                                004E   237 ;       PDT$L_RSTCDRP                   - CDRP address if above field is nonzero
                                004E   238 ;
                                004E   239 ;  OUPUTS:
                                004E   240 ;
                                004E   241 ;       CX$L_RST_STR_PDT                - 0
                                004E   242 ;       PDT$L_RSTCDRP                   - 0
                                004E   243 ;       CDRP deallocated
                                004E   244 ;
                                004E   245 ;-
                                004E   246
                                004E   247 CLEAN_RST_STR:
                                004E   248
                          1F    BB  004E   249          PUSHR   #^M<R0,R1,R2,R3,R4>     ; Save regs
                54  0000'CF     D0  0050   250          MOVL    CX$L_RST_STR_PDT,R4     ; Get saved PDT address
                          15    13  0055   251          BEQL    10$                     ; No suspended thread, branch
                50  0000'CF     D0  0057   252          MOVL    CX$L_RST_STR_CDRP,R0    ; Get CDRP address              [5-002]
                          0A    13  005C   253          BEQL    5$                      ; No CDRP, branch (shouldn't happen)
             00000000'GF      16  005E   254          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                0000'CF      D4  0064   255          CLRL    CX$L_RST_STR_CDRP       ; Show CDRP is gone             [5-002]
                0000'CF      D4  0068   256 5$:      CLRL    CX$L_RST_STR_PDT        ; Show suspended thread cleaned up
                          1F    BA  006C   257 10$:    POPR    #^M<R0,R1,R2,R3,R4>     ; Restore regs
                          05    05  006E   258          RSB
                                006F   259
                                006F   260
```

```
                                006F   262          .SBTTL  CLEAN_PORT_STATE,Return port to enabled state
                                006F   263 ;+
                                006F   264 ;
                                006F   265 ;  CLEAN_PORT_STATE
                                006F   266 ;
                                006F   267 ;  If the port state has been changed to either uninit/maint or enabled/maint,
                                006F   268 ;  return the port to the enabled state. The longword CX$L_PORT_STATE_PDT is
                                006F   269 ;  non-zero if this is the case.
                                006F   270 ;
                                006F   271 ;  INPUTS:
                                006F   272 ;
                                006F   273 ;  OUTPUTS:
                                006F   274 ;
                                006F   275 ;       All registers                    - Preserved
                                006F   276 ;
                                006F   277 ;-
                                006F   278
                                006F   279 CLEAN_PORT_STATE:
                                006F   280
                          3F    BB    006F   281          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
                    54    0000'CF   D0    0071   282          MOVL    CX$L_PORT_STATE_PDT,R4  ; Get address of PDT for port whose
                                        0076   283                                          ; state has been changed
                          08    13    0076   284          BEQL    10$                     ; None, branch
                                        0078   285          CLR_MAINT_STATE                 ; Return port to normal state
                    0000'CF     D4    007F   286          CLRL    CX$L_PORT_STATE_PDT     ; Show port state is back to enabled
                          3F    BA    0083   287 10$:      POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                          05    05    0085   288          RSB
                                        0086   289
                                        0086   290
                                        0086   291
```

```
                                0086   293          .SBTTL  CANCEL_CMD_BUF, Cancel all commands in command buffer
                                0086   294 ;+
                                0086   295 ;
                                0086   296 ;  CANCEL_CMD_BUF
                                0086   297 ;
                                0086   298 ;  Routine to cancel all commands in the command buffer. This involves
                                0086   299 ;  finding any CDRP's associated with commands, pulling the CDRP's off
                                0086   300 ;  any wait queues that may have been on, and deallocating them. Connection
                                0086   301 ;  type commands which don't have a CDRP associated with them, are not touched.
                                0086   302 ;
                                0086   303 ;  INPUTS:
                                0086   304 ;
                                0086   305 ;  OUTPUTS:
                                0086   306 ;
                                0086   307 ;-
                                0086   308          .ENABLE LSB
                                0086   309
                                0086   310 CANCEL_CMD_BUF:
                                0086   311
                          7E    D4    0086   312          CLRL    -(SP)                   ; Clear a counter
                53  0000'CF    D0    0088   313          MOVL    CX$L_CMDBUFBGN,R3       ; Get start of command buffer
                      01 A3     91    008D   314          CMPB    CDC$B_DONECNT(R3),-     ; Have all commands in the command
                         63           0090   315                  CDC$B_CMDCOUNT(R3)      ; buffer been completed?
                         5F     13    0091   316          BEQL    100$                    ; Yes, branch
                         0C     C1    0093   317          ADDL3   S^#CDC$K_CMDHDRSIZ,-    ; Get first command in buffer
                53  0000'CF           0095   318                  CX$L_CMDBUFBGN,R3       ;
          0000'CF    53    D0    0099   319          MOVL    R3,CX$L_CUR_CMD         ; Save it
                                009E   320
                                009E   321 CANCEL_CMD:
                                009E   322
                                009E   323          BITW    #<FLAG_M_CMD_DONE-      ; If command is done
                                009F   324                    !FLAG_M_CONN_LOST-    ; connection error,
                                009F   325                    !FLAG_M_SCS_ERR-      ; SCS error,
                                009F   326                    !FLAG_M_PWR_FAIL>,-   ; powerfail error,
          04 A3     B800 8F     009F   327                  CDC$W_FLAG(R3)          ;
                         2F     12    00A4   328          BNEQU   CANCEL_NXT_CMD          ; Then try next cmd
                                00A6   329
                                00A6   330          $DISPATCH -
                                00A6   331                  CDC$B_OPCODE(R3),-
                                00A6   332                  TYPE=B,-
                                00A6   333                  <-
                                00A6   334                  <CDC$CONNECT,CANCEL_NXT_CMD>,-
                                00A6   335                  <CDC$DISCONNECT,CANCEL_NXT_CMD>,-
                                00A6   336                  >
                                00AE   337
                55    0C A3     D0    00AE   338          MOVL    CDC$L_CDRP(R3),R5       ; See if CDRP has been allocated
                         21     13    00B2   339          BEQLU   CANCEL_NXT_CMD          ; No, try next command
                                00B4   340          SET_CMD_FLAG REC_WAIT           ; Indicate driver was suspended
                                00BA   341
                          0106  30    00BA   342          BSBW    CHECK_LINK_FQ           ; Is CDRP queued to an SCS queue?
                       03 50    E9    00BD   343          BLBC    R0,10$                  ; No, branch
                50     65       0F    00C0   344          REMQUE  CDRP$L_FQFL(R5),R0      ; Take CDRP off SCS wait queue
                                00C3   345
                          0110  30    00C3   346 10$:     BSBW    CHECK_LINK_PQ           ; Is CDRP queued to CDB pending queue?
                       04 50    E9    00C6   347          BLBC    R0,20$                  ; No, branch
                50     44 A5    0F    00C9   348          REMQUE  CDRP$L_PENDQFL(R5),R0   ; Take CDRP off CDB pending queue
                                00CD   349
```

```
                011D   30  00CD    350 20$:    BSBW    CLEAN_CDRP              ; Release all CDRP resources
                014D   30  00D0    351         BSBW    DEALLOC_CDRP            ; Deallocate the CDRP
                  00   11  00D3    352         BRB     CANCEL_NXT_CMD          ; Try next command
                               00D5    353
                               00D5    354 CANCEL_NXT_CMD:
                               00D5    355
                    6E   96  00D5    356         INCB    (SP)                    ; One more command cleaned up
          0000'DF   6E   91  00D7    357         CMPB    (SP),@CX$L_CMDBUFBGN    ; Are all commands are cleaned up?
                    14   13  00DC    358         BEQL    100$                    ; Yes, branch
                               00DE    359
      53  0000'CF   D0   00DE    360         MOVL    CX$L_CUR_CMD,R3         ; Restore addres of current command
        50    08 A3   3C   00E3    361         MOVZWL  CDC$W_TXTLEN(R3),R0     ; Get length of current command
            53  50   C0   00E7    362         ADDL2   R0,R3                   ; Get address of next command
      0000'CF   53   D0   00EA    363         MOVL    R3,CX$L_CUR_CMD         ; Save it
                FFAC   31   00EF    364         BRW     CANCEL_CMD              ; Go clean up the next command
                               00F2    365
                    8E   D5  00F2    366 100$:   TSTL    (SP)+                   ; Clean up stack
                    05   00F4    367         RSB
                               00F5    368
                               00F5    369         .DISABLE LSB
                               00F5    370
                               00F5    371
```

```
                        00F5   373          .SBTTL  FINISH_IO,      Complete the original I/O
                        00F5   374 ;+
                        00F5   375 ;
                        00F5   376 ;  FINISH I/O
                        00F5   377 ;
                        00F5   378 ;  All commands in the command buffer have been cleaned up. Finish the
                        00F5   379 ;  original I/O that the driver was working on.
                        00F5   380 ;
                        00F5   381 ;  INPUTS:
                        00F5   382 ;
                        00F5   383 ;  OUTPUTS:
                        00F5   384 ;
                        00F5   385 ;-
                        00F5   386
                        00F5   387 FINISH_IO:
                        00F5   388
 50   0830 8F    3C     00F5   389          MOVZWL  #SS$_CANCEL,R0          ; Say it was canceled
        51      D4     00FA   390          CLRL    R1                      ; No additional status
          FFC1'    30   00FC   391          BSBW    SIO_COMPLETE_ERR        ; Complete the original I/O
                   05   00FF   392          RSB
                        0100   393
                        0100   394
                        0100   395
```

```
                                0100    397          .SBTTL  CANCEL_MAPPED_BUFFERS, Unmap all mapped buffers
                                0100    398
                                0100    399 ;+
                                0100    400 ;
                                0100    401 ;  CANCEL MAPPED BUFFERS
                                0100    402 ;
                                0100    403 ;  Pull off any CDRP's which remain on the mapped buffer list. Unmap the
                                0100    404 ;  associated buffer and deallocate the CDRP.
                                0100    405 ;
                                0100    406 ;-
                                0100    407
                                0100    408 CANCEL_MAPPED_BUFFERS:
                                0100    409
    50   0000'CF    DE          0100    410          MOVAL   CX$Q_MAPLST,R0              ; Get head of map list
      55    00 B0    0F          0105    411 10$:     REMQUE  @(R0),R5                    ; Pull of a mapped buffer CDRP
            0C       1D          0109    412          BVS     20$                         ; Map queue empty, branch
      55    BC A5    DE          010B    413          MOVAL   -CDRP$L_PENDQFL(R5),R5      ; Back up to top of CDRP
            00DB     30          010F    414          BSBW    CLEAN_CDRP                  ; Clean up the CDRP
            010B     30          0112    415          BSBW    DEALLOC_CDRP                ; Deallocate the CDRP
              EE     11          0115    416          BRB     10$                         ; Try again
                     05          0117    417 20$:     RSB
                                 0118    418
                                 0118    419
```

```
                                0118   421          .SBTTL  CANCEL_CONNECTIONS, Clean up all remaining connections
                                0118   422 ;+
                                0118   423 ;
                                0118   424 ;  CANCEL_CONNECTIONS
                                0118   425 ;
                                0118   426 ;  The I/O is now complete. The last thing to do is discgard all connections.
                                0118   427 ;  This involves scanning the entire connection database for active CDB's,
                                0118   428 ;  cleaning up any CDRP's queued to CDB pending queues, and deallocating the
                                0118   429 ;  CDB's.
                                0118   430 ;
                                0118   431 ;  R7 is used as an index into the local port list of the driver. Each
                                0118   432 ;  entry in the local port list points to a CDB list for a particular local
                                0118   433 ;  port (PAA0, PAB0, etc).
                                0118   434 ;
                                0118   435 ;  R8 is used as an index into the CDB list. Each entry in the CDB list
                                0118   436 ;  points to a POSSIBLE CDB for a connection to a remote node.
                                0118   437 ;
                                0118   438 ;-
                                0118   439
                                0118   440 CANCEL_CONNECTIONS:
                                0118   441
      57   FFFFFFFF'8F    D0   0118   442          MOVL    #MAX_LOC_PORTS-1,R7     ; Get maximum number of local ports
        59   0000'CF47    D0   011F   443 10$:     MOVL    CX$L_LOCPRTLST[R7],R9   ; Get CDB list for this port
      58   FFFFFFFF'8F    D0   0125   444          MOVL    #MAX_NODES-1,R8         ; Get maximum number of remote nodes
           55    6948    D0   012C   445 20$:     MOVL    (R9)[R8],R5             ; Get addr of possible CDB
                   03    13   0130   446          BEQLU   30$                     ; No connection, try next node
                 0043    30   0132   447          BSBW    KILL_CDB                ; Clean up this connection
                F4 58    F4   0135   448 30$:     SOBGEQ  R8,20$                  ; Repeat for all remote ports
                E4 57    F4   0138   449          SOBGEQ  R7,10$                  ; Repeat for all local ports
                         05   013B   450          RSB
                              013C   451
                              013C   452
                              013C   453
```

```
                                013C   455          .SBTTL  START_POLLER,   Restart the configuration poller
                                013C   456 ;+
                                013C   457 ;
                                013C   458 ;  START_POLLER
                                013C   459 ;
                                013C   460 ;  If the poller is stopped, it should be restarted. In a non-cint
                                013C   461 ;  configuration, the ENABLE_POLLER simply results in an SS$_ILLIOFUNC
                                013C   462 ;  status being returned.
                                013C   463 ;
                                013C   464 ;  The PDT address is obtained by issuing a CONFIG_PTH call for PAA0/
                                013C   465 ;  remote port 0, and if that fails, PAB0/remote port 0. If both fail,
                                013C   466 ;  the poller can't be started.
                                013C   467 ;
                                013C   468 ;-
                                013C   469
                                013C   470 START_POLLER:
                                013C   471
                          55    02   D0  013C   472          MOVL    #2,R5                   ; Try this only twice
                     7E   30414150 8F   D0  013F   473          MOVL    #^A/PAA0/,-(SP)         ; Push local port name on stack
                          7E         7C  0146   474 10$:      CLRQ    -(SP)                   ; Use value of zero for remote port #
                                     0148   475          CONFIG_PTH -                    ; Issue config path
                                     0148   476                  STAADR=(SP),-           ; Input array
                                     0148   477                  OUTBUF=0                ; No output wanted
                          5E    0C   C0  0153   478          ADDL    #12,SP                  ; Clean off stack
                             0E 50   E8  0156   479          BLBS    R0,20$                  ; Success, branch
                     7E   30424150 8F   D0  0159   480          MOVL    #^A/PAB0/,-(SP)         ; Try again, this time with PAB0
                          E3 55   F5   0160   481          SOBGTR  R5,10$                  ; But not if PAB0 failed
                             8E   D5   0163   482          TSTL    (SP)+                   ; Clean off stack
                             10   11   0165   483          BRB     30$                     ; Can't start poller, branch
                     54   2C A1   D0  0167   484 20$:      MOVL    PB$L_PDT(R1),R4         ; Get PDT address
                                     0168   485          ENABLE_POLLER                   ; Start the configuration poller
                                05   0177   486 30$:      RSB
                                     0178   487
                                     0178   488
                                     0178   489
                                     0178   490
```

```
                     0178   492          .SBTTL  KILL_CDB,      Clean Up A Connection
                     0178   493 ;+
                     0178   494 ;
                     0178   495 ;  KILL_CDB
                     0178   496 ;
                     0178   497 ;  This routine cleans up all suspended threads and resources associated with
                     0178   498 ;  a connection.
                     0178   499 ;
                     0178   500 ;  If a CONNECT is in progress for the connection block, the connection can
                     0178   501 ;  NOT be cleaned up until the suspended connect call completes. Therefore,
                     0178   502 ;  set the CANCELED flag in the CDB status field and return.
                     0178   503 ;
                     0178   504 ;  If a CONNECT is not in progress for this connection block, then all
                     0178   505 ;  suspended threads (CDRP's) are removed from the CDB, all associated
                     0178   506 ;  resources are returned, and a DISCONNECT is issued. If the disconnect
                     0178   507 ;  succeeds, the CDB is deallocated and removed from the connection database.
                     0178   508 ;  Otherwise, the CDB remains in the database, and is removed when a subsequent
                     0178   509 ;  call eventually succeeds.
                     0178   510 ;
                     0178   511 ;  INPUTS:
                     0178   512 ;
                     0178   513 ;       R5              - CDB address
                     0178   514 ;
                     0178   515 ;  OUTPUTS:
                     0178   516 ;
                     0178   517 ;       R0-R2           - Destroyed
                     0178   518 ;
                     0178   519 ;-
                     0178   520
                     0178   521 KILL_CDB::
                     0178   522
                     0178   523          ASSUME  CDB$L_CDT+4 EQ CDB$L_PDT
                     0178   524
          04   A8    0178   525          BISW2   #CDB_M_CNCL,-           ; Show that a cancel has occurred
        0C A5         017A   526                  CDB$W_STATUS(R5)        ; for this connection
          00   E0    017C   527          BBS     #CDB_V_CONN,-           ; Branch if CONNECT in progress
     1E 0C A5         017E   528                  CDB$W_STATUS(R5),10$    ; for this connection
          01   E2    0181   529          BBSS    #CDB_V_DISC,-           ; Branch if DISCONNECT in progress
     19 0C A5         0183   530                  CDB$W_STATUS(R5),10$    ; for this connection
                     0186   531
        0017   30    0186   532          BSBW    CLEAN_CDB               ; Clean up connection resources
 53     1C A5   7D    0189   533          MOVQ    CDB$L_CDT(R5),R3        ; Restore CDT, PDT address
                     018D   534          DISCONNECT                      ; Issue a dissconnect
        24 B5   D4    0193   535          CLRL    @CDB$L_CDBSLOT(R5)      ; Clear addr of CDB in CONLST
     50   55   D0    0196   536          MOVL    R5,R0                   ; Set CDB addr for call
00000000'GF   16    0199   537          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDB
                     019F   538
                05   019F   539 10$:     RSB
                     01A0   540
                     01A0   541
```

```
                        01A0    543          .SBTTL  CLEAN_CDB,      Clean pending CDRP's off CDB
                        01A0    544 ;+
                        01A0    545 ;
                        01A0    546 ;  CLEAN_CDB
                        01A0    547 ;
                        01A0    548 ;  This routine pulls any pending CDRP's off the CDB and deallocates all
                        01A0    549 ;  resources associated with each CDRP.
                        01A0    550 ;
                        01A0    551 ;  INPUTS:
                        01A0    552 ;
                        01A0    553 ;       R5              - CDB address
                        01A0    554 ;
                        01A0    555 ;  OUTPUTS:
                        01A0    556 ;
                        01A0    557 ;       R0              - Destroyed
                        01A0    558 ;
                        01A0    559 ;--
                        01A0    560
                        01A0    561 CLEAN_CDB::
                        01A0    562
          2F   BB       01A0    563          PUSHR   #^M<R0,R1,R2,R3,R5>     ; Save regs
     53   55   D0       01A2    564          MOVL    R5,R3                   ; Save CDB address
55    2A B3   0F       01A5    565 10$:     REMQUE  @CDB$L_CDRPQFL(R3),R5   ; Pull CDRP off pending queue
        15    1D       01A9    566          BVS     30$                     ; Queue empty, branch
55    BC A5   DE       01AB    567          MOVAL   -CDRP$L_PENDQFL(R5),R5  ; Back up to FLINK in CDRP
       0011   30       01AF    568          BSBW    CHECK_LINK_FQ           ; See if CDRP is on an SCS queue
      03 50   E9       01B2    569          BLBC    R0,20$                  ; No, branch
55      65    0F       01B5    570          REMQUE  CDRP$L_FQFL(R5),R5      ; Pull CDRP off SCS wait queue
      0032    30       01B8    571 20$:     BSBW    CLEAN_CDRP              ; Clean resources associated with CDRP
      0062    30       01BB    572          BSBW    DEALLOC_CDRP            ; Deallocate the CDRP
        E5    11       01BE    573          BRB     10$                     ; Continue for all CDRP's on CDB
        2F    BA       01C0    574 30$:     POPR    #^M<R0,R1,R2,R3,R5>     ; Restore registers
              05       01C2    575          RSB
                       01C3    576
                       01C3    577
                       01C3    578
```

```
                        01C3    580          .SBTTL  CHECK_LINK_FQ,  Check for CDRP on resource queue
                        01C3    581 ;+
                        01C3    582 ;
                        01C3    583 ;  CHECK_LINK_FQ
                        01C3    584 ;
                        01C3    585 ;  This routine checks to see if a CDRP is linked to a resource wait queue.
                        01C3    586 ;  CDRP's can not be blindly removed from queues as it tends to cause VMS
                        01C3    587 ;  to crash.
                        01C3    588 ;
                        01C3    589 ;  INPUTS:
                        01C3    590 ;
                        01C3    591 ;       R5              - Address of FQFL in CDRP
                        01C3    592 ;
                        01C3    593 ;  OUTPUTS:
                        01C3    594 ;
                        01C3    595 ;       R0              - 1 if queued
                        01C3    596 ;                       - 0 otherwise
                        01C3    597 ;
                        01C3    598 ;-
                        01C3    599
                        01C3    600 CHECK_LINK_FQ::
                        01C3    601
   50    01    9A     01C3    602          MOVZBL  #1,R0                   ; Assume CDRP queued
04 A5    00    D1     01C6    603          CMPL    #0,CDRP$L_FQBL(R5)      ; Is link zero
         07    13     01CA    604          BEQLU   10$                     ; Yes, branch
55   04 B5    D1     01CC    605          CMPL    @4(R5),R5               ; Check for linkage into queue
         01    12     01D0    606          BNEQU   10$                     ; No, branch
               05     01D2    607          RSB                             ; Return
                      01D3    608
50    D4       01D3    609 10$:     CLRL    R0                      ; Say it is NOT queued
               05     01D5    610          RSB                             ; Return
                      01D6    611
                      01D6    612
                      01D6    613
```

```
                                01D6   615          .SBTTL  CHECK_LINK_PQ,  Check for CDRP on pending queue
                                01D6   616 ;+
                                01D6   617 ;
                                01D6   618 ;  CHECK_LINK_PQ
                                01D6   619 ;
                                01D6   620 ;  This routine checks to see if a CDRP is linked to a pending queue
                                01D6   621 ;  CDRP's can not be blindly removed from queues as it tends to cause VMS
                                01D6   622 ;  to crash.
                                01D6   623 ;
                                01D6   624 ;  INPUTS:
                                01D6   625 ;
                                01D6   626 ;       R5              - Address of FQFL in CDRP
                                01D6   627 ;
                                01D6   628 ;  OUTPUTS:
                                01D6   629 ;
                                01D6   630 ;       R0              - 1 if queued
                                01D6   631 ;                       - 0 otherwise
                                01D6   632 ;
                                01D6   633 ;-
                                01D6   634
                                01D6   635 CHECK_LINK_PQ::
                                01D6   636
48 A5    00    D1               01D6   637          CMPL    #0,CDRP$L_PENDQBL(R5)   ; Is pending queue flink non zero?
         0E    13               01DA   638          BEQLU   10$                     ; No, branch
50   44 A5    DE               01DC   639          MOVAL   CDRP$L_PENDQFL(R5),R0   ; Get Address of pending queue flink
50   48 B5    D1               01E0   640          CMPL    @CDRP$L_PENDQBL(R5),R0  ; Check for linkage into queue
         04    12               01E4   641          BNEQU   10$                     ; No, branch
    50   01    9A               01E6   642          MOVZBL  #1,R0                   ; Show CDRP is queued
               05               01E9   643          RSB
                                01EA   644
         50   D4               01EA   645 10$:     CLRL    R0                      ; Show CDRP is NOT queued
               05               01EC   646          RSB
                                01ED   647
                                01ED   648
                                01ED   649
```

```
                        01ED   651          .SBTTL  CLEAN_CDRP,     Deallocate CDRP resources
                        01ED   652 ;+
                        01ED   653 ;
                        01ED   654 ;  CLEAN_CDRP
                        01ED   655 ;
                        01ED   656 ;  This routine deallocates all resources held for all types of CDRPs.
                        01ED   657 ;
                        01ED   658 ;  INPUTS:
                        01ED   659 ;
                        01ED   660 ;       R5              - Address of for queue flink in CDRP
                        01ED   661 ;
                        01ED   662 ;  OUTPUTS:
                        01ED   663 ;
                        01ED   664 ;       All registers preserved
                        01ED   665 ;
                        01ED   666 ;-
                        01ED   667
                        01ED   668 CLEAN_CDRP::
                        01ED   669
          1F   BB       01ED   670          PUSHR   #^M<R0,R1,R2,R3,R4>     ; Save regs
54   4C A5   D0       01EF   671          MOVL    CDRP$L_PDT(R5),R4       ; Restore CDRP address
50   1C A5   D0       01F3   672          MOVL    CDRP$L_MSG_BUF(R5),R0   ; Buffer used?
          0C   13       01F7   673          BEQLU   10$                     ; No, branch
     50   20   C2       01F9   674          SUBL    #SCS$K_APPL_BASE - SCS$B_PPD,R0
                        01FC   675                                          ; Back up to top of buffer      [5-001]
          3B   91       01FC   676          CMPB    #DYN$C_CIDG,-           ; Is this a datagram?
        0A A0           01FE   677                  PPD$B_TYPE(R0)          ;
          18   13       0200   678          BEQLU   30$                     ; Yes, brancn
                        0202   679          DEALLOC_MSG_BUF                 ; Deallocate the message buffer
                        0205   680
2C A5   D5       0205   681 10$:    TSTL    CDRP$L_LBUFH_AD(R5)     ; Is there a mapping resource?
          03   13       0208   682          BEQLU   20$                     ; No, branch
                        020A   683          UNMAP                           ; Return mapping resources
                        020D   684
20 A5   D5       020D   685 20$:    TSTL    CDRP$L_RSPID(R5)        ; Any RSPID?
          0B   13       0210   686          BEQLU   40$                     ; No, branch
                        0212   687          DEALLOC_RSPID                   ; Return RSPID
          03   11       0218   688          BRB     40$
                        021A   689
                        021A   690 30$:    DEALLOC_DG_BUF                  ; Return buffer
                        021D   691
          1F   BA       021D   692 40$:    POPR    #^M<R0,R1,R2,R3,R4>     ; Restore regs
               05       021F   693          RSB
                        0220   694
                        0220   695
                        0220   696
                        0220   697
```

```
                                0220   699          .SBTTL  DEALLOC_CDRP,   Deallocate a CDRP appropriately
                                0220   700 ;+
                                0220   701 ;
                                0220   702 ;  DEALLOC_CDRP
                                0220   703 ;
                                0220   704 ;  This routine checks the type of CDRP allocated, and deallocates it
                                0220   705 ;  appropriately. Note that transfer type CDRP's have an IRP portion
                                0220   706 ;  at a negaitive offset from the start of the CDRP.
                                0220   707 ;
                                0220   708 ;  INPUTS:
                                0220   709 ;
                                0220   710 ;       R5                      - CDRP address
                                0220   711 ;
                                0220   712 ;  OUTPUTS:
                                0220   713 ;
                                0220   714 ;       All registers           - Preserved
                                0220   715 ;
                                0220   716 ;-
                                0220   717
                                0220   718 DEALLOC_CDRP::
                                0220   719
            0F   BB             0220   720          PUSHR   #^M<R0,R1,R2,R3>        ; Save regs
      50    55   D0             0222   721          MOVL    R5,R0                   ; Assume normal type CDRP
        2C  A5   D5             0225   722          TSTL    CDRP$L_LBUFH_AD(R5)     ; Is it a XFR type
            04   13             0228   723          BEQL    10$                     ; No, branch
      50    A0   A5   DE        022A   724          MOVAL   -IRP$K_CDRP(R5),R0      ; Back up to top of IRP
00000000'GF    16             022E   725 10$:     JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
            0F   BA             0234   726          POPR    #^M<R0,R1,R2,R3>        ; Restore regs
                 05             0236   727          RSB                             ; Return to caller
                                0237   728
                                0237   729
                                0237   730          .END
```

```
$$$CURSIZ                       =  000001C4
$$$NEWSIZ                       =  000001D0
BIT...                          =  0000000D
CAN$INITCINT                       00000032 R       01
CANCEL_CMD                         0000009E R       01
CANCEL_CMD_BUF                     00000086 R       01
CANCEL_CONNECTIONS                 00000118 R       01
CANCEL_MAPPED_BUFFERS              00000100 R       01
CANCEL_NXT_CMD                     000000D5 R       01
CDB$B_CDRPCNT                      00000028
CDB$B_RSTNADR                      0000000E
CDB$B_TYPE                         0000000A
CDB$K_LEN                          00000032
CDB$L_CDBSLOT                      00000024
CDB$L_CDRPQBL                      0000002E
CDB$L_CDRPQFL                      0000002A
CDB$L_CDT                          0000001C
CDB$L_PDT                          00000020
CDB$L_QBL                          00000004
CDB$L_QFL                          00000000
CDB$T_LPRTNAM                      00000016
CDB$W_SIZE                         00000008
CDB$W_STATUS                       0000000C
CDB_M_CNCL                      =  00000004
CDB_V_CONN                      =  00000000
CDB_V_DISC                      =  00000001
CDC$ARMCINTPATH                    0000001B
CDC$B_CFGCBLSTS                    00000021
CDC$B_CFGP0LBSTS                   0000002A
CDC$B_CFGP0STS                     0000001F
CDC$B_CFGP1LBSTS                   0000002B
CDC$B_CFGP1STS                     00000020
CDC$B_CMDCOUNT                     00000000
CDC$B_CNTFLG                       0000001F
CDC$B_COUNT                        00000015
CDC$B_CTPOPCODE                    00000014
CDC$B_DELAY                        00000019
CDC$B_DONECNT                      00000001
CDC$B_DSTPORT                      00000001
CDC$B_EXTEND                       0000001C
CDC$B_FDTCNT                       00000002
CDC$B_GARBAGE_FLAG                 00000014
CDC$B_NOACTFLAG                    0000001E
CDC$B_NODEADR                      00000014
CDC$B_OPCODE                       00000000
CDC$B_OTHERNODE                    0000001E
CDC$B_PATH                         0000001C
CDC$B_PKTMULT                      0000001F
CDC$B_PKTSIZ                       0000001D
CDC$B_POLLERFLAG                   0000001E
CDC$B_PROTREV                      00000036
CDC$B_PROTVER                      00000035
CDC$B_PROTYPE                      00000034
CDC$B_QNUMBER                      0000001F
CDC$B_RST_PORT                     00000015
CDC$B_SIOCNT                       00000003
CDC$B_TYPE                         00000014
CDC$CARRIER                        00000018
CDC$CONFIGDATA                     00000007
CDC$CONNECT                        00000008
CDC$C_CMDSPEC                      00000014
CDC$DISCONNECT                     00000009
CDC$GARBAGE                        0000000B
CDC$INITCINT                       0000001A
CDC$K_CMDHDRSIZ                    0000000C
CDC$K_CMDSPEC                      00000014
CDC$K_DATA                         00000014
CDC$LRGPACKET                      00000019
CDC$L_BUFADR                       00000024
CDC$L_BUFLENGTH                    00000020
CDC$L_BUFLNAME                     00000024
CDC$L_BUFLOFSET                    00000028
CDC$L_BUFRNAME                     0000002C
CDC$L_BUFROFSET                    00000030
CDC$L_CDB_AD                       0000000C
CDC$L_CDRP                         0000000C
CDC$L_CFGHWTYPE                    00000022
CDC$L_CFGPRTMSK                    00000026
CDC$L_CNTRDISCDG                   00000038
CDC$L_CNTRP0ACK                    00000020
CDC$L_CNTRP0NAK                    00000024
CDC$L_CNTRP0NORSP                  00000028
CDC$L_CNTRP1ACK                    0000002C
CDC$L_CNTRP1NAK                    00000030
CDC$L_CNTRP1NORSP                  00000034
CDC$L_CTPREF                       00000015
CDC$L_DISCONREAS                   00000019
CDC$L_LPORT                        00000010
CDC$L_PORTHDR                      00000004
CDC$L_R0STATUS                     0000000C
CDC$L_R1STATUS                     00000010
CDC$L_RESERV04                     00000004
CDC$L_RESERV08                     00000008
CDC$L_RESERV12                     0000000C
CDC$L_SEQNUM                       00000001
CDC$L_STARTADR                     00000020
CDC$L_SVAPTE                       00000018
CDC$MAINTSTATE                     0000000F
CDC$MAPBUF                         00000005
CDC$MAPMBUF                        00000010
CDC$NAK                            00000016
CDC$NOACK                          00000017
CDC$NOACT                          0000000E
CDC$NORSP                          00C00015
CDC$OVERSIZEPKT                    0000001D
CDC$POLLER                         0000000E
CDC$RDCINTPATH                     0000001C
CDC$READCNT                        0000000A
CDC$REQDATA                        00000004
CDC$REQMDATA                       00000013
CDC$RESET                          0C00000C
CDC$SELADR                         00000014
CDC$SENDDATA                       00000003
CDC$SENDDG                         00000002
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	00000237 (567.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000360 (864.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
BIT...	=0000000D	88 (1)	88 (1)
CAN$INITCINT	00000032-R	206 (4)	#-168 (3)
CANCEL_CMD	0000009E-R	321 (7)	#-364 (7)
CANCEL_CMD_BUF	00000086-R	310 (7)	#-173 (3)
CANCEL_CONNECTIONS	00000118-R	440 (10)	#-176 (3)
CANCEL_MAPPED_BUFFERS	00000100-R	408 (9)	#-175 (3)
CANCEL_NXT_CMD	000000D5-R	354 (7)	#-328 (7) 336 (7) #-339 (7) #-352 (7)
CDB$L_CDBSLOT	00000024		#-535 (12)
CDB$L_CDRPQFL	0000002A		565 (13)
CDB$L_CDT	0000001C		523 (12) #-533 (12)
CDB$L_PDT	00000020		523 (12)
CDB$W_STATUS	0000000C		#-526 (12) 528 (12) 530 (12)
CDB_M_CNCL	=00000004		#-525 (12)
CDB_V_CONN	=00000000		#-527 (12)
CDB_V_DISC	=00000001		#-529 (12)
CDC$B_CMDCOUNT	00000000		#-315 (7)
CDC$B_DONECNT	00000001		#-314 (7)
CDC$B_OPCODE	00000000		#-336 (7)
CDC$CONNECT	00000008		336 (7)
CDC$DISCONNECT	00000009		336 (7)
CDC$K_CMDHDRSIZ	0000000C		#-317 (7)
CDC$L_CDRP	0000000C		#-338 (7)
CDC$W_FLAG	00000004		#-327 (7) #-340 (7)
CDC$W_TXTLEN	00000008		#-361 (7)
CDRP$L_FQBL	=00000004		#-603 (14)
CDRP$L_FQFL	=00000000		344 (7) 570 (13)
CDRP$L_LBUFH_AD	=0000002C		#-681 (16) #-722 (17)
CDRP$L_MSG_BUF	=0000001C		#-672 (16)
CDRP$L_PDT	0000004C		#-671 (16)
CDRP$L_PENDQBL	00000048		#-637 (15) #-640 (15)
CDRP$L_PENDQFL	00000044		348 (7) 413 (9) 567 (13) 639 (15)
CDRP$L_RSPID	=00000020		#-685 (16)
CHECK_LINK_FQ	000001C3-R	600 (14)	#-342 (7) #-568 (13)
CHECK_LINK_PQ	000001D6-R	635 (15)	#-346 (7)
CLEAN_CDB	000001A0-R	561 (13)	#-532 (12)
CLEAN_CDRP	000001ED-R	668 (16)	#-350 (7) #-414 (9) #-571 (13)
CLEAN_PORT_STATE	0000006F-R	279 (6)	#-170 (3)
CLEAN_RST_STR	0000004E-R	247 (5)	#-169 (3)
CLEAR_CLER_REG	00000000-XR		#-214 (4)
CLEAR_GOU_BIT	00000000-XR		#-213 (4)
CLEAR_GO_BIT	00000000-XR		#-212 (4)
CLEAR_STS_REG	00000000-XR		#-215 (4)
COM$DRVDEALMEM	00000000-XR		254 (5) 537 (12) 725 (17)
CX$L_CINT_PDT	00000000-XR		#-209 (4) #-216 (4)
CX$L_CMDBUFBGN	00000000-XR		#-313 (7) #-318 (7) #-357 (7)
CX$L_CUR_CMD	00000000-XR		#-319 (7) #-360 (7) #-363 (7)
CX$L_LOCPRTLST	00000000-XR		#-443 (10)
CX$L_PORT_STATE_PDT	00000000-XR		#-282 (6) #-286 (6)
CX$L_RST_STR_CDRP	00000000-XR		#-252 (5) #-255 (5)
CX$L_RST_STR_PDT	00000000-XR		#-250 (5) #-256 (5)

CX$L_STATE	00000000-XR			166	(3)	172	(3)	#-180	(3)
CX$Q_MAPLST	00000000-XR			410	(9)				
CX_CANCELIO	00000000-R	163	(3)						
DEALLOC_CDRP	00000220-R	718	(17)	#-351	(7)	#-415	(9)	#-572	(13)
DYN$C_CIDG	=0000003B			#-676	(16)				
FINISH_IO	000000F5-R	387	(8)	#-174	(3)				
FLAG_M_CMD_DONE	=00008000			#-324	(7)				
FLAG_M_CONN_LOST	=00000800			#-325	(7)				
FLAG_M_PWR_FAIL	=00002000			#-326	(7)				
FLAG_M_REC_WAIT	=00000400			#-340	(7)				
FLAG_M_SCS_ERR	=00001000			#-326	(7)				
IRP$K_CDRP	=00000060			724	(17)				
KILL_CDB	00000178-R	521	(12)	#-447	(10)				
MAX_LOC_PORTS	00000000-XR			#-442	(10)				
MAX_NODES	00000000-XR			#-444	(10)				
MNT$NORMAL_STATE	00000000-XR			#-285	(6)				
MNT$POLLER_ON	00000000-XR			#-485	(11)				
PB$L_PDT	=0000002C			#-484	(11)				
PDT$L_DEALLOCDG	=0000001C			690	(16)				
PDT$L_DEALLOMSG	=00000020			679	(16)				
PDT$L_MAINTFCN	=00000078			285	(6)	485	(11)		
PDT$L_UNMAP	=00000064			683	(16)				
PPD$B_TYPE	0000000A			#-677	(16)				
SCS$B_PPD	=FFFFFFE0			#-674	(16)				
SCS$CONFIG_PTH	00000000-XR			477	(11)				
SCS$DEALL_RSPID	00000000-XR			687	(16)				
SCS$DISCONNECT	00000000-XR			534	(12)				
SCS$K_APPL_BASE	=00000000			#-674	(16)				
SIO_COMPLETE_ERR	00000000-XR			#-391	(8)				
SIZ...	=00000001	88	(1)	88	(1)				
SS$_CANCEL	=00000830			#-389	(8)				
START_POLLER	0000013C-R	470	(11)	#-177	(3)				
STATE_M_ALTGRBG	=00000004	88	(1)						
STATE_M_CANCEL	=00000008	88	(1)	#-179	(3)				
STATE_M_CMAP	=00000002	88	(1)						
STATE_M_FDT	=00000400	88	(1)						
STATE_M_NOTPROC	=00001000	88	(1)						
STATE_M_RMAP	=00000001	88	(1)						
STATE_M_SIO	=00000800	88	(1)						
STATE_M_STOP	=00000200	88	(1)						
STATE_V_ALTGRBG	=00000002	88	(1)						
STATE_V_CANCEL	=00000003	88	(1)	#-165	(3)				
STATE_V_CMAP	=00000001	88	(1)	#-171	(3)				
STATE_V_FDT	=0000000A	88	(1)						
STATE_V_NOTPROC	=0000000C	88	(1)						
STATE_V_RMAP	=00000000	88	(1)						
STATE_V_SIO	=0000000B	88	(1)						
STATE_V_STOP	=00000009	88	(1)						

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$BND	1	336 (7)	336 (7)
$BND1	1	336 (7)	336 (7)
$BND2	1	336 (7)	336 (7)
$CDCDEF	8	85 (1)	85 (1)
$CDRPDEF	6	65 (1)	65 (1)
$CDTDEF	4	66 (1)	66 (1)
$CINTREGDEF	3	78 (1)	78 (1)
$CXCDBDEF	2	86 (1)	86 (1)
$CXCDRPDEF	1	87 (1)	87 (1)
$CXSTATEDEF	1	88 (1)	88 (1)
$DEFINI	1	65 (1)	65 (1) 66 (1) 67 (1) 68 (1)
			69 (1) 70 (1) 71 (1) 72 (1)
			73 (1) 78 (1) 79 (1) 80 (1)
			85 (1) 86 (1) 87 (1)
$DISPATCH	2	331 (7)	331 (7)
$DSP1	1	336 (7)	336 (7)
$DSP2	1	336 (7)	336 (7)
$DYNDEF	7	67 (1)	67 (1)
$IODEF	17	68 (1)	68 (1)
$IRPDEF	7	69 (1)	69 (1)
$MOVEADR	1	477 (11)	477 (11)
$PAPDTDEF	5	79 (1)	79 (1)
$PBDEF	3	70 (1)	70 (1)
$PDTDEF	4	71 (1)	71 (1)
$PPDDEF	10	80 (1)	80 (1)
$SCSDEF	4	72 (1)	72 (1)
$SSDEF	21	73 (1)	73 (1)
$VIELD1	1		88 (1)
ASSUME	1		523 (12)
CLR_MAINT_STATE	1	285 (6)	285 (6)
CONFIG_PTR	1	476 (11)	476 (11)
DEALLOC_DG_BUF	1	690 (16)	690 (16)
DEALLOC_MSG_BUF	1	679 (16)	679 (16)
DEALLOC_RSPID	1	687 (16)	687 (16)
DISCONNECT	1	534 (12)	534 (12)
ENABLE_POLLER	1	485 (11)	485 (11)
SET_CMD_FLAG	1	340 (7)	340 (7)
UNMAP	1	683 (16)	683 (16)
_VIELD	1		88 (1)

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	90	00:00:00.34	00:00:01.19
Command processing	139	00:00:00.65	00:00:01.39
Pass 1	637	00:00:24.43	00:00:28.85
Symbol table sort	0	00:00:03.08	00:00:03.14

Pass 2	126	00:00:03.60	00:00:04.09
Symbol table output	38	00:00:00.26	00:00:00.28
Psect synopsis output	3	00:00:00.03	00:00:00.03
Cross-reference output	26	00:00:00.48	00:00:00.64
Assembler run totals	1062	00:00:32.87	00:00:39.62

The working set limit was 2400 pages.
112452 bytes (220 pages) of virtual memory were used to buffer the intermediate code.
There were 110 pages of symbol table space allocated to hold 2008 non-local and 33 local symbols.
730 source lines were read in Pass 1, producing 17 object records in Pass 2.
117 pages of virtual memory were used to define 45 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	3
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	13
DRB2:[SHULL.EVXCI.CXDRIVER]CXLIB.MLB;3	8
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	8
TOTALS (all libraries)	32

2391 GETS were required to define 32 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS/CROSS CXCANCEL+CXLIB/LIB+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB
```

Table of contents

(1) 74 DEFINITIONS
(2) 120 CX_STARTIO, Start I/O routine
(3) 434 SIO_CONFIG, Gather configuration data
(4) 501 SIO_CONNECT, Connect to remote
(5) 591 SIO_DISC, Disconnect from remote
(6) 694 SIO_MSG, Send message
(7) 761 SIO_DG, Send datagram
(8) 819 SIO_MAP, Map buffer
(9) 878 SIO_UNMAP, Unmap buffer
(10) 924 SIO_SNDDAT, Send data
(10) 925 SIO_REQDAT, Request data
(11) 983 SIO_SNDMDAT, Send maintenance data
(11) 984 SIO_REQMDAT, Request maintenance data
(12) 1040 SIO_READCNT, Read port counters
(13) 1133 SIO_RESET, Send reset
(14) 1167 SIO_START, Send start
(15) 1203 SIO_GARBAGE, Read the garbage buffer
(16) 1250 SIO_MOVCTP, Copy CTP data to MSG/DG buffer
(17) 1301 SIO_SETUP_SCS, Setup SCS interface
(18) 1339 MARK_CMD_DONE, Mark this command as done
(19) 1368 CHECK_VC, Check for virtual circuit
(19) 1369 GET_PDT_ADDR, Get local PDT address
(20) 1407 XFR_SETUP, Setup data XFR parameters
(21) 1480 MNT_XFR_SETUP, Setup maintenance transfer CDRP
(22) 1546 FIND_CDRP_XFR, Locate XFR CDRP on the map list
(23) 1611 ALLOC_CDRP_CTP, Allocate a CTP type CDRP
(24) 1659 ALLOC_CDRP_XFR, Allocate a CDRP for a XFR command
(25) 1721 ALLOC_CDB, Allocate connection block
(26) 1782 SET_RSP_STATUS, Set response status
(26) 1783 SET_RSP_STATUS_2, Set full response status
(27) 1820 CHECK_RST_STR, Prepare for arrival of reset or start packet
(28) 1867 CATCH_RST_STR, Catch reset or start packet

```
                        1          .TITLE  CXCMD   Command Start I/O
                        2          .IDENT  'V6-000'
                        3
                        4  ;*****************************************************************************
                        5  ;
                        6  ;                          COPYRIGHT (c) 1981, 1984 BY
                        7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
                        8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
                        9  ;
                       10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
                       11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
                       12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
                       13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
                       14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
                       15  ;
                       16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
                       17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
                       18  ;
                       19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
                       20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
                       21  ;
                       22  ;*****************************************************************************
                       23  ;
                       24  ;++
                       25  ;
                       26  ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
                       27  ;
                       28  ;
                       29  ; ABSTRACT:      Command STARTIO
                       30  ;
                       31  ; AUTHOR:        Richard Hennessy  27-Jul-1982
                       32  ;                Base Systems Diagnostic Engineering
                       33  ;
                       34  ; MODIFIED BY:   Jim Klumpp  6-MAY-83
                       35  ;
                       36  ;       6-000    Dave Shull      07-July-1984
                       37  ;                VMS V4 Modifications/Release
                       38  ;
                       39  ;       5-007    Dave Shull      07-July-1984
                       40  ;                Modified routine ALLOC_CDB to move all 4 characters of local
                       41  ;                port name (ie., PAA0) into CDB versus only the first 3
                       42  ;                characters.  VMS V4 expects the 4 character (unit number) to
                       43  ;                be set-up as ASCII equivalent versus as an integer.
                       44  ;
                       45  ;       5-006    Dave Shull      14-June-1984
                       46  ;                Modified routine CHECK_VC to not make 0 on PAx0 an integer but
                       47  ;                to leave it as an Ascii '0'.  This is needed in VMS V4.
                       48  ;
                       49  ;       5-005    Dave Shull      14-June-1984
                       50  ;                Fixed routines to pass start of PPD layer for SCS Read Counter
                       51  ;                call.
                       52  ;
                       53  ;       5-004    Dave Shull      24-March-1984
                       54  ;                Removed instruction 'SUBL2 #SCS$K_APPL_BASE-SCS$B_PPD,R2' from
                       55  ;                routine SIO_READCNT as the SCS read counter routine in V4 - FT1
                       56  ;                expects R2 to point to the Application Message in the Port
                       57  ;                Envelope.
```

```
                        0000     58 ;
                        0000     59 ;       5-003   Dave Shull      22-March-1984
                        0000     60 ;               Removed $PAPBDEF
                        0000     61 ;
                        0000     62 ;       5-002   Dave Shull      7-March-1984
                        0000     63 ;               Modified routine CATCH_RST_STR to save CDRP address in
                        0000     64 ;               CX$L_RST_STR_CDRP.
                        0000     65 ;
                        0000     66 ;       5-001   Dave Shull      9-Feb-1984
                        0000     67 ;               Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SYS$B_PPD
                        0000     68 ;               as in VMS V4 all SCS$ symbols reference from the Application
                        0000     69 ;               Data area versus Packet Pointer.
                        0000     70 ;--
                        0000     71
                        0000     72          .DEFAULT DISPLACEMENT,WORD
                        0000     73          .ENABLE SUPPRESSION
                        0000     74          .SBTTL  DFFINITIONS
                        0000     75
                        0000     76 ;
                        0000     77 ; Set PSECT to driver code
                        0000     78 ;
                        0000     79
                    00000000     80          .PSECT  $$$115_DRIVER,LONG
                        0000     81
                        0000     82
                        0000     83 ;  System Definitions (LIB.MLB):
                        0000     84
                        0000     85          $CDRPDEF                            ; Define CDRP offsets
                        0000     86          $CDTDEF                             ; Define CDT offsets
                        0000     87          $CIBHANDEF                          ; Define CI BUFFER HANDLE offsets
                        0000     88          $DYNDEF                             ; Define DYN offsets
                        0000     89          $IODEF                              ; Define I/O FUNCTION codes
                        0000     90          $IRPDEF                             ; Define IRP offsets
                        0000     91          $PBDEF          GLOBAL              ; Define PB ofsets
                        0000     92          $PDTDEF         GLOBAL              ; Define PDT offsets
                        0000     93          $SBDEF          GLOBAL              ; Define SB offsets
                        0000     94          $SCSDEF         GLOBAL              ; Define SCS packet offsets
                        0000     95          $SSDEF                              ; Define system status symbols
                        0000     96          $SYSAPDEF                           ; Define SYSAP offsets
                        0000     97          $UCBDEF                             ; Define UCB offsets
                        0000     98
                        0000     99
                        0000    100 ;  PADRIVER Definitions (PALIB.MLB):
                        0000    101
                        0000    102          $PAPDTDEF       GLOBAL             ; Define PAPDT offsets
                        0000    103          $PAREGDEF       GLOBAL             ; Define PA registers
                        0000    104          $PPDDEF         GLOBAL             ; Define PPD offsets
                        0000    105          $PAMAINTDEF                        ; Define PA maint definitions
                        0000    106
                        0000    107
                        0000    108 ;  CXDRIVER Definitions (CXLIB.MLB):
                        0000    109
                        0000    110          $CDCDEF                            ; Define CDC interface symbols
                        0000    111          $CINTREGDEF                        ; Define CINT registers
                        0000    112          $CTPDEF                            ; Define CTP symbols
                        0000    113          $CXCDBDEF                          ; Define CXCDB offsets
                        0000    114          $CXCDRPDEF                         ; Define CX CDRP extension
```

```
                        0000    115     $CXSTATEDEF                     ; Define CXDRIVER states
                        0000    116
                        0000    117
                        0000    118
```

```
                                0000   120          .SBTTL CX_STARTIO,      Start I/O routine
                                0000   121 ;+
                                0000   122 ;
                                0000   123 ;  CX_STARTIO
                                0000   124 ;
                                0000   125 ;  This routine and the individual command support routines provide the
                                0000   126 ;  actual interface to SCS. The class driver commands are converted to the
                                0000   127 ;  appropriate SCS function call.
                                0000   128 ;
                                0000   129 ;  Calling sequence:
                                0000   130 ;
                                0000   131 ;       JMPed to through the driver dispatch table by IOC$INITIATE
                                0000   132 ;
                                0000   133 ;  INPUTS:
                                0000   134 ;
                                0000   135 ;       R3      IRP address
                                0000   136 ;       R4      CCB address
                                0000   137 ;       R5      UCB address
                                0000   138 ;
                                0000   139 ;  IMPLICIT INPUTS:
                                0000   140 ;
                                0000   141 ;       Command buffer described by IRP has been preprocessed by the FDT
                                0000   142 ;       routines.
                                0000   143 ;
                                0000   144 ;  OUTPUTS:
                                0000   145 ;
                                0000   146 ;       R0                      - First longword of I/O status
                                0000   147 ;       R1                      - Second longword of I/O status
                                0000   148 ;
                                0000   149 ;  IMPLICIT OUTPUTS:
                                0000   150 ;
                                0000   151 ;       Various CI traffic, filled response buffer.
                                0000   152 ;
                                0000   153 ;-
                                0000   154
                                0000   155 CX_STARTIO::
                                0000   156
                                0000   157 ; The driver state flag CMAP (command buffer mapped) is used to synchronize
                                0000   158 ; access to the class driver's data structures.
                                0000   159
                                0000   160          SET_DRVR_STATE CMAP             ; Command buffer is mapped
                                0005   161          SET_DRVR_STATE SIO              ; Mark state as SIO
0000'CF    53    7D              000E   162          MOVQ    R3,CX$X_IRPSAV          ; Save IRP, CCB address
0008'CF    55    D0              0013   163          MOVL    R5,CX$X_IRPSAV+8        ; Save UCB address
     0000'CF    94              0018   164          CLRB    CX$B_CMDERRCNT          ; Clear SIO error counter
     0000'CF    C1              001C   165          ADDL3   CX$L_CMDBUFBGN,-        ; Get address of first command
     53    0C                    0020   166                  #CDC$K_CMDHDRSIZ,R3     ; in command buffer
0000'CF    53    D0              0022   167          MOVL    R3,CX$L_CUR_CMD         ; Save it
                                0027   168
                                0027   169
                                0027   170 SIO_DISPATCH:
                                0027   171
                                0027   172 ;  In many cases, when we give up control to SCS, there is a possibility that
                                0027   173 ;  our connection error routine may be entered. The connection error routine
                                0027   174 ;  will determine which commands were trying to use the disconnected/broken
                                0027   175 ;  connection, and mark the appropriate status in the flags word for those
                                0027   176 ;  commands.
```

```
                                0027   177
                                0027   178          BITW    -                               ; Any of these conditions for command?
                             B3 0027   179                  #<FLAG_M_CMD_DONE-              ;   Command done
                                0028   180                   !FLAG_M_CONN_LOST>,-           ;   Connection lost
                   04 A3  8800 8F      181                  CDC$W_FLAG(R3)                  ;
                          0A   13 002D   182          BEQL    30$                             ; If no errors, branch
                      51  0000'CF  D0 002F   183          MOVL    CX$L_CMDBUFBGN,R1               ; Get top of cmd buffer
                        01 A1  96 0034   184          INCB    CDC$B_DONECNT(R1)               ; Increment the done count
                           44  11 0037   185          BRB     SIO_NXT_CMD                     ; Try the next command
                               0039   186
                               0039   187 30$:       SET_CMD_FLAG CMD_START                  ; This command has been started
                               003F   188          $DISPATCH -                             ; Dispatch off command opcode
                               003F   189                  CDC$B_OPCODE(R3),-
                               003F   190                  TYPE=B,-
                               003F   191                  <-
                               003F   192                  <CDC$SENDMSG,SIO_MSG>,-         ; Send message
                               003F   193                  <CDC$SENDDG,SIO_DG>,-           ; Send datagram
                               003F   194                  <CDC$SENDDATA,SIO_SNDDAT>,-     ; Send data
                               003F   195                  <CDC$REQDATA,SIO_REQDAT>,-      ; Request data
                               003F   196                  <CDC$MAPBUF,SIO_MAP>,-          ; Map a buffer
                               003F   197                  <CDC$UNMAPBUF,SIO_UNMAP>,-      ; Unmap a buffer
                               003F   198                  <CDC$CONFIGDATA,SIO_CONFIG>,-   ; Gather configuration data
                               003F   199                  <CDC$CONNECT,SIO_CONNECT>,-     ; Connect
                               003F   200                  <CDC$DISCONNECT,SIO_DISC>,-     ; Disconnect
                               003F   201                  <CDC$READCNT,SIO_READCNT>,-     ; Read the performance counters
                               003F   202                  <CDC$GARBAGE,SIO_GARBAGE>,-     ; Get garbage
                               003F   203                  <CDC$RESET,SIO_RESET>,-         ; Send a reset
                               003F   204                  <CDC$START,SIO_START>,-         ; Send a start
                               003F   205                  <CDC$POLLER,SIO_POLLER>,-       ; Turn config poller on/off
                               003F   206                  <CDC$MAINTSTATE,SIO_MAINT_STATE>,- ; Set port to maintenance state
                               003F   207                  <CDC$MAPMBUF,SIO_NXT_CMD>,-     ; Map maintenance buffer
                               003F   208                  <CDC$UNMAPMBUF,SIO_NXT_CMD>,-   ; Unmap maintenance buffer
                               003F   209                  <CDC$SENDMDATA,SIO_SNDMDAT>,-   ; Send maintenance data
                               003F   210                  <CDC$REQMDATA,SIO_REQMDAT>,-    ; Request maintenance data
                               003F   211                  <CDC$SELADR,SIO_SELADR>,-       ; Change port address
                               003F   212                  <CDC$NORSP,SIO_NORSP>,-         ; Arm CINT NORSP logic
                               003F   213                  <CDC$NAK,SIO_NAK>,-             ; Arm CINT NAK logic
                               003F   214                  <CDC$NOACK,SIO_NOACK>,-         ; Arm CINT NOACK logic
                               003F   215                  <CDC$CARRIER,SIO_CARRIER>,-     ; Arm CINT CARRIER logic
                               003F   216                  <CDC$LRGPACKET,SIO_LRGPACKET>,- ; Arm CINT LRGPACKET logic
                               003F   217                  <CDC$INITCINT,SIO_INITCINT>,-   ; Initialize CINT logic
                               003F   218                  <CDC$ARMCINTPATH,SIO_ARMPATH>,- ; Arm CINT received path logic
                               003F   219                  <CDC$RDCINTPATH,SIO_READPATH>,- ; Read CINT received path status
                               003F   220                  <CDC$OVERSIZEPKT,SIO_OVERSIZEPKT>,- ; Arm CINT oversized packet logi
                               003F   221                  >
                               007D   222
                               007D   223
                               007D   224 SIO_INV_OPCODE:                                  ; Unimplemented opcodes
                               007D   225
                               007D   226
                               007D   227 ;  SIO_NXT_CMD
                               007D   228 ;
                               007D   229 ;  Calculate the address of the next command in the buffer, update the
                               007D   230 ;  startio counter, check to see if all commands are started/done and
                               007D   231 ;  take the appropriate action.
                               007D   232
                               007D   233 SIO_NXT_CMD::
```

```
                         007D
53   0000'CF       D0    007D            MOVL     CX$L_CUR_CMD,R3           ; Reset current command ptr
50     08 A3       3C    0082            MOVZWL   CDC$W_TXTLEN(R3),R0       ; Get length of current command
     53   50       C0    0086            ADDL2    R0,R3                     ; Get address of next command
0000'CF   53       D0    0089            MOVL     R3,CX$L_CUR_CMD           ; Save it
                         008E
                         008E      ;  Determine if all the commands in the command buffer are done. Commands
                         008E      ;  are done if they completed successfull, or if they failed. Commands may
                         008E      ;  be started (in the case of sending a message to a responder) but not
                         008E      ;  done (the message hasn't been returned yet).
                         008E
                         008E            ASSUME CDC$B_CMDCOUNT EQ 0
                         008E
51   0000'CF       D0    008E            MOVL     CX$L_CMDBUFBGN,R1         ; Point to cmd counter bytes
       03 A1       96    0093            INCB     CDC$B_SIOCNT(R1)          ; One more command started
  61   03 A1       91    0096            CMPB     CDC$B_SIOCNT(R1),(R1)     ; Are all commands started?
                         009A            BNEQW    SIO_DISPATCH              ; No, branch
  61   01 A1       91    009F            CMPB     CDC$B_DONECNT(R1),(R1)    ; Are all commands done?
          55       13    00A3            BEQLU    SIO_COMPLETE              ; Yes, branch
                         00A5
                         00A5      ;  We have started all the commands requested. Some are not done yet as the
                         00A5      ;  response from the responder has not yet been returned. Throw this thread
                         00A5      ;  away and let the message/datagram input routine complete the I/O.
                         00A5
                         00A5      SIO_POOF:
                         00A5
53   0000'CF       7D    00A5            MOVQ     CX$X_IRPSAV,R3            ; Restore R3,R4,R5
55   0008'CF       D0    00AA            MOVL     CX$X_IRPSAV+8,R5          ;
                         00AF            CLR_DRVR_STATE  SIO                ; Indicate driver has started all
                         00B8            SET_DRVR_STATE  NOTPROC            ; commands but is awaiting input
     0100 8F       AA    00C1            BICW     #UCB$M_BSY,-              ; Clear BUSY so user process can STOP
       64 A5             00C5                     UCB$W_STS(R5)             ; the I/O.
                   05    00C7            RSB
                         00C8
                         00C8
                         00C8      ;  SIO_NXT_CMD_ERR
                         00C8      ;
                         00C8      ;  Calculate the address of the next command in the buffer. Update STARTIO
                         00C8      ;  counter. Check if we are done. If done, complete this QIO by queuing the
                         00C8      ;  I/O to REQCOM code.
                         00C8
                         00C8      SIO_NXT_CMD_ERR:
                         00C8
53   0000'CF       D0    00C8            MOVL     CX$L_CUR_CMD,R3           ; Reset current command ptr
  52   08 A3       3C    00CD            MOVZWL   CDC$W_TXTLEN(R3),R2       ; Get length of current command
     53   52       C0    00D1            ADDL2    R2,R3                     ; Calculate address of next command
0000'CF   53       D0    00D4            MOVL     R3,CX$L_CUR_CMD           ; Save it
                         00D9
                         00D9      ;  Determine if all the commands in the command buffer are done. Commands
                         00D9      ;  are done if they completed successfull, or if they failed. Commands may
                         00D9      ;  be started (in the case of sending a message to a responder) but not
                         00D9      ;  done (the message hasn't been returned yet).
                         00D9
52   0000'CF       D0    00D9            MOVL     CX$L_CMDBUFBGN,R2         ; Point to cmd counter bytes
       03 A2       96    00DE            INCB     CDC$B_SIOCNT(R2)          ; One more command started
  62   03 A2       91    00E1            CMPB     CDC$B_SIOCNT(R2),(R2)     ; Have all commands been started?
                         00F5            BNEQW    10$                       ; No, branch
```

```
                0000'CF   96   00EA    291 10$:    INCB    CX$B_CMDERRCNT          ; One more command completed in error
                0000'CF   81   00EE    292         ADDB3   CX$B_CMDERRCNT,-        ; Calculate total commands done
                   01 A2       00F2    293                 CDC$B_DONECNT(R2),R3    ;
                   53   91     00F5    294         CMPB    R3,(R2)                 ; Have all commands been done
                        AB   12 00F8    295         BNEQU   SIO_POOF                ; No, throw away thread
                                00FA    296
                                00FA    297
                                00FA    298 ;  SIO_COMPLETE
                                00FA    299 ;
                                00FA    300 ;  At this point we are finished with all processing for all commands in the
                                00FA    301 ;  command buffer and all responses from the responders have returned. All
                                00FA    302 ;  resources used have been deallocated or saved for future use (XFR CDRPs).
                                00FA    303 ;  Do a REQCOM to complete this I/O
                                00FA    304
                                00FA    305 SIO_COMPLETE::                          ; Normal QIO completion
                                00FA    306
                   50   01   3C 00FA    307         MOVZWL  #SS$_NORMAL,R0          ; Set a return status for the QIO
                        51   D4 00FD    308         CLRL    R1                      ; Only R0 is valid here
                                00FF    309
                                00FF    310
                                00FF    311 ;  ERROR COMPLETION
                                00FF    312 ;
                                00FF    313 ;  Leave status in R0,R1 alone. IOC$REQCOM will handle it.
                                00FF    314
                                00FF    315 SIO_COMPLETE_ERR::
                                00FF    316
                                00FF    317
                                00FF    318 ;  IOPOST will copy the modified command buffer and the response buffer back
                                00FF    319 ;  into the user buffers. It will also deallocate the system buffers that we
                                00FF    320 ;  used during command processing
                                00FF    321
                             CA 00FF    322         BICL2   #<STATE_M_CMAP!-        ; Clear the driver state
                                0100    323                 STATE_M_RMAP>,-         ;
                0000'CF   03    0100    324                 CX$L_STATE              ;
                53   0000'CF   7D 0104   325         MOVQ    CX$X_IRPSAV,R3          ; Restore R3,R4,R5
                55   0008'CF   D0 0109   326         MOVL    CX$X_IRPSAV+8,R5        ;
                                010E    327         REQCOM                          ; End of I/O
                                0114    328
                                0114    329
                                0114    330
                                0114    331 ;*****************************************************************************
                                0114    332 ;
                                0114    333 ;  START I/O ERROR PATHS
                                0114    334 ;
                                0114    335 ;  Note that errors occurring at START I/O, unlike FDT, do not cause the I/O
                                0114    336 ;  to be aborted.
                                0114    337
                                0114    338
                                0114    339 NODE_UNKNOWN:
                                0114    340 CHECK_VC_ERR::
                                0114    341
                                0114    342         SET_CMD_FLAG SCS_ERR            ; Indicate SCS error
                      060E   30 011A    343         BSBW    SET_RSP_STATUS          ; Set response status
                      FFA8   31 011D    344         BRW     SIO_NXT_CMD_ERR         ; Go execute next command
                                0120    345
                                0120    346 CHECK_CONN_ERR:
                                0120    347
```

```
                        0120    348          SET_CMD_FLAG CONN_LOST          ; Indicate connection was lost
                0604    30  0126    349          BSBW    SET_RSP_STATUS_2    ; Set response status
                FF9C    31  0129    350          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            012C    351
                            012C    352 BUFFER_NOT_MAPPED:
                            012C    353
                05FE    30  012C    354          BSBW    SET_RSP_STATUS_2    ; Set response status
                FF96    31  012F    355          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            0132    356
                            0132    357 CONNECT_ERR:
                            0132    358
          53    0000'CF   D0  0132    359          MOVL    CX$L_CUR_CMD,R3     ; Restore address of current command
                            0137    360          SET_CMD_FLAG SCS_ERR        ; Indicate SCS error
00000294 8F      50   D1  013D    361          CMPL    R0,#SS$_REJECT      ; Was connection rejected?
                   05   13  0144    362          BEQL    10$                 ; Yes, branch
                05E2    30  0146    363          BSBW    SET_RSP_STATUS      ; Set response status
                   03   11  0149    364          BRB     20$                 ; Branch
                05DF    30  014B    365 10$:     BSBW    SET_RSP_STATUS_2    ; R1 status valid also
              24 B5     D4  014E    366 20$:     CLRL    @CDB$L_CDBSLOT(R5)  ; Clear CDB address in CDB list
            50   55     D0  0151    367          MOVL    R5,R0               ; Get CDB address
     00000000'GF     16  0154    368          JSB     G^COM$DRVDEALMEM    ; Deallocate the CDB
                FF6B    31  015A    369          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            015D    370
                            015D    371 ALLOC_CDB_ERR:
                            015D    372 ALLOC_CDRP_ERR:
                            015D    373
                05CB    30  015D    374          BSBW    SET_RSP_STATUS      ; Set response status
                FF65    31  0160    375          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            0163    376
                            0163    377 ALLOC_MSG_ERR:
                            0163    378 ALLOC_DG_ERR::
                            0163    379
                            0163    380          SET_CMD_FLAG SCS_ERR        ; Indicate SCS error
              05BF      30  0169    381          BSBW    SET_RSP_STATUS      ; Set response status
              FE91'     30  016C    382          BSBW    DEALLOC_CDRP        ; Deallocate the CDRP
            0C A3       D4  016F    383          CLRL    CDC$L_CDRP(R3)      ; Indicate CDRP is gone
              FF53      31  0172    384          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            0175    385
                            0175    386 SEND_MSG_ERR:
                            0175    387 SEND_DG_ERR:
                            0175    388
                            0175    389          SET_CMD_FLAG SCS_ERR        ; Indicate SCS error
              05AD      30  017B    390          BSBW    SET_RSP_STATUS      ; Set resposne status
              FE7F'     30  017E    391          BSBW    CLEAN_CDRP          ; Throw away CDRP resources
              FE7C'     30  0181    392          BSBW    DEALLOC_CDRP        ; Deallocate the CDRP
            0C A3       D4  0184    393          CLRL    CDC$L_CDRP(R3)      ; Indicate CDRP is gone
              FF3E      31  0187    394          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            018A    395
                            018A    396 XFR_ERR:
                            018A    397
                            018A    398          SET_CMD_FLAG SCS_ERR        ; Indicate SCS error
                0598    30  0190    399          BSBW    SET_RSP_STATUS      ; Set response status
                FF32    31  0193    400          BRW     SIO_NXT_CMD_ERR     ; Go execute next command
                            0196    401
                            0196    402 MNT_XFR_ERR:
                            0196    403
                            0196    404          SET_CMD_FLAG SCS_ERR        ; Indicate SCS error
```

```
058C   30  019C   405          BSBW    SET_RSP_STATUS          ; Set response status
FF26   31  019F   406          BRW     SIO_NXT_CMD_ERR         ; Go execute next command
           01A2   407
           01A2   408 READ_COUNT_ERR:
           01A2   409
           01A2   410          SET_CMD_FLAG SCS_ERR            ; Indicate SCS error
0580   30  01A8   411          BSBW    SET_RSP_STATUS          ; Set response status
FE52'  30  01AB   412          BSBW    CLEAN_CDRP              ; Throw away CDRP resources
FE4F'  30  01AE   413          BSBW    DEALLOC_CDRP            ; Deallocate the CDRP
0C A3  D4  01B1   414          CLRL    CDC$L_CDRP(R3)          ; Indicate CDRP is gone
FF11   31  01B4   415          BRW     SIO_NXT_CMD_ERR         ; Go execute next command
           01B7   416
           01B7   417 SEND_RESET_ERR:
           01B7   418 SEND_START_ERR:
           01B7   419
           01B7   420          SET_CMD_FLAG SCS_ERR            ; Indicate SCS error
056B   30  01BD   421          BSBW    SET_RSP_STATUS          ; Set response status
FF05   31  01C0   422          BRW     SIO_NXT_CMD_ERR         ; Go execute next command
           01C3   423
           01C3   424 CHECK_RST_STR_ERR:
           01C3   425
           01C3   426 ;  END OF START I/O ERROR PATHS
           01C3   427 ;
           01C3   428 ;*****************************************************************************
           01C3   429
           01C3   430
           01C3   431
           01C3   432
```

```
                     01C3   434          .SBTTL  SIO_CONFIG,     Gather configuration data
                     01C3   435 ;+
                     01C3   436 ;
                     01C3   437 ;  SIO_CONFIG
                     01C3   438 ;
                     01C3   439 ;  Routine to gather to configuration information about a remote port on
                     01C3   440 ;  the CI. The routine CHECK_VC is called which performs the actual SCS
                     01C3   441 ;  CONFIG_PTH call. This routine returns the path block address, from which
                     01C3   442 ;  the system block and PDT address can be obtained. The informatione returned
                     01C3   443 ;  is as follows:
                     01C3   444 ;
                     01C3   445 ;       - Path 0/1 status        (Path block)
                     01C3   446 ;       - Cable status           (Path block)
                     01C3   447 ;       - Hardware type          (Path block)
                     01C3   448 ;       - Port function mask     (Path block)
                     01C3   449 ;       - Loopback DG status     (PDT)
                     01C3   450 ;       - System ID              (System block)
                     01C3   451 ;
                     01C3   452 ;  INPUTS:
                     01C3   453 ;
                     01C3   454 ;       R3                       - Address of current command
                     01C3   455 ;       CDC$B_DSTPORT(R3)        - Remote port number
                     01C3   456 ;       CDC$L_LPORT(R3)          - Local port name
                     01C3   457 ;
                     01C3   458 ;  OUTPUTS:
                     01C3   459 ;
                     01C3   460 ;       R0-R5                    - Destroyed
                     01C3   461 ;
                     01C3   462 ;-
                     01C3   463
                     01C3   464 SIO_CONFIG:
                     01C3   465
          03DA   30  01C3   466          BSBW    CHECK_VC                ; Check for a VC to remote port
                     01C6   467          BLBCW   R0,NODE_UNKNOWN         ; Failure, branch
                     01CC   468
                     01CC   469          ASSUME  PB$B_P0_STS+1 EQ PB$B_P1_STS
                     01CC   470          ASSUME  CDC$B_CFGP0STS+1 EQ CDC$B_CFGP1STS
                     01CC   471
                     01CC   472          CALC_RSP DST=R4                 ; Get address of response buffer area
                     01D5   473
       29 A1     B0  01D5   474          MOVW    PB$B_P0_STS(R1),-       ; Copy path status fields,
       1F A4         01D8   475                  CDC$B_CFGP0STS(R4)      ;
       28 A1     90  01DA   476          MOVB    PB$B_CBL_STS(R1),-      ; cable status field,
       21 A4         01DD   477                  CDC$B_CFGCBLSTS(R4)     ;
       14 A1     D0  01DF   478          MOVL    PB$L_RPORT_TYP(R1),-    ; hardware type field,
       22 A4         01E2   479                  CDC$L_CFGHWTYPE(R4)     ;
       1C A1     D0  01E4   480          MOVL    PB$L_RPORT_FCN(R1),-    ; port function mask
       26 A4         01E7   481                  CDC$L_CFGPRTMSK(R4)     ;
                     01E9   482
                     01E9   483          ASSUME  PDT$B_P0_LBSTS+1 EQ PDT$B_P1_LBSTS
                     01E9   484          ASSUME  CDC$B_CFGP0LBSTS+1 EQ CDC$B_CFGP1LBSTS
                     01E9   485
  50   2C A1     D0  01E9   486          MOVL    PB$L_PDT(R1),R0         ; Get PDT address
     0180 C0     B0  01ED   487          MOVW    PDT$B_P0_LBSTS(R0),-    ; Copy loopback datagram
       2A A4         01F1   488                  CDC$B_CFGP0LBSTS(R4)    ; status
                     01F3   489
  50   30 A1     D0  01F3   490          MOVL    PB$L_SB_LINK(R1),R0     ; Get system block address
```

```
                18 A0    7D  01F7   491          MOVQ    SB$B_SYSTEMID(R0),-     ; Copy system ID field
                34 A4        01FA   492                  CDC$T_SYSTEMID(R4)      ;
                            01FC   493
                 038D   30  01FC   494          BSBW    MARK_CMD_DONE
                 FE7B   31  01FF   495          BRW     SIO_NXT_CMD
                            0202   496
                            0202   497
                            0202   498
                            0202   499
```

```
                     0202   501          .SBTTL  SIO_CONNECT,    Connect to remote
                     0202   502 ;+
                     0202   503 ;
                     0202   504 ;  SIO_CONNECT
                     0202   505 ;
                     0202   506 ;  Routine to establish a connection to a responder at the specified remote
                     0202   507 ;  port through the specified local port.
                     0202   508 ;
                     0202   509 ;  Because the connection call is suspended until the SCS protocol messages
                     0202   510 ;  can be sent to and from the remote port, it is possible for the driver to
                     0202   511 ;  be canceled before the connect call completes. If this occurs, the cancel
                     0202   512 ;  code will set the CANCELED ilag in the connection block (CDB). This flag
                     0202   513 ;  must be checked when the connection thread is resumed.
                     0202   514 ;
                     0202   515 ;  If a cancel occurs during the suspended connect call, and the connect call
                     0202   516 ;  completes successfully, a disconnect should be issued and the CDB cleaned
                     0202   517 ;  up. If a cancel occurs during the suspended connect call and the connect
                     0202   518 ;  call fails, the CDB should be cleaned up, but no disconnect should be
                     0202   519 ;  issued.
                     0202   520 ;
                     0202   521 ;  If the connect call fails, and no cancel has occurred, then the error status
                     0202   522 ;  is copied to the response buffer, and the next command in the command buffer
                     0202   523 ;  is executed.
                     0202   524 ;
                     0202   525 ;  INPUTS:
                     0202   526 ;
                     0202   527 ;       R3              - Address of current command
                     0202   528 ;
                     0202   529 ;  OUTPUTS:
                     0202   530 ;
                     0202   531 ;       R0-R5           - Destroyed
                     0202   532 ;
                     0202   533 ;-
                     0202   534
                     0202   535 SIO_CONNECT:
                     0202   536
 048D   30           0202   537          BSBW    ALLOC_CDB               ; Allocate a connection block
                     0205   538          BLBCW   R0,ALLOC_CDB_ERR        ; Error, branch
   01   A8           020B   539          BISW2   #CDB_M_CONN,-           ; Show connect is in progress
0C A5                020D   540                  CDB$W_STATUS(R5)
                     020F   541
                     020F   542          CONNECT -                               ; Issue the SCS connect call
                     020F   543                  MSGADR=CX$MSG_IN,-              ; Message input routine
                     020F   544                  DGADR=CX$DG_IN,-                ; Datagram input routine
                     020F   545                  ERRADR=CX$CONNECT_ERR,-         ; Connection error entry
                     020F   546                  LPRNAM=CONTROLLER_NAME,-        ; Local process name
                     020F   547                  RPRNAM=RESPONDER_NAME,-         ; Remote process name
                     020F   548                  RSYSID=CDC$T_SYSTEMID(R3),-     ; Destination system ID
                     020F   549                  RSTADR=CDB$B_RSTNADR(R5),-      ; Remote node/local port name
                     020F   550                  INITCR=CDC$W_MSGCREDIT(R3),-    ; Message credits
                     020F   551                  MINSCR=CDC$W_MINSENDCR(R3),-    ; Minimum send credits required
                     020F   552                  INITDG=CDC$W_DGCREDIT(R3),-     ; Datagram credits
                     020F   553                  CONDAT=CDC$T_CONDAT(R3),-       ; Connection data
                     020F   554                  AUXSTR=(R5)                     ; Addr of CDB for connection
                     0248   555
   01   AA           0248   556          BICW2   #CDB_M_CONN,-           ; Connect no longer in progress
0C A5                024A   557                  CDB$W_STATUS(R5)
```

```
                    02   E0  024C   558          BBS     #CDB_V_CNCL,-           ; If cancel occurred during suspended
                2A 0C A5         024E   559                  CDB$W_STATUS(R5),100$   ; connect call, branch
                         0251   560          BLBCW   R0,CONNECT_ERR          ; Branch if connect call failed
                         0257   561
                         0257   562
                         0257   563          ASSUME  CDB$L_CDT+4 EQ CDB$L_PDT
                         0257   564
              1C A5   53 7D  0257   565          MOVQ    R3,CDB$L_CDT(R5)        ; Copy CDT, PDT address to CDB
                         025B   566
           53   0000'CF  D0  025B   567          MOVL    CX$L_CUR_CMD,R3         ; Reset pointer to current command
                         0260   568          CALC_RSP  DST=R1                ; Get addr of response area for cmd
              0C A1   50 D0  0269   569          MOVL    R0,CDC$L_ROSTATUS(R1)   ; Set status
                    10 A1  D4  026D   570          CLRL    CDC$L_RISTATUS(R1)      ;
        14 A1   62   30 28  0270   571          MOVC3   #48,(R2),CDC$K_DATA(R1) ; Copy accept data to response buffer
                         0275   572
                   0314  30  0275   573          BSBW    MARK_CMD_DONE
                   FE02  31  0278   574          BRW     SIO_NXT_CMD
                         027B   575
                         027B   576
                   0A 50 E9  027B   577 100$:    BLBC    R0,120$                 ; If connect call failed, branch
                      02 A8  027E   578          BISW2   #CDB_M_DISC,-           ; Show disconnect is in progress
                   0C A5         0280   579                  CDB$W_STATUS(R5)
                         0282   580          DISCONNECT                      ; Issue a disconnect
                         0288   581
                   24 B5  D4  0288   582 120$:    CLRL    @CDB$L_CDBSLOT(R5)      ; Remove CDB from connection list
                50   55  D0  028B   583          MOVL    R5,R0                   ; Copy CDB address
         00000000'GF   16  028E   584          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDB
                      05  0294   585          RSB                             ; Return to SCS
                         0295   586
                         0295   587
                         0295   588
                         0295   589
```

```
                        0295   591          .SBTTL  SIO_DISC,       Disconnect from remote
                        0295   592 ;+
                        0295   593 ;
                        0295   594 ;  SIO_DISC
                        0295   595 ;
                        0295   596 ;  This routine performs the necessary functions to disconnect from a remote
                        0295   597 ;  process. The flow is:
                        0295   598 ;
                        0295   599 ;       - Check for an active connection to remote
                        0295   600 ;       - Set up SCS interface
                        0295   601 ;       - Issue disconnect call
                        0295   602 ;       - Clean up resources for connection (CDB, etc.)
                        0295   603 ;       - Update command and response status
                        0295   604 ;       - Dispatch next command
                        0295   605 ;
                        0295   606 ;  INPUTS:
                        0295   607 ;
                        0295   608 ;       R3                      - Address of current command
                        0295   609 ;
                        0295   610 ;  OUTPUTS:
                        0295   611 ;
                        0295   612 ;       R0-R5                   - Destroyed
                        0295   613 ;
                        0295   614 ;-
                        0295   615
                        0295   616 SIO_DISC:
                        0295   617
      FD68'   30        0295   618          BSBW    CHECK_CONN              ; Check for a connection
    26 50     E9        0298   619          BLBC    R0,100$                 ; No, error, branch
                        0298   620
   55   51    D0        0298   621          MOVL    R1,R5                   ; Get CDB address
53    1C A5    7D       029E   622          MOVQ    CDB$L_CDT(R5),R3        ; Get CDT, PDT address
                        02A2   623
                        02A2   624 ;  Issue the disconnect call. SCS check to see that the CDT is in a proper
                        02A2   625 ;  state to attempt a disconnect. If no, error status (SS$_ILLCDTST) is
                        02A2   626 ;  returned. Otherwise, SCS sends a disconnection request/response to the
                        02A2   627 ;  remote and suspends this thread until the SCS traffic completes.
                        02A2   628 ;  When this thread is resumed, SCS returns here with the following inputs:
                        02A2   629 ;
                        02A2   630 ;       R0      - Status (SS$_ALRDYCLOSED,SS$_ILLCDTST,SS$_NORMAL)
                        02A2   631 ;       R1-R3   - Destroyed
                        02A2   632 ;       R4      - PDT address
                        02A2   633 ;       R5      - CDB address
                        02A2   634
                        02A2   635          DISCONNECT                      ; Issue disconnect call
          02    E0      02A8   636          BBS     #CDB_V_CNCL,-           ; If CANCEL occurred during suspension
  14 0C A5              02AA   637                  CDB$W_STATUS(R5),200$   ; of disconnect call, branch
    11 50     E9        02AD   638          BLBC    R0,250$                 ; Disconnect failed, branch
                        02B0   639
                        02B0   640 ;  The connection is gone. Clean up all the resources associated with this
                        02B0   641 ;  connection. This includes any CDRP's on the CDB pending queue, and the
                        02B0   642 ;  memory allocated to the CDB itself.
                        02B0   643
      FD4D'   30        02B0   644          BSBW    CLEAN_CDB               ; Clean up it's resources
53  0000'CF   D0        02B3   645          MOVL    CX$L_CUR_CMD,R3         ; Reset address of current command
      0470    30        02B8   646          BSBW    SET_RSP_STATUS          ; Set response status
                        02BB   647
```

```
                02CE   30  02BB   648          BSBW    MARK_CMD_DONE
                FDBC   31  02BE   649          BRW     SIO_NXT_CMD
                           02C1   650
                           02C1   651
                           02C1   652
                           02C1   653 ;  NO ACTIVE CONNECTION EXISTS
                           02C1   654 ;
                           02C1   655 ;  No active CDB could be found for the specified combination of local port
                           02C1   656 ;  name and remote port number. Set the appropriate response status in the
                           02C1   657 ;  response buffer.
                           02C1   658
                           02C1   659 100$:
                           02C1   660
                           02C1   661
                           02C1   662 ;  CONNECTION IS IN STATE OF TRANSITION
                           02C1   663 ;
                           02C1   664 ;  An active connection fot the specified combination of local and remote port
                           02C1   665 ;  was found, however, the connection is in a state of transition. That is, a
                           02C1   666 ;  connect or disconnect has been issued for this connection, but the suspended
                           02C1   667 ;  thread has not yet been resumed.
                           02C1   668
                           02C1   669 150$:
                           02C1   670
                           02C1   671
                           02C1   672 ;  CANCEL OCCURRED DURING SUSPENDED DISCONNECT CALL
                           02C1   673 ;
                           02C1   674 ;  The state of the CDB indicates that the driver was canceled during the
                           02C1   675 ;  suspended disconnect call. Clean up the CDB by pulling all CDRP's off its
                           02C1   676 ;  pending queue, deallocating them, and the CDB itself. Finish by RSB'ing,
                           02C1   677 ;  as the original I/O must already have been completed.
                           02C1   678
                           02C1   679 200$:
                           02C1   680
                           02C1   681
                           02C1   682 ;  DISCONNECT FAILED
                           02C1   683 ;
                           02C1   684 ;  SCS detected that the CDT associated with this connection was in a state
                           02C1   685 ;  that prevented a dissconnect from being issued at this time. Restore the
                           02C1   686 ;  CDB to active state, and set the appropriate status in the response buffer.
                           02C1   687
                           02C1   688 250$:
                           02C1   689
                           02C1   690
                           02C1   691
                           02C1   692
```

```
                        02C1    694          .SBTTL  SIO_MSG,        Send message
                        02C1    695 ;+
                        02C1    696 ;
                        02C1    697 ;  SIO_MSG
                        02C1    698 ;
                        02C1    699 ;  This routine sends a CTP request message to the specified remote port.
                        02C1    700 ;  One or more CTP responses are expected to be returned for this request.
                        02C1    701 ;  The message input routine, when it senses all responses are received for
                        02C1    702 ;  this request, marks the command as done, increments the done count in the
                        02C1    703 ;  command buffer header, and finishes the I/O if appropriate.
                        02C1    704 ;
                        02C1    705 ;  In a CINT configuration, the control process may want to know the path
                        02C1    706 ;  that the incoming packet was received on. If this is the case, the
                        02C1    707 ;  appropriate flag bit will be set. This routine then must arm the CINT logic
                        02C1    708 ;  to detect the path over which the incoming packet was received BEFORE it
                        02C1    709 ;  sends the message to the remote node (UUT).
                        02C1    710 ;
                        02C1    711 ;       - Check for active connection to remote port
                        02C1    712 ;       - Allocate a CDRP for the command
                        02C1    713 ;       - Allocate message buffer
                        02C1    714 ;       - Copy CTP data into message buffer
                        02C1    715 ;       - Send the message buffer
                        02C1    716 ;       - Update status
                        02C1    717 ;       - Dispatch next command
                        02C1    718 ;
                        02C1    719 ;  INPUTS:
                        02C1    720 ;
                        02C1    721 ;       R3                      - Address of current command
                        02C1    722 ;
                        02C1    723 ;  OUTPUTS:
                        02C1    724 ;
                        02C1    725 ;       R0-R5                   - Destroyed
                        02C1    726 ;
                        02C1    727 ;-
                        02C1    728
                        02C1    729 SIO_MSG:
                        02C1    730
FD3C'   30              02C1    731          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        02C4    732          BLBCW   R0,CHECK_CONN_ERR       ; No connection, error, branch
                        02CA    733
0376    30              02CA    734          BSBW    ALLOC_CDRP_CTP          ; Allocate a CTP type CDRP
                        02CD    735          BLBCW   R0,ALLOC_CDRP_ERR       ; Error, branch
                        02D3    736
02A4    30              02D3    737          BSBW    SIO_SETUP_SCS           ; Setup SCS interface
                        02D6    738
                        02D6    739          ALLOC_MSG_BUF                   ; Attempt to allocate message buffer
                        02D9    740          BLBCW   R0,ALLOC_MSG_ERR        ; Error, branch
                        02DF    741
0264    30              02DF    742          BSBW    SIO_MOVCTP              ; Copy message text into message buffer
                        02E2    743
045F    30              02E2    744          BSBW    CHECK_RST_STR           ; Prepare to receive a reset or start
                        02E5    745          BLBCW   R0,CHECK_RST_STR_ERR    ; packet if appropriate
                        02EB    746
                        02EB    747 10$:     SEND_MSG_BUF                    ; Send the message
                        02EE    748          BLBCW   R0,SEND_MSG_ERR         ; Error, branch
                        02F4    749
  00    E1              02F4    750          BBC     #FLAG_V_NO_RESPONSE,-   ; Branch if a response IS expected
```

```
                03 04 A3        02F6    751                     CDC$W_FLAG(R3),20$      ; for this message
                    0290    30  02F9    752             BSBW    MARK_CMD_DONE           ; Otherwise, this command is done
                                02FC    753
                    FD7E    31  02FC    754 20$:        BRW     SIO_NXT_CMD
                                02FF    755
                                02FF    756
                                02FF    757
                                02FF    758
                                02FF    759
```

```
                        02FF    761          .SBTTL  SIO_DG,         Send datagram
                        02FF    762 ;+
                        02FF    763 ;
                        02FF    764 ;  SIO_DG
                        02FF    765 ;
                        02FF    766 ;  This routine sends a CTP request datagram to the specified remote port.
                        02FF    767 ;  One or more CTP responses are expected to be returned for this request.
                        02FF    768 ;  The datagram input routine, when it senses all responses are received for
                        02FF    769 ;  this request, marks the command as done, increments the done count in the
                        02FF    770 ;  command buffer header, and finishes the I/O if appropriate.
                        02FF    771 ;
                        02FF    772 ;  In sending datagrams, SCS expects to see the length of the application DG
                        02FF    773 ;  in R1. The flow in this routine is as follows:
                        02FF    774 ;
                        02FF    775 ;       - Check for active connection to remote port
                        02FF    776 ;       - Allocate a CDRP for the command
                        02FF    777 ;       - Allocate datagram buffer
                        02FF    778 ;       - Copy CTP data into datagram buffer
                        02FF    779 ;       - Send the datagram buffer
                        02FF    780 ;       - Update status
                        02FF    781 ;       - Dispatch next command
                        02FF    782 ;
                        02FF    783 ;  INPUTS:
                        02FF    784 ;
                        02FF    785 ;       R3                      - Address of current command
                        02FF    786 ;
                        02FF    787 ;  OUTPUTS:
                        02FF    788 ;
                        02FF    789 ;       R0-R5                   - Destroyed
                        02FF    790 ;
                        02FF    791
                        02FF    792 SIO_DG:
                        02FF    793
            FCFE'   30  02FF    794          BSBW    CHECK_ACTV_CONN        ; Check for an active connection
                        0302    795          BLBCW   R0,CHECK_CONN_ERR      ; No connection, error, branch
                        0308    796
            0338    30  0308    797          BSBW    ALLOC_CDRP_CTP         ; Get a CTP type CDRP
                        030B    798          BLBCW   R0,ALLOC_CDRP_ERR      ; Error, branch
                        0311    799
            0266    30  0311    800          BSBW    SIO_SETUP_SCS          ; Setup SCS interface
                        0314    801
                        0314    802          ALLOC_DG_BUF                   ; Allocate a datagram buffer
                        0317    803          BLBCW   R0,ALLOC_DG_ERR        ; Error, branch
                        031D    804
            0226    30  031D    805          BSBW    SIO_MOVCTP             ; Copy datagram    t to datagram buffer
                        0320    806
    51  08 A3   3C      0320    807          MOVZWL  CDC$W_TXTLEN(R3),R1    ; Calculate the application datagram
        51  14  C2      0324    808          SUBL    #CDC$K_CMDSPEC,R1      ; length, passed to SCS in R1
                        0327    809
                        0327    810          SEND_DG_BUF #SYSAP$C_DISPPO    ; Send DG, dispose of buffer to pool
                        032D    811          BLBCW   R0,SEND_DG_ERR         ; Error, branch
                        0333    812
            FD47    31  0333    813          BRW     SIO_NXT_CMD
                        0336    814
                        0336    815
                        0336    816
                        0336    817
```

```
                        0336   819          .SBTTL  SIO_MAP,        Map buffer
                        0336   820 ;+
                        0336   821 ;
                        0336   822 ;  SIO_MAP
                        0336   823 ;
                        0336   824 ;  This routine maps a buffer to be used for data transfers. The flow is
                        0336   825 ;  as follows:
                        0336   826 ;
                        0336   827 ;       - Set up the CDRP with map call information
                        0336   828 ;       - Issue the map call
                        0336   829 ;       - Queue CDRP to map list of XFR CDRP's
                        0336   830 ;       - Dispath next command
                        0336   831 ;
                        0336   832 ;  INPUTS:
                        0336   833 ;
                        0336   834 ;       R3                      - Address of current command
                        0336   835 ;
                        0336   836 ;  OUTPUTS:
                        0336   837 ;
                        0336   838 ;       R0-R5                   - Destroyed
                        0336   839 ;
                        0336   840 ;-
                        0336   841
                        0336   842 SIO_MAP:
                        0336   843
              FCC7'  30 0336   844          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        0339   845          BLBCW   R0,CHECK_CONN_ERR       ; No, branch
                        033F   846
              0330   30 033F   847          BSBW    ALLOC_CDRP_XFR          ; Allocate a transfer type CDRP
                        0342   848          BLBCW   R0,ALLOC_CDRP_ERR       ; Error, branch
                        0348   849
           52   24 A3   D0 0348   850          MOVL    CDC$L_BUFADR(R3),R2     ; Get address of buffer to map
D0 A5      52   FE00 8F   AB 034C   851          BICW3   #^XFE00,R2,-            ; Copy byte offset to CDRP
                        0353   852                  CDRP$W_BOFF(R5)         ;
              20 A3   D0 0353   853          MOVL    CDC$L_BUFLENGTH(R3),-   ; Copy byte count to CDRP
              D2 A5             0356   854                  CDRP$L_BCNT(R5)         ;
              18 A3   D0 0358   855          MOVL    CDC$L_SVAPTE(R3),-      ; Copy SVAPTE to CDRP
              CC A5             035B   856                  CDRP$L_SVAPTE(R5)
                        035D   857
               021A   30 035D   858          BSBW    SIO_SETUP_SCS           ; Setup SCS interface
                        0360   859
                        0360   860          MAP_IRP                         ; Issue the map call
              44 A5   0E 0363   861          INSQUE  CDRP$L_PENDQFL(R5),-    ; Place it on the map list queue
            0000'CF             0366   862                  CX$Q_MAPLST             ;
              0C A3   D4 0369   863          CLRL    CDC$L_CDRP(R3)          ; Show CDRP is gone in cmd buffer
                        036C   864
                        036C   865          CALC_RSP DST=R1                 ; Get response buffer slot address
              D2 A5   D0 0375   866          MOVL    CDRP$L_BCNT(R5),-       ; Copy buffer length
              20 A1             0378   867                  CDC$L_BUFLENGTH(R1)     ;
           50   2C A5   D0 037A   868          MOVL    CDRP$L_LBUFH_AD(R5),R0  ; Get local buffer handle address
              04 A0   D0 037E   869          MOVL    CIBHAN$L_BNAME(R0),-    ; Copy SCS buffer name
              24 A1             0381   870                  CDC$L_BUFLNAME(R1)      ;
                        0383   871
               0206   30 0383   872          BSBW    MARK_CMD_DONE
               FCF4   31 0386   873          BRW     SIO_NXT_CMD
                        0389   874
                        0389   875
```

0389 876

```
                                0389   878          .SBTTL  SIO_UNMAP,      Unmap buffer
                                0389   879 ;+
                                0389   880 ;
                                0389   881 ;  SIO_UNMAP
                                0389   882 ;
                                0389   883 ;  This routine unmaps the specified buffer. This flow is:
                                0389   884 ;
                                0389   885 ;       - Get the CDRP for the mapped buffer
                                0389   886 ;       - Set up the SCS interface data
                                0389   887 ;       - Place CDRP on pending CDB queue
                                0389   888 ;       - Issue the unmap call
                                0389   889 ;       - Deallocate the CDRP
                                0389   890 ;       - Dispatch the next command
                                0389   891 ;
                                0389   892 ;  INPUTS:
                                0389   893 ;
                                0389   894 ;       R3                      - Address of current command
                                0389   895 ;
                                0389   896 ;  OUTPUTS:
                                0389   897 ;
                                0389   898 ;       R0-R5                   - Destroyed
                                0389   899 ;
                                0389   900 ;-
                                0389   901
                                0389   902 SIO_UNMAP:
                                0389   903
                     0276   30  0389   904          BSBW    FIND_CDRP_XFR           ; Find the mapped buffer CDRP
                                038C   905          BLBCW   R0,BUFFER_NOT_MAPPED    ; Error, branch
                                0392   906
                54   4C A5  D0  0392   907          MOVL    CDRP$L_PDT(R5),R4       ; Restore PDT address
                                0396   908          UNMAP                           ; Unmap the buffer
                                0399   909
                                0399   910          CALC_RSP  DST=R1                ; Get addr of response area
                     D2 A5  D0  03A2   911          MOVL    CDRP$L_BCNT(R5),-       ; Copy buffer length
                     20 A1      03A5   912                  CDC$L_BUFLENGTH(R1)     ;
                50   20 A5  D0  03A7   913          MOVL    CDRP$L_LBUFH_AD(R5),R0  ; Get start of buffer handle
                     04 A0  D0  03AB   914          MOVL    CIBHAN$L_BNAME(R0),-    ; SCS buffer name
                     24 A1      03AE   915                  CDC$L_BUFLNAME(R1)
                      FC4D'     30  03B0   916          BSBW    DEALLOC_CDRP            ; Deallocate the CDRP
                                03B3   917
                      01D6     30  03B3   918          BSBW    MARK_CMD_DONE
                      FC64     31  03B6   919          BRW     SIO_NXT_CMD
                                03B9   920
                                03B9   921
                                03B9   922
```

```
                        03B9    924          .SBTTL  SIO_SNDDAT,     Send data
                        03B9    925          .SBTTL  SIO_REQDAT,     Request data
                        03B9    926 ;+
                        03B9    927 ;
                        03B9    928 ;  SIO_SNDDAT
                        03B9    929 ;  SIO_REQDAT
                        03B9    930 ;
                        03B9    931 ;  This routine performs a buffer read and write operations to the remote
                        03B9    932 ;  process. The flow is as follows:
                        03B9    933 ;
                        03B9    934 ;       - Check for an active connection to remote
                        03B9    935 ;       - Find the mapped buffer CDRP
                        03B9    936 ;       - Set up the SCS interface in CDRP
                        03B9    937 ;       - Issue the request/send data call
                        03B9    938 ;       - Update the command/response status
                        03B9    939 ;       - Dispatch the next command
                        03B9    940 ;
                        03B9    941 ;  INPUTS:
                        03B9    942 ;
                        03B9    943 ;       R3                      - Address of current command
                        03B9    944 ;
                        03B9    945 ;  OUTPUTS:
                        03B9    946 ;
                        03B9    947 ;       R0-R5                   - Destroyed
                        03B9    948 ;
                        03B9    949 ;-
                        03B9    950
                        03B9    951 SIO_SNDDAT:
                        03B9    952 SIO_REQDAT:
                        03B9    953
      FC44'     30      03B9    954          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        03BC    955          BLBCW   R0,CHECK_CONN_ERR       ; No connection, branch
                        03C2    956
      023D      30      03C2    957          BSBW    FIND_CDRP_XFR           ; Find the mapped buffer CDRP
                        03C5    958          BLBCW   R0,BUFFER_NOT_MAPPED    ; Error, branch
                        03CB    959
      01AC      30      03CB    960          BSBW    SIO_SETUP_SCS           ; Setup SCS interface
      01EF      30      03CE    961          BSBW    XFR_SETUP               ; Setup the XFR CDRP
                        03D1    962
        63      91      03D1    963          CMPB    CDC$B_OPCODE(R3),-      ; Maintenance read?
        04              03D3    964                  #CDC$_REQDATA
        11      13      03D4    965          BEQL    10$                     ; Yes, branch
                        03D6    966
                        03D6    967          SEND_DATA                       ; Initiate a buffer write
        0F      11      03E5    968          BRB     20$                     ; Branch
                        03E7    969
                        03E7    970 10$:     REQUEST_DATA                    ; Initiate a buffer read
                        03F6    971
    44 A5       0E      03F6    972 20$:     INSQUE  CDRP$L_PENDFL(R5),-     ; Insert CDRP back on map list
0000'CF                 03F9    973                  CXS$Q_MAPLST            ;
    0C A3       D4      03FC    974          CLRL    CDC$L_CDRP(R3)          ; Show CDRP is gone
                        03FF    975          BLBCW   R0,XFR_ERR              ; Transfer error, branch
                        0405    976
      0184      30      0405    977          BSBW    MARK_CMD_DONE
      FC72      31      0408    978          BRW     SIO_NXT_CMD
                        040B    979
                        040B    980
```

040B 981

```
                        040B   983          .SBTTL  SIO_SNDMDAT,    Send maintenance data
                        040B   984          .SBTTL  SIO_REQMDAT,    Request maintenance data
                        040B   985 ;+
                        040B   986 ;
                        040B   987 ;  SIO_SNDMDAT
                        040B   988 ;  SIO_REQMDAT
                        040B   989 ;
                        040B   990 ;  This routine performs maintenance reads and writes. The local mapped buffer
                        040B   991 ;  is a standard buffer type, whereas the remote buffer is is described by its
                        040B   992 ;  physical address.
                        040B   993 ;
                        040B   994 ;  INPUTS:
                        040B   995 ;
                        040B   996 ;       R3                          - Address of current command
                        040B   997 ;
                        040B   998 ;  OUTPUTS:
                        040B   999 ;
                        040B  1000 ;       R0-R5                       - Destroyed
                        040B  1001 ;
                        040B  1002 ;-
                        040B  1003
                        040B  1004 SIO_SNDMDAT:
                        040B  1005 SIO_REQMDAT:
                        040B  1006
                FBF2'   30  040B  1007          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        040E  1008          BLBCW   R0,CHECK_CONN_ERR       ; No connection, branch
                        0414  1009
                01EB    30  0414  1010          BSBW    FIND_CDRP_XFR           ; Find the mapped buffer CDRP
                        0417  1011          BLBCW   R0,BUFFER_NOT_MAPPED    ; Error, branch
                        041D  1012
                015A    30  041D  1013          BSBW    SIO_SETUP_SCS           ; Setup SCS interface
                01C8    30  0420  1014          BSBW    MNT_XFR_SETUP           ; Setup the maintenance transfer CDRP
                        0423  1015
                  63    91  0423  1016          CMPB    CDC$B_OPCODE(R3),-      ; Maintenance read?
                  13        0425  1017                  #CDC$REQMDATA
                  12    13  0426  1018          BEQL    10$                     ; Yes, branch
                        0428  1019
              53  01 A3    90  0428  1020          MOVB    CDC$B_DSTPORT(R3),R3    ; Pass remote port number
                        042C  1021          WRITE_MAINT_DATA                ; Perform maintenance write
                  10    11  0438  1022          BRB     20$
                        043A  1023
              53  01 A3    90  043A  1024 10$:    MOVB    CDC$B_DSTPORT(R3),R3    ; Pass remote port number
                        043E  1025          READ_MAINT_DATA                 ; Perform maintenance read
                        044A  1026
                44 A5   0E  044A  1027 20$:    INSQUE  CDRP$L_PENDQFL(R5),-    ; Insert CDRP back on map list
                0000'CF     044D  1028                  CX$Q_MAPLST             ;
              53  0000'CF   D0  0450  1029          MOVL    CX$L_CUR_CMD,R3         ; Get address of current command
                0C A3       D4  0455  1030          CLRL    CDC$L_CDRP(R3)          ; Show CDRP is gone
                            0458  1031          BLBCW   R0,MNT_XFR_ERR          ; Error, branch
                        045E  1032
                012B    30  045E  1033          BSBW    MARK_CMD_DONE           ; Mark this command done
                FC19    31  0461  1034          BRW     SIO_NXT_CMD             ; Go execute next command
                            0464  1035
                            0464  1036
                            0464  1037
                            0464  1038
```

```
                     0464  1040          .SBTTL  SIO_READCNT,    Read port counters
                     0464  1041 ;+
                     0464  1042 ;
                     0464  1043 ;  SIO_READCNT
                     0464  1044 ;
                     0464  1045 ;  This routine reads the port's performance counters, and either sets them
                     0464  1046 ;  up for future counts or releases the counters. The contents of the counters
                     0464  1047 ;  are returned in the response buffer.
                     0464  1048 ;
                     0464  1049 ;  Part of the interface to SCS is a datagram buffer, which is used to issue
                     0464  1050 ;  the read counter request to the port. It is also used to pass the counter's
                     0464  1051 ;  contents back to the driver. Note that the address of the port queue flink
                     0464  1052 ;  in the datagram buffer is passed to SCS - NOT the start of the application
                     0464  1053 ;  data area.
                     0464  1054 ;
                     0464  1055 ;       - Check for active virtual circuit to remote node
                     0464  1056 ;       - Allocate a datagram buffer
                     0464  1057 ;       - Set up SCS interface
                     0464  1058 ;       - Issue the read counters call
                     0464  1059 ;       - Fill in appropriate status
                     0464  1060 ;       - Dispatch next command
                     0464  1061 ;
                     0464  1062 ;  INPUTS:
                     0464  1063 ;
                     0464  1064 ;       R3                          - Address of current command
                     0464  1065 ;       CDC$B_DSTPORT(R3)           - Value of node for future monitoring,
                     0464  1066 ;                                     255 for all nodes
                     0464  1067 ;
                     0464  1068 ;  OUTPUTS:
                     0464  1069 ;
                     0464  1070 ;       R0-R5                       - Destroyed
                     0464  1071 ;
                     0464  1072 ;-
                     0464  1073
                     0464  1074 SIO_READCNT:
                     0464  1075
         0139   30   0464  1076          BSBW    CHECK_VC                ; Get PDT address for local port
                     0467  1077          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                     046D  1078
         01D3   30   046D  1079          BSBW    ALLOC_CDRP_CTP          ; Allocate a CTP type CDRP
                     0470  1080          BLBCW   R0,ALLOC_CDRP_ERR       ; Error, branch
                     0476  1081
   55    0C A3   D0  0476  1082          MOVL    CDC$L_CDRP(R3),R5       ; Get CDRP address
   4C A5    54   D0  047A  1083          MOVL    R4,CDRP$L_PDT(R5)       ; Save PDT address in CDRP
                     047E  1084
                     047E  1085          ALLOC_DG_BUF                    ; Allocate a datagram buffer
                     0481  1086          BLBCW   R0,ALLOC_DG_ERR         ; Error, branch
      52    20   C2  0487  1087          SUBL2   #SCS$K_APPL_BASE-SCS$B_PPD,R2   ;                       [5-005]
                     048A  1088                                          ; Adjust R2 to point to PPD Layer
                     048A  1089
                     048A  1090 ;*************************************************************************
                     048A  1091 ;
                     048A  1092 ;  ASSUMPTION IS BEING MADE HERE...USE ASSUME MACRO
                     048A  1093 ;
                     048A  1094 ;*************************************************************************
                     048A  1095
         1F A3   95  048A  1096          TSTB    CDC$B_CNTFLG(R3)        ; Should we release the counters?
```

```
                21   13  048D  1097          BEQL    20$                     ; Yes, branch
                                 048F  1098
             FF 8F   91  048F  1099          CMPB    #255,-                  ; Monitor all nodes
             1E A3       0492  1100                  CDC$B_OTHERNODE(R3)      ;
                0E   13  0494  1101          BEQL    10$                     ; Yes, branch
                                 0496  1102
                                 0496  1103          READ_COUNTERS   -       ; Read the counters
                                 0496  1104                  RSTADR=CDC$B_OTHERNODE(R3),- ; Remote port to monitor
                                 0496  1105                  LPRNAM=CONTROLLER_NAME       ; CXDRIVER process name
                0F   11  04A2  1106          BRB     30$                     ; Branch
                                 04A4  1107
                                 04A4  1108 10$:      READ_COUNTERS   -       ; Read the counters
                                 04A4  1109                  RSTADR=0,-      ; Monitor ALL remote ports
                                 04A4  1110                  LPRNAM=CONTROLLER_NAME  ; CXDRIVER process name
                03   11  04AE  1111          BRB     30$
                                 04B0  1112
                                 04B0  1113
                                 04B0  1114 20$:      RLS_COUNTERS            ; Read and release
                                 04B3  1115
                                 04B3  1116
                                 04B3  1117 30$:      BLBCW   R0,READ_COUNT_ERR       ; Error, branch
                3F   BB  04B9  1118          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
          10 A2  1C   28  04BB  1119          MOVC3   #28,PPD$L_PO_ACK(R2),-  ; Copy counter values to response
              20 A3       04BF  1120                  CDC$L_CNTRPOACK(R3)     ; buffer
                3F   BA  04C1  1121          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                                 04C3  1122
             FB3A'   30  04C3  1123          BSBW    CLEAN_CDRP              ; Clean CDRP resources (DG buffer)
             FB37'   30  04C6  1124          BSBW    DEALLOC_CDRP            ; Deallocate the CDRP
                                 04C9  1125
             00C0    30  04C9  1126          BSBW    MARK_CMD_DONE
             FBAE    31  04CC  1127          BRW     SIO_NXT_CMD
                                 04CF  1128
                                 04CF  1129
                                 04CF  1130
                                 04CF  1131
```

```
                        04CF  1133          .SBTTL  SIO_RESET,      Send reset
                        04CF  1134 ;+
                        04CF  1135 ;
                        04CF  1136 ;  SIO_RESET
                        04CF  1137 ;
                        04CF  1138 ;  Routine to send a maintenance reset to the specified remote port.
                        04CF  1139 ;
                        04CF  1140 ;  INPUTS:
                        04CF  1141 ;
                        04CF  1142 ;       R3                          - Address of current command
                        04CF  1143 ;
                        04CF  1144 ;  OUTPUTS:
                        04CF  1145 ;
                        04CF  1146 ;       R0-R5                       - Destroyed
                        04CF  1147 ;
                        04CF  1148 ;-
                        04CF  1149
                        04CF  1150 SIO_RESET:
                        04CF  1151
              00CE   30 04CF  1152          BSBW    CHECK_VC                    ; Get PDT address for local port
                        04D2  1153          BLBCW   R0,CHECK_VC_ERR             ; Error, branch
                        04D8  1154
        50    1C A3   9A 04D8  1155          MOVZBL  CDC$B_EXTEND(R3),R0         ; Make force flag a longword
                        04DC  1156          MRESET  -                           ; Send maintenance reset
                        04DC  1157                  RSTADR=CDC$B_OTHERNODE(R3),- ; Remote port number
                        04DC  1158                  FLAG=R0                     ; Force flag
                        04E6  1159          BLBCW   R0,SEND_RESET_ERR           ; Error, branch
                        04EC  1160
              009D   30 04EC  1161          BSBW    MARK_CMD_DONE
              FB88   31 04EF  1162          BRW     SIO_NXT_CMD
                        04F2  1163
                        04F2  1164
                        04F2  1165
```

```
                              04F2          .SBTTL  SIO_START,      Send start
                              04F2     ;+
                              04F2     ;
                              04F2     ;  SJO_RESET
                              04F2     ;
                              04F2     ;  Routine to send a maintenance start to the specified remote port.
                              04F2     ;
                              04F2     ;  INPUTS:
                              04F2     ;
                              04F2     ;       R3                      - Address of current command
                              04F2     ;
                              04F2     ;  OUTPUTS:
                              04F2     ;
                              04F2     ;       R0-R5                   - Destroyed
                              04F2     ;
                              04F2     ;-
                              04F2
                              04F2     SIO_START:
                              04F2
              00AB     30     04F2          BSBW    CHECK_VC                  ; Get PDT address for local port
                              04F5          BLBCW   R0,CHECK_VC_ERR           ; Error, branch
                              04FB
 50    1C A3    01     8D     04FB          XORB3   #1,CDC$B_EXTEND(R3),R0    ; Change polarity of start flag
          50    50     9A     0500          MOVZBL  R0,R0                     ; Make it a longword
                              0503          MSTART  -                         ; Send maintenance start
                              0503                  RSTADR=CDC$B_OTHERNODE(R3),- ; Remote port number
                              0503                  FLAG=R0,-                    ; Start flag
                              0503                  START=CDC$L_STARTADR(R3)     ; Starting address
                              0511          BLBCW   R0,SEND_START_ERR         ; Error, branch
                              0517
              0072     30     0517          BSBW    MARK_CMD_DONE
              FB60     31     051A          BRW     SIO_NXT_CMD
                              051D
                              051D
                              051D
```

```
                                051D  1203          .SBTTL  SIO_GARBAGE,    Read the garbage buffer
                                051D  1204 ;+
                                051D  1205 ;
                                051D  1206 ;  SIO_GARBAGE
                                051D  1207 ;
                                051D  1208 ;  Routine to check the garbage buffer and, if unempty, to copy the
                                051D  1209 ;  entry in the garbage buffer to the response buffer.
                                051D  1210 ;
                                051D  1211 ;  This mechanism is used to catch any extra packets that are sent by
                                051D  1212 ;  the remote node.
                                051D  1213 ;
                                051D  1214 ;  INPUTS:
                                051D  1215 ;
                                051D  1216 ;       R3                          - Address of current command
                                051D  1217 ;
                                051D  1218 ;  OUTPUTS:
                                051D  1219 ;
                                051D  1220 ;       R0-R5                       - Destroyed
                                051D  1221 ;
                                051D  1222 ;-
                                051D  1223
                                051D  1224 SIO_GARBAGE:
                                051D  1225
                                051D  1226          CALC_RSP DST=R2                     ; Get address of response slot
                  11 0000'CF    0526  1227          BBCC    #0,CX$B_GARBAGE_FLAG,10$; Branch if garbage buffer empty
                     3C   BB    052C  1228          PUSHR   #^M<R2,R3,R4,R5>            ; Save regs
                                052E  1229
                                052E  1230 ;  TEMPORARY - Increase response buffer size for this response. (Garbage
                                052E  1231 ;              buffer size is 128, not 10 bytes.
                                052E  1232
                0000'CF   0A    052E  1233          MOVC3   #10,CX$T_GARBAGE_BUF,-  ; Copy entry from garbage buffer
                      18 A2     0533  1234                  CDC$T_GARBAGE(R2)       ; to response buffer
                         3C BA  0535  1235          POPR    #^M<R2,R3,R4,R5>        ; Restore regs
                         01 90  0537  1236          MOVB    #CDC$_GARBAGE,-         ; Show garbage buffer was full
                      14 A2     0539  1237                  CDC$B_GARBAGE_FLAG(R2)
                         03 11  053B  1238          BRB     20$
                                053D  1239
                                053D  1240          ASSUME  CDC$_NOGARBAGE EQ 0
                                053D  1241
                      14 A2 94  053D  1242 10$:     CLRB    CDC$B_GARBAGE_FLAG(R2)  ; Show garbage buffer was empty
                                0540  1243
                       0049 30  0540  1244 20$:     BSBW    MARK_CMD_DONE
                       F837 31  0543  1245          BRW     SIO_NXT_CMD
                                0546  1246
                                0546  1247
                                0546  1248
```

```
                                0546  1250          .SBTTL  SIO_MOVCTP,     Copy CTP data to MSG/DG buffer
                                0546  1251 ;+
                                0546  1252 ;
                                0546  1253 ;  SIO_MOVCTP
                                0546  1254 ;
                                0546  1255 ;  This routine copies CTP text from a class driver command block into the
                                0546  1256 ;  application data area of an SCS message/datagram buffer. The flow is:
                                0546  1257 ;
                                0546  1258 ;       - Clear the application area in the MSG/DG buffer
                                0546  1259 ;       - Copy the CTP text into the buffer
                                0546  1260 ;
                                0546  1261 ;  INPUTS:
                                0546  1262 ;
                                0546  1263 ;       R2                      - Address of application data area in target SCS buf
                                0546  1264 ;       R3                      - Address of current command
                                0546  1265 ;
                                0546  1266 ;  OUTPUTS:
                                0546  1267 ;
                                0546  1268 ;       All registers preserved
                                0546  1269 ;       Text copied
                                0546  1270 ;
                                0546  1271 ;-
                                0546  1272
                                0546  1273 SIO_MOVCTP:
                                0546  1274
                                0546  1275 ;  Clear out entire message or datagram buffer.
                                0546  1276
                          3F    BB  0546  1277          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
              50    52    20    C3  0548  1278          SUBL3   #SCS$K_APPL_BASE-SCS$B_PPD,R2,R0
                                054C  1279                                          ; Back up to port header in blk [5-001]
          51    00000000'GF    3C  054C  1280          MOVZWL  G^SCS$GW_MAXMSG,R1      ; Assume message
                          3C    91  0553  1281          CMPB    #DYN$C_CIMSG,-          ; Is this a MSG?
                             0A A0    0555  1282                  PPD$B_TYPE(R0)          ;
                          07    13  0557  1283          BEQLU   10$                     ; Yes, branch
          51    00000000'GF    3C  0559  1284          MOVZWL  G^SCS$GW_MAXDG,R1       ; Set DG data area size
                          0C    BB  0560  1285 10$:     PUSHR   #^M<R2,R3>              ; Save regs
    62    51    00    FE AF   00    2C  0562  1286          MOVC5   #0,,#0,R1,(R2)          ; Clear the data area in buffer
                          0C    BA  0569  1287          POPR    #^M<R2,R3>              ; Restore regs
                                056B  1288
                                056B  1289 ;  Copy the CTP request information into the message buffer. The size to
                                056B  1290 ;  be copied is equal to the text length minus the command overhead.
                                056B  1291
                    50    08 A3   3C  056B  1292          MOVZWL  CDC$W_TXTLEN(R3),R0     ; Get text length
                       50   14    C2  056F  1293          SUBL2   S^#CDC$K_CMDSPEC,R0     ; Subtract CDC command overhead
                    14 A3   50    28  0572  1294          MOVC3   R0,CDC$K_CMDSPEC(R3),-  ; Copy CDT info into MSG/DG buffer
                            62        0576  1295                  (R2)                    ;
                          3F    BA  0577  1296          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore registers
                          05    05  0579  1297          RSB
                                057A  1298
                                057A  1299
```

```
                        057A  1301          .SBTTL  SIO_SETUP_SCS,  Setup SCS interface
                        057A  1302 ;+
                        057A  1303 ;
                        057A  1304 ;  SIO_SETUP_SCS
                        057A  1305 ;
                        057A  1306 ;  This routine sets up standard registers for interface to SCS routines.
                        057A  1307 ;  IT IS ESSENTIAL THAT AN ACTIVE CONNECTION EXISTS BEFORE CALLING THIS
                        057A  1308 ;  ROUTINE. This can be accomplished by calling the routine CHECK_ACTV_CONN
                        057A  1309 ;  prior to calling this routine.
                        057A  1310 ;
                        057A  1311 ;  INPUTS·
                        057A  1312 ;
                        057A  1313 ;       R1                      - CDB address (should be in active state)
                        057A  1314 ;       R3                      - Address of current command
                        057A  1315 ;       CDB$L_CDT(R1)           - CDT address
                        057A  1316 ;       CDB$L_PDT(R1)           - PDT address
                        057A  1317 ;       CDC$L_CDRP(R3)          - CDRP address
                        057A  1318 ;
                        057A  1319 ;  OUTPUTS:
                        057A  1320 ;
                        057A  1321 ;       R4                      - PDT address
                        057A  1322 ;       R5                      - CDRP address
                        057A  1323 ;       CDRP$L_CDT(R5)          - CDT address
                        057A  1324 ;       CDRP$L_PDT(R5)          - PDT address
                        057A  1325 ;
                        057A  1326 ;-
                        057A  1327
                        057A  1328 SIO_SETUP_SCS::
                        057A  1329
55    0C A3     D0      057A  1330          MOVL    CDC$L_CDRP(R3),R5       ; Get CDRP address
54    20 A1     D0      057E  1331          MOVL    CDB$L_PDT(R1),R4        ; Get PDT address
4C A5   54      D0      0582  1332          MOVL    R4,CDRP$L_PDT(R5)       ; Save PDT address in CDRP
      1C A1     D0      0586  1333          MOVL    CDB$L_CDT(R1),-         ; Save CDT address in CDRP
      24 A5             0589  1334                  CDRP$L_CDT(R5)
                05      058B  1335          RSB
                        058C  1336
                        058C  1337
```

```
                        058C  1339          .SBTTL  MARK_CMD_DONE,  Mark this command as done
                        058C  1340 ;+
                        058C  1341 ;
                        058C  1342 ;  MARK_CMD_DONE
                        058C  1343 ;
                        058C  1344 ;  Routine called after completing the execution of a command in a
                        058C  1345 ;  command buffer. The flag bits SCS_DONE and CMD_DONE are set, and the
                        058C  1346 ;  done count for the overall command buffer is incremented.
                        058C  1347 ;
                        058C  1348 ;  INPUTS:
                        058C  1349 ;
                        058C  1350 ;       CX$L_CUR_CMD            - Address of current command
                        058C  1351 ;
                        058C  1352 ;  OUTPUTS:
                        058C  1353 ;
                        058C  1354 ;       R1                      - Destroyed
                        058C  1355 ;
                        058C  1356 ;-
                        058C  1357
                        058C  1358 MARK_CMD_DONE::
                        058C  1359
    53  0000'CF    D0   058C  1360          MOVL    CX$L_CUR_CMD,R3         ; Reset current cmd address
                        0591  1361          SET_CMD_FLAG CMD_DONE           ; Command completed successfully
    51  0000'CF    D0   0597  1362          MOVL    CX$L_CMDBUFBGN,R1       ; Get start of command buffer
          01 A1    96   059C  1363          INCB    CDC$B_DONECNT(R1)       ; Increment done count
                   05   059F  1364          RSB
                        05A0  1365
                        05A0  1366
```

```
                        05A0  1368          .SBTTL  CHECK_VC,        Check for virtual circuit
                        05A0  1369          .SBTTL  GET_PDT_ADDR,    Get local PDT address
                        05A0  1370 ;+
                        05A0  1371 ;
                        05A0  1372 ;  CHECK_VC
                        05A0  1373 ;  GET_PDT_ADDR
                        05A0  1374 ;
                        05A0  1375 ;  This routine checks for the existance of a virtual circuit to the
                        05A0  1376 ;  specified remote port over the specified local port.
                        05A0  1377 ;
                        05A0  1378 ;  INPUTS:
                        05A0  1379 ;
                        05A0  1380 ;       R3                          - Address of current command
                        05A0  1381 ;
                        05A0  1382 ;  OUTPUTS:
                        05A0  1383 ;
                        05A0  1384 ;       R0                          - Status
                        05A0  1385 ;       R1                          - Path block address
                        05A0  1386 ;       R4                          - PDT address (if success)
                        05A0  1387 ;
                        05A0  1388 ;-
                        05A0  1389
                        05A0  1390 CHECK_VC::
                        05A0  1391 GET_PDT_ADDR:
                        05A0  1392
7F   10 A3   D0         05A0  1393          MOVL    CDC$L_LPORT(R3),-(SP)   ; Put port name on stack
       7E   D4          05A4  1394          CLRL    -(SP)                   ; Clear next longword
7E   01 A3   9A         05A6  1395          MOVZBL  CDC$B_DSTPORT(R3),-(SP) ; Put local port # on stack
                        05AA  1396          CONFIG_PTH -                    ; Perform config path
                        05AA  1397                  STAADR=(SP),-           ; Input array
                        05AA  1398                  OUTBUF=0                ; Don't want output
   5E   0C   C0         05B5  1399          ADDL    #12,SP                  ; Clean up stack
     04 50   E9         05B8  1400          BLBC    R0,10$                  ; No VC, branch
54   2C A1   D0         05BB  1401          MOVL    PB$L_PDT(R1),R4         ; Get PDT address
                 05     05BF  1402 10$:     RSB
                        05C0  1403
                        05C0  1404
                        05C0  1405
```

```
                     05C0  1407          .SBTTL  XFR_SETUP,      Setup data XFR parameters
                     05C0  1408 ;+
                     05C0  1409 ;
                     05C0  1410 ;  XFR_SETUP
                     05C0  1411 ;
                     05C0  1412 ;  This routine sets up the SCS interface for data transfers. The interface
                     05C0  1413 ;  consists of the following fields in the CDRP:
                     05C0  1414 ;
                     05C0  1415 ;       - CDRP$L_CDT (Already setup)
                     05C0  1416 ;       - CDRP$L_LBOFF
                     05C0  1417 ;       - CDRP$L_RBOFF
                     05C0  1418 ;       - CDRP$L_LBUFH_AD
                     05C0  1419 ;       - CDRP$L_RBUFH_AD
                     05C0  1420 ;       - CDRP$L_XCT_LEN
                     05C0  1421 ;
                     05C0  1422 ;  In the buffer handles:
                     05C0  1423 ;
                     05C0  1424 ;       - CIBHAN$L_RCONID
                     05C0  1425 ;       - CIBHAN$L_LCONID
                     05C0  1426 ;       - CIBHAN$L_BNAME  (in both local and remote)
                     05C0  1427 ;
                     05C0  1428 ;
                     05C0  1429 ;  INPUTS:
                     05C0  1430 ;
                     05C0  1431 ;       R1                          - CDB address
                     05C0  1432 ;       R3                          - Address of current command
                     05C0  1433 ;       R4                          - PDT address
                     05C0  1434 ;       R5                          - CDRP (XFR) address
                     05C0  1435 ;       CDRP$L_CDT                  - CDT address
                     05C0  1436 ;
                     05C0  1437 ;  OUTPUTS:
                     05C0  1438 ;
                     05C0  1439 ;       R0-R2                       - Destroyed
                     05C0  1440 ;
                     05C0  1441 ;-
                     05C0  1442
                     05C0  1443 XFR_SETUP:
                     05C0  1444
      28 A3    D0    05C0  1445          MOVL    CDC$L_BUFLOFSET(R3),-   ; Set local byte offset
      30 A5          05C3  1446                  CDRP$L_LBOFF(R5)        ;
      30 A3    D0    05C5  1447          MOVL    CDC$L_BUFROFSET(R3),-   ; Set remote byte offset
      38 A5          05C8  1448                  CDRP$L_RBOFF(R5)        ;
50    1C A1    D0    05CA  1449          MOVL    CDB$L_CDT(R1),R0        ; Get CDT addtess
52    2C A5    D0    05CE  1450          MOVL    CDRP$L_LBUFH_AD(R5),R2  ; Get local buffer handle address
      14 A0    D0    05D2  1451          MOVL    CDT$L_RCONID(R0),-      ; Copy REMOTE conid into LOCAL
      08 A2          05D5  1452                  CIBHAN$L_RCONID(R2)     ; buffer handle
52    34 A5    D0    05D7  1453          MOVL    CDRP$L_RBUFH_AD(R5),R2  ; Get remote buffer handle address
      2C A3    D0    05DB  1454          MOVL    CDC$L_BUFRNAME(R3),-    ; Set destination buffer name
      04 A2          05DE  1455                  CIBHAN$L_BNAME(R2)      ;
      18 A0    D0    05E0  1456          MOVL    CDT$L_LCONID(R0),-      ; Copy LOCAL conid into REMOTE
      08 A2          05E3  1457                  CIBHAN$L_RCONID(R2)     ; buffer handle
      20 A3    D0    05E5  1458          MOVL    CDC$L_BUFLENGTH(R3),-   ; Set transfer length
      3C A5          05E8  1459                  CDRP$L_XCT_LEN(R5)      ;
                     05EA  1460
                     05EA  1461 ;  Fill in the CDRP flag specifying whether to use 512 or 576 byte packets
                     05EA  1462 ;  when performing the buffer trasnfer. The flag is the MSB of the CDRP flags
                     05EA  1463 ;  byte and is used as follows:
```

```
                     05EA  1464 ;
                     05EA  1465 ;       0       - 512 byte packets
                     05EA  1466 ;       1       - 576 byte packets
                     05EA  1467 ;
                     05EA  1468 ;       ASSUME  CDC$PS512 EQ 0
                     05EA  1469 ;
                     05EA  1470 ;       CLRB    CDRP$B_FLAGS(R5)        ; Assume 512 byte packets
                     05EA  1471 ;       TSTB    CDC$PKTSIZ(R2)          ; 512 byte packets?
                     05EA  1472 ;       BEQL    25$                     ; Yes, branch
                     05EA  1473 ;       MOVB    #^X80,CDRP$B_FLAGS(R5)  ; Set up for 576 byte packets
                     05EA  1474 ;25$:
                     05EA  1475
                05   05EA  1476         RSB
                     05EB  1477
                     05EB  1478
```

```
                                05EB  1480          .SBTTL  MNT_XFR_SETUP,  Setup maintenance transfer CDRP
                                05EB  1481 ;+
                                05EB  1482 ;
                                05EB  1483 ;  MNT_XFR_SETUP
                                05EB  1484 ;
                                05EB  1485 ;  This routine sets up the SCS interface for maintenance data transfers. The
                                05EB  1486 ;  interface consists of the following fields in the CDRP:
                                05EB  1487 ;
                                05EB  1488 ;       - CDRP$L_LBOFF
                                05EB  1489 ;       - CDRP$L_LBUFH_AD
                                05EB  1490 ;       - CDRP$L_RBUFH_AD
                                05EB  1491 ;       - CDRP$L_XCT_LEN
                                05EB  1492 ;
                                05EB  1493 ;  In the local buffer handle:
                                05EB  1494 ;
                                05EB  1495 ;       - CIBHAN$L_BOFF    (standard buffer offset)
                                05EB  1496 ;       - CIBHAN$L_BNAME   (standard buffer name)
                                05EB  1497 ;
                                05EB  1498 ;  In the remote buffer handle:
                                05EB  1499 ;
                                05EB  1500 ;       - CIBHAN$L_BOFF   (physical address of maintenance buffer)
                                05EB  1501 ;       - CIBHAN$L_BNAME (value of 1)
                                05EB  1502 ;
                                05EB  1503 ;
                                05EB  1504 ;  INPUTS:
                                05EB  1505 ;
                                05EB  1506 ;       R1                          - CDB address
                                05EB  1507 ;       R3                          - Address of current command
                                05EB  1508 ;       R4                          - PDT address
                                05EB  1509 ;       R5                          - CDRP (XFR) address
                                05EB  1510 ;
                                05EB  1511 ;  OUTPUTS:
                                05EB  1512 ;
                                05EB  1513 ;       R0-R2                       - Destroyed
                                05EB  1514 ;
                                05EB  1515 ;-
                                05EB  1516
                                05EB  1517 MNT_XFR_SETUP:
                                05EB  1518
      28 A3    D0               05EB  1519          MOVL    CDC$L_BUFLOFSET(R3),-   ; Set local byte offset
      30 A5                     05EE  1520                  CDRP$L_LBOFF(R5)
52    34 A5    D0               05F0  1521          MOVL    CDRP$L_RBUFH_AD(R5),R2  ; Get remote buffer handle address
      2C A3    D0               05F4  1522          MOVL    CDC$L_BUFRNAME(R3),-    ; Copy physical address of maintenace
         62                     05F7  1523                  CIBHAN$L_BOFF(R2)       ; buffer to remote buffer handle
04 A2    01    D0               05F8  1524          MOVL    #1,CIBHAN$L_BNAME(R2)   ; Fill in remote buffer name (1)
      20 A3    D0               05FC  1525          MOVL    CDC$L_BUFLENGTH(R3),-   ; Set transfer length
      3C A5                     05FF  1526                  CDRP$L_XCT_LEN(R5)
                                0601  1527
                                0601  1528 ;  Fill in the CDRP flag specifying whether to use 512 or 576 byte packets
                                0601  1529 ;  when performing the buffer trasnfer. The flag is the MSB of the CDRP flags
                                0601  1530 ;  byte and is used as follows:
                                0601  1531 ;
                                0601  1532 ;       0          - 512 byte packets
                                0601  1533 ;       1          - 576 byte packets
                                0601  1534 ;
                                0601  1535 ;       ASSUME  CDC$PS512 EQ 0
                                0601  1536 ;
```

```
                     0601  1537 ;        CLRB    CDRP$B_FLAGS(R5)         ; Assume 512 byte packets
                     0601  1538 ;        TSTB    CDC$PKTSIZ(R2)           ; 512 byte packets?
                     0601  1539 ;        BEQL    25$                      ; Yes, branch
                     0601  1540 ;        MOVB    #^X80,CDRP$B_FLAGS(R5)   ; Set up for 576 byte packets
                     0601  1541 ;25$:
                     0601  1542
                  05 0601  1543         RSB
                     0602  1544
```

```
                        0602  1546          .SBTTL  FIND_CDRP_XFR,  Locate XFR CDRP on the map list
                        0602  1547 ;+
                        0602  1548 ;
                        0602  1549 ;  FIND_CDRP_XFR
                        0602  1550 ;
                        0602  1551 ;  This routine will locate the data xfr CDRP that corresponds to the BUFFER
                        0602  1552 ;  NAME supplied with SEND_DATA, REQUEST_DATA, UNMAP class driver commands.
                        0602  1553 ;
                        0602  1554 ;  The CDRP is removed from the mapped buffer queue.
                        0602  1555 ;
                        0602  1556 ;  INPUTS:
                        0602  1557 ;
                        0602  1558 ;       R3                      - Address of current command
                        0602  1559 ;
                        0602  1560 ;  OUTPUTS:
                        0602  1561 ;
                        0602  1562 ;       R0                      - Status
                        0602  1563 ;       R5                      - CDRP address (if success)
                        0602  1564 ;       CDC$L_CDRP(R3)          - CDRP address
                        0602  1565 ;       All other registers preserved
                        0602  1566 ;
                        0602  1567 ;-
                        0602  1568
                        0602  1569          .ENABLE LSB
                        0602  1570
                        0602  1571 FIND_CDRP_XFR:
                        0602  1572
          06   BB       0602  1573          PUSHR   #^M<R1,R2>              ; Save regs
55  0000'CF   DE       0604  1574          MOVAL   CX$Q_MAPLST,R5          ; Get address of map list
     50   55  D0       0609  1575          MOVL    R5,R0                   ; Save listhead address
     55   65  D0       060C  1576 10$:     MOVL    (R5),R5                 ; Get next CDRP on list
     50   55  D1       060F  1577          CMPL    R5,R0                   ; Back at start again?
          24  13       0612  1578          BEQL    20$                     ; Error, branch
                        0614  1579
 00000044 8F   C3       0614  1580          SUBL3   #CDRP$L_PENDQFL,-       ; Back up to to of CDRP
     51   55            061A  1581          R5,R1                   ;
       2C A1   D0       061C  1582          MOVL    CDRP$L_LBUFH_AD(R1),-   ; Get local buffer address
          52            061F  1583          R2                      ;
       04 A2   D1       0620  1584          CMPL    CIBHAN$L_BNAME(R2),-    ; Compare buffer names
       24 A3            0623  1585          CDC$L_BUFLNAME(R3)      ;
          E5   12       0625  1586          BNEQU   10$                     ; No match, loop
                        0627  1587
                        0627  1588 ;  Found a match. Copy the CDRP address to the command buffer and also
                        0627  1589 ;  return its address in R5. Return with success status.
                        0627  1590
    50   44 A1   0F     0627  1591          REMQUE  CDRP$L_PENDQFL(R1),R0   ; Remove CDRP from map queue
  0C A3   51   D0       062B  1592          MOVL    R1,CDC$L_CDRP(R3)       ; Copy CDRP addr to cmd buffer
     55   51   D0       062F  1593          MOVL    R1,R5                   ; Return CDRP address in R5
     50   01   3C       0632  1594          MOVZWL  #SS$_NORMAL,R0          ; Set successful status
          06   BA       0635  1595          POPR    #^M<R1,R2>              ; Restore regs
               05       0637  1596          RSB
                        0638  1597
                        0638  1598 ;  CDRP could not be found on the appropriate queue. Return with error
                        0638  1599 ;  status.
                        0638  1600
          06   BA       0638  1601 20$:     POPR    #^M<R1,R2>              ; Restore regs
50  0334 8F   3C       063A  1602          MOVZWL  #SS$_DEVREQERR,R0       ; Set bad parameter error
```

```
                     51   01   3C  063F  1603          MOVZWL  #CX$_CDRPNFND,R1        ; and the bad buffer name
                          05   0642  1604          RSB
                               0643  1605
                               0643  1606          .DISABLE LSB
                               0643  1607
                               0643  1608
                               0643  1609
```

```
                                0643  1611          .SBTTL  ALLOC_CDRP_CTP, Allocate a CTP type CDRP
                                0643  1612 ;+
                                0643  1613 ;
                                0643  1614 ;  ALLOC_CDRP_CTP
                                0643  1615 ;
                                0643  1616 ;  Routine to allocate a normal sized CDRP used to interface to SCS for
                                0643  1617 ;  sending messages and datagrams. The flow is:
                                0643  1618 ;
                                0643  1619 ;       - Allocate CDRP from nonpaged pool
                                0643  1620 ;       - Initialize the appropriate fields in the CDRP
                                0643  1621 ;       - Copy CDRP address into command block
                                0643  1622 ;
                                0643  1623 ;  INPUTS:
                                0643  1624 ;
                                0643  1625 ;       R3                      - Address of current command
                                0643  1626 ;
                                0643  1627 ;  OUTPUTS:
                                0643  1628 ;
                                0643  1629 ;       R0                      - Status (SS$_NORMAL or SS$_INSFMEM)
                                0643  1630 ;       R5                      - CDRP address
                                0643  1631 ;       CDC$L_CDRP(R3)          - CDRP address
                                0643  1632 ;       All other registers     - Preserved
                                0643  1633 ;
                                0643  1634 ;-
                                0643  1635
                                0643  1636 ALLOC_CDRP_CTP:
                                0643  1637
                         0E     0643  1638          PUSHR   #^M<R1,R2,R3>           ; Save regs
                 51      50 8F  0645  1639          MOVZBL  #CDRP$K_CTPLEN,R1       ; Length of CTP type CDRP
          00000000'GF           0649  1640          JSB     G^EXE$ALONONPAGED       ; Ask VMS for some memory
                 18      50     064F  1641          BLBC    R0,40$                  ; Error, branch
                  F9AB'         0652  1642          BSBW    CLEAR_BUFFER            ; Clear the entire CDRP
           08 A2        51     0655  1643          MOVW    R1,CDRP$W_CDRPSIZE(R2)  ; Set size of CDRP
                        39     0659  1644          MOVB    #DYN$C_CDRP,-           ; Set structure type
                  0A    A2     065B  1645                  CDRP$B_CD_TYPE(R2)
              55        52     065D  1646          MOVL    R2,R5                   ; Return CDRP address in R5
                        0E     0660  1647          POPR    #^M<R1,R2,R3>           ; Restore regs
           0C A3        55     0662  1648          MOVL    R5,CDC$L_CDRP(R3)       ; Save CDRP address in cmd buffer
              50        01     0666  1649          MOVZWL  #SS$_NORMAL,R0          ; Return success status
                               0669  1650          RSB
                               066A  1651
        50      0124 8F        066A  1652 40$:     MOVZWL  #SS$_INSFMEM,R0         ; Set appropriate error status
                        0E     066F  1653          POPR    #^M<R1,R2,R3>           ; Restore regs
                               0671  1654          RSB                             ; Return with error
                               0672  1655
                               0672  1656
                               0672  1657
```

```
                                0672  1659          .SBTTL  ALLOC_CDRP_XFR, Allocate a CDRP for a XFR command
                                0672  1660 ;+
                                0672  1661 ;
                                0672  1662 ;  ALLOC_CDRP_XFR
                                0672  1663 ;
                                0672  1664 ;  This routine allocates and initiates a transfer type CDRP to be used
                                0672  1665 ;  for data transfer operations. The CDRP consists of and IRP portion
                                0672  1666 ;  which is at a negative offset from the start of the CDRP. The flow is
                                0672  1667 ;  as follows:
                                0672  1668 ;
                                0672  1669 ;       - Allocate IRP/CDRP from NONPAGED pool
                                0672  1670 ;       - Setup size/type fields to make this block an IRP/CDRP
                                0672  1671 ;       - Initialize appropriate CDRP fields
                                0672  1672 ;       - Copy addr of CDRP portion of the IRP/CDRP into command block
                                0672  1673 ;
                                0672  1674 ;  INPUTS:
                                0672  1675 ;
                                0672  1676 ;       R3                      - Address of current command
                                0672  1677 ;
                                0672  1678 ;  OUTPUTS:
                                0672  1679 ;
                                0672  1680 ;       R0                      - Status
                                0672  1681 ;       R5                      - CDRP address
                                0672  1682 ;       CDC$L_CDRP(R3)          - CDRP address
                                0672  1683 ;       CDRP is properly initialized
                                0672  1684 ;
                                0672  1685 ;-
                                0672  1686
                                0672  1687 ALLOC_CDRP_XFR:
                                0672  1688
                      06    BB  0672  1689          PUSHR   #^M<R1,R2>              ; Save regs
51    000000C8 8F     D0  0674  1690          MOVL    #<CDRP$K_XFRLEN+-       ; Length of IRP/CDRP
                                067B  1691                  IRP$C_CDRP>,R1          ;
      00000000'GF     16  067B  1692          JSB     G^EXE$ALONONPAGED       ; Allocate the CDRP
            36 50     E9  0681  1693          BLBC    R0,20$                  ; Error, branch
             F979'    30  0684  1694          BSBW    CLEAR_BUFFER            ; Clear the whole block
   08 A2     51      B0  0687  1695          MOVW    R1,IRP$W_SIZE(R2)       ; Set size of entire block
                0A     90  068B  1696          MOVB    #DYN$C_IRP,-            ; set type
          0A A2           068D  1697                  IRP$B_TYPE(R2)          ;
   53   0000'CF     D0  068F  1698          MOVL    CX$X_IRPSAV,R3          ; Get IRP addr
          0B A3     D0  0694  1699          MOVL    IRP$B_RMOD(R3),-        ; Set access mode
          0B A2           0697  1700                  IRP$B_RMOD(R2)
   55     60 A2     DE  0699  1701          MOVAL   IRP$C_CDRP(R2),R5       ; Get address of CDRP portion
                39     90  069D  1702          MOVB    #DYN$C_CDRP,-           ; Set type of structure
          0A A5           069F  1703                  CDRP$B_CD_TYPE(R5)      ;
   53   0000'CF     D0  06A1  1704          MOVL    CX$L_CUR_CMD,R3         ; Reset ptr to current cmd
   0C A3     55     D0  06A6  1705          MOVL    R5,CDC$L_CDRP(R3)       ; Save CDRP address in cmd buffer
          50 A5     DE  06AA  1706          MOVAL   CDRP$T_CX_LBUFHNDL(R5),- ; Set addr of local buffer handle
          2C A5           06AD  1707                  CDRP$L_LBUFH_AD(R5)
          5C A5     DE  06AF  1708          MOVAL   CDRP$T_CX_RBUFHNDL(R5),- ; Set remote buffer handle addr
          34 A5           06B2  1709                  CDRP$L_RBUFH_AD(R5)
                06     BA  06B4  1710          POPR    #^M<R1,R2>              ; Restore regs
          50    01     3C  06B6  1711          MOVZWL  #SS$_NORMAL,R0          ; Set success status
                05     05  06B9  1712          RSB
                           06BA  1713
                06     BA  06BA  1714 20$:     POPR    #^M<R1,R2>              ; Restore regs
   50    0124 8F     3C  06BC  1715          MOVZWL  #SS$_INSFMEM,R0         ; Set error status
```

```
 05  06C1  1716          RSB
     06C2  1717
     06C2  1718
     06C2  1719
```

```
                                06C2  1721          .SBTTL  ALLOC_CDB,      Allocate connection block
                                06C2  1722 ;+
                                06C2  1723 ;
                                06C2  1724 ;  Routine to allocate and initialize a connection block, and link the
                                06C2  1725 ;  connection block to the driver's connection database.
                                06C2  1726 ;
                                06C2  1727 ;  INPUTS:
                                06C2  1728 ;
                                06C2  1729 ;       R3                          - Address of current command
                                06C2  1730 ;
                                06C2  1731 ;  OUTPUTS:
                                06C2  1732 ;
                                06C2  1733 ;       R0                          - Status
                                06C2  1734 ;       R5                          - CDB address (if success)
                                06C2  1735 ;       All other registers         - Preserved
                                06C2  1736 ;
                                06C2  1737 ;-
                                06C2  1738
                                06C2  1739 ALLOC_CDB:
                                06C2  1740
                         0E     BB  06C2  1741          PUSHR   #^M<R1,R2,R3>           ; Save regs
                     51  32     9A  06C4  1742          MOVZBL  #CDB$K_LEN,R1           ; Size to CDB to allocate
                00000000'GF     16  06C7  1743          JSB     G^EXE$ALONONPAGED       ; Allocate a CDB from nonpaged pool
                    51 50     E9  06CD  1744          BLBC    R0,10$                  ; Error, branch
                                06D0  1745
                       F92D'    30  06D0  1746          BSBW    CLEAR_BUFFER            ; Clear the CDB
                08 A2   51     B0  06D3  1747          MOVW    R1,CDB$W_SIZE(R2)       ; Set CDB size
                        33     90  06D7  1748          MOVB    #DYN$C_CDB,-            ; Set type field
                     0A A2         06D9  1749                  CDB$B_TYPE(R2)
                53   08 AE     D0  06DB  1750          MOVL    8(SP),R3                ; Restore command address
                                06DF  1751
                                06DF  1752          ASSUME  CDB$B_RSTNADR+8 EQ CDB$T_LPRTNAM
                                06DF  1753
                    01 A3     9A  06DF  1754          MOVZBL  CDC$B_DSTPORT(R3),-     ; Fill in l.o. remote port number
                    0E A2         06E2  1755                  CDB$B_RSTNADR(R2)       ;
                    12 A2     D4  06E4  1756          CLRL    4+CDB$B_RSTNADR(R2)     ; Clear h.o. remote port number
                    10 A3     D0  06E7  1757          MOVL    CDC$L_LPORT(R3),-       ; Fill in local port name       [5-007]
                    16 A2         06EA  1758                  CDB$T_LPRTNAM(R2)       ;
                    28 A2     94  06EC  1759          CLRB    CDB$B_CDRPCNT(R2)       ; Clear pending CDRP count
                    2A A2     9E  06EF  1760          MOVAB   CDB$L_CDRPQFL(R2),-     ; Init pending queue listhead
                    2A A2         06F2  1761                  CDB$L_CDRPQFL(R2)
                    2A A2     9E  06F4  1762          MOVAB   CDB$L_CDRPQFL(R2),-
                    2E A2         06F7  1763                  CDB$L_CDRPQBL(R2)
                                06F9  1764
                                06F9  1765          CALC_CDB_LIST DST=R0            ; Get addr of CDB list for local port
                 51   01 A3     9A  0707  1766          MOVZBL  CDC$B_DSTPORT(R3),R1    ; Get remote port number
                  6041   52     D0  070B  1767          MOVL    R2,(R0)[R1]             ; Write CDB address to CDB list
                0C A3    52     D0  070F  1768          MOVL    R2,CDC$L_CDB_AD(R3)     ; Copy CDB addr into cmd buf
              24 A2    6041     DE  0713  1769          MOVAL   (R0)[R1],-              ; Point the CDB back to its slot
                                0718  1770                  CDB$L_CDBSLOT(R2)       ; in the CDB list
                   55   52     D0  0718  1771          MOVL    R2,R5                   ; Return CDB address in R5
                        0E     BA  071B  1772          POPR    #^M<R1,R2,R3>           ; Restore regs
                   50   01     D0  071D  1773          MOVL    #SS$_NORMAL,R0          ; Set success status
                                05  0720  1774          RSB
                                0721  1775
                        0E     BA  0721  1776 10$:     POPR    #^M<R1,R2,R3>           ; Restore regs
           50   00000124 8F     D0  0723  1777          MOVL    #SS$_INSFMEM,R0         ; Set error status
```

```
                05  072A  1778          RSB                             ; Return
                    072B  1779
                    072B  1780
```

```
                                072B  1782          .SBTTL  SET_RSP_STATUS, Set response status
                                072B  1783          .SBTTL  SET_RSP_STATUS_2, Set full response status
                                072B  1784 ;+
                                072B  1785 ;
                                072B  1786 ;  SET RESPONSE STATUS
                                072B  1787 ;  SET RESPONSE STATUS_2
                                072B  1788 ;
                                072B  1789 ;  Routine to copy the response status from R0 to the R0 status field
                                072B  1790 ;  for the current command. The alternate entry copies the R1 status also.
                                072B  1791 ;
                                072B  1792 ;  INPUTS:
                                072B  1793 ;
                                072B  1794 ;       R3                          - Address of current command
                                072B  1795 ;
                                072B  1796 ;  OUTPUTS:
                                072B  1797 ;
                                072B  1798 ;       R1                          - Destroyed (for SET_RSP_STATUS)
                                072B  1799 ;                                   - Preserved (for SET_RSP_STATUS_2)
                                072B  1800 ;       All other registers         - Preserved
                                072B  1801 ;
                                072B  1802 ;-
                                072B  1803
                                072B  1804 SET_RSP_STATUS:
                                072B  1805
                          51    D4  072B  1806          CLRL    R1                          ; No secondary status
                                072D  1807
                                072D  1808 SET_RSP_STATUS_2:
                                072D  1809
                             50 DD  072D  1810          PUSHL   R0                          ; Save R0 status
                          50 02 A3  3C  072F  1811          MOVZWL  CDC$W_RSPOFSET(R3),R0       ; Get offset into rsp buffer for cmd
                     50 0000'CF     C0  0733  1812          ADDL2   CX$L_RSPBUFBGN,R0           ; Get address of response slot
                        0C A0  6E   D0  0738  1813          MOVL    (SP),CDC$L_ROSTATUS(R0)     ; Copy R0 status to response slot
                        10 A0  51   D0  073C  1814          MOVL    R1,CDC$L_RTSTATUS(R0)       ; Copy R1 status
                             50 8ED0  0740  1815          POPL    R0                          ; Restore R0
                                05  0743  1816          RSB
                                    0744  1817
                                    0744  1818
```

```
                                0744  1820          .SBTTL  CHECK_RST_STR,  Prepare for arrival of reset or start packet
                                0744  1821 ;+
                                0744  1822 ;
                                0744  1823 ;  CHECK_RST_STR
                                0744  1824 ;
                                0744  1825 ;  If a CTP generate reset or start message is being sent to the remote
                                0744  1826 ;  responder, prepare for the arrival of the reset or start packet. This
                                0744  1827 ;  involves allocating a CDRP and issuing a CATCH_RST or CATCH_STR call to
                                0744  1828 ;  PADRIVER. PADRIVER suspends the thread until a reset or start packet is
                                0744  1829 ;  received, at which point the thread is resumed. The reset or start packet
                                0744  1830 ;  is then placed in the response buffer.
                                0744  1831 ;
                                0744  1832 ;  This routine issues a subroutine call to a routine that actually issues
                                0744  1833 ;  the CATCH_RST or CATCH_STR. This allows the CXDRIVER thread to maintain
                                0744  1834 ;  control and send the actual CTP request.
                                0744  1835 ;
                                0744  1836 ;  INPUTS:
                                0744  1837 ;
                                0744  1838 ;       R2                      - Address of CTP request
                                0744  1839 ;       R3                      - Address of current command
                                0744  1840 ;       R4                      - PDT address
                                0744  1841 ;
                                0744  1842 ;  OUTPUTS:
                                0744  1843 ;
                                0744  1844 ;-
                                0744  1845
                                0744  1846 CHECK_RST_STR:
                                0744  1847
          62    06    91        0744  1848          CMPB    #CTP$GENRSTREQ,(R2)     ; CTP generate reset request
                09    13        0747  1849          BEQL    10$                     ; Yes, branch
          62    07    91        0749  1850          CMPB    #CTP$GENSTRREQ,(R2)     ; CTP generate start request
                04    13        074C  1851          BEQL    10$                     ; Yes, branch
          50    01    D0        074E  1852          MOVL    #1,R0                   ; This is not a CTP generate reset or
                      05        0751  1853          RSB                             ; start request - return success status
                                0752  1854
                24    BB        0752  1855 10$:     PUSHR   #^M<R2,R5>              ; Save regs
    51  00000050  8F    D0      0754  1856          MOVL    #CDRP$K_CTPLEN,R1       ; Get length of CDRP to allocate
        00000000'GF    16        075B  1857          JSB     G^EXE$ALONONPAGED       ; Allocate a CDRP
             09    50    E9      0761  1858          BLBC    R0,20$                  ; Error, branch
          55    52    D0        0764  1859          MOVL    R2,R5                   ; Get CDRP address in R5
       08 A5    51    B0        0767  1860          MOVW    R1,CDRP$W_CDRPSIZE(R5)  ; Fill in size field
                03    10        076B  1861          BSBB    CATCH_RST_STR           ; Catch a reset or start packet
                24    BA        076D  1862 20$:     POPR    #^M<R2,R5>              ; Restore regs
                      05        076F  1863          RSB
                                0770  1864
                                0770  1865
```

```
                                0770  1867          .SBTTL  CATCH_RST_STR,  Catch reset or start packet
                                0770  1868 ;+
                                0770  1869 ;
                                0770  1870 ;  CATCH_RST_STR
                                0770  1871 ;
                                0770  1872 ;  This piece of code is called as a subroutine because the CATCH_RST
                                0770  1873 ;  macro suspends the current thread. This allows CXDRIVER to send a CTP
                                0770  1874 ;  generate reset or start request to the UUT and still be notified when
                                0770  1875 ;  the reset or start packet arrives.
                                0770  1876 ;
                                0770  1877 ;  When the suspended thread is resumed at the instruction after the
                                0770  1878 ;  CATCH_RST macro, R2 contains the address of the reset or start packet.
                                0770  1879 ;  The packet's contents can then be copied into the response buffer.
                                0770  1880 ;  The discgarding of the reset or start packet is done by padriver.
                                0770  1881 ;
                                0770  1882 ;  It the CATCH_RST macro returns error status, it means another sysap was
                                0770  1883 ;  waiting for a start or reset packet.
                                0770  1884 ;
                                0770  1885 ;  The PDT address of the local port is saved before issuing the CATCH_RST
                                0770  1886 ;  macro so that if a reset or start packet never arrives, the suspended
                                0770  1887 ;  thread can be cleaned up, and the CDRP deallocated bu CANCEL.
                                0770  1888 ;
                                0770  1889 ;  INPUTS:
                                0770  1890 ;
                                0770  1891 ;       R4                      - PDT address
                                0770  1892 ;       R5                      - CDRP address
                                0770  1893 ;
                                0770  1894 ;  OUTPUTS:
                                0770  1895 ;
                                0770  1896 ;       R0                      - Status
                                0770  1897 ;       R1,R2,R5                - Destroyed
                                0770  1898 ;       R3,R4, other registers  - Perserved
                                0770  1899 ;       CDRP                    - Deallocated
                                0770  1900 ;
                                0770  1901 ;-
                                0770  1902
                                0770  1903 CATCH_RST_STR:
                                0770  1904
      0000'CF    54    D0       0770  1905          MOVL    R4,CX$L_RST_STR_PDT     ; Save PDT address for local port
      0000'CF    55    D0       0775  1906          MOVL    R5,CX$L_RST_STR_CDRP    ; Save CDRP address             [5-002]
                                077A  1907          CATCH_RST                       ; Wait for reset or start packet
           0000'CF    D4       0780  1908          CLRL    CX$L_RST_STR_PDT        ; Clear saved PDT address
             0C 50    E8       0784  1909          BLBS    R0,5$                   ; Success, branch
          50    55    D0       0787  1910          MOVL    R5,R0                   ; Get CDRP address
    00000000'GF    16       078A  1911          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                50    D4       0790  1912          CLRL    R0                      ; Return error status
                      05       0792  1913          RSB
                               0793  1914
                54    DD       0793  1915 5$:     PUSHL   R4                      ; Save PDT address
           0000'CF    C1       0795  1916          ADDL3   CX$L_RSPBUFBGN,-        ; Point to area in response buffer
50    00000080 8F               0799  1917                  #CDC$_PACKETDATA,R0     ; for saving the reset or start packet
             80    82    7D       079F  1918          MOVQ    (R2)+,(R0)+             ; Save the reset or start packet
             80    82    7D       07A2  1919          MOVQ    (R2)+,(R0)+             ; in the response buffer
             80    82    7D       07A5  1920          MOVQ    (R2)+,(R0)+
             80    82    7D       07A8  1921          MOVQ    (R2)+,(R0)+
                54 8ED0    07AB  1922          POPL    R4                      ; Restore PDT address
             50    55    D0       07AE  1923 10$:     MOVL    R5,R0                   ; Get CDRP address
```

```
                00000000'GF   16  07B1  1924          JSB     G^CUM$DRVDEALMEM        ; Deallocate the CDRP
                              05  07B7  1925          RSB
                                  07B8  1926
                                  07B8  1927
                                  07B8  1928          .END
```

```
$$$CURSIZ                       = 000001C4
$$$NEWSIZ                       = 000001D0
ALLOC_CDB                         000006C2 R      01
ALLOC_CDB_ERR                     0000015D R      01
ALLOC_CDRP_CTP                    00000643 R      01
ALLOC_CDRP_ERR                    0000015D R      01
ALLOC_CDRP_XFR                    00000672 R      01
ALLOC_DG_ERR                      00000163 RG     01
ALLOC_MSG_ERR                     00000163 R      01
BIT...                          = 0000000D
BUFFER_NOT_MAPPED                 0000012C R      01
CATCH_RST_STR                     00000770 R      01
CDB$B_CDRPCNT                     00000028
CDB$B_RSTNADR                     0000000E
CDB$B_TYPE                        0000000A
CDB$K_LEN                         00000032
CDB$L_CDBSLOT                     00000024
CDB$L_CDRPQBL                     0000002E
CDB$L_CDRPQFL                     0000002A
CDB$L_CDT                         0000001C
CDB$L_PDT                         00000020
CDB$L_QBL                         00000004
CDB$L_QFL                         00000000
CDB$T_LPRTNAM                     00000016
CDB$W_SIZE                        00000008
CDB$W_STATUS                      0000000C
CDB_M_CONN                      = 00000001
CDB_M_DISC                      = 00000002
CDB_V_CNCL                      = 00000002
CDC$ARMCINTPATH                   0000001B
CDC$B_CFGCBLSTS                   00000021
CDC$B_CFGP0LBSTS                  0000002A
CDC$B_CFGP0STS                    0000001F
CDC$B_CFGP1LBSTS                  00000028
CDC$B_CFGP1STS                    00000020
CDC$B_CMDCOUNT                    00000000
CDC$B_CNTFLG                      0000001F
CDC$B_COUNT                       00000015
CDC$B_CTPOPCODE                   00000014
CDC$B_DELAY                       00000019
CDC$B_DONECNT                     00000001
CDC$B_DSTPORT                     00000001
CDC$B_EXTEND                      0000001C
CDC$B_FOTCNT                      00000002
CDC$B_GARBAGE_FLAG                00000014
CDC$B_NOACTFLAG                   0000001E
CDC$B_NODEADR                     00000014
CDC$B_OPCODE                      00000000
CDC$B_OTHERNODE                   0000001E
CDC$B_PATH                        0000001C
CDC$B_PKTMULT                     0000001F
CDC$B_PKTSIZ                      0000001D
CDC$B_POLLERFLAG                  0000001E
CDC$B_PROTREV                     00000036
CDC$B_PROTVER                     00000035
CDC$B_PROTYPE                     00000034
CDC$B_QNUMBER                     0000001F
CDC$B_RST_PORT                    00000015
CDC$B_SIOCNT                      00000003
CDC$B_TYPE                        00000014
CDC$CARRIER                       00000018
CDC$CONFIGDATA                    00000007
CDC$CONNECT                       00000008
CDC$C_CMDSPEC                     00000014
CDC$DISCONNECT                    00000009
CDC$GARBAGE                       0000000B
CDC$INITCINT                      0000001A
CDC$K_CMDHDRSIZ                   0000000C
CDC$K_CMDSPEC                     00000014
CDC$K_DATA                        00000014
CDC$LRGPACKET                     0000C019
CDC$L_BUFADR                      00000024
CDC$L_BUFLENGTH                   00000020
CDC$L_BUFLNAME                    00000024
CDC$L_BUFLOFSET                   00000028
CDC$L_BUFRNAME                    0000002C
CDC$L_BUFROFSET                   00000030
CDC$L_CDB_AD                      0000000C
CDC$L_CDRP                        0000000C
CDC$L_CFGHWTYPE                   00000022
CDC$L_CFGPRTMSK                   00000026
CDC$L_CNTRDISCDG                  00000038
CDC$L_CNTRPOACK                   00000020
CDC$L_CNTRPONAK                   00000024
CDC$L_CNTRPONORSP                 00000028
CDC$L_CNTRP1ACK                   0000002C
CDC$L_CNTRP1NAK                   00000030
CDC$L_CNTRP1NORSP                 00000034
CDC$L_CTPREF                      00000015
CDC$L_DISCONREAS                  00000019
CDC$L_LPORT                       00000010
CDC$L_PORTHDR                     00000004
CDC$L_R0STATUS                    0000000C
CDC$L_R1STATUS                    00000010
CDC$L_RESERV04                    00000004
CDC$L_RESERV08                    00000008
CDC$L_RESERV12                    0000000C
CDC$L_SEQNUM                      00000001
CDC$L_STARTADR                    00000020
CDC$L_SVAPTE                      00000018
CDC$MAINTSTATE                    0000000F
CDC$MAPBUF                        00000005
CDC$MAPMBUF                       00000010
CDC$NAK                           00C00016
CDC$NOACK                         00000017
CDC$NOACT                         0000000E
CDC$NORSP                         00000015
CDC$OVERSIZEPKT                   0000001D
CDC$POLLER                        0000000E
CDC$RDCINTPATH                    0000001C
CDC$READCNT                       0000000A
CDC$REQDATA                       00000004
CDC$REQMDATA                      00000013
CDC$RESET                         0000000C
```

CDC$SELADR		00000014			CHECK_CONN		********	X 01
CDC$SENDDATA		00000003			CHECK_CONN_ERR		00000120	R 01
CDC$SENDDG		00000002			CHECK_RST_STR		00000744	R 01
CDC$SENDMDATA		00000012			CHECK_RST_STR_ERR		000001C3	R 01
CDC$SENDMSG		00000001			CHECK_VC		000005A0	RG 01
CDC$START		0000000D			CHECK_VC_ERR		00000114	RG 01
CDC$T_CONDAT		00000040			CIBHAN$L_BNAME	=	00000004	
CDC$T_ENDMSG		00000072			CIBHAN$L_BOFF	=	00000000	
CDC$T_GARBAGE		00000018			CIBHAN$L_RCONID	=	00000008	
CDC$T_LOCPROCNAM		00000014			CLEAN_CDB		********	X 01
CDC$T_PACKETDATA		00000080			CLEAN_CDRP		********	X 01
CDC$T_REMPROCNAM		00000024			CLEAR_BUFFER		********	X 01
CDC$T_SYSTEMID		00000034			COM$DRVDEALMEM		********	X 01
CDC$UNMAPBUF		00000006			CONNECT_ERR		00000132	R 01
CDC$UNMAPMBUF		00000011			CONTROLLER_NAME		********	X 01
CDC$W_CMDOFSET		00000015			CTP$ACTCOUNT		00000006	
CDC$W_CMDSEQNUM		00000017			CTP$BUFLENGTH		0000000C	
CDC$W_DGCREDIT		0000003E			CTP$BUFLNAME		00000010	
CDC$W_FLAG		00000004			CTP$BUFLOFSET		00000014	
CDC$W_MAXDG		00000014			CTP$BUFMAPREQ		00000001	
CDC$W_MAXMSG		00000018			CTP$BUFMAPRSP		00000041	
CDC$W_MINSENDCR		0000003C			CTP$BUFRNAME		00000018	
CDC$W_MSGCREDIT		0000003A			CTP$BUFROFSET		0000001C	
CDC$W_POLLINT		0000001C			CTP$BUFTYPE		00000005	
CDC$W_POLLNUM		00000020			CTP$BUFUNMREQ		00000002	
CDC$W_REPEAT		0000001A			CTP$BUFUNMRSP		00000042	
CDC$W_RESERV10		0000000A			CTP$CDATPREV		00000002	
CDC$W_RSPBUFOFSET		00000004			CTP$CDATPTYPE		00000000	
CDC$W_RSPCNT		00000006			CTP$CDATPVERS		00000001	
CDC$W_RSPOFSET		00000002			CTP$CFGPOSTS		0000000B	
CDC$W_TXTLEN		00000008			CTP$CFGP1STS		0000000C	
CDC$_GARBAGE	=	00000001	G		CTP$CNTFLG		0000000A	
CDC$_NOGARBAGE	=	00000000	G		CTP$CNTRDISCDG		00000024	
CDRP$B_CD_TYPE	=	0000000A			CTP$CNTRP0ACK		0000000C	
CDRP$B_FLAGS		00000040			CTP$CNTRP0NAK		00000010	
CDRP$C_BT_LEN	=	00000040			CTP$CNTRP0NRSP		00000014	
CDRP$K_CTPLEN		00000050			CTP$CNTRP1ACK		00000018	
CDRP$K_CX_LEN		00000068			CTP$CNTRP1NAK		0000001C	
CDRP$K_XFRLEN		00000068			CTP$CNTRP1NRSP		00000020	
CDRP$L_BCNT	=	FFFFFFD2			CTP$CONFIGREQ		00000009	
CDRP$L_CDT	=	00000024			CTP$CONFIGRSP		00000049	
CDRP$L_LBOFF	=	00000030			CTP$CONNECTREQ		0000000B	
CDRP$L_LBUFH_AD	=	0000002C			CTP$CONNECTRSP		0000004B	
CDRP$L_PDT		0000004C			CTP$COUNTSREQ		0000000A	
CDRP$L_PENDQBL		00000048			CTP$COUNTSRSP		0000004A	
CDRP$L_PENDQFL		00000044			CTP$DELAY		00000005	
CDRP$L_RBOFF	=	00000038			CTP$EXTEND		00C00008	
CDRP$L_RBUFH_AD	=	00000034			CTP$FINISHREQ		0000000C	
CDRP$L_SVAPTE	=	FFFFFFCC			CTP$FINISHRSP		0000004C	
CDRP$L_XCT_LEN	=	0000003C			CTP$FMASK		00000006	
CDRP$T_CX_LBUFHNDL		00000050			CTP$FUNCTREQ		00000000	
CDRP$T_CX_RBUFHNDL		0000005C			CTP$FUNCTRSP		00000040	
CDRP$W_BOFF	=	FFFFFFD0			CTP$GENCONST		00000009	
CDRP$W_CDRPSIZE	=	00000008			CTP$GENDATA		0000000E	
CDT$L_LCONID	=	00000018			CTP$GENDGRREQ		00000005	
CDT$L_RCONID	=	00000014			CTP$GENDGRRSP		00000045	
CHECK_ACTV_CONN		********	X	01	CTP$GENFUNCT		00000008	

```
CTP$GENLENGTH                   0000000C
CTP$GENMSGREQ                   00000004
CTP$GENMSGRSP                   00000044
CTP$GENRSTREQ                   00000006
CTP$GENRSTRSP                   00000046
CTP$GENSTRREQ                   00000007
CTP$GENSTRRSP                   00000047
CTP$IMAGEDATA                   0000000E
CTP$MAXCMDOPC                   00000011
CTP$MBUFMAPREQ                  0000000D
CTP$MBUFMAPRSP                  0000004D
CTP$MBUFUNMREQ                  0000000E
CTP$MBUFUNMRSP                  0000004E
CTP$MOVBUFREQ                   00000003
CTP$MOVBUFRSP                   00000043
CTP$MOVETYPE                    00000008
CTP$MOVMBUFREQ                  0000000F
CTP$MOVMBUFRSP                  0000004F
CTP$MSTATEREQ                   00000010
CTP$MSTATERSP                   00000050
CTP$NOACTFLAG                   0000000A
CTP$NOACTREQ                    00000008
CTP$NOACTRSP                    00000048
CTP$OPCODE                      00000000
CTP$OPEXPAND                    000000FF
CTP$OTHERNODE                   00000009
CTP$PKTMULT                     0000000B
CTP$PKTSIZ                      0000000A
CTP$REFERENCE                   00000001
CTP$REPCOUNT                    00000006
CTP$RESERV10                    0000000A
CTP$RESERV11                    0000000B
CTP$RESERV12                    0000000C
CTP$RESERV20                    00000014
CTP$RESERV5                     00000005
CTP$RESERV6                     00000006
CTP$RESERV7                     00000007
CTP$RESERV9                     00000009
CTP$REVISION                  = 00000000  G
CTP$STARTADR                    0000000C
CTP$STATUS                      00000005
CTP$VERSION                   = 00000003  G
CX$B_CMDERRCNT                  ********  X   01
CX$B_GARBAGE_FLAG               ********  X   01
CX$CONNECT_ERR                  ********  X   01
CX$DG_IN                        ********  X   01
CX$L_CMDBUFBGN                  ********  X   01
CX$L_CUR_CMD                    ********  X   01
CX$L_LOCPRTLST                  ********  X   01
CX$L_RSPBUFBGN                  ********  X   01
CX$L_RST_STR_CDRP               ********  X   01
CX$L_RST_STR_PDT                ********  X   01
CX$L_STATE                      ********  X   01
CX$MSG_IN                       ********  X   01
CX$Q_MAPLST                     ********  X   01
CX$T_GARBAGE_BUF                ********  X   01
CX$X_IRPSAV                     ********  X   01
CX$_CDRPNFND                  = 00000001  G
CX_STARTIO                      00000000 RG   01
DEALLOC_CDRP                    ********  X   01
DYN$C_CDB                     = 00000033
DYN$C_CDRP                    = 00000039
DYN$C_CIMSG                   = 0000003C
DYN$C_IRP                     = 0000000A
EXE$ALONONPAGED                 ********  X   01
FIND_CDRP_XFR                   00000602 R    01
FLAG_M_CMD_DONE               = 00008000
FLAG_M_CMD_START              = 00004000
FLAG_M_CONN_LOST              = 00000800
FLAG_M_SCS_ERR                = 00001000
FLAG_V_NO_RESPONSE            = 00000000
GET_PDT_ADDR                    000005A0 R    01
IOC$REQCOM                      ********  X   01
IRP$B_RMOD                    = 0000000B
IRP$B_TYPE                    = 0000000A
IRP$C_CDRP                    = 00000060
IRP$W_SIZE                    = 00000008
MARK_CMD_DONE                   0000058C RG   01
MNT$CATCH_RST                 = 00000007  G
MNT$CATCH_STR                 = 00000008  G
MNT$MAINT_READ                = 00000005  G
MNT$MAINT_STATE               = 00000002  G
MNT$MAINT_WRITE               = 00000006  G
MNT$NORMAL_STATE              = 00000004  G
MNT$POLLER_OFF                = 00000000  G
MNT$POLLER_ON                 = 00000001  G
MNT$UNINIT_STATE              = 00000003  G
MNT_XFR_ERR                     00000196 R    01
MNT_XFR_SETUP                   000005EB R    01
NODE_UNKNOWN                    00000114 R    01
PA_CNF                          00000000  G
PA_CNF_M_ADPTYP               = 000000FF  G
PA_CNF_M_CIBPE                = 00008000  G
PA_CNF_M_CRD                  = 00010000  G
PA_CNF_M_CTO                  = 00002000  G
PA_CNF_M_CXTER                = 00040000  G
PA_CNF_M_CXTMO                = 00100000  G
PA_CNF_M_MAINT                = 00004000  G
PA_CNF_M_MXTFLT               = 08000000  G
PA_CNF_M_NOCI                 = 00001000  G
PA_CNF_M_PARFLT               = 80000000  G
PA_CNF_M_PDN                  = 00800000  G
PA_CNF_M_PFD                  = 00000100  G
PA_CNF_M_PUP                  = 00400000  C
PA_CNF_M_RDS                  = 00020000  G
PA_CNF_M_RDTO                 = 00080000  G
PA_CNF_M_TDEAD                = 00000200  G
PA_CNF_M_TFAIL                = 00000400  G
PA_CNF_M_URDFLT               = 20000000  G
PA_CNF_M_WSQFLT               = 40000000  G
PA_CNF_M_XMTFLT               = 04000000  G
PA_CNF_S_ADPTYP               = 00000008  G
PA_CNF_V_ADPTYP               = 00000000  G
PA_CNF_V_CIBPE                = 0000000F  G
```

```
PA_CNF_V_CRD                =  00000010  G
PA_CNF_V_CTO                =  0000000D  G
PA_CNF_V_CXTER              =  00000012  G
PA_CNF_V_CXTMO              =  00000014  G
PA_CNF_V_MAINT              =  0000000E  G
PA_CNF_V_MXTFLT             =  0000001B  G
PA_CNF_V_NOCI               =  0000000C  G
PA_CNF_V_PARFLT             =  0000001F  G
PA_CNF_V_PDN                =  00000017  G
PA_CNF_V_PFD                =  00000008  G
PA_CNF_V_PUP                =  00000016  G
PA_CNF_V_RDS                =  00000011  G
PA_CNF_V_RDTO               =  0C000013  G
PA_CNF_V_TDEAD              =  00000009  G
PA_CNF_V_TFAIL              =  0000000A  G
PA_CNF_V_URDFLT             =  0000001D  G
PA_CNF_V_WSQFLT             =  0000001E  G
PA_CNF_V_XMTFLT             =  0000001A  G
PA_CQ0                         00000908  G
PA_CQ0_M_CQC                =  00000001  G
PA_CQ0_V_CQC                =  00000000  G
PA_CQ1                         0000090C  G
PA_CQ1_M_CQC                =  00000001  G
PA_CQ1_V_CQC                =  00000000  G
PA_CQ2                         00000910  G
PA_CQ2_M_CQC                =  00000001  G
PA_CQ2_V_CQC                =  00000000  G
PA_CQ3                         00000914  G
PA_CQ3_M_CQC                =  00000001  G
PA_CQ3_V_CQC                =  00000000  G
PA_C_DCACHESZ               =  00000003  G
PA_C_LSINDX                 =  00000800  G
PA_C_LSLENGTH               =  00000800  G
PA_C_MLACHESZ               =  00000003  G
PA_C_UCODEST                =  00000400  G
PA_C_WCSSIZ                 =  00000C00  G
PA_DFQ                         00000928  G
PA_DFQ_M_DFQC               =  00000001  G
PA_DFQ_V_DFQC               =  00000000  G
PA_MADR                        00000014  G
PA_MDATR                       00000018  G
PA_MFQ                         0000092C  G
PA_MFQ_M_MFQC               =  00000001  G
PA_MFQ_V_MFQC               =  00000000  G
PA_MTC                         00000930  G
PA_MTC_M_MTC                =  00000001  G
PA_MTC_V_MTC                =  00000000  G
PA_MTEC                        00000934  G
PA_PDC                         00000920  G
PA_PDC_M_PDC                =  00000001  G
PA_PDC_V_PDC                =  00000000  G
PA_PEC                         0000091C  G
PA_PEC_M_PEC                =  00000001  G
PA_PEC_V_PEC                =  00000000  G
PA_PESR                        0000093C  G
PA_PFAR                        00000938  G
PA_PIC                         00000924  G
PA_PIC_M_PIC                =  00000001  G
PA_PIC_V_PIC                =  00000000  G
PA_PMC                         00000004  G
PA_PMC_M_CSPE               =  00004000  G
PA_PMC_M_IPE                =  00000400  G
PA_PMC_M_LSPE               =  00002000  G
PA_PMC_M_MIE                =  00000004  G
PA_PMC_M_MIF                =  00000008  G
PA_PMC_M_MIN                =  00000001  G
PA_PMC_M_MTD                =  00000002  G
PA_PMC_M_OPE                =  00000200  G
PA_PMC_M_PE                 =  00008000  G
PA_PMC_M_PSA                =  00000040  G
PA_PMC_M_RBPE               =  00001000  G
PA_PMC_M_RSVD               =  00000020  G
PA_PMC_M_UNIN               =  00000080  G
PA_PMC_M_WP                 =  00000010  G
PA_PMC_M_XBPE               =  00000100  G
PA_PMC_M_XMPE               =  00000800  G
PA_PMC_V_CSPE               =  0000000E  G
PA_PMC_V_IPE                =  0000000A  G
PA_PMC_V_LSPE               =  0000000D  G
PA_PMC_V_MIE                =  00000002  G
PA_PMC_V_MIF                =  00000003  G
PA_PMC_V_MIN                =  00000000  G
PA_PMC_V_MTD                =  00000001  G
PA_PMC_V_OPE                =  00000009  G
PA_PMC_V_PE                 =  0000000F  G
PA_PMC_V_PSA                =  00000006  G
PA_PMC_V_RBPE               =  0000000C  G
PA_PMC_V_RSVD               =  00000005  G
PA_PMC_V_UNIN               =  00000007  G
PA_PMC_V_WP                 =  00000004  G
PA_PMC_V_XBPE               =  00000008  G
PA_PMC_V_XMPE               =  0000000B  G
PA_PPR                         00000940  G
PA_PPR_M_BUFLEN             =  0FFF0000  G
PA_PPR_M_MNODE              =  10000000  G
PA_PPR_M_PN                 =  000000FF  G
PA_PPR_S_BUFLEN             =  0000000C  G
PA_PPR_S_PN                 =  00000008  G
PA_PPR_V_BUFLEN             =  00000010  G
PA_PPR_V_MNODE              =  0000001C  G
PA_PPR_V_PN                 =  00000000  G
PA_PQBBR                       00000904  G
PA_PS                          00000900  G
PA_PSR                         00000918  G
PA_PSR_M_PSC                =  00000001  G
PA_PSR_V_PSC                =  00000000  G
PA_PS_M_DSE                 =  00000010  G
PA_PS_M_MFQE                =  00000002  G
PA_PS_M_MSE                 =  00000020  G
PA_PS_M_MTE                 =  80000000  G
PA_PS_M_PDC                 =  00000004  G
PA_PS_M_PIC                 =  00000008  G
PA_PS_M_RQA                 =  00000001  G
PA_PS_M_SE                  =  00000040  G
```

```
PA_PS_V_DSE                =  00000004   G
PA_PS_V_MFQE               =  00000001   G
PA_PS_V_MSE                =  00000005   G
PA_PS_V_MTE                =  0000001F   G
PA_PS_V_PDC                =  00000002   G
PA_PS_V_PIC                =  00000003   G
PA_PS_V_RQA                =  00000000   G
PA_PS_V_SE                 =  00000006   G
PB$B_CBL_STS               =  00000028   G
PB$B_P0_STS                =  00000029   G
PB$B_P1_STS                =  0000002A   G
PB$B_PROTOCOL              =  00000048   G
PB$B_RSTATE                =  00000021   G
PB$B_RSTATION              =  0000000C   G
PB$B_RST_PORT              =  00000020   G
PB$B_SUBTYP                =  0000000B   G
PB$B_TYPE                  =  0000000A   G
PB$C_CI750                 =  00000002   G
PB$C_CI780                 =  00000002   G
PB$C_CINT                  =  00000007   G
PB$C_CLOSED                =  00000000   G
PB$C_DISAB                 =  00000001   G
PB$C_ENAB                  =  00000002   G
PB$C_HSC                   =  00000004   G
PB$C_KL10                  =  00000006   G
PB$C_LENGTH                =  00000054   G
PB$C_NI                    =  00000008   G
PB$C_OPEN                  =  00000003   G
PB$C_PS                    =  00000009   G
PB$C_PWR_FAIL              =  00004000   G
PB$C_ST_REC                =  00000002   G
PB$C_ST_SENT               =  00000001   G
PB$C_UNINIT                =  00000000   G
PB$C_VC_FAIL               =  00008000   G
PB$K_LENGTH                =  00000054   G
PB$L_BLINK                 =  00000004   G
PB$L_CDTLST                =  00000034   G
PB$L_DUETIME               =  0000003C   G
PB$L_FLINK                 =  00000000   G
PB$L_PDT                   =  0000002C   G
PB$L_RPORT_FCN             =  0000001C   G
PB$L_RPORT_REV             =  00000018   G
PB$L_RPORT_TYP             =  00000014   G
PB$L_SBLINK                =  00000030   G
PB$L_SCSMSG                =  00000040   G
PB$L_WAITQBL               =  0000003C   G
PB$L_WAITQFL               =  00000038   G
PB$M_CUR_CBL               =  00000001   G
PB$M_CUR_PS                =  00000001   G
PB$M_DUALPATH              =  80000000   G
PB$M_MAINT                 =  00000001   G
PB$M_TIM                   =  00000001   G
PB$S_LPORT_NAME            =  00000004   G
PB$S_PBDEF                 =  00000054   G
PB$S_PORT_TYP              =  0000001F   G
PB$S_RSTATION              =  00000006   G
PB$S_STATE                 =  00000002   G
PB$T_LPORT_NAME            =  00000024   G
PB$V_CUR_CBL               =  00000000   G
PB$V_CUR_PS                =  00000000   G
PB$V_DUALPATH              =  0000001F   G
PB$V_MAINT                 =  00000000   G
PB$V_PORT_TYP              =  00000000   G
PB$V_STATE                 =  00000001   G
PB$V_TIM                   =  00000000   G
PB$W_RETRY                 =  00000022   G
PB$W_SIZE                  =  00000008   G
PB$W_STATE                 =  00000012   G
PB$W_STS                   =  00000044   G
PB$W_VCFAIL_RSN            =  00000046   G
PDT$B_DQIMAP                  00000154   G
PDT$B_HSHUT_DG                000001B0   G
PDT$B_MAX_PORT                0000017C   G
PDT$B_NXT_PORT                0000017E   G
PDT$B_PO_LBSTS                00000180   G
PDT$B_P1_LBSTS                00000181   G
PDT$B_PDT_TYPE             =  00000007   G
PDT$B_PLOGMAP                 00000134   G
PDT$B_PORTMAP                 00000114   G
PDT$B_PORT_NUM                0000017D   G
PDT$B_REQIDPS                 0000017F   G
PDT$B_SUBTYP               =  0000000B   G
PDT$B_TYPE                 =  0000000A   G
PDT$C_HSHUT_SIZ            =  00000014   G
PDT$C_LENGTH               =  000000E4   G
PDT$C_PA                   =  00000001   G
PDT$C_PALENGTH             =  00000360
PDT$C_PAREGBASE               000000E4   G
PDT$C_PAREGEND                00000110   G
PDT$C_PE                   =  00000003   G
PDT$C_PQB                  =  000001E0
PDT$C_PS                   =  00000004   G
PDT$C_PU                   =  00000002   G
PDT$C_SCSBASE              =  0000000C   G
PDT$C_SCSEND               =  00000084   G
PDT$K_LENGTH               =  000000E4   G
PDT$K_SCSBASE              =  0000000C   G
PDT$K_SCSEND               =  00000084   G
PDT$L_ACCEPT               =  0000000C   G
PDT$L_ADP                  =  000000E0   G
PDT$L_ALLOCDG              =  00000010   G
PDT$L_ALLOCMSG             =  00000014   G
PDT$L_CNF                     000000E4   G
PDT$L_CNTCDRP              =  00C000D4   G
PDT$L_CONNECT              =  00000018   G
PDT$L_CQ0                     000C00F0   G
PDT$L_CQ1                     000000F4   G
PDT$L_DCONNECT             =  00000028   G
PDT$L_DEALLOCDG            =  0000001C   G
PDT$L_DEALLOMSG            =  00000020   G
PDT$L_DEALRGMSG            =  00000024   G
PDT$L_DFQ                     000000FC   G
PDT$L_DFQHDR                  00000208   G
PDT$L_DGHDRSZ                 00000190   G
```

```
PDT$L_DGNETHD                   00000194  G
PDT$L_DGOVRHD                 = 000000B8  G
PDT$L_DQELOGOUT                 000002E0  G
PDT$L_FLINK                   = 00000000  G
PDT$L_GPTBASE                   0000022C  G
PDT$L_GPTLEN                    00000230  G
PDT$L_LBDG                      00000184  G
PDT$L_MAINTFCN                = 00000078  G
PDT$L_MAP                     = 0000002C  G
PDT$L_MAPBYPASS               = 00000030  G
PDT$L_MAPIRP                  = 00000034  G
PDT$L_MAPIRPBYP               = 00000038  G
PDT$L_MAXBCNT                 = 000000BC  G
PDT$L_MFQ                       00000100  G
PDT$L_MFQHDR                    0000020C  G
PDT$L_MQELOGOUT                 00000320  G
PDT$L_MRESET                  = 00000070  G
PDT$L_MSGHDRSZ                = 000000B4  G
PDT$L_MSTART                  = 00000074  G
PDT$L_MTC                       00000104  G
PDT$L_PFAR                      00000108  G
PDT$L_PMC                       000000E8  G
PDT$L_POLLERDUE                 0000018C  G
PDT$L_POLLSWEEP               = 000000D8  G
PDT$L_POOLDUE                   00000188  G
PDT$L_PPR                       0000010C  G
PDT$L_PS                        000000EC  G
PDT$L_PSR                       000000F8  G
PDT$L_QUEUEDG                 = 0000003C  G
PDT$L_QUEUEMDGS               = 00000040  G
PDT$L_RCHMSGBUF               = 00000044  G
PDT$L_RCLMSGBUF               = 00000048  G
PDT$L_READCOUNT               = 00000068  G
PDT$L_REJECT                  = 0000004C  G
PDT$L_REQDATA                 = 00000050  G
PDT$L_RLSCOUNT                = 0000006C  G
PDT$L_SENDDATA                = 00000054  G
PDT$L_SENDDG                  = 00000058  G
PDT$L_SENDMSG                 = 0000005C  G
PDT$L_SENDRGDG                = 0000007C  G
PDT$L_SNDCNTMSG               = 00000060  G
PDT$L_SPTBASE                   00000224  G
PDT$L_SPTLEN                    00000228  G
PDT$L_STOP_VCS                = 00000080  G
PDT$L_UCBO                    = 000000DC  G
PDT$L_UNMAP                   = 00000064  G
PDT$L_VBDT                      0000021C  G
PDT$L_VPQB                      00000218  G
PDT$L_WAITQBL                 = 000000B0  G
PDT$L_WAITQFL                 = 000000AC  G
PDT$M_CNTBSY                  = 00000001  G
PDT$M_CNTRLS                  = 00000002  G
PDT$M_CUR_LBS                 = 00000001  G
PDT$M_LBDG                    = 00000004  G
PDT$M_PRV_LBS                 = 00000002  G
PDT$M_PUP                     = 00000002  G
PDT$M_PWF_CLNUP               = 00000001  G
PDT$M_SNGLHOST                = 00000001  G
PDT$M_X_LBS                   = 00000004  G
PDT$Q_COMQ2                     000001F0  G
PDT$Q_COMQ3                     000001F8  G
PDT$Q_COMQBASE                  000001E0  G
PDT$Q_COMQH                     000001E8  G
PDT$Q_COMQL                     000001E0  G
PDT$Q_DFREEQ                    000001D0  G
PDT$Q_FORMPB                    00000174  G
PDT$Q_MFREEQ                    000001D8  G
PDT$Q_RSPQ                      00000200  G
PDT$Q_TEMP_RSPQ                 00000198  G
PDT$S_CNTOWNER                = 00000010  G
PDT$S_PDTDEF                  = 000000E4  G
PDT$T_CNTOWNER                = 000000C4  G
PDT$V_CNTBSY                  = 00000000  G
PDT$V_CNTRLS                  = 00000001  G
PDT$V_CUR_LBS                 = 00000000  G
PDT$V_LBDG                    = 00000002  G
PDT$V_PRV_LBS                 = 00000001  G
PDT$V_PUP                     = 00000001  G
PDT$V_PWF_CLNUP               = 00000000  G
PDT$V_SNGLHOST                = 00000000  G
PDT$V_X_LBS                   = 00000002  G
PDT$W_BDTLEN                    00000220  G
PDT$W_DQELEN                    00000210  G
PDT$W_FLAGS                   = 000000C0  G
PDT$W_LPORT_STS                 00000110  G
PDT$W_MQELEN                    00000214  G
PDT$W_PBCOUNT                   00000112  G
PDT$W_PORTCHAR                = 00000004  G
PDT$W_SIZE                    = 00000008  G
PPD$B_DEF_ST                    0000001C  G
PPD$B_FLAGS                     0000000F  G
PPD$B_HWVERS                    00000034  G
PPD$B_LBDATA                    00000012  G
PPD$B_LCB_0                     00000012  G
PPD$B_LCB_LPORT                 00000010  G
PPD$B_LCB_NPORT                 0000000F  G
PPD$B_LCB_OPC                   00000011  G
PPD$B_LCB_PORT                  0000000E  G
PPD$B_OPC                       0000000E  G
PPD$B_PORT                      0000000C  G
PPD$B_PROTOCOL                  0000001A  G
PPD$B_RSTATE                    00000025  G
PPD$B_RST_PORT                  00000024  G
PPD$B_STATUS                    00C0000D  G
PPD$B_SWFLAG                    0000000B  G
PPD$B_SYSTEMID                  00000014  G
PPD$B_TYPE                      0000000A  G
PPD$C_ACK                     = 00000002  G
PPD$C_ACK_LEN                 = 00000004  G
PPD$C_CACHECLR                = 00008000  G
PPD$C_CACHE_LEN               = 00000002  G
PPD$C_CNFREC                  = 00000023  G
PPD$C_DATREC                  = 00000031  G
PPD$C_DGREC                   = 00000021  G
```

```
PPD$C_DISAB                     =  00000001   G
PPD$C_ELOG                      =  00000005   G
PPD$C_ENAB                      =  00000002   G
PPD$C_HOSTSHUT                  =  00000006   G
PPD$C_HSHUT_LEN                 =  00000002   G
PPD$C_IDREC                     =  0000002B   G
PPD$C_INVTC                     =  00000018   G
PPD$C_LBDAT_LEN                 =  00000030   G
PPD$C_LBREC                     =  0000002D   G
PPD$C_LC_LENGTH                    00000046   G
PPD$C_LCB_DATA                     00000013   G
PPD$C_LENGTH                       00000012   G
PPD$C_MCNFREC                   =  00000029   G
PPD$C_MDATREC                   =  00000033   G
PPD$C_MIN_DGSIZ                    00000050   G
PPD$C_MSGREC                    =  00000022   G
PPD$C_OSEQ                      =  00000005   G
PPD$C_PRT_BASE                  =  00000000   G
PPD$C_PRT_ELOG                  =  00000001   G
PPD$C_PSAUTO                    =  00000000   G
PPD$C_PSPO                      =  00000001   G
PPD$C_PSP1                      =  00000002   G
PPD$C_PTHACK                    =  00000000   G
PPD$C_PTHARB                    =  00000003   G
PPD$C_PTHNAK                    =  00000001   G
PPD$C_PTHNO_RSP                 =  00000002   G
PPD$C_RDCNT                     =  0000001A   G
PPD$C_REQDAT                    =  00000008   G
PPD$C_REQDAT0                   =  00000008   G
PPD$C_REQDAT1                   =  00000009   G
PPD$C_REQDAT2                   =  0000000A   G
PPD$C_REQID                     =  00000005   G
PPD$C_REQMDAT                   =  0000000E   G
PPD$C_RETCNF                    =  00000003   G
PPD$C_RETDAT                    =  00000011   G
PPD$C_SCS_DG                    =  00000003   G
PPD$C_SCS_MSG                   =  00000004   G
PPD$C_SETCKT                    =  00000019   G
PPD$C_SNDDAT                    =  00000010   G
PPD$C_SNDDG                     =  00000001   G
PPD$C_SNDLB                     =  0000000D   G
PPD$C_SNDMDAT                   =  00000012   G
PPD$C_SNDMSG                    =  00000002   G
PPD$C_SNDRST                    =  00000006   G
PPD$C_SNDSTRT                   =  00000007   G
PPD$C_STABO                     =  00000004   G
PPD$C_STACK                     =  00000001   G
PPD$C_STACK_LEN                 =  0000003E   G
PPD$C_START                     =  00000000   G
PPD$C_START_LEN                 =  0000003E   G
PPD$C_STINVDP                   =  00000002   G
PPD$C_STPSV                     =  00000000   G
PPD$C_STURC                     =  00000003   G
PPD$C_STURP                     =  00000001   G
PPD$C_TYPACCV                   =  00000004   G
PPD$C_TYPBLV                    =  00000003   G
PPD$C_TYPBMSE                   =  00000006   G
PPD$C_TYPINVBN                  =  00000002   G
PPD$C_TYPNP                     =  00000005   G
PPD$C_TYPOK                     =  00000000   G
PPD$C_TYPOTHER                  =  00000007   G
PPD$C_TYPVCC                    =  00000001   G
PPD$C_UNINIT                    =  00000000   G
PPD$C_VCDCL                     =  00000006   G
PPD$K_LB_LENGTH                    00000046   G
PPD$K_LENGTH                       00000012   G
PPD$L_BLINK                        00000004   G
PPD$L_DG_DISC                      00000028   G
PPD$L_FLINK                        00000000   G
PPD$L_IN_VCD                       00000018   G
PPD$L_LBCRC                        00000042   G
PPD$L_PO_ACK                       00000010   G
PPD$L_PO_NAK                       00000014   G
PPD$L_PO_NRSP                      00000018   G
PPD$L_P1_ACK                       0000001C   G
PPD$L_P1_NAK                       00000020   G
PPD$L_P1_NRSP                      00000024   G
PPD$L_REC_BOFF                     00000028   G
PPD$L_REC_NAME                     00000024   G
PPD$L_RPORT_FCN                    00000020   G
PPD$L_RPORT_REV                    0000001C   G
PPD$L_RPORT_TYP                    00000018   G
PPD$L_SND_BOFF                     00000020   G
PPD$L_SND_NAME                     0000001C   G
PPD$L_ST_ADDR                      00000018   G
PPD$L_XCT_LEN                      00000018   G
PPD$M_CST                       =  00008000   G
PPD$M_DISPOSE                   =  00000001   G
PPD$M_DQI                       =  00001000   G
PPD$M_DS                        =  00000001   G
PPD$M_DSTART                    =  00000080   G
PPD$M_DUALPATH                  =  80000000   G
PPD$M_ERR                       =  00000001   G
PPD$M_FORCE                     =  00000080   G
PPD$M_MAINT                     =  00000001   G
PPD$M_NR                        =  00004000   G
PPD$M_NS                        =  00002000   G
PPD$M_P                         =  00000080   G
PPD$M_POSTS                     =  00000006   G
PPD$M_P1STS                     =  00000018   G
PPD$M_RSP                       =  00000001   G
PPD$Q_CURTIME                      00000048   G
PPD$Q_NODENAME                     00000040   G
PPD$Q_SWINCARN                     00C00028   G
PPD$Q_XCT_ID                       00000010   G
PPD$S_M                         =  00000003   G
PPD$S_PORT_TYP                  =  0000001F   G
PPD$S_PS                        =  00000002   G
PPD$S_PSTS                      =  00000002   G
PPD$S_PTH0                      =  00000002   G
PPD$S_PTH1                      =  00000002   G
PPD$S_RP                        =  00000002   G
PPD$S_SP                        =  00000002   G
PPD$S_STATE                     =  00000002   G
```

```
PPD$S_STSST                =  00000004   G
PPD$S_STSTYP               =  00000003   G
PPD$T_HWTYPE                  00000030   G
PPD$T_SWTYPE                  00000020   G
PPD$T_SWVERS                  00000024   G
PPD$V_CST                  =  0000000F   G
PPD$V_DISPOSE              =  00000000   G
PPD$V_DQI                  =  0000000C   G
PPD$V_DS                   =  00000000   G
PPD$V_DSTART               =  00000007   G
PPD$V_DUALPATH             =  0000001F   G
PPD$V_ERR                  =  00000000   G
PPD$V_FORCE                =  00000007   G
PPD$V_M                    =  00000004   G
PPD$V_MAINT                =  00000000   G
PPD$V_NR                   =  0000000E   G
PPD$V_NS                   =  0000000D   G
PPD$V_P                    =  00000007   G
PPD$V_PORT_TYP             =  00000000   G
PPD$V_PS                   =  00000001   G
PPD$V_PSTS                 =  00000009   G
PPD$V_PTH0                 =  00000001   G
PPD$V_PTH1                 =  00000003   G
PPD$V_RP                   =  00000001   G
PPD$V_RSP                  =  00000000   G
PPD$V_SP                   =  00000004   G
PPD$V_STATE                =  00000001   G
PPD$V_STSST                =  00000001   G
PPD$V_STSTYP               =  00000005   G
PPD$W_LCB_LEN7                00000000   G
PPD$W_LENGTH                  00000010   G
PPD$W_MASK                    00000010   G
PPD$W_MAXDG                   0000001C   G
PPD$W_MAXMSG                  0000001E   G
PPD$W_MTYPE                   00000012   G
PPD$W_M_VAL                   00000014   G
PPD$W_SIZE                    00000008   G
READ_COUNT_ERR                000001A2 R      01
RESPONDER_NAME                ********   X    01
SB$B_ENBMSK                =  0000005A   G
SB$B_HWVERS                =  00000038   G
SB$B_SUBTYP                =  0000000B   G
SB$B_SYSTEMID              =  00000018   G
SB$B_TYPE                  =  0000000A   G
SB$C_LENGTH                =  00000060   G
SB$K_LENGTH                =  00000060   G
SB$L_BLINK                 =  00000004   G
SB$L_CSB                   =  0000005C   G
SB$L_DDB                   =  00000054   G
SB$L_FLINK                 =  00000000   G
SB$L_PBBL                  =  00000010   G
SB$L_PBCONNX               =  00000014   G
SB$L_PBFL                  =  0000000C   G
SB$Q_SWINCARN              =  0000002C   G
SB$S_ENBMSK                =  00000002   G
SB$S_HWTYPE                =  00000004   G
SB$S_HWVERS                =  0000000C   G
SB$S_NODENAME                 00000010   G
SB$S_SBDEF                 =  00000060   G
SB$S_SWINCARN              =  00000008   G
SB$S_SWTYPE                =  00000004   G
SB$S_SWVERS                =  00000004   G
SB$S_SYSTEMID              =  00000006   G
SB$T_HWTYPE                =  00000034   G
SB$T_NODENAME              =  00000044   G
SB$T_SWTYPE                =  00000024   G
SB$T_SWVERS                =  00000028   G
SB$W_MAXDG                 =  00000020   G
SB$W_MAXMSG                =  00000022   G
SB$W_SIZE                  =  00000008   G
SB$W_TIMEOUT               =  00000058   G
SCS$ALLOC_RSPID               ********     X    01
SCS$B_CON_DAT              =  00000024   G
SCS$B_PPD                  =  FFFFFFE0   G
SCS$CONFIG_PTH                ********     X    01
SCS$CONNECT                   ********     X    01
SCS$C_ACCP_REQ             =  00000002   G
SCS$C_ACCP_REQL            =  00000042   G
SCS$C_ACCP_RSP             =  00000003   G
SCS$C_ACCP_RSPL            =  00000012   G
SCS$C_APPL_BASE            =  00000000   G
SCS$C_APPL_DG              =  0000000B   G
SCS$C_APPL_MSG             =  0000000A   G
SCS$C_CON_BASE             =  00000004   G
SCS$C_CON_REQ              =  00000000   G
SCS$C_CON_REQL             =  00000042   G
SCS$C_CON_RSP              =  00000001   G
SCS$C_CON_RSPL             =  00000012   G
SCS$C_CR_REQ               =  00000008   G
SCS$C_CR_REQL              =  00000012   G
SCS$C_CR_RSP               =  00000009   G
SCS$C_CR_RSPL              =  0000000E   G
SCS$C_DISC_REQ             =  00000006   G
SCS$C_DISC_REQL            =  00000012   G
SCS$C_DISC_RSP             =  00000007   G
SCS$C_DISC_RSPL            =  0000000E   G
SCS$C_OVHD                 =  0000000E   G
SCS$C_REJ_REQ              =  00000004   G
SCS$C_REJ_REQL             =  00000012   G
SCS$C_REJ_RSP              =  00000005   G
SCS$C_REJ_RSPL             =  0000000E   G
SCS$C_STDISC               =  00000019   G
SCS$C_STINSFCR             =  00000021   G
SCS$C_STNOMAT              =  00C00000A  G
SCS$C_STNORMAL             =  00000001   G
SCS$C_STNORS               =  00000012   G
SCS$DISCONNECT                ********     X    01
SCS$GW_MAXDG                  ********     X    01
SCS$GW_MAXMSG                 ********     X    01
SCS$K_APPL_BASE            =  00000000   G
SCS$K_CON_BASE             =  00000004   G
SCS$K_STDISC               =  00000019   G
SCS$K_STINSFCR             =  00000021   G
SCS$K_STNOMAT              =  0000000A   G
```

```
SCS$K_STNORMAL          =  00000001  G
SCS$K_STNORS            =  00000012  G
SCS$L_DST_CONID         =  FFFFFFF8  G
SCS$L_LCONID            =  FFFFFFF0  G
SCS$L_REC_BOFF          =  00000008  G
SCS$L_REC_NAME          =  00000004  G
SCS$L_RSPID             =  FFFFFFF4  G
SCS$L_SND_BOFF          =  00000000  G
SCS$L_SND_NAME          =  FFFFFFFC  G
SCS$L_SRC_CONID         =  FFFFFFFC  G
SCS$L_XCT_LEN           =  FFFFFFF8  G
SCS$S_CON_DAT           =  00000010  G
SCS$S_DST_PROC          =  00000010  G
SCS$S_PPD               =  00000010  G
SCS$S_SCSDEF            =  00000054  G
SCS$S_SCSDEF1           =  0000001C  G
SCS$S_SRC_PROC          =  00000010  G
SCS$T_DST_PROC          =  00000004  G
SCS$T_SRC_PROC          =  00000014  G
SCS$W_CREDIT            =  FFFFFFF6  G
SCS$W_LENGTH            =  FFFFFFF0  G
SCS$W_MIN_CR            =  00000000  G
SCS$W_MTYPE             =  FFFFFFF4  G
SCS$W_STATUS            =  00000002  G
SEND_DG_ERR               00000175  R     01
SEND_MSG_ERR              00000175  R     01
SEND_RESET_ERR            000001B7  R     01
SEND_START_ERR            000001B7  R     01
SET_RSP_STATUS            0000072B  R     01
SET_RSP_STATUS_2          0000072D  R     01
SIO_ARMPATH               ********    X   01
SIO_CARRIER               ********    X   01
SIO_COMPLETE              000000FA  RG    01
SIO_COMPLETE_ERR          000000FF  RG    01
SIO_CONFIG                000001C3  R     01
SIO_CONNECT               00000202  R     01
SIO_DG                    000002FF  R     01
SIO_DISC                  00000295  R     01
SIO_DISPATCH              0000002F  R     01
SIO_GARBAGE               0000051D  R     01
SIO_INITCINT              ********    X   01
SIO_INV_OPCODE            0000007D  R     01
SIO_LRGPACKET             ********    X   01
SIO_MAINT_STATE           ********    X   01
SIO_MAP                   00000336  R     01
SIO_MOVCTP                00000546  R     01
SIO_MSG                   000002C1  R     01
SIO_NAK                   ********    X   01
SIO_NOACK                 ********    X   01
SIO_NORSP                 ********    X   01
SIO_NXT_CMD               0000007D  RG    01
SIO_NXT_CMD_ERR           000000C8  R     01
SIO_OVERSIZEPKT           ********    X   01
SIO_POLLER                ********    X   01
SIO_POOF                  000000A5  R     01
SIO_READCNT               00000464  R     01
SIO_READPATH              ********    X   01
SIO_REQDAT                00C0003B9 R     01
SIO_REQMDAT               0000040B  R     01
SIO_RESET                 000004CF  R     01
SIO_SELADR                ********    X   01
SIO_SETUP_SCS             0000057A  RG    01
SIO_SNDDAT                000003B9  R     01
SIO_SNDMDAT               0000040B  R     01
SIO_START                 000004F2  R     01
SIO_UNMAP                 00000389  R     01
SIZ...                  =  00000001
SS$_DEVREQERR           =  00000334
SS$_INSFMEM             =  00000124
SS$_NORMAL              =  00000001
SS$_REJECT              =  00000294
STATE_M_ALTGRBG         =  00000004
STATE_M_CANCEL          =  00000008
STATE_M_CMAP            =  00000002
STATE_M_FDT             =  00000400
STATE_M_NOTPROC         =  00001000
STATE_M_RMAP            =  00000001
STATE_M_SIO             =  00000800
STATE_M_STOP            =  00000200
STATE_V_ALTGRBG         =  00000002
STATE_V_CANCEL          =  00000003
STATE_V_CMAP            =  00000001
STATE_V_FDT             =  0000000A
STATE_V_NOTPROC         =  0000000C
STATE_V_RMAP            =  00000000
STATE_V_SIO             =  0000000B
STATE_V_STOP            =  00000009
SYSAP$C_DISPPO          =  00000002
UCB$M_BSY               =  00000100
UCB$W_STS               =  00000064
XFR_ERR                   0000018A  R     01
XFR_SETUP                 000005C0  R     01
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	000007B8 (1976.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000944 (2372.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
ALLOC_CDB	000006C2-R	1739 (25)	#-537 (4)
ALLOC_CDB_ERR	0000015D-R	371 (2)	#-538 (4)
ALLOC_CDRP_CTP	00000643-R	1636 (23)	#-1079 (12) #-734 (6) #-797 (7)
ALLOC_CDRP_ERR	0000015D-R	372 (2)	#-1080 (12) #-735 (6) #-798 (7) #-848 (8)
ALLOC_CDRP_XFR	00000672-R	1687 (24)	#-847 (8)
ALLOC_DG_ERR	00000163-R	378 (2)	#-1086 (12) #-803 (7)
ALLOC_MSG_ERR	00000163-R	377 (2)	#-740 (6)
BIT...	=0000000D	115 (1)	115 (1)
BUFFER_NOT_MAPPED	0000012C-R	352 (2)	#-1011 (11) #-905 (9) #-958 (10)
CATCH_RST_STR	00000770-R	1903 (28)	#-1861 (27)
CDB$B_CDRPCNT	00000028		#-1759 (25)
CDB$B_RSTNADR	0000000E		1752 (25) #-1755 (25) #-1756 (25) 554 (4)
CDB$B_TYPE	0000000A		#-1749 (25)
CDB$K_LEN	00000032		#-1742 (25)
CDB$L_CDBSLOT	00000024		#-1770 (25) #-366 (2) #-582 (4)
CDB$L_CDRPQBL	0000002E		#-1763 (25)
CDB$L_CDRPQFL	0000002A		1760 (25) #-1761 (25) 1762 (25)
CDB$L_CDT	0000001C		#-1333 (17) #-1449 (20) 563 (4) #-565 (4)
			#-622 (5)
CDB$L_PDT	00000020		#-1331 (17) 563 (4)
CDB$T_LPRTNAM	00000016		1752 (25) #-1758 (25)
CDB$W_SIZE	00000008		#-1747 (25)
CDB$W_STATUS	0000000C		#-540 (4) #-557 (4) 559 (4) #-579 (4)
			637 (5)
CDB_M_CONN	=00000001		#-539 (4) #-556 (4)
CDB_M_DISC	=00000002		#-578 (4)
CDB_V_CNCL	=00000002		#-558 (4) #-636 (5)
CDC$ARMCINTPATH	0000001B		221 (2)
CDC$B_CFGCBLSTS	00000021		#-477 (3)
CDC$B_CFGP0LBSTS	0000002A		484 (3) #-488 (3)
CDC$B_CFGP0STS	0000001F		470 (3) #-475 (3)
CDC$B_CFGP1LBSTS	0000002B		484 (3)
CDC$B_CFGP1STS	00000020		470 (3)
CDC$B_CMDCOUNT	00000000		245 (2)
CDC$B_CNTFLG	0000001F		#-1096 (12)
CDC$B_DONECNT	00000001		#-1363 (18) #-184 (2) #-251 (2) #-293 (2)
CDC$B_DSTPORT	00000001		#-1020 (11) #-1024 (11) #-1395 (19) #-1754 (25)
			#-1766 (25)
CDC$B_EXTEND	0000001C		#-1155 (13) #-1189 (14)
CDC$B_GARBAGE_FLAG	00000014		#-1237 (15) #-1242 (15)
CDC$B_OPCODE	00000000		#-1016 (11) #-221 (2) #-963 (10)
CDC$B_OTHERNODE	0000001E		#-1100 (12) 1105 (12) 1158 (13) 1194 (14)
CDC$B_SIOCNT	00000003		#-248 (2) #-249 (2) #-288 (2) #-289 (2)
CDC$CARRIER	00000018		221 (2)
CDC$CONFIGDATA	00000007		221 (2)
CDC$CONNECT	00000008		221 (2)
CDC$DISCONNECT	00000009		221 (2)
CDC$GARBAGE	0000000B		221 (2)
CDC$INITCINT	0000001A		221 (2)
CDC$K_CMDHDRSIZ	0000000C		#-166 (2)

CDC$K_CMDSPEC	00000014	#-1293	(16)	1294	(16)	#-808	(7)			
CDC$K_DATA	00000014	571	(4)							
CDC$LRGPACKET	00000019	221	(2)							
CDC$L_BUFADR	00000024	#-850	(8)							
CDC$L_BUFLENGTH	00000020	#-1458	(20)	#-1525	(21)	#-853	(8)	#-867	(8)	
		#-912	(9)							
CDC$L_BUFLNAME	00000024	#-1585	(22)	#-870	(8)	#-915	(9)			
CDC$L_BUFLOFSET	00000028	#-1445	(20)	#-1519	(21)					
CDC$L_BUFRNAME	0000002C	#-1454	(20)	#-1522	(21)					
CDC$L_BUFROFSET	00000030	#-1447	(20)							
CDC$L_CDB_AD	0000000C	#-1768	(25)							
CDC$L_CDRP	0000000C	#-1030	(11)	#-1082	(12)	#-1330	(17)	#-1592	(22)	
		#-1648	(23)	#-1705	(24)	#-383	(2)	#-393	(2)	
		#-414	(2)	#-863	(8)	#-974	(10)			
CDC$L_CFGHWTYPE	00000022	#-479	(3)							
CDC$L_CFGPRTMSK	00000026	#-481	(3)							
CDC$L_CNTRPOACK	00000020	1120	(12)							
CDC$L_LPORT	00000010	#-1393	(19)	#-1757	(25)	#-1765	(25)			
CDC$L_R0STATUS	0000000C	#-1813	(26)	#-569	(4)					
CDC$L_R1STATUS	00000010	#-1814	(26)	#-570	(4)					
CDC$L_STARTADR	00000020	#-1194	(14)							
CDC$L_SVAPTE	00000018	#-855	(8)							
CDC$MAINTSTATE	0000000F	221	(2)							
CDC$MAPBUF	00000005	221	(2)							
CDC$MAPMBUF	00000010	221	(2)							
CDC$NAK	00000016	221	(2)							
CDC$NOACK	00000017	221	(2)							
CDC$NORSP	00000015	221	(2)							
CDC$OVERSIZEPKT	0000001D	221	(2)							
CDC$POLLER	0000000E	221	(2)							
CDC$RDCINTPATH	0000001C	221	(2)							
CDC$READCNT	0000000A	221	(2)							
CDC$REQDATA	00000004	221	(2)	#-964	(10)					
CDC$REQMDATA	00000013	#-1017	(11)	221	(2)					
CDC$RESET	0000000C	221	(2)							
CDC$SELADR	00000014	221	(2)							
CDC$SENDDATA	00000003	221	(2)							
CDC$SENDDG	00000002	221	(2)							
CDC$SENDMDATA	00000012	221	(2)							
CDC$SENDMSG	00000001	221	(2)							
CDC$START	0000000D	221	(2)							
CDC$T_CONDAT	00000040	554	(4)							
CDC$T_GARBAGE	00000018	1234	(15)							
CDC$T_PACKETDATA	00000080	#-1917	(28)							
CDC$T_SYSTEMID	00000034	#-492	(3)	554	(4)					
CDC$UNMAPBUF	00000006	221	(2)							
CDC$UNMAPMBUF	00000011	221	(2)							
CDC$W_DGCREDIT	0000003E	#-554	(4)							
CDC$W_FLAG	00000004	#-1361	(18)	#-181	(2)	#-187	(2)	#-342	(2)	
		#-348	(2)	#-360	(2)	#-380	(2)	#-389	(2)	
		#-398	(2)	#-404	(2)	#-410	(2)	#-420	(2)	
		751	(6)							
CDC$W_MINSENDCR	0000003C	#-554	(4)							
CDC$W_MSGCREDIT	0000003A	#-554	(4)							
CDC$W_RSPOFSET	00000002	#-1226	(15)	#-1811	(26)	#-472	(3)	#-568	(4)	
		#-865	(8)	#-910	(9)					
CDC$W_TXTLEN	00000008	#-1292	(16)	#-236	(2)	#-278	(2)	#-807	(7)	

```
CDC$_GARBAGE                    =00000001                       #-1236  (15)
CDC$_NOGARBAGE                  =00000000                         1240  (15)
CDRP$B_CD_TYPE                  =0000000A                       #-1645  (23)  #-1703  (24)
CDRP$K_CTPLEN                    00000050                       #-1639  (23)  #-1856  (27)
CDRP$K_XFRLEN                    00000068                       #-1690  (24)
CDRP$L_BCNT                     =FFFFFFD2                       #-854   (8)   #-866   (8)   #-911   (9)
CDRP$L_CDT                      =00000024                       #-1334  (17)
CDRP$L_LBOFF                    =00000030                       #-1446  (20)  #-1520  (21)
CDRP$L_LBUFH_AD                 =0000002C                       #-1450  (20)  #-1582  (22)  #-1707  (24)  #-868   (8)
                                                                #-913   (9)
CDRP$L_PDT                       0000004C                       #-1083  (12)  #-1332  (17)  #-907   (9)
CDRP$L_PENDQFL                   00000044                         1027  (11)  #-1580  (22)    1591  (22)    861   (8)
                                                                  972  (10)
CDRP$L_RBOFF                    =00000038                       #-1448  (20)
CDRP$L_RBUFH_AD                 =00000034                       #-1453  (20)  #-1521  (21)  #-1709  (24)
CDRP$L_SVAPTE                   =FFFFFFCC                       #-856   (8)
CDRP$L_XCT_LEN                  =0000003C                       #-1459  (20)  #-1526  (21)
CDRP$T_CX_LBUFHNDL               00000050                         1706  (24)
CDRP$T_CX_RBUFHNDL               0000005C                         1708  (24)
CDRP$W_BOFF                     =FFFFFFD0                       #-852   (8)
CDRP$W_CDRPSIZE                 =00000008                       #-1643  (23)  #-1860  (27)
CDT$L_LCONID                    =00000018                       #-1456  (20)
CDT$L_RCONID                    =00000014                       #-1451  (20)
CHECK_ACTV_CONN                  00000000-XR                    #-1007  (11)  #-731   (6)   #-794   (7)   #-844   (8)
                                                                #-954   (10)
CHECK_CONN                       00000000-XR                    #-618   (5)
CHECK_CONN_ERR                   0000012D-R    346     (2)      #-1008  (11)  #-732   (6)   #-795   (7)   #-845   (8)
                                                                #-955   (10)
CHECK_RST_STR                    00000744-R    1846    (27)     #-744   (6)
CHECK_RST_STR_ERR                000001C3-R    424     (2)      #-745   (6)
CHECK_VC                         000005A0-R    1390    (19)     #-1076  (12)  #-1152  (13)  #-1186  (14)  #-466   (3)
CHECK_VC_ERR                     00000114-R    340     (2)      #-1077  (12)  #-1153  (13)  #-1187  (14)
CIBHAN$L_BNAME                  =00000004                       #-1455  (20)  #-1524  (21)  #-1584  (22)  #-869   (8)
                                                                #-914   (9)
CIBHAN$L_BOFF                   =00000000                       #-1523  (21)
CIBHAN$L_RCONID                 =00000008                       #-1452  (20)  #-1457  (20)
CLEAN_CDB                        00000000-XR                    #-644   (5)
CLEAN_CDRP                       00000000-XR                    #-1123  (12)  #-391   (2)   #-412   (2)
CLEAR_BUFFER                     00000000-XR                    #-1642  (23)  #-1694  (24)  #-1746  (25)
COM$DRVDEALMEM                   00000000-XR                      1911  (28)    1924  (28)    368   (2)     584   (4)
CONNECT_ERR                      00000132-R    357     (2)      #-560   (4)
CONTROLLER_NAME                  00000000-XR                      1105  (12)    1110  (12)    554   (4)
CTP$GENRSTREQ                    00000006                       #-1848  (27)
CTP$GENSTRREQ                    00000007                       #-1850  (27)
CTP$REVISION                    =00000000       112     (1)
CTP$VERSION                     =00000003       112     (1)
CX$B_CMDERRCNT                   00000000-XR                    #-164   (2)   #-291   (2)   #-292   (2)
CX$B_GARBAGE_FLAG                00000000-XR                      1227  (15)
CX$CONNECT_ERR                   00000000-XR                       554  (4)
CX$DG_IN                         00000000-XR                       554  (4)
CX$L_CMDBUFBGN                   00000000-XR                    #-1362  (18)  #-165   (2)   #-183   (2)   #-247   (2)
                                                                #-287   (2)
CX$L_CUR_CMD                     00000000-XR                    #-1029  (11)  #-1360  (18)  #-167   (2)   #-1704  (24)
                                                                #-235   (2)   #-238   (2)   #-277   (2)   #-280   (2)
                                                                #-359   (2)   #-567   (4)   #-645   (5)
CX$L_LOCPRTLST                   00000000-XR                    #-1765  (25)
CX$L_RSPBUFBGN                   00000000-XR                    #-1226  (15)  #-1812  (26)  #-1916  (28)  #-472   (3)
```

```
                                                        #-568     (4)     #-865    (8)     #-910    (9)
CX$L_RST_STR_CDRP              00000000-XR              #-1906    (28)
CX$L_RST_STR_PDT               00000000-XR              #-1905    (28)    #-1908   (28)
CX$L_STATE                     00000000-XR              #-160     (2)     #-161    (2)     #-262    (2)     #-263    (2)
                                                        #-324     (2)
CX$MSG_IN                      00000000-XR                554     (4)
CX$Q_MAPLST                    00000000-XR               1028     (11)      1574   (22)      862    (8)       973    (10)
CX$T_GARBAGE_BUF               00000000-XR               1233     (15)
CX$X_IRPSAV                    00000000-XR              #-162     (2)     #-163    (2)     #-1698   (24)    #-260    (2)
                                                        #-261     (2)     #-325    (2)     #-326    (2)
CX$_CDRPNFND                  =00000001                 #-1603    (22)
CX_STARTIO                     00000000-R    155   (2)
DEALLOC_CDRP                   00000000-XR              #-1124    (12)    #-382    (2)     #-392    (2)     #-413    (2)
                                                        #-916     (9)
DYN$C_CDB                     =00000033                 #-1748    (25)
DYN$C_CDRP                    =00000039                 #-1644    (23)    #-1702   (24)
DYN$C_CIMSG                   =0000003C                 #-1281    (16)
DYN$C_IRP                     =0000000A                 #-1696    (24)
EXE$ALONONPAGED                00000000-XR               1640     (23)      1692   (24)      1743   (25)      1857   (27)
FIND_CDRP_XFR                  00000602-R    1571  (22) #-1010    (11)    #-904    (9)     #-957    (10)
FLAG_M_CMD_DONE               =00008000                 #-1361    (18)    #-180    (2)
FLAG_M_CMD_START              =00004000                 #-187     (2)
FLAG_M_CONN_LOST              =00000800                 #-180     (2)     #-348    (2)
FLAG_M_SCS_ERR                =00001000                 #-342     (2)     #-360    (2)     #-380    (2)     #-389    (2)
                                                        #-398     (2)     #-404    (2)     #-410    (2)     #-420    (2)
FLAG_V_NO_RESPONSE            =00000000                 #-750     (6)
GET_PDT_ADDR                   000005A0-R    1391  (19)
IOC$REQCOM                     00000000-XR                327     (2)
IRP$B_RMOD                    =0000000B                 #-1699    (24)    #-1700   (24)
IRP$B_TYPE                    =0000000A                 #-1697    (24)
IRP$C_CDRP                    =00000060                 #-1691    (24)      1701   (24)
IRP$W_SIZE                    =00000008                 #-1695    (24)
MARK_CMD_DONE                  0000058C-R    1358  (18) #-1033    (11)    #-1126   (12)    #-1161   (13)    #-1197   (14)
                                                        #-1244    (15)    #-494    (3)     #-573    (4)     #-648    (5)
                                                        #-752     (6)     #-872    (8)     #-918    (9)     #-977    (10)
MNT$CATCH_RST                 =00000007      105   (1)  #-1907    (28)
MNT$CATCH_STR                 =00000008      105   (1)
MNT$MAINT_READ                =00000005      105   (1)  #-1025    (11)
MNT$MAINT_STATE               =00000002      105   (1)
MNT$MAINT_WRITE               =00000006      105   (1)  #-1021    (11)
MNT$NORMAL_STATE              =00000004      105   (1)
MNT$POLLER_OFF                =00000000      105   (1)
MNT$POLLER_ON                 =00000001      105   (1)
MNT$UNINIT_STATE              =00000003      105   (1)
MNT_XFR_ERR                    00000196-R    402   (2)  #-1031    (11)
MNT_XFR_SETUP                  000005EB-R    1517  (21) #-1014    (11)
NODE_UNKNOWN                   00000114-R    339   (2)  #-467     (3)
PB$B_CBL_STS                  =00000028                 #-476     (3)
PB$B_P0_STS                   =00000029                   469     (3)     #-474    (3)
PB$B_P1_STS                   =0000002A                   469     (3)
PB$L_PDT                      =0000002C                 #-1401    (19)    #-486    (3)
PB$L_RPORT_FCN                =0000001C                 #-480     (3)
PB$L_RPORT_TYP                =00000014                 #-478     (3)
PB$L_SBLINK                   =00000030                 #-490     (3)
PDT$B_P0_LBSTS                 00000180                   483     (3)     #-487    (3)
PDT$B_P1_LBSTS                 00000181                   483     (3)
PDT$L_ALLOCDG                 =00000010                  1021     (11)      1025   (11)      1085   (12)      802    (7)
```

```
PDT$L_ALLOCMSG                  =00000014                       739     (6)     967     (10)    970     (10)
PDT$L_MAINTFCN                  =00000078                       1021    (11)    1025    (11)    1907    (28)
PDT$L_MAPIRP                    =00000034                       860     (8)
PDT$L_MRESET                    =00000070                       1158    (13)
PDT$L_MSTART                    =00000074                       1194    (14)
PDT$L_READCOUNT                 =00000068                       1105    (12)    1110    (12)
PDT$L_REQDATA                   =00000050                       970     (10)
PDT$L_RLSCOUNT                  =0000006C                       1114    (12)
PDT$L_SENDDATA                  =00000054                       967     (10)
PDT$L_SENDDG                    =00000058                       810     (7)
PDT$L_SENDMSG                   =0000005C                       747     (6)
PDT$L_UNMAP                     =00000064                       908     (9)
PPD$B_TYPE                       0000000A                       #-1282  (16)
PPD$L_PO_ACK                     00000010                       1119    (12)
READ_COUNT_ERR                   000001A2-R     408     (2)     #-1117  (12)
RESPONDER_NAME                   00000000-XR                    554     (4)
SB$B_SYSTEMID                   =00000018                       #-491   (3)
SCS$ALLOC_RSPID                  00000000-XR                    967     (10)    970     (10)
SCS$B_PPD                       =FFFFFFE0                       #-1087  (12)    #-1278  (16)
SCS$CONFIG_PTH                   00000000-XR                    1398    (19)
SCS$CONNECT                      00000000-XR                    554     (4)
SCS$DISCONNECT                   00000000-XR                    580     (4)     635     (5)
SCS$GW_MAXDG                     00000000-XR                    #-1284  (16)
SCS$GW_MAXMSG                    00000000-XR                    #-1280  (16)
SCS$K_APPL_BASE                 =00000000                       #-1087  (12)    #-1278  (16)
SEND_DG_ERR                      00000175-R     387     (2)     #-811   (7)
SEND_MSG_ERR                     00000175-R     386     (2)     #-748   (6)
SEND_RESET_ERR                   000001B7-R     417     (2)     #-1159  (13)
SEND_START_ERR                   000001B7-R     418     (2)     #-1195  (14)
SET_RSP_STATUS                   0000072B-R     1804    (26)    #-343   (2)     #-363   (2)     #-374   (2)     #-381   (2)
                                                                #-390   (2)     #-399   (2)     #-405   (2)     #-411   (2)
                                                                #-421   (2)     #-646   (5)
SET_RSP_STATUS_2                 0000072D-R     1808    (26)    #-349   (2)     #-354   (2)     #-365   (2)
SIO_ARMPATH                      00000000-XR                    221     (2)
SIO_CARRIER                      00000000-XR                    221     (2)
SIO_COMPLETE                     000000FA-R     305     (2)     #-252   (2)
SIO_COMPLETE_ERR                 000000FF-R     315     (2)
SIO_CONFIG                       000001C3-R     464     (3)     221     (2)
SIO_CONNECT                      00000202-R     535     (4)     221     (2)
SIO_DG                           000002FF-R     792     (7)     221     (2)
SIO_DISC                         00000295-R     616     (5)     221     (2)
SIO_DISPATCH                     00000027-R     170     (2)     #-250   (2)
SIO_GARBAGE                      0000051D-R     1224    (15)    221     (2)
SIO_INITCINT                     00000000-XR                    221     (2)
SIO_INV_OPCODE                   0000007D-R     224     (2)
SIO_LRGPACKET                    00000000-XR                    221     (2)
SIO_MAINT_STATE                  00000000-XR                    221     (2)
SIO_MAP                          00000336-R     842     (8)     221     (2)
SIO_MOVCTP                       00000546-R     1273    (16)    #-742   (6)     #-805   (7)
SIO_MSG                          000002C1-R     729     (6)     221     (2)
SIO_NAK                          00000000-XR                    221     (2)
SIO_NOACK                        00000000-XR                    221     (2)
SIO_NORSP                        000000C0-XR                    221     (2)
SIO_NXT_CMD                      0000007D-R     233     (2)     #-1034  (11)    #-1127  (12)    #-1162  (13)    #-1198  (14)
                                                                #-1245  (15)    #-185   (2)     221     (2)     #-495   (3)
                                                                #-574   (4)     #-649   (5)     #-754   (6)     #-813   (7)
                                                                #-873   (8)     #-919   (9)     #-978   (10)
```

Command Start I/O

SIO_NXT_CMD_ERR	000000C8-R	275	(2)	#-344	(2)	#-350	(2)	#-355	(2)	#-369	(2)
				#-375	(2)	#-384	(2)	#-394	(2)	#-400	(2)
				#-406	(2)	#-415	(2)	#-422	(2)		
SIO_OVERSIZEPKT	00000000-XR			221	(2)						
SIO_POLLER	00000000-XR			221	(2)						
SIO_POOF	000000A5-R	258	(2)	#-295	(2)						
SIO_READCNT	00000464-R	1074	(12)	221	(2)						
SIO_READPATH	00000000-XR			221	(2)						
SIO_REQDAT	000003B9-R	952	(10)	221	(2)						
SIO_REQMDAT	0000040B-R	1005	(11)	221	(2)						
SIO_RESET	000004CF-R	1150	(13)	221	(2)						
SIO_SELADR	00000000-XR			221	(2)						
SIO_SETUP_SCS	0000057A-R	1328	(17)	#-1013	(11)	#-737	(6)	#-800	(7)	#-858	(8)
				#-960	(10)						
SIO_SNDDAT	000003B9-R	951	(10)	221	(2)						
SIO_SNDMDAT	0000040B-R	1004	(11)	221	(2)						
SIO_START	000004F2-R	1184	(14)	221	(2)						
SIO_UNMAP	00000389-R	902	(9)	221	(2)						
SIZ...	=00000001	115	(1)	115	(1)						
SS$_DEVREQERR	=00000334			#-1602	(22)						
SS$_INSFMEM	=00000124			#-1652	(23)	#-1715	(24)	#-1777	(25)		
SS$_NORMAL	=00000001			#-1594	(22)	#-1649	(23)	#-1711	(24)	#-1773	(25)
				#-307	(2)						
SS$_REJECT	=00000294			#-361	(2)						
STATE_M_ALTGRBG	=00000004	115	(1)								
STATE_M_CANCEL	=00000008	115	(1)								
STATE_M_CMAP	=00000002	115	(1)	#-160	(2)	#-322	(2)				
STATE_M_FDT	=00000400	115	(1)								
STATE_M_NOTPROC	=00001000	115	(1)	#-263	(2)						
STATE_M_RMAP	=00000001	115	(1)	#-323	(2)						
STATE_M_SIO	=00000800	115	(1)	#-161	(2)	#-262	(2)				
STATE_M_STOP	=00000200	115	(1)								
STATE_V_ALTGRBG	=00000002	115	(1)								
STATE_V_CANCEL	=00000003	115	(1)								
STATE_V_CMAP	=00000001	115	(1)								
STATE_V_FDT	=0000000A	115	(1)								
STATE_V_NOTPROC	=0000000C	115	(1)								
STATE_V_RMAP	=00000000	115	(1)								
STATE_V_SIO	=0000000B	115	(1)								
STATE_V_STOP	=00000009	115	(1)								
SYSAP$C_DISPPO	=00000002			#-810	(7)						
UCB$M_BSY	=00000100			#-264	(2)						
UCB$W_STS	=00000064			#-265	(2)						
XFR_ERR	0000018A-R	396	(2)	#-975	(10)						
XFR_SETUP	000005C0-R	1443	(20)	#-961	(10)						

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$BND	1	221 (2)	221 (2)
$BND1	1	221 (2)	221 (2)
$BND2	1	221 (2)	221 (2)
$CDCDEF	8	110 (1)	110 (1)
$CDRPDEF	6	85 (1)	85 (1)
$CDTDEF	4	86 (1)	86 (1)
$CIBHANDEF	1	87 (1)	87 (1)
$CINTREGDEF	3	111 (1)	111 (1)
$CTPDEF	7	112 (1)	112 (1)
$CXCDBDEF	2	113 (1)	113 (1)
$CXCDRPDEF	1	114 (1)	114 (1)
$CXSTATEDEF	1	115 (1)	115 (1)
$DEFINI	1	85 (1)	102 (1) 103 (1) 104 (1) 110 (1)
			111 (1) 112 (1) 113 (1) 114 (1)
			85 (1) 86 (1) 87 (1) 88 (1)
			89 (1) 90 (1) 91 (1) 92 (1)
			93 (1) 94 (1) 95 (1) 96 (1)
			97 (1)
$DISPATCH	2	189 (2)	189 (2)
$DSP1	1	221 (2)	221 (2)
$DSP2	1	221 (2)	221 (2)
$DYNDEF	7	88 (1)	88 (1)
$IODEF	17	89 (1)	89 (1)
$IRPDEF	7	90 (1)	90 (1)
$MOVEADR	1	1105 (12)	1105 (12) 1110 (12) 1158 (13) 1194 (14)
			1398 (19)
$PAMAINTDEF	1	105 (1)	105 (1)
$PAPDTDEF	5	102 (1)	102 (1)
$PAREGDEF	5	103 (1)	103 (1)
$PBDEF	3	91 (1)	91 (1)
$PDTDEF	4	92 (1)	92 (1)
$PPDDEF	10	104 (1)	104 (1)
$PUSHADR	1	554 (4)	554 (4)
$SBDEF	2	93 (1)	93 (1)
$SCSDEF	4	94 (1)	94 (1)
$SSDEF	21	95 (1)	95 (1)
$SYSAPDEF	1	96 (1)	96 (1)
$UCBDEF	16	97 (1)	97 (1)
$VIELD1	1		115 (1)
ALLOC_DG_BUF	1	802 (7)	1021 (11) 1025 (11) 1085 (12) 802 (7)
ALLOC_MSG_BUF	1	739 (6)	739 (6)
ASSUME	1		1240 (15) 1752 (25) 245 (2) 469 (3)
			470 (3) 483 (3) 484 (3) 563 (4)
BLBCW	1	467 (3)	1008 (11) 1011 (11) 1031 (11) 1077 (12)
			1080 (12) 1086 (12) 1117 (12) 1153 (13)
			1159 (13) 1187 (14) 1195 (14) 467 (3)
			538 (4) 560 (4) 732 (6) 735 (6)
			740 (6) 745 (6) 748 (6) 795 (7)
			798 (7) 803 (7) 811 (7) 845 (8)
			848 (8) 905 (9) 955 (10) 958 (10)

```
                                                       975   (10)
BNEQW                  1            250    (2)         250   (2)       290   (2)
CALC_CDB_LIST          1            1765   (25)        1765  (25)
CALC_RSP               1            472    (3)         1226  (15)      472   (3)       568   (4)       865   (8)
                                                       910   (9)
CATCH_RST              1            1907   (28)        1907  (28)
CLR_DRVR_STATE         1            262    (2)         262   (2)
CONFIG_PTH             1            1397   (19)        1397  (19)
CONNECT                2            543    (4)         543   (4)
DISCONNECT             1            580    (4)         580   (4)       635   (5)
MAP_IRP                1            860    (8)         860   (8)
MRESET                 1            1157   (13)        1157  (13)
MSTART                 1            1192   (14)        1192  (14)
READ_COUNTERS          1            1104   (12)        1104  (12)      1109  (12)
READ_MAINT_DATA        1            1025   (11)        1025  (11)
REQCOM                 1            327    (2)         327   (2)
REQUEST_DATA           1            970    (10)        970   (10)
RLS_COUNTERS           1            1114   (12)        1114  (12)
SEND_DATA              1            967    (10)        967   (10)
SEND_DG_BUF            1            810    (7)         810   (7)
SEND_MSG_BUF           1            747    (6)         747   (6)
SET_CMD_FLAG           1            187    (2)         1361  (18)      187   (2)       342   (2)       348   (2)
                                                       360   (2)       380   (2)       389   (2)       398   (2)
                                                       404   (2)       410   (2)       420   (2)
SET_DRVR_STATE         1            160    (2)         160   (2)       161   (2)       263   (2)
UNMAP                  1            908    (9)         908   (9)
WRITE_MAINT_DATA       1            1021   (11)        1021  (11)
_VIELD                 1                               115   (1)

                                 +--------------------------+
                                 ! Performance indicators   !
                                 +--------------------------+

Phase                       Page faults    CPU Time        Elapsed Time
-----                       -----------    --------        ------------
Initialization                   108       00:00:00.37     00:00:01.25
Command processing               124       00:00:00.62     00:00:01.34
Pass 1                           951       00:00:38.50     00:00:50.42
Symbol table sort                  0       00:00:04.38     00:00:05.19
Pass 2                           369       00:00:07.14     00:00:08.58
Symbol table output               31       00:00:00.70     00:00:01.17
Psect synopsis output              4       00:00:00.03     00:00:00.03
Cross-reference output            12       00:00:01.47     00:00:01.78
Assembler run totals            1603       00:00:53.22     00:01:09.77

The working set limit was 2250 pages.
177132 bytes (346 pages) of virtual memory were used to buffer the intermediate code.
There were 150 pages of symbol table space allocated to hold 2713 non-local and 73 local symbols.
1928 source lines were read in Pass 1, producing 53 object records in Pass 2.
185 pages of virtual memory were used to define 74 macros.
```

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	4
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	27
DRB2:[SHULL.EVXCI.CXDRIVER]CXLIB.MLB;3	17
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	9
TOTALS (all libraries)	57

3274 GETS were required to define 57 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS/CROSS CXCMD+CXLIB/LIB+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB
```

Table of contents

(1)	60	DEFINITIONS	
(2)	101	CX_STARTDATA,	Start command list processing
(3)	272	FDT_DISPATCH,	Check validity of individual commands
(4)	392	FDT_ABORT,	Abort the I/O for various reasons
(5)	460	FDT_CONFIG,	Gather configuration data
(6)	492	FDT_CONNECT,	Establish connection
(7)	533	FDT_DISCONNECT,	Disconnect routine
(8)	570	FDT_SENDMSG,	Send message
(9)	608	FDT_SENDDG,	Send datagram
(10)	646	FDT_MAP,	Map buffer
(11)	692	FDT_MNTMAP,	Map a maintenance buffer
(12)	748	FDT_UNMAP,	Unmap buffer
(13)	781	FDT_SNDDAT,	Send data
(13)	782	FDT_REQDAT,	Request data
(14)	817	FDT_SNDMDAT,	Send maintenance data
(14)	818	FDT_REQMDAT,	Request maintenance data
(15)	850	FDT_RDCNT,	Read performance counters
(16)	883	FDT_RESET,	Send reset
(16)	884	FDT_START,	Send start
(17)	913	FDT_POLLER,	Start/Stop poller
(17)	914	FDT_SELADR,	Change port address
(17)	915	FDT_NORSP,	Arm CINT NORSP logic
(17)	916	FDT_NAK,	Arm CINT NAK logic
(17)	917	FDT_NOACK,	Arm CINT NOACK logic
(17)	918	FDT_CARRIER,	Place carrier on wire
(18)	956	CHECK_LOC_PORT,	Check local port name
(19)	1005	ALLOC_RSP_SCS,	Allocate SCS response buffer space
(19)	1006	ALLOC_RSP_CTP,	Allocate CTP response buffer space
(20)	1088	FILL_CMD_REF,	Build a command reference number
(21)	1127	CHECK_CONN,	Check for connection
(22)	1170	CHECK_ACTV_CONN,	Check for an active connection
(23)	1216	CLEAR_BUFFER,	Clear buffer routine
(24)	1245	ALLOC_SEQNUM,	Create a sequence number

```
                        1         .TITLE  CXCMDFDT, STARTDATA FDT Routines
                        2         .IDENT  'V06-000'
                        3
                        4 ;*****************************************************************************
                        5 ;
                        6 ;                          COPYRIGHT (c) 1981, 1984 BY
                        7 ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
                        8 ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
                        9 ;
                       10 ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
                       11 ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
                       12 ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
                       13 ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
                       14 ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
                       15 ;
                       16 ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
                       17 ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
                       18 ;
                       19 ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
                       20 ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
                       21 ;
                       22 ;*****************************************************************************
                       23 ;
                       24 ;++
                       25 ;
                       26 ;
                       27 ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
                       28 ;
                       29 ;
                       30 ; ABSTRACT:      Command buffer FDT routines
                       31 ;
                       32 ; AUTHOR:        Richard Hennessy  27-Jul-1982
                       33 ;                Base Systems Diagnostic Engineering
                       34 ;
                       35 ; MODIFIED BY:   Jim Klumpp  6-MAY-83
                       36 ;                Base Systems Diagnostic Engineering
                       37 ;
                       38 ;       6-000    Dave Shull      07-July-1984
                       39 ;                VMS V4 Modifications/Release.  Changed CX_VERSION from 5 to 6.
                       40 ;
                       41 ;       5-002    Dave Shull      06-July-1984
                       42 ;                Removed instruction in routine CHECK_LOC_PORT to NOT clear
                       43 ;                the 4th byte of device PAx0.  VMS V4 expects the unit number,
                       44 ;                within the port name, to be passed as ASCII versus Integer.
                       45 ;
                       46 ;       5-001    Dave Shull      7-Feb-1984
                       47 ;                Change Reference of (MFPR       #PR$_TODR,(R2)) to
                       48 ;                MFPR #PR780$_TODR,(R2).  TODR register is considered a processor
                       49 ;                specific implementation register, as of VMS V4, but has the same
                       50 ;                internal register number.
                       51 ;
                       52 ;--
                       53
                       54 ;  Define CXDRIVER version for compatibility to control process.
                       55
00000006               56         CX_VERSION == 6
                       57
```

```
                        0000     58          .DEFAULT DISPLACEMENT,WORD
                        0000     59          .ENABLE SUPPRESSION
                        0000     60          .SBTTL  DEFINITIONS
                        0000     61
                        0000     62 ; Set PSECT to driver code
                        0000     63
                    00000000     64          .PSECT  $$$115_DRIVER,LONG
                        0000     65
                        0000     66 ;  System definitions (STARLET.MLB):
                        0000     67
                        0000     68          $PR780DEF                       ; Define 780 Specific registers
                        0000     69
                        0000     70 ;  System definitions (LIB.MLB):
                        0000     71
                        0000     72          $CDRPDEF                        ; Define CDRP offsets
                        0000     73          $CDTDEF                         ; Define CDT offsets
                        0000     74          $CIBHANDEF                      ; Define CI BUFFER HANDLE offsets
                        0000     75          $DYNDEF                         ; Define DYN offsets
                        0000     76          $IODEF                          ; Define I/O FUNCTION codes
                        0000     77          $IRPDEF                         ; Define IRP offsets
                        0000     78          $JIBDEF                         ; Define JIB offsets
                        0000     79          $PCBDEF                         ; Define PCB offsets
                        0000     80          $SCSDEF                         ; Define SCS packet offsets
                        0000     81          $SSDEF                          ; Define system status values
                        0000     82          $UCBDEF                         ; Define UCB offsets
                        0000     83
                        0000     84
                        0000     85 ;  PADRIVER definitions (PALIB.MLB):
                        0000     86
                        0000     87
                        0000     88
                        0000     89 ;  CXDRIVER definitions (CXLIB.MLB):
                        0000     90
                        0000     91          $CDCDEF                         ; Define CDC interface symbols
                        0000     92          $CTPDEF                         ; Define CTP symbols
                        0000     93          $CXCDBDEF                       ; Define CXCDB offsets
                        0000     94          $CXCDRPDEF                      ; Define CX CDRP extension
                        0000     95          $CXQIODEF                       ; Define QIO parameter offsets
                        0000     96          $CXSTATEDEF                     ; Define CXDRIVER states
                        0000     97
                        0000     98
                        0000     99
```

```
                        0000   101          .SBTTL  CX_STARTDATA,   Start command list processing
                        0000   102 ;+
                        0000   103 ;
                        0000   104 ;  CX_STARTDATA
                        0000   105 ;
                        0000   106 ;  This routine is called by the SYS$QIO system service at IPL IPL$ASTDEL.
                        0000   107 ;
                        0000   108 ;  This driver uses a buffered I/O scheme for passing a list of commands from
                        0000   109 ;  the user process to the driver. This routine performs the standard functions
                        0000   110 ;  required for buffered I/O operations. The flow is as follows:
                        0000   111 ;
                        0000   112 ;       - Check privilege of user process (DIAGNOSE)
                        0000   113 ;       - Check that user buffer doesn't exceed 64K
                        0000   114 ;       - Check user process quotas for the buffered I/O operation
                        0000   115 ;       - Allocate a system buffer from non-paged pool
                        0000   116 ;       - Set up the system buffer for a standard buffered I/O operation
                        0000   117 ;       - Copy the user buffer contents into the system buffer
                        0000   118 ;       - Initialize some driver internal state in preparation for the I/O
                        0000   119 ;       - Check for basic validity of the commands in the user buffer
                        0000   120 ;
                        0000   121 ;       - When all commands have been processed, the SYS$QIO system service
                        0000   122 ;         code is reentered to queue a request to the driver's STARTIO code
                        0000   123 ;         (JMP EXE$QIODRVPKT).
                        0000   124 ;       - If an error is found, processing is stopped and this command list
                        0000   125 ;         is returned to the user (JMP EXE$FINISHIO).
                        0000   126 ;
                        0000   127 ;  NOTE!! - Any errors found at FDT time cause the I/O to be aborted. This
                        0000   128 ;            helps simplify the control process' job.
                        0000   129 ;
                        0000   130 ;  INPUTS:
                        0000   131 ;
                        0000   132 ;       R0-R2           - Scratch
                        0000   133 ;       R3              - Address of the IRP (I/O request packet)
                        0000   134 ;       R4              - Address of the PCB (process control block)
                        0000   135 ;       R5              - Address of the UCB (unit control block)
                        0000   136 ;       R6              - Address of the CCB (channel control block)
                        0000   137 ;       R7              - Bit # of the I/O function code
                        0000   138 ;       R8              - Address of the FDT table entry for this code
                        0000   139 ;       R9              - Scratch
                        0000   140 ;       P1(AP)          - User process address of command/response buffer
                        0000   141 ;       P2(AP)          - Size (in bytes) of command/response buffer
                        0000   142 ;       P3(AP)          - Current version of diagnostice program
                        0000   143 ;
                        0000   144 ;       IPL             - IPL$_ASTDEL
                        0000   145 ;
                        0000   146 ;  OUTPUTS:
                        0000   147 ;
                        0000   148 ;       R0              - Status
                        0000   149 ;       R3-R8           - Preserved
                        0000   150 ;
                        0000   151 ;-
                        0000   152
                        0000   153 CX_STARTDATA::
                        0000   154
                        0000   155 ;  Save some registers which may get destroyed during the preprocessing of the
                        0000   156 ;  commands in the command buffer.
                        0000   157
```

```
                                0029   183 ;  buffer does not exceed 64K.
                                0029   184
                                0029   185          CMPL    BUFF_LEN(AP),-          ; Does buffer size exceed 64K?
0000FFF4 8F                     002C   186                  #<^X10000-^D12>
                                0031   187          BGTRW   BAD_LENGTH              ; Yes, error, branch
                                0036   188
                                0036   189 ;  Check the user's access rights to the buffer. If the user has no write
                                0036   190 ;  access EXE$READCHK completes the I/O with the appropriate error status.
                                0036   191
    50    6C    D0                0036   192          MOVL    BUFF_AD(AP),R0          ; Get PVA of user buffer
 51    04 AC    3C                0039   193          MOVZWL  BUFF_LEN(AP),R1         ; Get LENGTH of user buffer
00000000'GF    16                0043   194          JSB     G^EXE$READCHK           ; Check WRITE access to buffer
                                0043   195
                                0043   196 ;  Check that the processes buffered byte count quota is not exceeded by this
                                0043   197 ;  I/O. If it is, the process is put in a resource wait state. The MAXBUF
                                0043   198 ;  parameter is NOT tested.
                                0043   199
    51    0C    A0                0043   200          ADDW2   #12,R1                  ; Get total buffered I/O size
          53    DD                0046   201          PUSHL   R3                      ; Save R3
00000000'GF    16                0048   202          JSB     G^EXE$BUFQUOPRC         ; Check process quotas
          53 8EDO                 004E   203          POPL    R3                      ; Restore R3
                                0051   204          BLBCW   R0,FDT_ABORT            ; Exceeded quota, branch
                                0057   205
                                0057   206 ;  Allocate a system buffer for performing the buffered I/O. The size needed
                                0057   207 ;  for the system buffer includes the 12 bytes of VMS header. Then clear the
                                0057   208 ;  entire system buffer except for the VMS header.
                                0057   209
          53    DD                0057   210          PUSHL   R3                      ; Save reg
00000000'GF    16                0059   211          JSB     G^EXE$ALLOCBUF          ; Attempt to allocate system buffer
          53 8EDO                 005F   212          POPL    R3                      ; Restore reg
                                0062   213          BLBCW   R0,FDT_ABORT            ; Error, abort the I/O
          06    BB                0068   214          PUSHR   #^M<R1,R2>              ; Save regs
```

```
      52    0C  C0  006A            ADDL    #12,R2                    ; Advance past VMS header
      51    0C  C2  006D            SUBL    #12,R1                    ; Length without VMS header
          03BC  30  0070            BSBW    CLEAR_BUFFER              ; Clear out buffer except VMS header
            06  BA  0073            POPR    #^M<R1,R2>                ; Restore registers
                    0075
                    0075    ; The system buffer has been successfull allocated. Fill in the appropriate
                    0075    ; fields in the IRP.
                    0075
   2C A3    52  D0  0075            MOVL    R2,IRP$L_SVAPTE(R3)       ; Write system buffer address to SVAPTE
   30 A3    51  B0  0079            MOVW    R1,IRP$W_BOFF(R3)         ; Write buffer length to BOFF
50  0080    C4  D0  007D            MOVL    PCB$L_JIB(R4),R0          ; Get JIB address
   20 A0    51  C2  0082            SUBL2   R1,JIB$L_BYTCNT(R0)       ; Subtract buffer size from byte count
                A8  0086            BISW2   #<IRP$M_BUFIO!-           ; Indicate buffered I/O,
                    0087                    IRP$M_FUNC>,-             ; read operation,
   2A A3    03      0087                    IRP$W_STS(R3)             ; in status field
                    008A
                    008A    ; Fill in the system buffer appropriately. This includes copying the contents
                    008A    ; of the user buffer into the system buffer.
                    008A
   62    0C A2  DE  008A            MOVAL   12(R2),(R2)               ; Address of system buffer data area
   04 A2    6C  D0  008E            MOVL    BUFF_AD(AP),4(R2)         ; User buffer address
            3C  BB  0092            PUSHR   #^M<R2,R3,R4,R5>          ; Save regs
      04 AC     28  0094            MOVC3   BUFF_LEN(AP),-            ; Copy the contents of the user buffer
0C A2    00 BC      0097                    @BUFF_AD(AP),12(R2)       ; into the system buffer
            3C  BA  009B            POPR    #^M<R2,R3,R4,R5>          ; Restore regs
                    009D
                    009D    ; The system buffer is divided up into command and response portions. Save
                    009D    ; the starting and ending address of the command and response buffers in
                    009D    ; driver space.
                    009D
      52    62  D0  009D            MOVL    (R2),R2                   ; Get address of command buffer
 0000'CF    52  D0  00A0            MOVL    R2,CX$L_CMDBUFBGN         ; Save it
      04 A2     C1  00A5            ADDL3   CDC$W_RSPBUFOFSET(R2),- ; Calculate end of command buffer AND
 0000'CF    52      00A8                    R2,CX$L_RSPBUFBGN         ; start of response buffer
      04 AC     C1  00AC            ADDL3   BUFF_LEN(AP),-            ; Calculate end of response buffer
 0000'CF    52      00AF                    R2,CX$L_RSPBUFEND
     0000'CF    C3  00B3            SUBL3   CX$L_RSPBUFBGN,-          ; Calculate the total size of the
     0000'CF        00B7                    CX$L_RSPBUFEND,-          ; response buffer.
     0000'CF        00BA                    CX$L_RSPCNTREM
                    00BD
                    00BD    ; Initiate various driver internal storage locations in preparation for
                    00BD    ; performing the I/O.
                    00BD
        037B    30  00BD            BSBW    ALLOC_SEQNUM              ; Get a sequence number
     0001'CF    B0  00C0            MOVW    1+CX$L_SEQNUM,-           ; Copy it to rsp seq num
     0002'CF        00C4                    2+CX$L_RSPSEQNUM
     0000'CF    D4  00C7            CLRL    CX$L_NXT_RSP              ; Start rsp buffer offsets at 0
     0000'CF    D4  00CB            CLRL    CX$B_CMDERRCNT            ; Start counting errors at 0
53   0000'CF    D0  00CF            MOVL    CX$L_CMDBUFBGN,R3         ; Recover start of cmd buffer
       01 A3    B4  00D4            CLRW    CDC$B_DONECNT(R3)         ; Clear counter bytes
       03 A3    94  00D7            CLRB    CDC$B_SIOCNT(R3)          ; Clear SIO count
      53    0C  C0  00DA            ADDL2   #CDC$K_CMDHDRSIZ,R3       ; Step over CDC header
 0000'CF    53  D0  00DD            MOVL    R3,CX$L_CUR_CMD           ; Set pointer to current cmd
                    00E2
                    00E2
                    00E2
```

```
                                00E2    272         .SBTTL  FDT_DISPATCH,   Check validity of individual commands
                                00E2    273 ;+
                                00E2    274 ;
                                00E2    275 ;  This section of code checks the basic validity of the commands in the
                                00E2    276 ;  command buffer. Any errors found case the I/O to be aborted.
                                00E2    277 ;
                                00E2    278 ;-
                                00E2    279
                                00E2    280 FDT_DISPATCH:
                                00E2    281
                    01 A3   91  00E2    282         CMPB    CDC$B_DSTPORT(R3),-     ; Valid destination node field?
                    00'8F       00E5    283                 #MAX_NODES              ;
                                00E7    284         BGTRUW  BAD_DESTINATION         ; Error, branch
                                00EC    285
                     0239   30  00EC    286         BSBW    CHECK_LOC_PORT          ; Check for valid local port name
                                00EF    287         BLBCW   R0,BAD_LOCALPORT        ; Error, branch
                                00F5    288
                    08 A3   B5  00F5    289 20$:    TSTW    CDC$W_TXTLEN(R3)        ; Valid text length?
                                00F8    290         BLEQUW  BAD_CMDLEN              ; No, branch
          50    08 A3   53   C1 00FD    291         ADDL3   R3,CDC$W_TXTLEN(R3),R0  ; Reasonable length in cmd?
              0000'CF   50   D1 0102    292         CMPL    R0,CX$L_CMDBUFEND       ;
                                0107    293         BGTRUW  BAD_CMDLEN              ; No, branch
                                010C    294
                                010C    295 30$:    $DISPATCH -                             ; Dispatch off the command
                                010C    296                 CDC$B_OPCODE(R3),TYPE=B,-       ; specific opcode
                                010C    297                 <-
                                010C    298                 <CDC$SENDMSG,FDT_SENDMSG>,-     ; Send message
                                010C    299                 <CDC$SENDDG,FDT_SENDDG>,-       ; Send datagram
                                010C    300                 <CDC$SENDDATA,FDT_SNDDAT>,-     ; Send date
                                010C    301                 <CDC$REQDATA,FDT_REQDAT>,-      ; Request data
                                010C    302                 <CDC$MAPBUF,FDT_MAP>,-          ; Map a buffer
                                010C    303                 <CDC$UNMAPBUF,FDT_UNMAP>,-      ; Unmap a buffer
                                010C    304                 <CDC$CONFIGDATA,FDT_CONFIG>,-   ; Gather configuration data
                                010C    305                 <CDC$CONNECT,FDT_CONNECT>,-     ; Connect
                                010C    306                 <CDC$DISCONNECT,FDT_DISCONNECT>,- ; Disconnect
                                010C    307                 <CDC$READCNT,FDT_RDCNT>,-       ; Read the performance counters
                                010C    308                 <CDC$GARBAGE,FDT_GARBAGE>,-     ; Get garbage
                                010C    309                 <CDC$RESET,FDT_RESET>,-         ; Send reset
                                010C    310                 <CDC$START,FDT_START>,-         ; Send start
                                010C    311                 <CDC$POLLER,FDT_POLLER>,-       ; Turn poller on/off
                                010C    312                 <CDC$MAINTSTATE,FDT_MAINT_STATE>,- ; Set port to maint state
                                010C    313                 <CDC$MAPMBUF,FDT_MNTMAP>,-      ; Map maintenance buffer
                                010C    314                 <CDC$UNMAPMBUF,FDT_NEXT_CMD>,-  ; Unmap maintenance buffer
                                010C    315                 <CDC$SENDMDATA,FDT_SNDMDAT>,-   ; Send maintenance data
                                010C    316                 <CDC$REQMDATA,FDT_REQMDAT>,-    ; Request maintenance data
                                010C    317                 <CDC$SELADR,FDT_SELADR>,-       ; Select port address
                                010C    318                 <CDC$NORSP,FDT_NORSP>,-         ; Arm CINT NORSP logic
                                010C    319                 <CDC$NAK,FDT_NAK>,-             ; Arm CINT NAK logic
                                010C    320                 <CDC$NOACK,FDT_NOACK>,-         ; Arm CINT NOACK logic
                                010C    321                 <CDC$CARRIER,FDT_CARRIER>,-     ; Arm CINT CARRIER logic
                                010C    322                 <CDC$LRGPACKET,FDT_LRGPACKET>,- ; Arm CINT LRGPACKET logic
                                010C    323                 <CDC$INITCINT,FDT_INITCINT>,-   ; Inititalize CINT logic
                                010C    324                 <CDC$ARMCINTPATH,FDT_ARMPATH>,- ; Arm CINT path select logic
                                010C    325                 <CDC$RDCINTPATH,FDT_READPATH>,- ; Read CINT received path
                                010C    326                 <CDC$OVERSIZEPKT,FDT_OVERSIZEPKT>,- ; Arm CINT oversized packet logi
                                010C    327                 >
                                014A    328
```

```
                        014A    329 ;  If the command contains an opcode not listed in the dispatch table listed
                        014A    330 ;  above, or an opcode that is currently unsupported, abort the I/O with the
                        014A    331 ;  appropriate status.
                        014A    332
                        014A    333 FDT_INV_OPCODE:
                        014A    334 FDT_UNSOPC:
                        014A    335
                  65 11 014A    336         BRB     BAD_CMDOPC              ; Abort the I/O with appropriate status
                        014C    337
                        014C    338
                        014C    339 ;  The following commands do not require any FDT processing. Fall through to
                        014C    340 ;  start preprocessing of the next command.
                        014C    341
                        014C    342 FDT_GARBAGE:
                        014C    343 FDT_READPATH:
                        014C    344 FDT_ARMPATH:
                        014C    345 FDT_INITCINT:
                        014C    346 FDT_MAINT_STATE:
                        014C    347 FDT_LRGPACKET:
                        014C    348 FDT_OVERSIZEPKT:
                        014C    349
                        014C    350 ;  FDT_NEXT_CMD
                        014C    351 ;
                        014C    352 ;  Calculate the address of the next command in the buffer, update various
                        014C    353 ;  states, check if we are done. If done, complete this FDT by JMPing to
                        014C    354 ;  EXE$QIODRVPKT which will bring the IPL up to FORK level and enter our
                        014C    355 ;  STARTIO code.
                        014C    356
                        014C    357 FDT_NEXT_CMD:
                        014C    358
53 0000'CF    D0        014C    359         MOVL    CX$L_CUR_CMD,R3         ; Reset current cmd ptr
51 0000'CF    D0        0151    360         MOVL    CX$L_CMDBUFBGN,R1       ; Point to cmd counter bytes
     02 A1    96        0156    361         INCB    CDC$B_FDTCNT(R1)        ; Increment FDT complete count
61   02 A1    91        0159    362         CMPB    CDC$B_FDTCNT(R1),(R1)   ; FDT complete?
        14    13        015D    363         BEQLU   20$                     ; Yes, branch
                        015F    364
53 0000'CF    D0        015F    365         MOVL    CX$L_CUR_CMD,R3         ; Reset current cmd ptr
50   08 A3    3C        0164    366         MOVZWL  CDC$W_TXTLEN(R3),R0     ; Get length of current command
   53   50    C0        0168    367         ADDL2   R0,R3                   ; Index to next command
0000'CF   53    D0      016B    368         MOVL    R3,CX$L_CUR_CMD         ; Save the pointer
       FF6F    31       0170    369         BRW     FDT_DISPATCH            ; Continue processing
                        0173    370
                        0173    371 ;  See if all commands are done. It is possible to complete a command buffer
                        0173    372 ;  without doing any start I/O.
                        0173    373
61   01 A1    91        0173    374 20$:    CMPB    CDC$B_DONECNT(R1),(R1)  ; Are we ALL done?
        08    12        0177    375         BNEQU   30$                     ; No, branch
   50   01    3C        0179    376         MOVZWL  #SS$_NORMAL,R0          ; Set success status
        51    D4        017C    377         CLRL    R1                      ;
      0047    31        017E    378         BRW     FDT_ABORT               ; Finish the I/O
                        0181    379
                        0181    380 ;  We are done with the FDT preprocessing. EXE$QIODRVPKT will now queue the
                        0181    381 ;  IRP for the driver, and since there is only one one request, STARTIO will
                        0181    382 ;  be entered immediately.
                        0181    383
53 0000'CF    7D        0181    384 30$:    MOVQ    CX$X_IRPSAV,R3          ; Restore R3,R4,R5
55 0008'CF    D0        0186    385         MOVL    CX$X_IRPSAV+8,R5        ;
```

```
                00000000'GF   17  018B   386          JMP     G^EXE$QIODRVPKT         ; Go start the I/O
                              0191   387
                              0191   388
                              0191   389
                              0191   390
```

```
                           0191            .SBTTL  FDT_ABORT,      Abort the I/O for various reasons
                           0191       ;+
                           0191       ;
                           0191       ;   FDT_ABORT
                           0191       ;
                           0191       ;   Any errors found at FDT time cause the I/O to be aborted. The appropriate
                           0191       ;   status is returned in R0 and R1.
                           0191       ;
                           0191       ;   The I/O can be finished (successfully) through this path if there is no
                           0191       ;   work to be done at start I/O time. This is the case for the maintenance
                           0191       ;   buffer map command.
                           0191       ;
                           0191       ;-
                           0191
                           0191       BAD_PRIV:                                       ; User has insufficient privilege
                           0191
          50    24     3C  0191            MOVZWL  #SS$_NOPRIV,R0                  ; Set appropriate status
                51     D4  0194            CLRL    R1                              ;
                30     11  0196            BRB     FDT_ABORT                       ; Abort the I/O
                           0198
                           0198       VERSION_MISMATCH:                               ; Mismatch of user process and driver
                           0198
   50    00000334 8F   D0  0198            MOVL    #SS$_DEVREQERR,R0               ; Indicate device error
          51    06     D0  019F            MOVL    #CX$_CDVERSION,R1               ; Version mismatch
                24     11  01A2            BRB     FDT_ABORT
                           01A4
                           01A4       FDT_PWRFAIL:                                    ; Power fail occurred
                           01A4
          0000'CF      94  01A4            CLRB    CX$B_PWRFAIL                    ; Clear power fail flag
     50   0364 8F      3C  01A8            MOVZWL  #SS$_POWERFAIL,R0               ; Set appropriate error status
                51     D4  01AD            CLRL    R1                              ;
                17     11  01AF            BRB     FDT_ABORT                       ; Abort the I/O
                           01B1
                           01B1
                           01B1       BAD_LENGTH:                                     ; User buffer is too large
                           01B1       BAD_DESTINATION:                                ; Invalid remote port number
                           01B1       BAD_LOCALPORT:                                  ; Invalid local port name
                           01B1       BAD_CMDLEN:                                     ; Invalid command text length
                           01B1       BAD_CMDOPC:                                     ; Invalid command opcode
                           01B1       BAD_RSPLEN:                                     ; Invalid response buffer length
                           01B1       BAD_BUFNAME:                                    ; Invalid buffer name
                           01B1       BAD_SYSID:                                      ; Invalid system ID
                           01B1
          50    14     3C  01B1            MOVZWL  #SS$_BADPARAM,R0                ; Set appropriate status
                51     D4  01B4            CLRL    R1                              ; No additional class driver status
                10     11  01B6            BRB     FDT_ABORT                       ; Abort the I/O
                           01B8
                           01B8       FDT_CONN_ERR:                                   ; Connection was lost
                           01B8
                           01B8            SET_CMD_FLAG CONN_LOST                  ; Tell user where error occurred
                08     11  01BE            BRB     FDT_ABORT                       ; Abort the I/O
                           01C0
                           01C0       FDT_CONN_ERR1:                                  ; Connection is already active
                           01C0
     50   0334 8F      3C  01C0            MOVZWL  #SS$_DEVREQERR,R0               ; Set system error status
          51    03     3C  01C5            MOVZWL  #CX$_CONN_ACTV,R1               ; Indicate connection was active
                           01C8
```

```
                        01C8    449 FDT_ABORT:
                        01C8    450
53   0000'CF    7D      01C8    451         MOVQ    CX$X_IRPSAV,R3          ; Restore IRP, PCB and UCB
55   0008'CF    D0      01CD    452         MOVL    CX$X_IRPSAV+8,R5        ; address
     0000'CF    D4      01D2    453         CLRL    CX$L_STATE              ; Wipe the state clean
 00000000'GF    17      01D6    454         JMP     G^EXE$FINISHIO          ; Abort the I/O
                        01DC    455
                        01DC    456
                        01DC    457
                        01DC    458
```

```
                        01DC    460          .SBTTL  FDT_CONFIG,     Gather configuration data
                        01DC    461 ;+
                        01DC    462 ;
                        01DC    463 ;  FDT_CONFIG
                        01DC    464 ;
                        01DC    465 ;  This routine preprocesses a CONFIG class driver command. The flow is
                        01DC    466 ;  as follows:
                        01DC    467 ;
                        01DC    468 ;       - Allocate response buffer space
                        01DC    469 ;       - Dispatch the next command
                        01DC    470 ;
                        01DC    471 ;  INPUTS:
                        01DC    472 ;
                        01DC    473 ;       R3              - Address of current command
                        01DC    474 ;
                        01DC    475 ;  OUTPUTS:
                        01DC    476 ;
                        01DC    477 ;       R0-R1           - Destroyed
                        01DC    478 ;
                        01DC    479 ;-
                        01DC    480
                        01DC    481 FDT_CONFIG:
                        01DC    482
          0173    30    01DC    483          BSBW    ALLOC_RSP_SCS          ; Allocate some resp buf space
                        01DF    484          BLBCW   R0,BAD_RSPLEN          ; Error, branch
                        01E5    485
          FF64    31    01E5    486          BRW     FDT_NEXT_CMD
                        01E8    487
                        01E8    488
                        01E8    489
                        01E8    490
```

```
                                01E8    492          .SBTTL  FDT_CONNECT,    Establish connection
                                01E8    493 ;+
                                01E8    494 ;
                                01E8    495 ;  FDT_CONNECT
                                01E8    496 ;
                                01E8    497 ;  This routine preprocesses a connect class driver command. The steps
                                01E8    498 ;  taken are:
                                01E8    499 ;
                                01E8    500 ;       - Allocate response save space
                                01E8    501 ;       - Check for non-zero system block address
                                01E8    502 ;       - Check for the existance of active connection
                                01E8    503 ;       - Go dispatch next class driver command
                                01E8    504 ;
                                01E8    505 ; INPUTS:
                                01E8    506 ;
                                01E8    507 ;       R3              - Address of current command
                                01E8    508 ;
                                01E8    509 ; OUTPUTS:
                                01E8    510 ;
                                01E8    511 ;       R0-R1           - Destroyed
                                01E8    512 ;
                                01E8    513 ;-
                                01E8    514
                                01E8    515 FDT_CONNECT:
                                01E8    516
                     0167   30  01E8    517          BSBW    ALLOC_RSP_SCS          ; Allocate some resp buf space
                                01EB    518          BLBCW   R0,BAD_RSPLEN          ; Error, branch
                                01F1    519
                 34 A3    D5  01F1    520          TSTL    CDC$T_SYSTEMID(R3)     ; Check l.o. 32 bits for zero
                    08    12  01F4    521          BNEQU   10$                    ; No, branch
                 38 A3    B5  01F6    522          TSTW    4+CDC$T_SYSTEMID(R3)   ; Check h.o 16 bits
                                01F9    523          BEQLW   BAD_SYSID              ; Illegal system ID, branch
                                01FE    524
                   01B0   30  01FE    525 10$:     BSBW    CHECK_CONN             ; Check for already existant connection
                                0201    526          BLBSW   R0,FDT_CONN_ERR1       ; Error, branch
                                0207    527
                   FF42   31  0207    528          BRW     FDT_NEXT_CMD
                                020A    529
                                020A    530
                                020A    531
```

```
                        020A    533          .SBTTL  FDT_DISCONNECT, Disconnect routine
                        020A    534 ;+
                        020A    535 ;
                        020A    536 ;  FDT_DISCONNECT
                        020A    537 ;
                        020A    538 ;  This routine does all the preprocessing for a disconnect command. The
                        020A    539 ;  steps are:
                        020A    540 ;
                        020A    541 ;       - Allocate response buffer space
                        020A    542 ;       - Check connection state
                        020A    543 ;       - Go dispatch the next command
                        020A    544 ;
                        020A    545 ;  INPUTS:
                        020A    546 ;
                        020A    547 ;       R3              - Address of current command
                        020A    548 ;
                        020A    549 ;  OUTPUTS:
                        020A    550 ;
                        020A    551 ;       R0-R2           - Destroyed
                        020A    552 ;
                        020A    553 ;-
                        020A    554
                        020A    555 FDT_DISCONNECT:
                        020A    556
0145    30              020A    557          BSBW    ALLOC_RSP_SCS          ; Allocate some resp buf space
                        020D    558          BLBCW   R0,BAD_RSPLEN          ; Error, branch
                        0213    559
01D5    30              0213    560          BSBW    CHECK_ACTV_CONN        ; Check for an active connection
                        0216    561          BLBCW   R0,FDT_CONN_ERR        ; No connection, error, branch
                        021C    562
FF2D    31              021C    563          BRW     FDT_NEXT_CMD
                        021F    564
                        021F    565
                        021F    566
                        021F    567
                        021F    568
```

```
                        021F    570          .SBTTL FDT_SENDMSG,     Send message
                        021F    571 ;+
                        021F    572 ;
                        021F    573 ;  FDT_SENDMSG
                        021F    574 ;
                        021F    575 ;  Routine to do all the preprocessing needed to send a message. This includes
                        021F    576 ;  the following steps:
                        021F    577 ;
                        021F    578 ;       - Allocate response-save area
                        021F    579 ;       - Check for existence of a connection to remote process
                        021F    580 ;       - Fill in the CTP command reference number
                        021F    581 ;       - Dispatch to next command
                        021F    582 ;
                        021F    583 ;  INPUTS:
                        021F    584 ;
                        021F    585 ;       R3              - Address of current command
                        021F    586 ;
                        021F    587 ;  OUTPUTS:
                        021F    588 ;
                        021F    589 ;       R0-R2           - Destroyed
                        021F    590 ;
                        021F    591 ;-
                        021F    592
                        021F    593 FDT_SENDMSG:
                        021F    594
                013A    30      021F    595          BSBW    ALLOC_RSP_CTP           ; Allocate some resp buf space
               8C 50    E9      0222    596          BLBC    R0,BAD_RSPLEN           ; Error, branch
                        0225    597
                01C3    30      0225    598          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
               8D 50    E9      0228    599          BLBC    R0,FDT_CONN_ERR         ; No connection, error, branch
                        022B    600
                0175    30      022B    601          BSBW    FILL_CMD_REF            ; Fill in CTP command reference number
                        022E    602
                FF1B    31      022E    603          BRW     FDT_NEXT_CMD            ; Go dispatch the next command
                        0231    604
                        0231    605
                        0231    606
```

```
                        0231    608         .SBTTL FDT_SENDDG,      Send datagram
                        0231    609 ;+
                        0231    610 ;
                        0231    611 ;  FDT_SENDDG
                        0231    612 ;
                        0231    613 ;  This routine will do all the preprocessing needed to send a datagram.
                        0231    614 ;  The flow is:
                        0231    615 ;
                        0231    616 ;       - Allocate response-save area
                        0231    617 ;       - Check for existence of a connection to remote process
                        0231    618 ;       - Fill in the CTP command reference number
                        0231    619 ;       - Go dispatch the next command
                        0231    620 ;
                        0231    621 ;  INPUTS:
                        0231    622 ;
                        0231    623 ;       R3              - Address of current command
                        0231    624 ;
                        0231    625 ;  OUTPUTS:
                        0231    626 ;
                        0231    627 ;       R0-R2           - Destroyed
                        0231    628 ;
                        0231    629 ;-
                        0231    630
                        0231    631 FDT_SENDDG:
                        0231    632
            0128    30  0231    633         BSBW    ALLOC_RSP_CTP           ; Allocate some resp buf space
                        0234    634         BLBCW   R0,BAD_RSPLEN           ; Error, branch
                        023A    635
            01AE    30  023A    636         BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        023D    637         BLBCW   R0,FDT_CONN_ERR         ; No connection, error, branch
                        0243    638
            015D    30  0243    639         BSBW    FILL_CMD_REF            ; Fill in CTP command reference number
                        0246    640
            FF03    31  0246    641         BRW     FDT_NEXT_CMD            ; Go dispatch the next command
                        0249    642
                        0249    643
                        0249    644
```

```
                                0249   646          .SBTTL  FDT_MAP,        Map buffer
                                0249   647 ;+
                                0249   648 ;
                                0249   649 ;  FDT_MAP
                                0249   650 ;
                                0249   651 ;  This routine does all preprocessing needed to do an SCS MAP call. The
                                0249   652 ;  flow is:
                                0249   653 ;
                                0249   654 ;       - Allocate response buffer space
                                0249   655 ;       - Check for connection to remote process
                                0249   656 ;       - Calculate the system virtual address of the first page table
                                0249   657 ;         entry for the buffer and write it the command buffer
                                0249   658 ;
                                0249   659 ;  INPUTS:
                                0249   660 ;
                                0249   661 ;       R3              - Address of current command
                                0249   662 ;
                                0249   663 ;  OUTPUTS:
                                0249   664 ;
                                0249   665 ;       R0-R3           - Destroyed
                                0249   666 ;
                                0249   667 ;-
                                0249   668
                                0249   669 FDT_MAP:
                                0249   670
                      0106   30 0249   671          BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                                024C   672          BLBCW   R0,BAD_RSPLEN           ; Error, branch
                                0252   673
                      0196   30 0252   674          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                                0255   675          BLBCW   R0,FDT_CONN_ERR         ; No connection, error, branch
                                025B   676
                         53    DD 025B   677          PUSHL   R3                      ; Save command address
                    52    24 A3    DO 025D   678          MOVL    CDC$L_BUFADR(R3),R2     ; Get address of buffer to map
                54    0004'CF    DO 0261   679          MOVL    CX$X_IRPSAV+4,R4        ; Restore PCB address
                  55    6C A4    DO 0265   680          MOVL    PCB$L_PHD(R4),R5        ; Get PHD address
                                026A   681          DSBINT  #IPL$_SYNCH             ; Raise IPL to synch for call
            00000000'GF    16 0274   682          JSB     G^MMG$PTEADRCHK         ; Calculate the SVAPTE for buffer
                                027A   683          ENBINT                          ; Restore IPL
                         50  8ED0 027D   684          POPL    R0                      ; Restore CDRP address
                18 A0    53    DO 0280   685          MOVL    R3,CDC$L_SVAPTE(R0)     ; Copy SVAPTE to command buffer
                                0284   686
                      FEC5   31 0284   687          BRW     FDT_NEXT_CMD
                                0287   688
                                0287   689
                                0287   690
```

```
                                0287   692          .SBTTL  FDT_MNTMAP,     Map a maintenance buffer
                                0287   693 ;+
                                0287   694 ;
                                0287   695 ;  FDT_MNTMAP
                                0287   696 ;
                                0287   697 ;  This routine performs the FDT processing for a maintenance buffer map
                                0287   698 ;  by doing the following things:
                                0287   699 ;
                                0287   700 ;       - Allocate response buffer space
                                0287   701 ;       - Calculate the physical address of the user buffer
                                0287   702 ;       - Mark the command done (no START I/O needed)
                                0287   703 ;       - Dispatch to next command
                                0287   704 ;
                                0287   705 ;  INPUTS:
                                0287   706 ;
                                0287   707 ;       R3                      - Address of current command
                                0287   708 ;
                                0287   709 ;  OUTPUTS:
                                0287   710 ;
                                0287   711 ;       R0-R2                   - Destroyed
                                0287   712 ;
                                0287   713 ;-
                                0287   714
                                0287   715 FDT_MNTMAP:
                                0287   716
                      0008   30 0287   717          BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                                028A   718          BLBCW   R0,BAD_RSPLEN           ; Error, branch
                                0290   719
                        52    24 A3   D0  0290   720          MOVL    CDC$L_BUFADR(R3),R2     ; Get address of buffer to map
                                52    DD  0294   721          PUSHL   R2                      ; Save buffer address
                          54  0004'CF   D0  0296   722          MOVL    CX$X_IRPSAV+4,R4        ; Restore PCB address
                          55    6C A4   D0  029B   723          MOVL    PCB$L_PHD(R4),R5        ; Get PHD address
                                029F   724          DSBINT  #IPL$_SYNCH             ; Raise IPL to synch for call
                   00000000'GF    16  02A9   725          JSB     G^MMG$PTEADRCHK         ; Calculate the SVAPTE for buffer
                                02AF   726          ENBINT                          ; Restore IPL
                            52 8ED0   02B2   727          POPL    R2                      ; Restore buffer address
                                02B5   728
                50  52    09   00  EE  02B5   729          EXTV    #0,#9,R2,R0             ; Extract the byte offset from user buf
                51  63    15   00  EE  02BA   730          EXTV    #0,#21,(R3),R1          ; Extract the PFN from the PTE
                    51    51   09  78  02BF   731          ASHL    #9,R1,R1                ; Shift the PFN
                    53  0000'CF   D0  02C3   732          MOVL    CX$L_CUR_CMD,R3         ; Reset current cmd ptr
                                02C8   733          CALC_RSP DST=R2                 ; Get response slot address
                24  A2    51   50  C9  02D1   734          BISL3   R0,R1,CDC$L_BUFLNAME(R2); Form the physical address of user buf
                      0C  A2    01  D0  02D6   735          MOVL    #1,CDC$L_R0STATUS(R2)   ; Set success R0 status in response
                      10  A2    01  D0  02DA   736          MOVL    #1,CDC$L_R1STATUS(R2)   ; Set success R1 status in response
                                02DE   737
                    51  0000'CF   D0  02DE   738          MOVL    CX$L_CMDBUFBGN,R1       ; Get starting addr of command buffer
                          01  A1   96  02E3   739          INCB    CDC$B_DONECNT(R1)       ; One more command done
                                02E6   740          SET_CMD_FLAG CMD_DONE           ; Set DONE bit for this command
                                02EC   741
                                02EC   742
                            FE5D   31  02EC   743          BRW     FDT_NEXT_CMD
                                02EF   744
                                02EF   745
                                02EF   746
```

```
                        02EF    748          .SBTTL  FDT_UNMAP,      Unmap buffer
                        02EF    749 ;+
                        02EF    750 ;
                        02EF    751 ;  FDT_UNMAP
                        02EF    752 ;
                        02EF    753 ;  This routine performs FDT preprocessing for the buffer unmap command.
                        02EF    754 ;  The flow is as follows:
                        02EF    755 ;
                        02EF    756 ;       - Allocate response buffer . ace
                        02EF    757 ;       - Locate the mapped buffer CDRP on the map list
                        02EF    758 ;       - Dispatch the next command
                        02EF    759 ;
                        02EF    760 ;  INPUTS:
                        02EF    761 ;
                        02EF    762 ;       R3                          - Address of current command
                        02EF    763 ;
                        02EF    764 ;  OUTPUTS:
                        02EF    765 ;
                        02EF    766 ;       R0-R2                       - Destroyed
                        02EF    767 ;
                        02EF    768 ;-
                        02EF    769
                        02EF    770 FDT_UNMAP:
                        02EF    771
          0060    30    02EF    772          BSBW    ALLOC_RSP_SCS          ; Allocate some resp buf space
                        02F2    773          BLBCW   R0,BAD_RSPLEN          ; Error, branch
                        02F8    774
          FE51    31    02F8    775          BRW     FDT_NEXT_CMD           ; Dispatch to next command
                        02FB    776
                        02FB    777
                        02FB    778
                        02FB    779
```

```
                        02FB   781          .SBTTL  FDT_SNDDAT,     Send data
                        02FB   782          .SBTTL  FDT_REQDAT,     Request data
                        02FB   783 ;+
                        02FB   784 ;
                        02FB   785 ;  FDT_SNDDAT
                        02FB   786 ;  FDT_REQDAT
                        02FB   787 ;
                        02FB   788 ;  This routine preprocesses a SEND_DATA class driver command. The steps taken
                        02FB   789 ;  are:
                        02FB   790 ;
                        02FB   791 ;       - Allocate response buffer space
                        02FB   792 ;       - Check for a connection to the remote process
                        02FB   793 ;       - Dispatch the next command
                        02FB   794 ;
                        02FB   795 ;  INPUTS:
                        02FB   796 ;
                        02FB   797 ;       R3                          - Address of current command
                        02FB   798 ;
                        02FB   799 ;  OUTPUTS:
                        02FB   800 ;
                        02FB   801 ;-
                        02FB   802
                        02FB   803 FDT_SNDDAT:
                        02FB   804 FDT_REQDAT:
                        02FB   805
                0054 30 02FB   806          BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                        02FE   807          BLBCW   R0,BAD_RSPLEN           ; Error, branch
                        0304   808
                00E4 30 0304   809          BSBW    CHECK_ACTV_CONN         ; Check for an active connection
                        0307   810          BLBCW   R0,FDT_CONN_ERR          ; No connection, error, branch
                        030D   811
                FE3C 31 030D   812          BRW     FDT_NEXT_CMD
                        0310   813
                        0310   814
                        0310   815
```

```
                     0310   817          .SBTTL  FDT_SNDMDAT,    Send maintenance data
                     0310   818          .SBTTL  FDT_REQMDAT,    Request maintenance data
                     0310   819 ;+
                     0310   820 ;
                     0310   821 ;  FDT_SNDMDAT
                     0310   822 ;  FDT_REQMDAT
                     0310   823 ;
                     0310   824 ;  Routines to send and request maintencance buffer transfers. The local
                     0310   825 ;  mapped buffer is the standard buffer type, wheras the remote buffer
                     0310   826 ;  handle is a maintenance type buffer. For maintenance buffers, the buffer
                     0310   827 ;  name is a 1, and the buffer offset is the physical address of the buffer.
                     0310   828 ;
                     0310   829 ;  INPUTS:
                     0310   830 ;
                     0310   831 ;       R3              - Address of current command
                     0310   832 ;
                     0310   833 ;  OUTPUTS:
                     0310   834 ;
                     0310   835 ;-
                     0310   836
                     0310   837 FDT_SNDMDAT:
                     0310   838 FDT_REQMDAT:
                     0310   839
                     0310   840
                003F 30 0310   841          BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                     0313   842          BLBCW   R0,BAD_RSPLEN           ; Error, branch
                     0319   843
                FE30 31 0319   844          BRW     FDT_NEXT_CMD
                     031C   845
                     031C   846
                     031C   847
                     031C   848
```

```
                        850          .SBTTL  FDT_RDCNT,      Read performance counters
                        851 ;+
                        852 ;
                        853 ;  FDT_RDCNT
                        854 ;
                        855 ;  This routine preprocesses a RDCNT class driver command. The flow is as
                        856 ;  follows:
                        857 ;
                        858 ;       - Allocate response buffer space
                        859 ;       - Allocate a CDRP to interface to SCS
                        860 ;       - Dispatch the next command
                        861 ;
                        862 ; INPUTS:
                        863 ;
                        864 ;       R3              - Address of current command
                        865 ;
                        866 ; OUTPUTS:
                        867 ;
                        868 ;       R0-R1           - Destroyed
                        869 ;
                        870 ;-
                        871
                        872 FDT_RDCNT:
                        873
                        874 ;       BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                        875 ;       BLBCW   R0,BAD_RSPLEN           ; Error, branch
                        876 ;
                        877 ;       BRW     FDT_NEXT_CMD
                        878
                        879
                        880
                        881
```

```
                        031C   883          .SBTTL  FDT_RESET,      Send reset
                        031C   884          .SBTTL  FDT_START,      Send start
                        031C   885 ;+
                        031C   886 ;
                        031C   887 ;  FDT_RESET
                        031C   888 ;  FDT_START
                        031C   889 ;
                        031C   890 ;  Routine to perform preprocessing on start and reset commands.
                        031C   891 ;
                        031C   892 ;  INPUTS:
                        031C   893 ;
                        031C   894 ;       R3                          - Address of current command
                        031C   895 ;
                        031C   896 ;  OUTPUTS:
                        031C   897 ;
                        031C   898 ;       R0-R2                       - Destroyed
                        031C   899 ;
                        031C   900 ;-
                        031C   901
                        031C   902 FDT_START:
                        031C   903 FDT_RESET:
                        031C   904
                        031C   905 ;       BSBW    ALLOC_RSP_SCS           ; Allocate some resp buf space
                        031C   906 ;       BLBC    R0,BAD_RSPLEN           ; Error, branch
                        031C   907 ;
                        031C   908 ;       BRW     FDT_NEXT_CMD
                        031C   909
                        031C   910
                        031C   911
```

```
                        031C   913          .SBTTL  FDT_POLLER,     Start/Stop poller
                        031C   914          .SBTTL  FDT_SELADR,     Change port address
                        031C   915          .SBTTL  FDT_NORSP,      Arm CINT NORSP logic
                        031C   916          .SBTTL  FDT_NAK,        Arm CINT NAK logic
                        031C   917          .SBTTL  FDT_NOACK,      Arm CINT NOACK logic
                        031C   918          .SBTTL  FDT_CARRIER,    Place carrier on wire
                        031C   919 ;+
                        031C   920 ;
                        031C   921 ;  FDT_POLLER
                        031C   922 ;  FDT_SELADR
                        031C   923 ;  FDT_NORSP
                        031C   924 ;  FDT_NAK
                        031C   925 ;  FDT_NOACK
                        031C   926 ;  FDT_CARRIER
                        031C   927 ;
                        031C   928 ;  FDT routine to handle CINT specific commands. Currently, the only thing
                        031C   929 ;  done is to allocate sufficient space in the response buffer.
                        031C   930 ;
                        031C   931 ;  INPUTS:
                        031C   932 ;
                        031C   933 ;       R3                          - Pointer to current command
                        031C   934 ;
                        031C   935 ;  OUTPUTS:
                        031C   936 ;
                        031C   937 ;       R0-R2                       - Destroyed
                        031C   938 ;-
                        031C   939
                        031C   940 FDT_POLLER:
                        031C   941 FDT_SELADR:
                        031C   942 FDT_NORSP:
                        031C   943 FDT_NAK:
                        031C   944 FDT_NOACK:
                        031C   945 FDT_CARRIER:
                        031C   946
0033    30              031C   947          BSBW    ALLOC_RSP_SCS          ; Allocate some resp buf space
                        031F   948          BLBCW   R0,BAD_RSPLEN          ; Error, branch
                        0325   949
FE24    31              0325   950          BRW     FDT_NEXT_CMD           ; Try next command
                        0328   951
                        0328   952
                        0328   953
                        0328   954
```

```
                     0328   956          .SBTTL  CHECK_LOC_PORT, Check local port name
                     0328   957 ;+
                     0328   958 ;
                     0328   959 ;  CHECK_LOC_PORT
                     0328   960 ;
                     0328   961 ;  Routine to check the validity of the local port name for a command in the
                     0328   962 ;  command buffer. The local port name is confined to be one of the following:
                     0328   963 ;
                     0328   964 ;       - First letter must be "A"
                     0328   965 ;       - Second letter must be "A" or "X"
                     0328   966 ;       - Third letter must be "A", "B", "C", etc. Number of local
                     0328   967 ;         adapters is limited by MAX_LOC_PORTS
                     0328   968 ;       - Fourth letter can be anything. Fourth letter is made into
                     0328   969 ;         a null.
                     0328   970 ;
                     0328   971 ;  INPUTS:
                     0328   972 ;
                     0328   973 ;       R3                      - Address of current command
                     0328   974 ;
                     0328   975 ;  OUTPUTS:
                     0328   976 ;
                     0328   977 ;       R0                      - Status
                     0328   978 ;
                     0328   979 ;-
                     0328   980
                     0328   981 CHECK_LOC_PORT:
                     0328   982
                     0328   983
4150 8F    10 A3  B1 0328   984          CMPW    CDC$L_LPORT(R3),#^A/PA/ ; Check first two letters for "PA"
              08  13 032E   985          BEQL    10$                     ; Valid, branch
5850 8F    10 A3  B1 0330   986          CMPW    CDC$L_LPORT(R3),#^A/PX/ ; Check first two letters for "PX"
              17  12 0336   987          BNEQ    20$                     ; Invalid name, branch
                     0338   988
                     0338   989
        41 8F     83 0338   990 10$:     SUBB3   #^X41,-                 ; Change ASCII to decimal
     50   12 A3       033B   991                  CDC$L_LPORT+2(R3),R0
              0F  1F 033E   992          BLSSU   20$                     ; Invalid name, branch
     FF'8F    50  91 0340   993          CMPB    R0,#MAX_LOC_PORTS-1     ; Port name greater than max
              09  1A 0344   994          BGTRU   20$                     ; Yes, error, branch
  11 A3   41 8F  90 0346   995          MOVB    #^A/A/,CDC$L_LPORT+1(R3); Change "PX" to "PA"
        50    01  D0 034B   996          MOVL    #1,R0                   ; Return success status
              05  05 034E   997          RSB
                     034F   998
              50  D4 034F   999 20$:     CLRL    R0                      ; Return error status
              05  05 0351  1000          RSB
                     0352  1001
                     0352  1002
                     0352  1003
```

```
                     0352  1005          .SBTTL  ALLOC_RSP_SCS,  Allocate SCS response buffer space
                     0352  1006          .SBTTL  ALLOC_RSP_CTP,  Allocate CTP response buffer space
                     0352  1007 ;+
                     0352  1008 ;
                     0352  1009 ;  ALLOC_RSP_SCS
                     0352  1010 ;  ALLOC_RSP_CTP
                     0352  1011 ;
                     0352  1012 ;  Routines to insure that enough space exists in the response buffer for
                     0352  1013 ;  the expected CTP or SCS responses which will result from the execution
                     0352  1014 ;  of the command.
                     0352  1015 ;
                     0352  1016 ;  INPUTS:
                     0352  1017 ;
                     0352  1018 ;       R3              - Address of current command
                     0352  1019 ;
                     0352  1020 ;  OUTPUTS:
                     0352  1021 ;
                     0352  1022 ;       R0,R1           - Destroyed
                     0352  1023 ;       CDC$W_RSPOFSET  - Addr of response save area for command
                     0352  1024 ;       CX$L_NXT_RSP    - Addr of next free area in response buffer
                     0352  1025 ;       CX$L_RSPCNTREM  - Total free bytes in response buffer
                     0352  1026 ;
                     0352  1027 ;-
                     0352  1028
                     0352  1029          .ENABLE LSB
                     0352  1030
                     0352  1031 ALLOC_RSP_SCS:
                     0352  1032
     50    01    9A  0352  1033          MOVZBL  #1,R0                   ; Number of responses expected
51   0000'CF    3C  0355  1034          MOVZWL  CX_MSG_RSP_SIZ,R1       ; Bytes per response
           2B    11  035A  1035          BRB     40$
                     035C  1036
                     035C  1037 ALLOC_RSP_CTP:
                     035C  1038
  50    1A A3    B0  035C  1039          MOVW    CDC$W_REPEAT(R3),R0     ; Assume all responses will be saved
           01    E0  0360  1040          BBS     #FLAG_V_SAVE_ALL,-      ; Branch of all responses are to be
     05 04 A3        0362  1041                  CDC$W_FLAG(R3),10$      ; saved
     50    01    9A  0365  1042          MOVZBL  S^#1,R0                 ; Not SAVALL, allocate 1 response area
           0E    11  0368  1043          BRB     30$                     ; Go select size
                     036A  1044
                     036A  1045 ;  All responses are to be saved. Because of a quirk in CTP, if the CTP request
                     036A  1046 ;  is either a generate message of datagram request, the number of responses
                     036A  1047 ;  expected is one greater than the CTP repeat count.
                     036A  1048
           04    91  036A  1049 10$:     CMPB    #CTP$GENMSGREQ,-        ; CTP generate message request?
        14 A3        036C  1050                  CDC$B_CTPOPCODE(R3)     ;
           06    13  036E  1051          BEQLU   20$                     ; Yes, branch
           05    91  0370  1052          CMPB    #CTP$GENDGRREQ,-        ; CTP generate datagram request?
        14 A3        0372  1053                  CDC$B_CTPOPCODE(R3)     ;
           02    12  0374  1054          BNEQU   30$                     ; No, branch
           50    B6  0376  1055 20$:     INCW    R0                      ; CTP generate msg or dg, add one
                     0378  1056
                     0378  1057 ;  The size of the response area needed is dependent upon the expected type
                     0378  1058 ;  of response (msg or dg).
                     0378  1059
51   0000'CF    3C  0378  1060 30$:     MOVZWL  CX_MSG_RSP_SIZ,R1       ; Assume MSG size block is needed
           02    E1  037D  1061          BBC     #FLAG_V_TYP_PKT_REC,-   ; If message, branch
```

```
                05 04 A3        037F  1062                    CDC$W_FLAG(R3),40$      ;
            51   0000'CF   3C   0382  1063            MOVZWL  CX_DG_RSP_SIZ,R1        ; Make DG size blocks
                           0387  1064
                           0387  1065 ; Now compute amount of response save space needed for this command
                           0387  1066
                51    50   A4  0387  1067 40$:       MULW2   R0,R1                   ; Size * Count
            0000'CF   51   A2  038A  1068            SUBW2   R1,CX$L_RSPCNTREM       ; See if enough space left
                      0F   1F  038F  1069            BLSSU   50$                     ; Not enough space, error, branch
                           0391  1070
                           0391  1071 ;  Copy starting offset of response save area for this command into command
                           0391  1072 ;  for use when input or results come in.
                           0391  1073
                0000'CF   B0  0391  1074            MOVW    CX$L_NXT_RSP,-          ; Put offset for response into
                   02 A3        0395  1075                    CDC$W_RSPOFSET(R3)      ; command buffer
            0000'CF   51   A0  0397  1076            ADDW2   R1,CX$L_NXT_RSP         ; Update offset into response buffer
                50    01   3C  039C  1077            MOVZWL  #SS$_NORMAL,R0          ; Set success status
                           05  039F  1078            RSB
                           03A0  1079
                      50   D4  03A0  1080 50$:       CLRL    R0                      ; Reture error status
                           05  03A2  1081            RSB
                           03A3  1082
                           03A3  1083            .DISABLE LSB
                           03A3  1084
                           03A3  1085
                           03A3  1086
```

```
                                03A3  1088          .SBTTL  FILL_CMD_REF,   Build a command reference number
                                03A3  1089 ;+
                                03A3  1090 ;
                                03A3  1091 ;  FILL_CMD_REF
                                03A3  1092 ;
                                03A3  1093 ;  This routine build a specific command reference number by taking the
                                03A3  1094 ;  following steps:
                                03A3  1095 ;
                                03A3  1096 ;       - Computing offset to current command
                                03A3  1097 ;       - Copying the sequence number word
                                03A3  1098 ;
                                03A3  1099 ;  A command reference number has the following format:
                                03A3  1100 ;
                                03A3  1101 ;       31              16  15              0
                                03A3  1102 ;       +-----------------------------------+
                                03A3  1103 ;       ! Sequence number !   CMD offset    !
                                03A3  1104 ;       +-----------------------------------+
                                03A3  1105 ;
                                03A3  1106 ;  INPUTS:
                                03A3  1107 ;
                                03A3  1108 ;       R3              - Address of current command
                                03A3  1109 ;
                                03A3  1110 ;  OUTPUTS:
                                03A3  1111 ;
                                03A3  1112 ;       CDC$L_CTPREF(R3)  - Generated number.
                                03A3  1113 ;
                                03A3  1114 ;-
                                03A3  1115
                                03A3  1116 FILL_CMD_REF:
                                03A3  1117
 15 A3    53    0000'CF    C3   03A3  1118          SUBL3   CX$L_CMDBUFBGN,R3,-     ; Calculate offset in bytes to
                                03AA  1119                  CDC$Q_CMDOFSET(R3)      ; current command
                 0001'CF    B0   03AA  1120          MOVW    1+CX$L_SEQNUM,-         ; Form sequence number
                   17 A3          03AE  1121                  CDC$W_CMDSEQNUM(R3)     ;
                              05   03B0  1122          RSB
                                03B1  1123
                                03B1  1124
                                03B1  1125
```

15
(9)

```
                                03B1  1127          .SBTTL  CHECK_CONN,     Check for connection
                                03B1  1128 ;+
                                03B1  1129 ;
                                03B1  1130 ;  CHECK_CONN
                                03B1  1131 ;
                                03B1  1132 ;  This routine checks for the presence of a connection to the specified remote
                                03B1  1133 ;  port over the specified local port. The connection may be in a state of
                                03B1  1134 ;  transition. Use the routine CHECK_ACTV_CONN when looking for a connection
                                03B1  1135 ;  that is active (not in transition).
                                03B1  1136 ;
                                03B1  1137 ;  Note that IPL is  raised to IPL$_SCS to synchronize access to the connection
                                03B1  1138 ;  database. This will keep the keep the connection database from changing
                                03B1  1139 ;  while stepping through it.
                                03B1  1140 ;
                                03B1  1141 ;  INPUTS:
                                03B1  1142 ;
                                03B1  1143 ;       R3                          - Address of current command
                                03B1  1144 ;       CDC$B_DSTPORT(R3)           - Destination port number
                                03B1  1145 ;       CDC$L_LPORT(R3)             - Local port device name
                                03B1  1146 ;
                                03B1  1147 ;  OUTPUTS:
                                03B1  1148 ;
                                03B1  1149 ;       R0                          - Status
                                03B1  1150 ;       R1                          - CDB address if success
                                03B1  1151 ;                                   - CX$_NOCONN otherwise
                                03B1  1152 ;
                                03B1  1153 ;-
                                03B1  1154
                                03B1  1155 CHECK_CONN::
                                03B1  1156
                                03B1  1157          DSBINT  #IPL$_SCS           ; Synch with connection database
                                03B8  1158          CALC_CDB_AD DST=R1          ; Get address of possible CDB
                          05    03D8  1159          BEQL    10$                 ; No, branch
                    50    01    03DA  1160          MOVZWL  #SS$_NORMAL,R0      ; Set success status
                          08    03DD  1161          BRB     20$                 ;
                0334  8F    03DF  1162 10$:        MOVZWL  #SS$_DEVREQERR,R0   ; Set class of error
                    51    02    03E4  1163          MOVZWL  #CX$_NOCONN,R1      ; No connection
                                03E7  1164 20$:        ENBINT                      ; Restore previous IPL
                                03EA  1165          RSB
                                03EB  1166
                                03EB  1167
                                03EB  1168
```

```
                                03EB  1170          .SBTTL  CHECK_ACTV_CONN, Check for an active connection
                                03EB  1171 ;+
                                03EB  1172 ;
                                03EB  1173 ;  CHECK_ACTV_CONN
                                03EB  1174 ;
                                03EB  1175 ;  This routine checks for the presence of an active connection to the
                                03EB  1176 ;  specified remote port over the specified local port.
                                03EB  1177 ;
                                03EB  1178 ;  Note that IPL is raised to IPL$_SCS to synchronize access to the connection
                                03EB  1179 ;  database. This will keep the keep the connection database from changing
                                03EB  1180 ;  while stepping through it.
                                03EB  1181 ;
                                03EB  1182 ;  INPUTS:
                                03EB  1183 ;
                                03EB  1184 ;       R3                          - Address of current command
                                03EB  1185 ;       CDC$B_DSTPORT(R3)           - Destination port number
                                03EB  1186 ;       CDC$L_LPORT(R3)             - Local port device name
                                03EB  1187 ;
                                03EB  1188 ;  OUTPUTS:
                                03EB  1189 ;
                                03EB  1190 ;       R0                          - Status
                                03EB  1191 ;       R1                          - CDB address if success
                                03EB  1192 ;                                   - CX$_NOCONN otherwise
                                03EB  1193 ;
                                03EB  1194 ;-
                                03EB  1195
                                03EB  1196 CHECK_ACTV_CONN::
                                03EB  1197
                                03EB  1198          DSBINT  #IPL$_SCS               ; Synch with connection database
                                03F5  1199          CALC_CDB_AD DST=R1              ; Get address of possible CDB
                      0F    13  0412  1200          BEQL    10$                     ; No, branch
                      00    E0  0414  1201          BBS     #CDB_V_CONN,-           ; Branch if CONNECT in progress for
                0A 0C A1          0416  1202                  CDB$W_STATUS(R1),10$    ; this connection
                      01    E0  0419  1203          BBS     #CDB_V_DISC,-           ; Branch if DISCONNECT in progress for
                05 0C A1          041B  1204                  CDB$W_STATUS(R1),10$    ; this connection
                 50   01    3C  041E  1205          MOVZWL  #SS$_NORMAL,R0          ; Set success status
                      08    11  0421  1206          BRB     20$                     ;
                                0423  1207
           50    0334 8F    3C  0423  1208 10$:     MOVZWL  #SS$_DEVREQERR,R0       ; Set class of error
                 51   02    3C  0428  1209          MOVZWL  #CX$_NOCONN,R1          ; No connection
                                042B  1210 20$:     ENBINT                          ; Restore previous IPL
                           05   042E  1211          RSB
                                042F  1212
                                042F  1213
                                042F  1214
```

```
                                042F  1216          .SBTTL  CLEAR_BUFFER,   Clear buffer routine
                                042F  1217 ;+
                                042F  1218 ;
                                042F  1219 ;  CLEAR_BUFFER
                                042F  1220 ;
                                042F  1221 ;  This routine clears a buffer.
                                042F  1222 ;
                                042F  1223 ;  INPUTS:
                                042F  1224 ;
                                042F  1225 ;       R1              - Buffer size
                                042F  1226 ;       R2              - Buffer starting address
                                042F  1227 ;
                                042F  1228 ;  OUTPUTS:
                                042F  1229 ;
                                042F  1230 ;       Buffer cleared
                                042F  1231 ;       All registers preserved
                                042F  1232 ;
                                042F  1233 ;-
                                042F  1234
                                042F  1235 CLEAR_BUFFER::
                                042F  1236
                          3F    BB  042F  1237          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
    62    51    00    FE AF     00    2C  0431  1238          MOVC5   #0,,#0,R1,(R2)          ; Clear out the specified buffer
                          3F    BA  0438  1239          POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                          05        043A  1240          RSB
                                043B  1241
                                043B  1242
                                043B  1243
```

```
                        043B  1245          .SBTTL  ALLOC_SEQNUM,   Create a sequence number
                        043B  1246 ;+
                        043B  1247 ;
                        043B  1248 ;  ALLOC_SEQNUM
                        043B  1249 ;
                        043B  1250 ;  This routine creates a random sequence by reading the time of day
                        043B  1251 ;  register and doing some munging on it.
                        043B  1252 ;
                        043B  1253 ;  INPUTS:
                        043B  1254 ;
                        043B  1255 ;       None
                        043B  1256 ;
                        043B  1257 ;  OUTPUTS:
                        043B  1258 ;
                        043B  1259 ;       CX$L_SEQNUM     - Sequence number
                        043B  1260 ;       All registers preserved
                        043B  1261 ;
                        043B  1262 ;-
                        043B  1263
                        043B  1264 ALLOC_SEQNUM:
                        043B  1265
          07    BB      043B  1266          PUSHR   #^M<R0,R1,R2>           ; Save registers
52  0000'CF    DE      043D  1267          MOVAL   CX$L_SEQNUM,R2          ; Make a work space
    62    1B    DB      0442  1268          MFPR    #PR780$_TODR,(R2)       ; Get a seed
    50    62    3C      0445  1269          MOVZWL  (R2),R0                 ; Use 16 bits
  51    02 A2    3C      0448  1270          MOVZWL  2(R2),R1                ; + another 16 bits
    51    50    C4      044C  1271          MULL2   R0,R1                   ; Multiply
  50    01 A2    3C      044F  1272          MOVZWL  1(R2),R0                ; Add in a fudge factor
    62    51    50    C1      0453  1273          ADDL3   R0,R1,(R2)              ; + save the value
01 A2    10F3    8F    A0      0457  1274          ADDW2   #^X10F3,1(R2)           ; MUNG the middle 16 bits
          07    BA      045D  1275          POPR    #^M<R0,R1,R2>           ; Restore regs
          05    05      045F  1276          RSB
                        0460  1277
                        0460  1278
                        0460  1279          .END
```

```
ALLOC_RSP_CTP                   00C0035C R      01
ALLOC_RSP_SCS                   00000352 R      01
ALLOC_SEQNUM                    0000043B R      01
BAD_BUFNAME                     00000181 R      01
BAD_CMDLEN                      00000181 R      01
BAD_CMDOPC                      00000181 R      01
BAD_DESTINATION                 00000181 R      01
BAD_LENGTH                      00000181 R      01
BAD_LOCALPORT                   00000181 R      01
BAD_PRIV                        00000191 R      01
BAD_RSPLEN                      00000181 R      01
BAD_SYSID                       00000181 R      01
BIT...                      =   0000000D
BUFF_AD                     =   00000000   G
BUFF_LEN                    =   00000004   G
CDB$B_CDRPCNT                   00000028
CDB$B_RSTNADR                   0000000E
CDB$B_TYPE                      0000000A
CDB$K_LEN                       00000032
CDB$L_CDBSLOT                   00000024
CDB$L_CDRPQBL                   0000002E
CDB$L_CDRPQFL                   0000002A
CDB$L_CDT                       0000001C
CDB$L_PDT                       00000020
CDB$L_QBL                       00000004
CDB$L_QFL                       00000000
CDB$T_LPRTNAM                   00000016
CDB$W_SIZE                      00000008
CDB$W_STATUS                    0000000C
CDB_V_CONN                  =   00000000
CDB_V_DISC                  =   00000001
CDC$ARMCINTPATH                 00000018
CDC$B_CFGCBLSTS                 00000021
CDC$B_CFGPOLBSTS                0000002A
CDC$B_CFGPOSTS                  0000001F
CDC$B_CFGP1LBSTS                00000028
CDC$B_CFGP1STS                  00000020
CDC$B_CMDCOUNT                  00000000
CDC$B_CNTFLG                    0000001F
CDC$B_COUNT                     00000015
CDC$B_CTPOPCODE                 00000014
CDC$B_DELAY                     00000019
CDC$B_DONECNT                   00000001
CDC$B_DSTPORT                   00000001
CDC$B_EXTEND                    0000001C
CDC$B_FDTCNT                    00000002
CDC$B_GARBAGE_FLAG              00000014
CDC$B_NOACTFLAG                 0000001E
CDC$B_NODEADR                   00000014
CDC$B_OPCODE                    00000000
CDC$B_OTHERNODE                 0000001E
CDC$B_PATH                      0000001C
CDC$B_PKTMULT                   0000001F
CDC$B_PKTSIZ                    0000001D
CDC$B_POLLERFLAG                0000001E
CDC$B_PROTREV                   00000036
CDC$B_PROTVER                   00000035
CDC$B_PROTYPE                   00000034
CDC$B_QNUMBER                   0000001F
CDC$B_RST_PORT                  00000015
CDC$B_SIOCNT                    00000003
CDC$B_TYPE                      00000014
CDC$CARRIER                     00000018
CDC$CONFIGDATA                  00000007
CDC$CONNECT                     00000008
CDC$C_CMDSPEC                   00000014
CDC$DISCONNECT                  00000009
CDC$GARBAGE                     0000000B
CDC$INITCINT                    0000001A
CDC$K_CMDHDRSIZ                 0000000C
CDC$K_CMDSPEC                   00000014
CDC$K_DATA                      00000014
CDC$_RGPACKET                   00000019
CDC$L_BUFADR                    00000024
CDC$L_BUFLENGTH                 00000020
CDC$L_BUFLNAME                  00000024
CDC$L_BUFLOFSET                 00000028
CDC$L_BUFRNAME                  0000002C
CDC$L_BUFROFSET                 00000030
CDC$L_CDB_AD                    0000000C
CDC$L_CDRP                      00000000
CDC$L_CFGHWTYPE                 00000022
CDC$L_CFGPRTMSK                 00000026
CDC$L_CNTRDISCDG                00000038
CDC$L_CNTRPOACK                 00000020
CDC$L_CNTRPONAK                 00000024
CDC$L_CNTRPONORSP               00000028
CDC$L_CNTRP1ACK                 0000002C
CDC$L_CNTRP1NAK                 00000030
CDC$L_CNTRP1NORSP               00000034
CDC$L_CTPREF                    00000015
CDC$L_DISCONREAS                00C00019
CDC$L_LPORT                     00000010
CDC$L_PORTHDR                   00000004
CDC$L_ROSTATUS                  0000000C
CDC$L_R1STATUS                  00000010
CDC$L_RESERV04                  00000004
CDC$L_RESERV08                  00000008
CDC$L_RESERV12                  0000000C
CDC$L_SEQNUM                    00000001
CDC$L_STARTADR                  00000020
CDC$L_SVAPTE                    00000018
CDC$MAINTSTATE                  0000000F
CDC$MAPBUF                      00C00005
CDC$MAPMBUF                     00000010
CDC$NAK                         00000016
CDC$NOACK                       00000017
CDC$NOACT                       0000000E
CDC$NORSP                       00000015
CDC$OVERSIZEPKT                 0000001D
CDC$POLLER                      0000000E
CDC$RDCINTPATH                  0000001C
CDC$READCNT                     0000000A
CDC$REQDATA                     00000004
```

CDC$REQMDATA	00000013			CTP$BUFUNMRSP	00000042
CDC$RESE	0000000C			CTP$CDATPREV	00000002
CDC$SELADR	00000014			CTP$CDATPTYPE	00000000
CDC$SENDDATA	00000003			CTP$CDATPVERS	00000001
CDC$SENDDG	00000002			CTP$CFGPOSTS	0000000B
CDC$SENDMDATA	00000012			CTP$CFGP1STS	0000000C
CDC$SENDMSG	00000001			CTP$CNTFLG	0000000A
CDC$START	0000000D			CTP$CNTRDISCDG	00000024
CDC$T_CONDAT	00000040			CTP$CNTRPOACK	0000000C
CDC$T_ENDMSG	00000072			CTP$CNTRPONAK	00000010
CDC$T_GARBAGE	00000018			CTP$CNTRPONRSP	00000014
CDC$T_LOCPROCNAM	00000014			CTP$CNTRP1ACK	00000018
CDC$T_PACKETDATA	00000080			CTP$CNTRP1NAK	0000001C
CDC$T_REMPROCNAM	00000064			CTP$CNTRP1NRSP	00000020
CDC$T_SYSTEMID	00000034			CTP$CONFIGREQ	00000009
CDC$UNMAPBUF	00000006			CTP$CONFIGRSP	00000049
CDC$UNMAPMBUF	00000011			CTP$CONNECTREQ	0000000B
CDC$W_CMDOFSET	00000015			CTP$CONNECTRSP	0000004B
CDC$W_CMDSEQNUM	00000017			CTP$COUNTSREQ	0000000A
CDC$W_DGCREDIT	0000003E			CTP$COUNTSRSP	0000004A
CDC$W_FLAG	00000004			CTP$DELAY	00000005
CDC$W_MAXDG	00000014			CTP$EXTEND	00000008
CDC$W_MAXMSG	00000018			CTP$FINISHREQ	0000000C
CDC$W_MINSENDCR	0000003C			CTP$FINISHRSP	0000004C
CDC$W_MSGCREDIT	0000003A			CTP$FMASK	00000006
CDC$W_POLLINT	0000001C			CTP$FUNCTREQ	00000000
CDC$W_POLLNUM	00000020			CTP$FUNCTRSP	00000040
CDC$W_REPEAT	0000001A			CTP$GENCONST	00000009
CDC$W_RESERV10	0000000A			CTP$GENDATA	0000000E
CDC$W_RSPBUFOFSET	00000004			CTP$GENDGRREQ	00000005
CDC$W_RSPCNT	00000006			CTP$GENDGRRSP	00000045
CDC$W_RSPOFSET	00000002			CTP$GENFUNCT	00000008
CDC$W_TXTLEN	00000008			CTP$GENLENGTH	0000000C
CDRP$B_FLAGS	00000040			CTP$GENMSGREQ	00000004
CDRP$C_BT_LEN =	00000040			CTP$GENMSGRSP	00000044
CDRP$K_CTPLEN	00000050			CTP$GENRSTREQ	00000006
CDRP$K_CX_LEN	00000068			CTP$GENRSTRSP	00000046
CDRP$K_XFRLEN	00000068			CTP$GENSTRREQ	00000007
CDRP$L_PDT	0000004C			CTP$GENSTRRSP	00000047
CDRP$L_PENDQBL	00000048			CTP$IMAGEDATA	0000000E
CDRP$L_PENDQFL	00000044			CTP$MAXCMDOPC	00000011
CDRP$T_CX_LBUFHNDL	00000050			CTP$MBUFMAPREQ	0000000D
CDRP$T_CX_RBUFHNDL	0000005C			CTP$MBUFMAPRSP	0000004D
CHECK_ACTV_CONN	000003EB	RG	01	CTP$MBUFUNMREQ	0000000E
CHECK_CONN	000003B1	RG	01	CTP$MBUFUNMRSP	0000004E
CHECK_LOC_PORT	00000328	R	01	CTP$MOVBUFREQ	00000003
CLEAR_BUFFER	0000042F	RG	01	CTP$MOVBUFRSP	00C00043
CTP$ACTCOUNT	00000006			CTP$MOVETYPE	00000008
CTP$BUFLENGTH	0000000C			CTP$MOVMBUFREQ	0000000F
CTP$BUFLNAME	00000010			CTP$MOVMBUFRSP	0000004F
CTP$BUFLOFSET	00000014			CTP$MSTATEREQ	00000010
CTP$BUFMAPREQ	00000001			CTP$MSTATERSP	00000050
CTP$BUFMAPRSP	00000041			CTP$NOACTFLAG	0000000A
CTP$BUFRNAME	00000018			CTP$NOACTREQ	00000008
CTP$BUFROFSET	0000001C			CTP$NOACTRSP	00000048
CTP$BUFTYPE	00000005			CTP$OPCODE	00000000
CTP$BUFUNMREQ	00000002			CTP$OPEXPAND	000000FF

CTP$OTHERNODE		00000009			FDT_MAINT_STATE		0000014C	R	01
CTP$PKTMULT		0000000B			FDT_MAP		00000249	R	01
CTP$PKTSIZ		0000000A			FDT_MNTMAP		00000287	R	01
CTP$REFERENCE		00000001			FDT_NAK		0000031C	R	01
CTP$REPCOUNT		00000006			FDT_NEXT_CMD		0000014C	R	01
CTP$RESERV10		0000000A			FDT_NOACK		0000031C	R	01
CTP$RESERV11		0000000B			FDT_NORSP		0000031C	R	01
CTP$RESERV12		0000000C			FDT_OVERSIZEPKT		0000014C	R	01
CTP$RESERV20		00000014			FDT_POLLER		0000031C	R	01
CTP$RESERV5		00000005			FDT_PWRFAIL		000001A4	R	01
CTP$RESERV6		00000006			FDT_RDCNT		0000031C	R	01
CTP$RESERV7		00000007			FDT_READPATH		0000014C	R	01
CTP$RESERV9		00000009			FDT_REQDAT		000002FB	R	01
CTP$REVISION	=	00000000	G		FDT_REQMDAT		00000310	R	01
CTP$STARTADR		0000000C			FDT_RESET		0000031C	R	01
CTP$STATUS		00000005			FDT_SELADR		0000031C	R	01
CTP$VERSION	=	00000003	G		FDT_SENDDG		00000231	R	01
CX$B_CMDERRCNT		********	X	01	FDT_SENDMSG		0000021F	R	01
CX$B_PWRFAIL		********	X	01	FDT_SNDDAT		000002FB	R	01
CX$L_CMDBUFBGN		********	X	01	FDT_SNDMDAT		00000310	R	01
CX$L_CMDBUFEND		********	X	01	FDT_START		0000031C	R	01
CX$L_CUR_CMD		********	X	01	FDT_UNMAP		000002EF	R	01
CX$L_LOCPRTLST		********	X	01	FDT_UNSOPC		0000014A	R	01
CX$L_NXT_RSP		********	X	01	FIL[_CMD_REF		000003A3	R	01
CX$L_RSPBUFBGN		********	X	01	FLAG_M_CMD_DONE	=	00008000		
CX$L_RSPBUFEND		********	X	01	FLAG_M_CONN_LOST	=	00000800		
CX$L_RSPCNTREM		********	X	01	FLAG_V_SAVE_ALL	=	00000001		
CX$L_RSPSEQNUM		********	X	01	FLAG_V_TYP_PKT_REC	=	00000002		
CX$L_SEQNUM		********	X	01	IPL$_SCS		********	X	01
CX$L_STATE		********	X	01	IPL$_SYNCH		********	X	01
CX$X_IRPSAV		********	X	01	IRP$[_SVAPTE	=	0000002C		
CX$_CDVERSION	=	00000006	G		IRP$M_BUFIO	=	00000001		
CX$_CONN_ACTV	=	00000003	G		IRP$M_FUNC	=	00000002		
CX$_NOCONN	=	00000002	G		IRP$W_BOFF	=	00000030		
CX_DG_RSP_SIZ		********	X	01	IRP$W_STS	=	0000002A		
CX_MSG_RSP_SIZ		********	X	01	JIB$L_BYTCNT	=	00000020		
CX_STARTDATA		00000000	RG	01	MAX_LOC_PORTS		********	X	01
CX_VERSION	=	0000C006	G		MAX_NODES		********	X	01
DIAG_VERSION	=	00000008	G		MMG$PTEADRCHK		********	X	01
EXE$ALLOCBUF		********	X	01	PCB$L_JIB	=	00000080		
EXE$BUFQUOPRC		********	X	01	PCB$L_PHD	=	0000006C		
EXE$FINISHIO		********	X	01	PCB$Q_PRIV	=	00000084		
EXE$QIODRVPKT		********	X	01	PR$_IPL		********	X	01
EXE$READCHK		********	X	01	PR780$_TODR	=	0000001B		
FDT_ABORT		000001C8	R	01	PRV$V_DIAGNOSE		********	X	01
FDT_ARMPATH		0000014C	R	01	SIZ...	=	00000001		
FDT_CARRIER		0000031C	R	01	SS$_BADPARAM	=	00C00014		
FDT_CONFIG		000001DC	R	01	SS$_DEVREQERR	=	00000334		
FDT_CONNECT		000001E8	R	01	SS$_NOPRIV	=	00000024		
FDT_CONN_ERR		000001B8	R	01	SS$_NORMAL	=	00000001		
FDT_CONN_ERR1		000001C0	R	01	SS$_POWERFAIL	=	00000364		
FDT_DISCONNECT		0000020A	R	01	STATE_M_ALTGRBG	=	00000004		
FDT_DISPATCH		000000E2	R	01	STATE_M_CANCEL	=	00000008		
FDT_GARBAGE		0000014C	R	01	STATE_M_CMAP	=	00000002		
FDT_INITCINT		0000014C	R	01	STATE_M_FDT	=	00000400		
FDT_INV_OPCODE		0000014A	R	01	STATE_M_NOTPROC	=	00001000		
FDT_LRGPACKET		0000014C	R	01	STATE_M_RMAP	=	00000001		

```
STATE_M_SIO                     = 00000800
STATE_M_STOP                    = 00000200
STATE_V_ALTGRBG                 = 00000002
STATE_V_CANCEL                  = 00000003
STATE_V_CMAP                    = 00000001
STATE_V_FDT                     = 0000000A
STATE_V_NOTPROC                 = 0000000C
STATE_V_RMAP                    = 00000000
STATE_V_SIO                     = 00000008
STATE_V_STOP                    = 00000009
VERSION_MISMATCH                  00000198 R      01
```

+----------------+
! Psect synopsis !
+----------------+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	00000460 (1120.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	000000FF (255.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
ALLOC_RSP_CTP	0000035C-R	1037 (19)	#-595 (8) #-633 (9)
ALLOC_RSP_SCS	00000352-R	1031 (19)	#-483 (5) #-517 (6) #-557 (7) #-671 (10) #-717 (11) #-772 (12) #-806 (13) #-841 (14) #-947 (17)
ALLOC_SEQNUM	0000043B-R	1264 (24)	#-258 (2)
BAD_BUFNAME	000001B1-R	432 (4)	
BAD_CMDLEN	000001B1-R	429 (4)	#-290 (3) #-293 (3)
BAD_CMDOPC	000001B1-R	430 (4)	#-336 (3)
BAD_DESTINATION	000001B1-R	427 (4)	#-284 (3)
BAD_LENGTH	000001B1-R	426 (4)	#-187 (2)
BAD_LOCALPORT	000001B1-R	428 (4)	#-287 (3)
BAD_PRIV	00000191-R	406 (4)	#-166 (2)
BAD_RSPLEN	000001B1-R	431 (4)	#-484 (5) #-518 (6) #-558 (7) #-596 (8) #-634 (9) #-672 (10) #-718 (11) #-773 (12) #-807 (13) #-842 (14) #-948 (17)
BAD_SYSID	000001B1-R	433 (4)	#-523 (6)
BIT...	=0000000D	96 (1)	96 (1)
BUFF_AD	=00000000	95 (1)	#-192 (2) #-235 (2) 238 (2)
BUFF_LEN	=00000004	95 (1)	#-185 (2) #-193 (2) #-237 (2) #-249 (2)
CDB$W_STATUS	0000000C		1202 (22) 1204 (22)
CDB_V_CONN	=00000000		#-1201 (22)
CDB_V_DISC	=00000001		#-1203 (22)
CDC$ARMCINTPATH	0000001B		327 (3)
CDC$B_CTPOPCODE	00000014		#-1050 (19) #-1053 (19)
CDC$B_DONECNT	00000001		#-264 (2) #-374 (3) #-739 (11)
CDC$B_DSTPORT	00000001		#-1158 (21) #-1199 (22) #-282 (3)
CDC$B_FDTCNT	00000002		#-361 (3) #-362 (3)
CDC$B_OPCODE	00000000		#-327 (3)
CDC$B_SIOCNT	00000003		#-265 (2)
CDC$CARRIER	00000018		327 (3)
CDC$CONFIGDATA	00000007		327 (3)
CDC$CONNECT	00000008		327 (3)
CDC$DISCONNECT	00000009		327 (3)
CDC$GARBAGE	0000000B		327 (3)
CDC$INITCINT	0000001A		327 (3)
CDC$K_CMDHDRSIZ	0000000C		#-266 (2)
CDC$LRGPACKET	00000019		327 (3)
CDC$L_BUFADR	00000024		#-678 (10) #-720 (11)
CDC$L_BUFLNAME	00000024		#-734 (11)
CDC$L_LPORT	00000010		#-1158 (21) #-1199 (22) #-984 (18) #-986 (18) #-991 (18) #-995 (18)
CDC$L_ROSTATUS	0000000C		#-735 (11)
CDC$L_RISTATUS	00000010		#-736 (11)
CDC$L_SVAPTE	00000018		#-685 (10)
CDC$MAINTSTATE	0000000F		327 (3)
CDC$MAPBUF	00000005		327 (3)
CDC$MAPMBUF	00000010		327 (3)
CDC$NAK	00000016		327 (3)
CDC$NOACK	00000017		327 (3)
CDC$NORSP	00000015		327 (3)

```
CDC$OVERSIZEPKT                 0000001D                        327     (3)
CDC$POLLER                      0000000E                        327     (3)
CDC$RDCINTPATH                  0000001C                        327     (3)
CDC$READCNT                     0000000A                        327     (3)
CDC$REQDATA                     00000004                        327     (3)
CDC$REQMDATA                    00000013                        327     (3)
CDC$RESET                       0000000C                        327     (3)
CDC$SELADR                      00000014                        327     (3)
CDC$SENDDATA                    00000003                        327     (3)
CDC$SENDDG                      00000002                        327     (3)
CDC$SENDMDATA                   00000012                        327     (3)
CDC$SENDMSG                     00000001                        327     (3)
CDC$START                       0000000D                        327     (3)
CDC$T_SYSTEMID                  00000034                    #-520       (6)     #-522       (6)
CDC$UNMAPBUF                    00000006                        327     (3)
CDC$UNMAPMBUF                   00000011                        327     (3)
CDC$W_CMDOFSET                  00000015                    #-1119      (20)
CDC$W_CMDSEQNUM                 00000017                    #-1121      (20)
CDC$W_FLAG                      00000004                      1041      (19)      1062      (19)    #-441       (4)     #-740       (11)
CDC$W_REPEAT                    0000001A                    #-1039      (19)
CDC$W_RSPBUFOFSET               00000004                    #-247       (2)
CDC$W_RSPOFSET                  00000002                    #-1075      (19)    #-733       (11)
CDC$W_TXTLEN                    00000008                    #-289       (3)     #-291       (3)     #-366       (3)
CHECK_ACTV_CONN                 000003EB-R      1196    (22)#-560       (7)     #-598       (8)     #-636       (9)     #-674       (10)
                                                            #-809       (13)
CHECK_CONN                      000003B1-R      1155    (21)#-525       (6)
CHECK_LOC_PORT                  00000328-R      981     (18)#-286       (3)
CLEAR_BUFFER                    0000042F-R      1235    (23)#-217       (2)
CTP$GENDGRREQ                   00000005                    #-1052      (19)
CTP$GENMSGREQ                   00000004                    #-1049      (19)
CTP$REVISION                    =00000000       92      (1)
CTP$VERSION                     =00000003       92      (1)
CX$B_CMDERRCNT                   00000000-XR                #-262       (2)
CX$B_PWRFAIL                     00000000-XR                #-179       (2)     #-420       (4)
CX$L_CMDBUFBGN                   00000000-XR                #-1118      (20)    #-246       (2)     #-263       (2)     #-360       (3)
                                                            #-738       (11)
CX$L_CMDBUFEND                   00000000-XR                #-292       (3)
CX$L_CUR_CMD                     00000000-XR                #-267       (3)     #-359       (3)     #-365       (3)     #-368       (3)
                                                            #-732       (11)
CX$L_LOCPRTLST                   00000000-XR                #-1158      (21)    #-1199      (22)
CX$L_NXT_RSP                     00000000-XR                #-1074      (19)    #-1076      (19)    #-261       (2)
CX$L_RSPBUFBGN                   00000000-XR                #-248       (2)     #-251       (2)     #-733       (11)
CX$L_RSPBUFEND                   00000000-XR                #-250       (2)     #-252       (2)
CX$L_RSPCNTREM                   00000000-XR                #-1068      (19)    #-253       (2)
CX$L_RSPSEQNUM                   00000000-XR                #-260       (2)
CX$L_SEQNUM                      00000000-XR                #-1120      (20)      1267      (24)    #-259       (2)
CX$L_STATE                       00000000-XR                #-453       (4)
CX$X_IRPSAV                      00000000-XR                #-158       (2)     #-159       (2)     #-384       (3)     #-385       (3)
                                                            #-451       (4)     #-452       (4)     #-679       (10)    #-722       (11)
CX$_CDVERSION                   =00000006                   #-415       (4)
CX$_CONN_ACTV                   =00000003                   #-447       (4)
CX$_NOCONN                      =00000002                   #-1163      (21)    #-1209      (22)
CX_DG_RSP_SIZ                    00000000-XR                #-1063      (19)
CX_MSG_RSP_SIZ                   00000000-XR                #-1034      (19)    #-1060      (19)
CX_STARTDATA                     00000000-R      153     (2)
CX_VERSION                      =00000006       56      (1)     #-173       (2)
DIAG_VERSION                    =00000008       95      (1)     #-172       (2)
```

```
EXE$ALLOCBUF                    00000000-XR                    211     (2)
EXE$BUFQUOPRC                   00000000-XR                    202     (2)
EXE$FINISHIO                    00000000-XR                    454     (4)
EXE$QIODRVPKT                   00000000-XR                    386     (3)
EXE$READCHK                     00000000-XR                    194     (2)
FDT_ABORT                       000001C8-R   449     (4)     #-204   (2)     #-213   (2)     #-378   (3)     #-410   (4)
                                                             #-416   (4)     #-423   (4)     #-437   (4)     #-442   (4)
FDT_ARMPATH                     0000014C-R   344     (3)       327   (3)
FDT_CARRIER                     0000031C-R   945     (17)      327   (3)
FDT_CONFIG                      000001DC-R   481     (5)       327   (3)
FDT_CONNECT                     000001E8-R   515     (6)       327   (3)
FDT_CONN_ERR                    000001B8-R   439     (4)     #-561   (7)     #-599   (8)     #-637   (9)     #-675   (10)
                                                             #-810   (13)
FDT_CONN_ERR1                   000001C0-R   444     (4)     #-526   (6)
FDT_DISCONNECT                  0000020A-R   555     (7)       327   (3)
FDT_DISPATCH                    000000E2-R   280     (3)     #-369   (3)
FDT_GARBAGE                     0000014C-R   342     (3)       327   (3)
FDT_INITCINT                    0000014C-R   345     (3)       327   (3)
FDT_INV_OPCODE                  0000014A-R   333     (3)
FDT_LRGPACKET                   0000014C-R   347     (3)       327   (3)
FDT_MAINT_STATE                 0000014C-R   346     (3)       327   (3)
FDT_MAP                         00000249-R   669     (10)      327   (3)
FDT_MNTMAP                      00000287-R   715     (11)      327   (3)
FDT_NAK                         0000031C-R   943     (17)      327   (3)
FDT_NEXT_CMD                    0000014C-R   357     (3)       327   (3)     #-486   (5)     #-528   (6)     #-563   (7)
                                                             #-603   (8)     #-641   (9)     #-687   (10)    #-743   (11)
                                                             #-775   (12)    #-812   (13)    #-844   (14)    #-950   (17)
FDT_NOACK                       0000031C-R   944     (17)      327   (3)
FDT_NORSP                       0000031C-R   942     (17)      327   (3)
FDT_OVERSIZEPKT                 0000014C-R   348     (3)       327   (3)
FDT_POLLER                      0000031C-R   940     (17)      327   (3)
FDT_PWRFAIL                     000001A4-R   418     (4)     #-180   (2)
FDT_RDCNT                       0000031C-R   872     (15)      327   (3)
FDT_READPATH                    0000014C-R   343     (3)       327   (3)
FDT_REQDAT                      000002FB-R   804     (13)      327   (3)
FDT_REQMDAT                     00000310-R   838     (14)      327   (3)
FDT_RESET                       0000031C-R   903     (16)      327   (3)
FDT_SELADR                      0000031C-R   941     (17)      327   (3)
FDT_SENDDG                      00000231-R   631     (9)       327   (3)
FDT_SENDMSG                     0000021F-R   593     (8)       327   (3)
FDT_SNDDAT                      000002FB-R   803     (13)      327   (3)
FDT_SNDMDAT                     00000310-R   837     (14)      327   (3)
FDT_START                       0000031C-R   902     (16)      327   (3)
FDT_UNMAP                       000002EF-R   770     (12)      327   (3)
FDT_UNSOPC                      0000014A-R   334     (3)
FILE_CMD_REF                    000003A3-R   1116    (20)    #-601   (8)     #-639   (9)
FLAG_M_CMD_DONE                 =00008000                    #-740   (11)
FLAG_M_CONN_LOST                =00000800                    #-441   (4)
FLAG_V_SAVE_ALL                 =00000001                    #-1040  (19)
FLAG_V_TYP_PKT_REC              =00000002                    #-1061  (19)
IPL$_SCS                         00000000-XR                 #-1157  (21)    #-1198  (22)
IPL$_SYNCH                       00000000-XR                 #-681   (10)    #-724   (11)
IRP$L_SVAPTE                    =0000002C                    #-223   (2)
IRP$M_BUFIO                     =00000001                    #-227   (2)
IRP$M_FUNC                      =00000002                    #-228   (2)
IRP$W_BOFF                      =00000030                    #-224   (2)
IRP$W_STS                       =0000002A                    #-229   (2)
```

```
JIB$L_BYTCNT                    =00000020
MAX_LOC_PORTS                    00000000-XR
MAX_NODES                        00000000-XR
MMG$PTEADRCHK                    00000000-XR
PCB$L_JIB                       =00000080
PCB$L_PHD                       =0000006C
PCB$Q_PRIV                      =00000084
PR$_IPL                          00000000-XR

PR780$_TODR                     =0000001B
PRV$V_DIAGNOSE                   00000000-XR
SIZ...                          =00000001       96      (1)
SS$_BADPARAM                    =00000014
SS$_DEVREQERR                   =00000334
SS$_NOPRIV                      =00000024
SS$_NORMAL                      =00000001
SS$_POWERFAIL                   =00000364
STATE_M_ALTGRBG                 =00000004       96      (1)
STATE_M_CANCEL                  =00000008       96      (1)
STATE_M_CMAP                    =00000002       96      (1)
STATE_M_FDT                     =00000400       96      (1)
STATE_M_NOTPROC                 =00001000       96      (1)
STATE_M_RMAP                    =00000001       96      (1)
STATE_M_SIO                     =00000800       96      (1)
STATE_M_STOP                    =00000200       96      (1)
STATE_V_ALTGRBG                 =00000002       96      (1)
STATE_V_CANCEL                  =00000003       96      (1)
STATE_V_CMAP                    =00000001       96      (1)
STATE_V_FDT                     =0000000A       96      (1)
STATE_V_NOTPROC                 =0000000C       96      (1)
STATE_V_RMAP                    =00000000       96      (1)
STATE_V_SIO                     =0000000B       96      (1)
STATE_V_STOP                    =00000009       96      (1)
VERSION_MISMATCH                 00000198-R     412     (4)     #-174   (2)

#-226   (2)
#-993   (18)
#-283   (3)
  682   (10)    725     (11)
#-225   (2)
#-680   (10)    #-723   (11)
  165   (2)
#-1157  (21)    #-1164  (21)    #-1198  (22)    #-1210  (22)
#-681   (10)    #-683   (10)    #-724   (11)    #-726   (11)
#-1268  (24)
#-165   (2)
  96    (1)
#-435   (4)
#-1162  (21)    #-1208  (22)    #-414   (4)     #-446   (4)
#-408   (4)
#-1077  (19)    #-1160  (21)    #-1205  (22)    #-376   (3)
#-421   (4)
```

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$BND	1	327 (3)	327 (3)
$BND1	1	327 (3)	327 (3)
$BND2	1	327 (3)	327 (3)
$CDCDEF	8	91 (1)	91 (1)
$CDRPDEF	6	72 (1)	72 (1)
$CDTDEF	4	73 (1)	73 (1)
$CIBHANDEF	1	74 (1)	74 (1)
$CTPDEF	7	92 (1)	92 (1)
$CXCDBDEF	2	93 (1)	93 (1)
$CXCDRPDEF	1	94 (1)	94 (1)
$CXQIODEF	1	95 (1)	95 (1)
$CXSTATEDEF	1	96 (1)	96 (1)
$DEFINI	1	68 (1)	68 (1) 72 (1) 73 (1) 74 (1)
			75 (1) 76 (1) 77 (1) 78 (1)
			79 (1) 80 (1) 81 (1) 82 (1)
			91 (1) 92 (1) 93 (1) 94 (1)
$DISPATCH	2	296 (3)	296 (3)
$DSP1	1	327 (3)	327 (3)
$DSP2	1	327 (3)	327 (3)
$DYNDEF	7	75 (1)	75 (1)
$IODEF	17	76 (1)	76 (1)
$IRPDEF	7	77 (1)	77 (1)
$JIBDEF	3	78 (1)	78 (1)
$PCBDEF	7	79 (1)	79 (1)
$PR780DEF	1	68 (1)	68 (1)
$SCSDEF	4	80 (1)	80 (1)
$SSDEF	21	81 (1)	81 (1)
$UCBDEF	16	82 (1)	82 (1)
$VIELD1	1		96 (1)
BEQLW	1	523 (6)	523 (6)
BGTRUW	1	284 (3)	284 (3) 293 (3)
BGTRW	1	187 (2)	187 (2)
BLBCW	1	204 (2)	204 (2) 213 (2) 287 (3) 484 (5)
			518 (6) 558 (7) 561 (7) 634 (9)
			637 (9) 672 (10) 675 (10) 718 (11)
			773 (12) 807 (13) 810 (13) 842 (14)
			948 (17)
BLBSW	1	526 (6)	526 (6)
BLEQUW	1	290 (3)	290 (3)
BNEQW	1	174 (2)	174 (2) 180 (2)
CALC_CDB_AD	1	1158 (21)	1158 (21) 1199 (22)
CALC_RSP	1	733 (11)	733 (11)
DSBINT	1	681 (10)	1157 (21) 1198 (22) 681 (10) 724 (11)
ENBINT	1	683 (10)	1164 (21) 1210 (22) 683 (10) 726 (11)
IFPRIV	1	165 (2)	165 (2)
SET_CMD_FLAG	1	441 (4)	441 (4) 740 (11)
_VIELD	1		96 (1)

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	94	00:00:00.33	00:00:01.30
Command processing	135	00:00:00.64	00:00:01.37
Pass 1	735	00:00:29.22	00:00:37.50
Symbol table sort	0	00:00:03.49	00:00:05.73
Pass 2	211	00:00:04.92	00:00:05.78
Symbol table output	41	00:00:00.30	00:00:00.39
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	46	00:00:00.90	00:00:01.08
Assembler run totals	1266	00:00:39.84	00:00:53.21

The working set limit was 2550 pages.
132173 bytes (259 pages) of virtual memory were used to buffer the intermediate code.
There were 120 pages of symbol table space allocated to hold 2248 non-local and 47 local symbols.
1279 source lines were read in Pass 1, producing 20 object records in Pass 2.
142 pages of virtual memory were used to define 53 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	1
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	12
DRB2:[SHULL.EVXCI.CXDRIVER]CXLIB.MLB;3	16
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	8
TOTALS (all libraries)	37

2600 GETS were required to define 37 macros.

There were no errors, warnings or information messages.

```
MACRO/LIS/CROSS CXCMDFDT+CXLIB/LIB+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB
```

Table of contents

(1) 47 DEFINITIONS
(3) 73 UNIT INITIALIZATION ROUTINE
(4) 189 UNSOLICITED INTERRUPT ROUTINE
(5) 207 HIPL_ALLPOOL
(6) 251 CX_END, The END of CXDRIVER

```
                        0000     1          .TITLE  CXINIT  Initialization and Misc. Routines
                        0000     2          .IDENT  'V6-000'
                        0000     3
                        0000     4  ;*****************************************************************************
                        0000     5  ;
                        0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
                        0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
                        0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
                        0000     9  ;
                        0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
                        0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
                        0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
                        0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
                        0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
                        0000    15  ;
                        0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
                        0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
                        0000    18  ;
                        0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
                        0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
                        0000    21  ;
                        0000    22  ;*****************************************************************************
                        0000    23  ;
                        0000    24  ;++
                        0000    25  ;
                        0000    26  ; FACILITY:     CI DIAGNOSTIC CLASS DRIVER
                        0000    27  ;
                        0000    28  ;
                        0000    29  ; ABSTRACT:     This module contains the miscellaneous routines:
                        0000    30  ;
                        0000    31  ;                       - Unit initialization routine
                        0000    32  ;                       - Unsolicited interrupt routine
                        0000    33  ;
                        0000    34  ; AUTHOR:       Richard Hennessy  27-Jul-1982
                        0000    35  ;               Base Systems Diagnostic Engineering
                        0000    36  ;
                        0000    37  ; MODIFIED BY:  Jim Klumpp  6-MAY-83
                        0000    38  ;               Base Systems Diagnostic Engineering
                        0000    39  ;
                        0000    40  ;       6-000   Dave Shull      07-July-1984
                        0000    41  ;               VMS V4 Modifications/Release
                        0000    42  ;
                        0000    43  ;--
                        0000    44
                        0000    45          .DEFAULT DISPLACEMENT,WORD
                        0000    46          .ENABLE  SUPPRESSION
                        0000    47          .SBTTL   DEFINITIONS
                        0000    48
                        0000    49  ; Set PSECT to driver code
                        0000    50
                    00000000    51          .PSECT   $$$115_DRIVER,LONG
                        0000    52
                        0000    53
                        0000    54  ;  System definitions (LIB.MLB):
                        0000    55
                        0000    56          $CDRPDEF                        ; Define CDRP offsets
                        0000    57          $IRPDEF                         ; Define IRP offsets
```

```
0000     58          $SCSDEF                         ; Define SCS offsets
0000     59          $UCBDEF                         ; Define UCB offsets
0000     60
0000     61
0000     62 ;  CXDRIVER definitions (CXLIB.MLB):
0000     63
0000     64          $CDCDEF                         ; Define CDC interface offsets
0000     65          $CXCDBDEF                       ; Define CXCDB offsets
0000     66          $CXCDRPDEF                      ; Define CX CDRP extension
0000     67          $CXSTATEDEF                     ; Define CXDRIVER states
0000     68
0000     69
0000     70
```

```
                        0000     72
                        0000     73          .SBTTL  UNIT INITIALIZATION ROUTINE
                        0000     74
                        0000     75 ;+
                        0000     76 ;
                        0000     77 ; CX_UNIT_INIT
                        0000     78 ;
                        0000     79 ; This code perfoms initialization procedures for CXDRIVER such as
                        0000     80 ; setting up queue listheads and various other data structures.
                        0000     81 ;
                        0000     82 ; INPUTS:
                        0000     83 ;
                        0000     84 ;       R5      - UCB address
                        0000     85 ;
                        0000     86 ;  OUTPUTS:
                        0000     87 ;
                        0000     88 ;       All registers preserved
                        0000     89 ;
                        0000     90 ;-
                        0000     91
                00000001 0000    92 INIT_FLAG:       .BLKB   1               ; Execute initialization routine
                        0001     93                                          ; only once (on driver loading)
                        0001     94
                        0001     95 CX_UNIT_INIT::
                        0001     96
                        0001     97 ;*****************************************************************************
                        0001     98 ;
                        0001     99 ;  SOFTWARE DEBUG
                        0001    100 ;
                        0001    101 ;       JSB     G^INI$BRK
                        0001    102 ;
                        0001    103 ;  END SOFTWARE DEBUG
                        0001    104 ;
                        0001    105 ;*****************************************************************************
                        0001    106
 56 FB AF    00    E2   0001    107         BBSS    #0,INIT_FLAG,50$        ; On powerfail, do NOT execute this
                        0006    108                                         ; routine.
                        0006    109
          0000'CF   DE  0006    110         MOVAL   CX$Q_MAPLST,-           ; Initialize the mapped
          0000'CF       000A    111                 CX$Q_MAPLST             ; buffer listhead
          0000'CF   DE  000D    112         MOVAL   CX$Q_MAPLST,-
          0004'CF       0011    113                 4+CX$Q_MAPLST
                        0014    114
                        0014    115 ;  Calculate response buffer entry size using SCS global parameters
                        0014    116
 00000000'GF    A1    0014    117         ADDW3   G^SCS$GW_MAXMSG,-       ; MSG size
                14      001A    118                 #CDC$K_DATA,-
          0000'CF       001B    119                 CX_MSG_RSP_SIZ
 00000000'GF    A1    001F    120         ADDW3   G^SCS$GW_MAXDG,-        ; DG size
                '4      0024    121                 #CDC$K_DATA,-
          0000'CF       0025    122                 CX_DG_RSP_SIZ
                        0028    123
                        0028    124 ;  Allocate a CDB list block for each possible local port. Write the address
                        0028    125 ;  of each CDB list into the local port list. The data structures look as
                        0028    126 ;  follows:
                        0028    127 ;
                        0028    128 ;  LOCAL PORT LIST              CDB LIST(S)
```

```
                                  0028      ;
                                  0028      ;                               ----------------
                                  0028      ;                               :              :
                                  0028      ;                               :  VMS header  :
                                  0028      ;                               :              :
                                  0028      ;       -------------           ----------------
                                  0028      ;    :      PAA0     : ------> : Remote node 0 : ------> CDB for PAA0/Remote node 0
                                  0028      ;       -------------           ----------------
                                  0028      ;    :      PAB0     :         : Remote node 1 :
                                  0028      ;       -------------           ----------------
                                  0028      ;             .                        .
                                  0028      ;             .                        .
                                  0028      ;             .                        .
                                  0028      ;       -------------           ----------------
                                  0028      ;    :      PAn0     :         : Remote node n :
                                  0028      ;       -------------           ----------------
                                  0028
                        1F   BB   0028              PUSHR   #^M<R0,R1,R2,R3,R4>        ; Save registers
             53   0000'CF    DE   002A              MOVAL   CX$L_LOCPRTLST,R3          ; Point to top of local port list
             54     FF'8F    9A   002F              MOVZBL  #<MAX_LOC_PORTS-1>,R4      ; Loop for all possible local ports
             51     0C'8F    9A   0033      10$:    MOVZBL  #<12+<4*MAX_NODES>>,R1     ; Size of CDB list (including ovrhd)
                  007E'CF    16   0037              JSB     HIPL_ALLPOOL               ; Allocate a CDB list
                    22 50    E9   003B              BLBC    R0,20$                     ; Error, branch
                        3F   BB   003E              PUSHR   #^M<R0,R1,R2,R3,R4,R5>     ; Save registers
   62    51    00  FE AF  00 2C   0040              MOVC5   #0,.,#0,R1,(R2)            ; Clear the CDB list (except for ovrhd)
                        3F   BA   0047              POPR    #^M<R0,R1,R2,R3,R4,R5>     ; Restore registers
              08 A2     51   B0   0049              MOVW    R1,IRP$W_SIZE(R2)          ; Fill in size field
        6344     0C A2       DE   004D              MOVAL   12(R2),(R3)[R4]            ; Write CDB list addr to loc prt list
                    DE 54    F4   0052              SOBGEQ  R4,10$                     ; Loop
                        1F   BA   0055              POPR    #^M<R0,R1,R2,R3,R4>        ; Restore registers
                                  0057
                                  0057      ;*******************************************************************
                                  0057      ;
                                  0057      ; I should check CMAP,RMAP etc and if there is anything there do the
                                  0057      ;  appropriate deallocations etc.
                                  0057      ;
                                  0057      ;*******************************************************************
                                  0057
                  0000'CF    D4   0057              CLRL    CX$L_STATE                 ; Clear driver state
                        10   A8   005B              BISW    #UCB$M_ONLINE,-            ; Set the unit ONLINE
                    64 A5         005D                      UCB$W_STS(R5)
                             05   005F              RSB
                                  0060
                                  0060
                                  0060      ;  A memory allocation error has occurred in the init code. Deallocate all
                                  0060      ;  CDB lists allocated up until this time and Leave CXDRIVER offline.
                                  0060
             54     FF'8F    9A   0060      20$:    MOVZBL  #<MAX_LOC_PORTS-1>,R4      ; Loop for all possible local ports
             50      6344    D0   0064      30$:    MOVL    (R3)[R4],R0                ; Get a CDB list
                        10   13   0068              BEQL    40$                        ; None, branch
                      6344   D4   006A              CLRL    (R3)[R4]                   ; Indicate CDB list is gone
             50     F4 A0    DE   006D              MOVAL   -12(R0),R0                 ; Back up to top of header
              00000000'GF    16   0071              JSB     G^COM$DRVDEALMEM           ; Deallocate the CDB list
                    EA 54    F4   0077              SOBGEQ  R4,30$                     ; Loop
                        1F   BA   007A      40$:    POPR    #^M<R0,R1,R2,R3,R4>        ; Restore registers
                             05   007C      50$:    RSB
                                  007D
```

```
                        007D   186
                        007D   187
```

```
                     007D   189          .SBTTL  UNSOLICITED INTERRUPT ROUTINE
                     007D   190
                     007D   191 ;+
                     007D   192 ;
                     007D   193 ;  CX_UNSOLINT
                     007D   194 ;
                     007D   195 ;  Since CXDRIVER is a class driver, which uses PADRIVER to interface to
                     007D   196 ;  CI hardware, no interrupts should be processed.
                     007D   197 ;
                     007D   198 ;-
                     007D   199
                     007D   200 CX_UNSOLINT::
                     007D   201
                  05 007D   202          RSB
                     007E   203
                     007E   204
                     007E   205
```

```
                        007E    207          .SBTTL  HIPL_ALLPOOL
                        007E    208
                        007E    209 ;+
                        007E    210 ;
                        007E    211 ;  HIPL_ALLPOOL
                        007E    212 ;
                        007E    213 ;  This subroutine allocates nonpaged pool from IPL greater than
                        007E    214 ;  fork IPL.  It is assumed taht the current IPL was reached by
                        007E    215 ;  reasing IPL from a level less than, or equal to, IPL$SYNCH.
                        007E    216 ;  Otherwise this code path might interrupt a pool allocation
                        007E    217 ;  already in progress and corrupt pool.  In particular, this
                        007E    218 ;  routine may be called as a result of a SYSGEN controller/unit
                        007E    219 ;  init, INITADP controller/unit init, but not on power fail
                        007E    220 ;  recovery.
                        007E    221 ;
                        007E    222 ; Inputs:
                        007E    223 ;
                        007E    224 ;       R1                      -# bytes of pool to allocate
                        007E    225 ;
                        007E    226 ; Outputs:
                        007E    227 ;
                        007E    228 ;       R0                      -0/1 for fail/success
                        007E    229 ;       R1                      -# bytes actually allocated
                        007E    230 ;       R2                      -Addr of buffer allocated
                        007E    231 ;       Other registers         -Preserved
                        007E    232 ;-
                        007E    233
                        007E    234          .ENABL  LSB
                        007E    235
                        007E    236 HIPL_ALLPOOL:
                        007E    237
          7E    53    7D      007E    238          MOVQ    R3,-(SP)                ; Save caller's registers
54  00000000'GF    DE      0081    239          MOVAL   G^EXE$GL_NONPAGED,R4    ; Get addr of pool header area
              64    DD      0088    240          PUSHL   (R4)                    ; Save IPL from header
64  00000000'8F    DB      008A    241          MFPR    #PR$_IPL,(R4)           ; Put current IPL in header
    00000000'GF    16      0091    242          JSB     G^EXE$ALONONPAGED       ; Allocate from nonpaged pool
              64  8ED0      0097    243          POPL    (R4)                    ; Restore old IPL in header
          53    8E    7D      009A    244          MOVQ    (SP)+,R3                ; Restore caller's registers
                      05      009D    245          RSB                             ; Return
                              009E    246
                              009E    247          .DSABL  LSB
                              009E    248
                              009E    249
```

```
                                009E    251          .SBTTL  CX_END, The END of CXDRIVER
                                009E    252
                                009E    253 CX_END::
                                009E    254
                                009E    255          .END
```

BIT...	= 0000000D	CDC$L_BUFADR	00000024
CDB$B_CDRPCNT	00000028	CDC$L_BUFLENGTH	00000020
CDB$B_RSTNADR	0000000E	CDC$L_BUFLNAME	00000024
CDB$B_TYPE	0000000A	CDC$L_BUFLOFSET	00000028
CDB$K_LEN	00000032	CDC$L_BUFRNAME	0000002C
CDB$L_CDBSLOT	00000024	CDC$L_BUFROFSET	00000030
CDB$L_CDRPQBL	0000002E	CDC$L_CDB_AD	0000000C
CDB$L_CDRPQFL	0000002A	CDC$L_CDRP	0000000C
CDB$L_CDT	0000001C	CDC$L_CFGHWTYPE	00000022
CDB$L_PDT	00000020	CDC$L_CFGPRTMSK	00000026
CDB$L_QBL	00000004	CDC$L_CNTRDISCDG	00000038
CDB$L_QFL	00000000	CDC$L_CNTRPOACK	00000020
CDB$T_LPRTNAM	00000016	CDC$L_CNTRPONAK	00000024
CDB$W_SIZE	00000008	CDC$L_CNTRPONORSP	00000028
CDB$W_STATUS	0000000C	CDC$L_CNTRP1ACK	0000002C
CDC$ARMCINTPATH	0000001B	CDC$L_CNTRP1NAK	00000030
CDC$B_CFGCBLSTS	00000021	CDC$L_CNTRP1NORSP	00000034
CDC$B_CFGPOLBSTS	0000002A	CDC$L_CTPREF	00000015
CDC$B_CFGPOSTS	0000001F	CDC$L_DISCONREAS	00000019
CDC$B_CFGP1LBSTS	0000002B	CDC$L_LPORT	00000010
CDC$B_CFGP1STS	00000020	CDC$L_PORTHDR	00000004
CDC$B_CMDCOUNT	00000000	CDC$L_R0STATUS	0000000C
CDC$B_CNTFLG	0000001F	CDC$L_R1STATUS	00000010
CDC$B_COUNT	00000015	CDC$L_RESERV04	00000004
CDC$B_CTPOPCODE	00000014	CDC$L_RESERV08	00000008
CDC$B_DELAY	00000019	CDC$L_RESERV12	0000000C
CDC$B_DONECNT	00000001	CDC$L_SEQNUM	00000001
CDC$B_DSTPORT	00000001	CDC$L_STARTADR	000C0020
CDC$B_EXTEND	0000001C	CDC$L_SVAPTE	00000018
CDC$B_FDTCNT	00000002	CDC$MAINTSTATE	0000000F
CDC$B_GARBAGE_FLAG	00000014	CDC$MAPBUF	00000005
CDC$B_NOACTFLAG	0000001E	CDC$MAPMBUF	00000010
CDC$B_NODEADR	00000014	CDC$NAK	00000016
CDC$B_OPCODE	00000000	CDC$NOACK	00000017
CDC$B_OTHERNODE	0000001E	CDC$NOACT	0000000E
CDC$B_PATH	0000001C	CDC$NORSP	00000015
CDC$B_PKTMULT	0000001F	CDC$OVERSIZEPKT	0000001D
CDC$B_PKTSIZ	00000010	CDC$POLLER	0000000E
CDC$B_POLLERFLAG	0000001E	CDC$RDCINTPATH	0000001C
CDC$B_PROTREV	00000036	CDC$READCNT	0000000A
CDC$B_PROTVER	00000035	CDC$REQDATA	00000004
CDC$B_PROTYPE	00000034	CDC$REQMDATA	00000013
CDC$B_QNUMBER	0000001F	CDC$RESET	0000000C
CDC$B_RST_PORT	00000015	CDC$SELADR	00000014
CDC$B_SIOCNT	00000003	CDC$SENDDATA	00000003
CDC$B_TYPE	00000014	CDC$SENDDG	00000002
CDC$CARRIER	00000018	CDC$SENDMDATA	00C00012
CDC$CONFIGDATA	00000007	CDC$SENDMSG	00000001
CDC$CONNECT	00000008	CDC$START	0000000D
CDC$C_CMDSPEC	00000014	CDC$T_CONDAT	00000040
CDC$DISCONNECT	00000009	CDC$T_ENDMSG	00000072
CDC$GARBAGE	00000008	CDC$T_GARBAGE	00000018
CDC$INITCINT	0000001A	CDC$T_LOCPROCNAM	00000014
CDC$K_CMDHDRSIZ	0000000C	CDC$T_PACKETDATA	00000080
CDC$K_CMDSPEC	00000014	CDC$T_REMPROCNAM	00000024
CDC$K_DATA	00000014	CDC$T_SYSTEMID	00000034
CDC$LRGPACKET	00000019	CDC$UNMAPBUF	00000006

```
CDC$UNMAPMBUF                   00000011
CDC$W_CMDOFSET                  00000015
CDC$W_CMDSEQNUM                 00000017
CDC$W_DGCREDIT                  0000003E
CDC$W_FLAG                      00000004
CDC$W_MAXDG                     00000014
CDC$W_MAXMSG                    00000018
CDC$W_MINSENDCR                 0000003C
CDC$W_MSG_REDIT                 0000003A
CDC$W_POLLINT                   0000001C
CDC$W_POLLNUM                   00000020
CDC$W_REPEAT                    0000001A
CDC$W_RESERV10                  0000000A
CDC$W_RSPBUFOFSET               00000004
CDC$W_RSPCNT                    00000006
CDC$W_RSPOFSET                  00000002
CDC$W_TXTLEN                    00000008
CDRP$B_FLAGS                    00000040
CDRP$C_BT_LEN                 = 00000040
CDRP$K_CTPLEN                   00000050
CDRP$K_CX_LEN                   00000068
CDRP$K_XFRLEN                   00000068
CDRP$L_PDT                      0000004C
CDRP$L_PENDQBL                  00000048
CDRP$L_PENDQFL                  00000044
CDRP$T_CX_LBUFHNDL              00000050
CDRP$T_CX_RBUFHNDL              0000005C
COM$DRVDEALMEM                  ********    X   01
CX$L_LOCPRTLST                  ********    X   01
CX$L_STATE                      ********    X   01
CX$Q_MAPLST                     ********    X   01
CX_DG_RSP_SIZ                   ********    X   01
CX_END                          0000009E RG     01
CX_MSG_RSP_SIZ                  ********      X 01
CX_UNIT_INIT                    00000001 RG     01
CX_UNSOLINT                     0000007D RG     01
EXE$ALONONPAGED                 ********      X 01
EXE$GL_NONPAGED                 ********      X 01
HIPL_ALLPOOL                    0000007E R      01
INIT_FLAG                       00000000 R      01
IRP$W_SIZE                    = 00000008
MAX_LOC_PORTS                   ********      X 01
MAX_NODES                       ********      X 01
PR$_IPL                         ********      X 01
SCS$GW_MAXDG                    ********      X 01
SCS$GW_MAXMSG                   ********      X 01
SIZ...                        = 00000001
STATE_M_ALTGRBG               = 00000004
STATE_M_CANCEL                = 00000008
STATE_M_CMAP                  = 00000002
STATE_M_FDT                   = 00000400
STATE_M_NOTPROC               = 0C001000
STATE_M_RMAP                  = 00000001
STATE_M_SIO                   = 00000800
STATE_M_STOP                  = 00000200
STATE_V_ALTGRBG               = 00000002
STATE_V_CANCEL                = 00000003
STATE_V_CMAP                  = 00000001
STATE_V_FDT                   = 0000000A
STATE_V_NOTPROC               = 0000000C
STATE_V_RMAP                  = 00000000
STATE_V_SIO                   = 0000000B
STATE_V_STOP                  = 00000009
UCB$M_ONLINE                  = 00000010
UCB$W_STS                     = 00000064
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$115_DRIVER	0000009E (158.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000080 (128.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
BIT...	=0000000D	67 (1)	67 (1)
CDC$K_DATA	00000014		#-118 (3) #-121 (3)
COM$DRVDEALMEM	00000000-XR		181 (3)
CX$L_LOCPRTLST	00000000-XR		147 (3)
CX$L_STATE	00000000-XR		#-167 (3)
CX$Q_MAPLST	00000000-XR		110 (3) #-111 (3) 112 (3) #-113 (3)
CX_DG_RSP_SIZ	00000000-XR		#-122 (3)
CX_END	0000009E-R	253 (6)	
CX_MSG_RSP_SIZ	00000000-XR		#-119 (3)
CX_UNIT_INIT	00000001-R	95 (3)	
CX_UNSOLINT	0000007D-R	200 (4)	
EXE$ALONONPAGED	00000000-XR		242 (5)
EXE$GL_NONPAGED	00000000-XR		239 (5)
HIPL_ALLPOOL	0000007E-R	236 (5)	150 (3)
INIT_FLAG	00000000-R	92 (3)	107 (3)
IRP$W_SIZE	=00000008		#-155 (3)
MAX_LOC_PORTS	00000000-XR		#-148 (3) #-176 (3)
MAX_NODES	00000000-XR		#-149 (3)
PR$_IPL	00000000-XR		#-241 (5)
SCS$GW_MAXDG	00000000-XR		#-120 (3)
SCS$GW_MAXMSG	00000000-XR		#-117 (3)
SIZ...	=00000001	67 (1)	67 (1)
STATE_M_ALTGRBG	=00000004	67 (1)	
STATE_M_CANCEL	=00000008	67 (1)	
STATE_M_CMAP	=00000002	67 (1)	
STATE_M_FDT	=00000400	67 (1)	
STATE_M_NOTPROC	=00001000	67 (1)	
STATE_M_RMAP	=00000001	67 (1)	
STATE_M_SIO	=00000800	67 (1)	
STATE_M_STOP	=00000200	67 (1)	
STATE_V_ALTGRBG	=00000002	67 (1)	
STATE_V_CANCEL	=00000003	67 (1)	
STATE_V_CMAP	=00000001	67 (1)	
STATE_V_FDT	=0000000A	67 (1)	
STATE_V_NOTPROC	=0000000C	67 (1)	
STATE_V_RMAP	=00000000	67 (1)	
STATE_V_SIO	=0000000B	67 (1)	
STATE_V_STOP	=00000009	67 (1)	
UCB$M_ONLINE	=00000010		#-168 (3)
UCB$W_STS	=00000064		#-169 (3)

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$CDCDEF	8	64 (1)	64 (1)
$CDRPDEF	6	56 (1)	56 (1)
$CXCDBDEF	2	65 (1)	65 (1)
$CXCDRPDEF	1	66 (1)	66 (1)
$CXSTATEDEF	1	67 (1)	67 (1)
$DEFINI	1	56 (1)	56 (1) 57 (1) 58 (1) 59 (1)
			64 (1) 65 (1) 66 (1)
$IRPDEF	7	57 (1)	57 (1)
$SCSDEF	4	58 (1)	58 (1)
$UCBDEF	16	59 (1)	59 (1)
$VIELD1	1		67 (1)
_VIELD	1		67 (1)

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	93	00:00:00.33	00:00:01.29
Command processing	104	00:00:00.63	00:00:01.41
Pass 1	365	00:00:10.72	00:00:12.83
Symbol table sort	0	00:00:01.33	00:00:02.49
Pass 2	55	00:00:01.52	00:00:02.85
Symbol table output	22	00:00:00.15	00:00:00.19
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	13	00:00:00.21	00:00:00.54
Assembler run totals	658	00:00:14.93	00:00:21.72

The working set limit was 1500 pages.
46560 bytes (91 pages) of virtual memory were used to buffer the intermediate code.
There were 50 pages of symbol table space allocated to hold 876 non-local and 5 local symbols.
255 source lines were read in Pass 1, producing 14 object records in Pass 2.
58 pages of virtual memory were used to define 20 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	0
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	4
DRB2:[SHULL.EVXCI.CXDRIVER]CXLIB.MLB;3	4
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	5
TOTALS (all libraries)	13

1033 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

```
1
1)
MACRO/LIS/CROSS CXINIT+CXLIB/LIB+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB
```

Table of contents

(1)	56	DEFINITIONS	
(2)	92	CX$MSG_IN,	Message input routine
(2)	93	CX$DG_IN,	Datagram input routine
(3)	291	CX$CONNECT_ERR,	Connection error routine
(4)	400	CLEAN_CMD,	Clean up command resources
(5)	500	SET_SEQ_NUM,	Set Response Sequence Number
(7)	532	COPY_RSP,	Copy CTP response to response buffer
(8)	593	DEAL_MSG,	Deallocate an SCS Message Buffer
(9)	622	RE_QUEUE_DG,	Queue DG buffer to free queue

```
0000     1          .TITLE  CXINPUT, DRIVER INPUT routines
0000     2          .IDENT  'V6-000'
0000     3
0000     4  ;*****************************************************************************
0000     5  ;
0000     6  ;                          COPYRIGHT (c) 1981, 1984 BY
0000     7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
0000     8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
0000     9  ;
0000    10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
0000    11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
0000    12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
0000    13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
0000    14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
0000    15  ;
0000    16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
0000    17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
0000    18  ;
0000    19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
0000    20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
0000    21  ;
0000    22  ;*****************************************************************************
0000    23  ;
0000    24  ;+
0000    25  ;
0000    26  ;
0000    27  ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
0000    28  ;
0000    29  ;
0000    30  ; ABSTRACT:      This module contains the message and datagram input routines
0000    31  ;                and the connection error routine.
0000    32  ;
0000    33  ; AUTHOR:        Richard Hennessy  27-Jul-1982
0000    34  ;                Base Systems Diagnostic Engineering
0000    35  ;
0000    36  ; MODIFIED BY:   Jim Klumpp   6-MAY-83
0000    37  ;                Base Systems Diagnostic Engineering
0000    38  ;
0000    39  ;       6-000    Dave Shull       07-July-1984
0000    40  ;                VMS V4 Modifications/Release
0000    41  ;
0000    42  ;       5-001    Dave Shull       2-Feb-1984
0000    43  ;                Modified COPY_RSP routine as per change in VMS V4 $SCSDEF
0000    44  ;                library change.  SCS$C_APPL_BASE is now 0 (versus 32)
0000    45  ;                Change ASSUME from 32 to 0
0000    46  ;
0000    47  ;       5-002    Dave Shull       9-Feb-1984
0000    48  ;                Modified use of SCS$K_APPL_BASE to SCS$K_APPL_BASE-SYS$B_PPD
0000    49  ;                as in VMS V4 all SCS$ symbols reference from the Application
0000    50  ;                Data area versus Packet Pointer.
0000    51  ;
0000    52  ;--
0000    53
0000    54          .DEFAULT DISPLACEMENT,WORD
0000    55          .ENABLE  SUPPRESSION
0000    56          .SBTTL   DEFINITIONS
0000    57
```

```
                     58 ; Set PSECT to driver code
                     59
00000000             60       .PSECT  $$$115_DRIVER,LONG
                     61
                     62
                     63 ;  System definitions (LIB.MLB):
                     64
                     65          $CDRPDEF                        ; Define CDRP inputs
                     66          $CDTDEF                         ; Define CDT offsets
                     67          $DYNDEF                         ; Define DYN offsets
                     68          $IRPDEF                         ; Define IRP offsets
                     69          $PDTDEF                         ; Define PDT offsets
                     70          $SCSDEF                         ; Define SCS packet offsets
                     71          $SSDEF                          ; Define system status symbols
                     72
                     73
                     74 ;  PADRIVER definitions (PALIB.MLB):
                     75
                     76          $PAPDTDEF                       ; Define PAPDT offsets
                     77          $PAREGDEF                       ; Define PAREG offsets
                     78          $PPDDEF                         ; Define PPD offsets
                     79
                     80
                     81 ;  CXDRIVER definitions (CXLIB.MLB):
                     82
                     83          $CDCDEF                         ; Define CDC interface symbols
                     84          $CINTREGDEF                     ; Define CINT regsiters
                     85          $CTPDEF                         ; Define CTP symbols
                     86          $CXCDBDEF                       ; Define CXCDB offsets
                     87          $CXCDRPDEF                      ; Define CX CDRP extension
                     88          $CXSTATEDEF                     ; Define CXDRIVER states
                     89
                     90
```

```
                     0000    92          .SBTTL  CX$MSG_IN,      Message input routine
                     0000    93          .SBTTL  CX$DG_IN,       Datagram input routine
                     0000    94 ;+
                     0000    95 ;
                     0000    96 ;  CX$MSG_IN
                     0000    97 ;  CX$DG_IN
                     0000    98 ;
                     0000    99 ;  This routine is entered when a message or datagram is received over one
                     0000   100 ;  of the class drivers' connections. The flow is as follows:
                     0000   101 ;
                     0000   102 ;       - Validate the command reference number in the received MSG/DG
                     0000   103 ;       - Update response received count
                     0000   104 ;       - Copy message buffer info into response buffer
                     0000   105 ;       - Check to see if the command is done
                     0000   106 ;       - If so, complete I/O by JMPing to REQCOM
                     0000   107 ;
                     0000   108 ;  INPUTS:
                     0000   109 ;
                     0000   110 ;       IPL             - IPL$_SCS (fork IPL)
                     0000   111 ;
                     0000   112 ;       R1              - Length of application data in MSG/DG buffer
                     0000   113 ;       R2              - MSG/DG buffer address
                     0000   114 ;       R3              - CDT address
                     0000   115 ;       R4              - PDT address
                     0000   116 ;
                     0000   117 ;  Outputs
                     0000   118 ;
                     0000   119 ;
                     0000   120 ;       R0-R2           - Destroyed
                     0000   121 ;
                     0000   122 ;-
                     0000   123
                     0000   124 CX$MSG_IN::
                     0000   125 CX$DG_IN::
                     0000   126
                     0000   127 ;  Check for the presence of a response buffer. If one is not present,
                     0000   128 ;  deallocate this message buffer and return.
                     0000   129
                     0000   130          BBCW    #STATE_V_CMAP,-         ; Branch if no command buffer present
                     0000   131                  CX$L_STATE,100$         ;
                     0009   132
                     0009   133 ;  Calculate the address of a POSSIBLE corresponding command. Check to see that
                     0009   134 ;  the response is in fact a response to one of the current commands in the
                     0009   135 ;  command buffer.
                     0009   136
          50   01 A2  3C  0009   137          MOVZWL  CTP$REFERENCE(R2),R0    ; Get command offset from reference #
55   50   0000'CF  C1  000D   138          ADDL3   CX$L_CMDBUFBGN,R0,R5    ; Get address of possible command
     0000'CF   55  D1  0013   139          CMPL    R5,CX$L_CMDBUFEND       ; Is address outside command buffer?
                       0018   140          BGEQUW  200$                    ; Yes, error, branch
     0000'CF   55  D1  001D   141          CMPL    R5,CX$L_CMDBUFBGN       ;
                       0022   142          BLEQUW  200$                    ; Yes, error, branch
          01 A2  D1  0027   143          CMPL    CTP$REFERENCE(R2),-     ; Does reference number match that
          15 A5      002A   144                  CDC$L_CTPREF(R5)        ; in the command buffer?
                     002C   145          BNEQW   200$                    ; No, error, branch
                     0031   146
                     0031   147 ;  Check to see if the command is done. If so, this is an extra response to
                     0031   148 ;  the command.
```

```
                        0031   149
                        0031   150          BBSW    #FLAG_V_CMD_DONE,-      ; If command is already done, branch
                        0031   151                  CDC$W_FLAG(R5),250$      ;
                        0039   152
                        0039   153 ;  Check if command wants to SAVE ALL received responses.
                        0039   154
          01    E0      0039   155          BBS     #FLAG_V_SAVE_ALL,-      ; Save all responses, branch
    06 04 A5            003B   156                  CDC$W_FLAG(R5),10$      ;
                        003E   157
                        003E   158 ;  If we don't want to save all responses, then overlay each response that
                        003E   159 ;  arrives at the beginning of the response slot for this command. Otherwise,
                        003E   160 ;  we must calculate the position in the response slot by determining the
                        003E   161 ;  number of responses already received.
                        003E   162
    54  02 A5   3C      003E   163          MOVZWL  CDC$W_RSPOFSET(R5),R4   ; Get starting offset of response slot
          1A    11      0042   164          BRB     30$                     ; Continue
                        0044   165
                        0044   166 ;  The control process wants all responses to be saved. The proper location
                        0044   167 ;  to save this response is dependent upon the number of responses already
                        0044   168 ;  received, and the size of the expected responses.
                        0044   169
    0000'CF   A5        0044   170 10$:    MULW3   CX_MSG_RSP_SIZ,-        ; Assume messages expected. Calculate
    54  06 A5           0048   171                  CDC$W_RSPCNT(R5),R4     ; offset into response slot
          02    E1      004B   172          BBC     #FLAG_V_TYP_PKT_REC,-   ; Assumption correct, branch
    07 04 A5            004D   173                  CDC$W_FLAG(R5),20$      ;
    0000'CF   A5        0050   174          MULW3   CX_DG_RSP_SIZ,-         ; Data rams expected. Calculate offset
    54  06 A5           0054   175                  CDC$W_RSPCNT(R5),R4     ; into response slot
                        0057   176
                        0057   177 ;  Add the offset of the response slot from the top of the response buffer to
                        0057   178 ;  the offset into the response slot (just calculated) to obtain the total
                        0057   179 ;  offset from the top of the response buffer to the location where the current
                        0057   180 ;  response belongs.
                        0057   181
  54    02 A5   A0      0057   182 20$:    ADDW2   CDC$W_RSPOFSET(R5),R4   ; Get TOTAL offset into response buffer
    54  54      3C      005B   183          MOVZWL  R4,R4                   ; Change to longword
                        005E   184
                        005E   185 ;  Finally, by adding the address of the top of the response buffer to the
                        005E   186 ;  total offset into the response buffer for this response, we get the address
                        005E   187 ;  where this response belongs.
                        005E   188
54  0000'CF   C0        005E   189 30$:    ADDL2   CX$L_RSPBUFBGN,R4       ; Get address for this response
                        0063   190
                        0063   191 ;  Allocate a response sequence number and mark the relative order of receipt
                        0063   192 ;  of this response. Then copy the MSG or DG into the response buffer.
                        0063   193
        0180    30      0063   194          BSBW    SET_SEQ_NUM             ; Set sequence number for this response
        0188    30      0066   195          BSBW    COPY_RSP                ; Copy MSG or DG into response buffer
                        0069   196
                        0069   197 ;  Deallocate the buffer used for the response. For datagram, which have no
                        0069   198 ;  flow control, queue this buffer to the free queue for receipt of more
                        0069   199 ;  responses. For messages, let SCS dispose of the buffer.
                        0069   200
50  52    20    C3      0069   201          SUBL3   #SCS$K_APPL_BASE - SCS$B_PPD,R2,R0
                        006D   202                                          ; Back up to PPD header in buffer[5-002]
          3B    91      006D   203          CMPB    #DYN$C_CIDG,-           ; Is this response a datagram?
    0A A0               006F   204                  PPD$B_TYPE(R0)          ;
          05    13      0071   205          BEQLU   40$                     ; Yes, branch
```

```
                 018A   30   0073   206          BSBW    DEAL_MSG                ; Return MSG buf to SCS
                   03   11   0076   207          BRB     50$                     ;
                 018D   30   0078   208 40$:     BSBW    RE_QUEUE_DG             ; Return DG to DG free queue
                        0078   209
               06 A5    B6   007B   210 50$:     INCW    CDC$W_RSPCNT(R5)        ; Update response received count
                        007E   211
                        007E   212 ;  CTP generate message and datagram requests require one more response to
                        007E   213 ;  be returned than is specified by the repeat count, CTP$REPEAT. This is
                        007E   214 ;  due to the fact that the last response sent is considered a CTP protocol
                        007E   215 ;  MSG/DG rather than part of the requested action specified in the request.
                        007E   216
                   04   91   007E   217          CMPB    #CTP$GENMSGREQ,-        ; Is this a GENMSG CTP opcode?
                14 A5        0080   218                  CDC$B_CTPOPCODE(R5)      ;
                   06   13   0082   219          BEQLU   60$                     ; Yes, branch
                   05   91   0084   220          CMPB    #CTP$GENDGRREQ,-        ; Is this a GENDG CTP opcode?
                14 A5        0086   221                  CDC$B_CTPOPCODE(R5)      ;
                   0B   12   0088   222          BNEQU   70$                     ; No, branch
          50   1A A5   01   A1   008A   223 60$:     ADDW3   #1,CDC$W_REPEAT(R5),R0  ; Expect one extra response
               06 A5   50   B1   008F   224          CMPW    R0,CDC$W_RSPCNT(R5)     ; Have all responses been received?
                   24   12   0093   225          BNEQU   80$                     ; No, branch
                        0095   226
                        0095   227 ;  All responses for this command have been received. This command is finished.
                        0095   228 ;  Deallocate the CDRP since we are done with it.
                        0095   229
          50   0C A5   D0   0095   230 70$:     MOVL    CDC$L_CDRP(R5),R0       ; Get CDRP address
               0C A5   D4   0099   231          CLRL    CDC$L_CDRP(R5)          ; Show CDRP is gone
        00000000'GF    16   009C   232          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDRP
                        00A2   233          SET_CMD_FLAG CMD_DONE,REG=R5    ; Mark this command as done
          55   0000'CF   D0   00A8   234          MOVL    CX$L_CMDBUFBGN,R5       ; Point to top of command buffer
                01 A5   96   00AD   235          INCB    CDC$B_DONECNT(R5)       ; One more command done
                01 A5   91   00B0   236          CMPB    CDC$B_DONECNT(R5),-     ; Are all commands done?
                   65        00B3   237                  CDC$B_CMDCOUNT(R5)      ;
                             00B4   238          BEQLW   SIO_COMPLETE            ; Yes, branch to complete the I/O
                   05   00B9   239 80$:     RSB                             ; Wait for more input
                        00BA   240
                        00BA   241
                        00BA   242 ;  COMMAND BUFFER NOT MAPPED
                        00BA   243 ;  CONNECTION IS IN TRANSITION STATE
                        00BA   244 ;  COMMAND REFERENCE NUMBER DOES NOT MATCH
                        00BA   245 ;  EXTRA RESPONSE RECEIVED
                        00BA   246 ;
                        00BA   247 ;  Input was received over a connection, but no command buffer was mapped.
                        00BA   248 ;  Discard the received MSG or DG buffer.
                        00BA   249 ;
                        00BA   250 ;  The connection over which the message was received is in a state of
                        00BA   251 ;  transition.
                        00BA   252 ;
                        00BA   253 ;  The command reference number in the received response does not match
                        00BA   254 ;  the CRN in the command.
                        00BA   255
                        00BA   256 100$:
                        00BA   257 150$:
                        00BA   258 200$:
                        00BA   259
          50   52   20   C3   00BA   260          SUBL3   #SCS$K_APPL_BASE - SCS$B_PPD,R2,R0
                        00BE   261                                               ; Back up to PPD header in buffer[5-002]
        0000'CF   00   E2   00BE   262          BBSS    #0,CX$B_GARBAGE_FLAG,-  ; Branch if garbage buffer full
```

```
                0C      00C3            105$
                3F  BB  00C4     PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
   60    0080 8F    28  00C6     MOVC3   #128,(R0),-             ; Copy response to garbage buffer
         0000'CF        00CB             CX$T_GARBAGE_BUF
                3F  BA  00CE     POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
                        00D0
                3B  91  00D0 105$: CMPB  #DYN$C_CIDG,-           ; Is this response a datagram?
             0A A0      00D2             PPD$B_TYPE(R0)          ;
                04  13  00D4     BEQLU   110$                    ; Yes, branch
              0127  30  00D6     BSBW    DEAL_MSG                ; Return MSG buf to SCS
                    05  00D9     RSB
                        00DA
              012B  30  00DA 110$: BSBW  RE_QUEUE_DG             ; Return DG to DG free queue
                    05  00DD     RSB
                        00DE
                        00DE
                        00DE ;  EXTRA RESPONSE RECEIVED
                        00DE ;
                        00DE ;  An extra response was received for this command. Mark the appropriate
                        00DE ;  status in the command, then discard this MSG/DG buffer.
                        00DE
                        00DE 250$: SET_CMD_FLAG EXTRA_INPUT      ; Indicate extra input received
                D4  11  00E4     BRB     100$
                        00E6
                        00E6
                        00E6
                        00E6
```

```
00E6   291          .SBTTL  CX$CONNECT_ERR, Connection error routine
00E6   292 ;+
00E6   293 ;
00E6   294 ;  CX$CONNECT_ERR
00E6   295 ;
00E6   296 ;  This routine is invoked by SCS when a connection is going down. Connections
00E6   297 ;  can go down for any of the following reasons:
00E6   298 ;
00E6   299 ;       - The virtual circuit breaks
00E6   300 ;       - Remote process issues a disconnect (should never happen)
00E6   301 ;
00E6   302 ;  This routine handles the broken connection by performing the following
00E6   303 ;  actions:
00E6   304 ;
00E6   305 ;  IF A COMMAND BUFFER IS PRESENT:
00E6   306 ;
00E6   307 ;       - Marks any commands for the broken connection as CONN_ERR
00E6   308 ;       - Copies the broken status into the NEXT response area associated
00E6   309 ;         with the each cmd using the broken connection
00E6   310 ;       - Cleans up any CDRPs used by errored commands
00E6   311 ;       - If the command buffer is completed, finish the QIO
00E6   312 ;
00E6   313 ;  IN ALL CASES:
00E6   314 ;
00E6   315 ;       - Cleans up any other active commands on this connection
00E6   316 ;       - Issues a disconnect to SCS
00E6   317 ;       - Deallocates the CDB
00E6   318 ;       - RSB back to SCS connection error code
00E6   319 ;
00E6   320 ;  INPUTS:
00E6   321 ;
00E6   322 ;       R0              - Status (SS$_DISCONNECT, SS$_VCBROKEN)
00E6   323 ;       R1              - Secondary status (Disconnect reason or
00E6   324 ;                         SS$_CNTRLERR, SS$_ABORT, SS$_POWERFAIL)
00E6   325 ;       R3              - CDT address
00E6   326 ;       R4              - PDT address
00E6   327 ;
00E6   328 ;  OUTPUTS:
00E6   329 ;
00E6   330 ;       R0-R5           - Destroyed
00E6   331 ;
00E6   332 ;-
00E6   333
00E6   334 CX$CONNECT_ERR::
00E6   335
00E6   336 ;*****************************************************************************
00E6   337 ;
00E6   338 ;  SOFTWARE DEBUG
00E6   339 ;
00E6   340 ;  This driver initiates all disconnections. Thus, if things are working
00E6   341 ;  properly, this routine should never be entered.
00E6   342 ;
00E6   343 ;       JSB     G^INI$BRK
00E6   344 ;
00E6   345 ;  END SOFTWARE DEBUG
00E6   346 ;
00F6   347 ;*****************************************************************************
```

```
                                00E6    348
                                00E6    349          ASSUME  CDC$B_CMDCOUNT EQ 0
                                00E6    350
                     08   BB    00E6    351          PUSHR   #^M<R0,R1,R3>            ; Save regs
                     01   E1    00E8    352          BBC     #STATE_V_CMAP,-         ; Branch if no command buffer present
                  25 0000'CF          00EA    353                  CX$L_STATE,CLEAN_CONN
                53   0000'CF   D0    00EE    354          MOVL    CX$L_CMDBUFBGN,R3       ; Get start of command buffer
                  63   01 A3   91    00F3    355          CMPB    CDC$B_DONECNT(R3),(R3)  ; All commands done?
                       1A      13    00F7    356          BEQL    CLEAN_CONN              ; Yes, START I/O will finish the I/O
                               00F9    357
                               00F9    358
                               00F9    359 CLEAN_CMD_BUF:
                               00F9    360
                       52      94    00F9    361          CLRB    R2                      ; Clear counter
                          53   0C    C0    00FB    362          ADDL    #CDC$K_CMDHDRSIZ,R3     ; Get first cmd in cmd buffer
                             0049    30    00FE    363 10$:     BSBW    CLEAN_CMD               ; Clean command if for this connection
                       52      96    0101    364          INCB    R2                      ; One more command handled
                0000'DF    52      91    0103    365          CMPB    R2,@CX$L_CMDBUFBGN      ; Have all commands been handled?
                       09      13    0108    366          BEQL    CLEAN_CONN              ; Yes, branch
                  50   08 A3   3C    010A    367          MOVZWL  CDC$W_TXTLEN(R3),R0     ; Get length of this command
                     53   50   C0    010E    368          ADDL    R0,R3                   ; Get address of next command
                       EB      11    0111    369          BRB     10$                     ; Try next command
                               0113    370
                               0113    371
                               0113    372 CLEAN_CONN:
                               0113    373
                       0B      BA    0113    374          POPR    #^M<R0,R1,R3>           ; Restore regs
                55   5C A3     D0    0115    375          MOVL    CDT$L_AUXSTRUC(R3),R5   ; Get CDB address
                       02      A8    0119    376          BISW2   #CDB_M_DISC,-           ; Show DISCONNECT in progress for this
                     0C A5           011B    377                  CDB$W_STATUS(R5)        ; connection
                               011D    378          DISCONNECT                      ; Issue the disconnect
                     24 B5     D4    0123    379          CLRL    @CDB$L_CDBSLOT(R5)      ; Remove CDB from connection database
                  50   55     D0    0126    380          MOVL    R5,R0                   ; Get CDB address
            00000000'GF      16    0129    381          JSB     G^COM$DRVDEALMEM        ; Deallocate the CDB
                               012F    382
                               012F    383
                               012F    384 FINISH_IO:
                               012F    385
                       01      E1    012F    386          BBC     #STATE_V_CMAP,-         ; Branch if command buffer was NOT
                14 0000'CF           0131    387                  CX$L_STATE,10$          ; present
            50   0000'CF     D0    0135    388          MOVL    CX$L_CMDBUFBGN,R0       ; Get start of command buffer
                    01 A0    81    013A    389          ADDB3   CDC$B_DONECNT(R0),-     ; Get total commands done, failed
            51   0000'CF           013D    390                  CX$B_CMDERRCNT,R1       ;
                 60   51     91    0141    391          CMPB    R1,(R0)                 ; All command done or failed?
                       03      12    0144    392          BNEQ    10$                     ; No, branch
                          FEB7'    30    0146    393          BSBW    SIO_COMPLETE            ; Finish off the QIO
                               05    0149    394 10$:     RSB                             ; Return to SCS
                               014A    395
                               014A    396
                               014A    397
                               014A    398
```

```
                                014A  400          .SBTTL  CLEAN_CMD,      Clean up command resources
                                014A  401 ;+
                                014A  402 ;
                                014A  403 ;  CLEAN_CMD
                                014A  404 ;
                                014A  405 ;  This routine detemines if the specified command is associated with the
                                014A  406 ;  failed connection, and cleans up the command if appropriate.
                                014A  407 ;
                                014A  408 ;  Commands which are already done or have already resulted in error are
                                014A  409 ;  ignored.
                                014A  410 ;
                                014A  411 ;  Connection type commands (CONNECT, DISCONNECT) are ignored.
                                014A  412 ;
                                014A  413 ;  Buffer transfer type commands (CDRP$L_LBUFH_AD <> 0) are handled somewhat
                                014A  414 ;  differently. The CDRP is NOT deallocated. Instead, the CDRP is returned to
                                014A  415 ;  the mapped buffer queue.
                                014A  416 ;
                                014A  417 ;  INPUTS:
                                014A  418 ;
                                014A  419 ;       R3                      - Command address
                                014A  420 ;       (SP)                    - Return address
                                014A  421 ;       4(SP)                   - R0 connection error status
                                014A  422 ;       8(SP)                   - R1 connection error status
                                014A  423 ;       12(SP)                  - CDT address of failed connection
                                014A  424 ;
                                014A  425 ;  OUTPUTS:
                                014A  426 ;
                                014A  427 ;       R0-R1,R5                - Destroyed
                                014A  428 ;
                                014A  429 ;-
                                014A  430
                                01.A  431 CLEAN_CMD:
                                014A  432
                          B3    014A  433          BITW    #<FLAG_M_CMD_DONE-      ; Check command for done,
                                014B  434                    !FLAG_M_SCS_ERR-      ; SCS error,
                                014B  435                    !FLAG_M_CONN_LOST>,-  ; or connection error
04 A3   9800 8F                 014B  436                    CDC$W_FLAG(R3)        ; in the flags word
                                0150  437          BNEQW   40$                   ; One of these conditions exist, branch
                                0155  438
                      63  91    0155  439          CMPB    CDC$B_OPCODE(R3),-    ; Connect command?
                      08        0157  440                  #CDC$_CONNECT
                                0158  441          BEQLW   40$                   ; Yes, branch
                      63  91    015D  442          CMPB    CDC$B_OPCODE(R3),-    ; Disconnect command?
                      09        015F  443                  #CDC$_DISCONNECT
                                0160  444          BEQLW   40$                   ; Yes, branch
                                0165  445
                  FE98'   30    0165  446          BSBW    CHECK_CONN            ; Get possible CDB address
                7A 50     E9    0168  447          BLBC    R0,40$                ; No connection, branch
                                016B  448
0C AE           1C A1     D1    016B  449          CMPL    CDB$L_CDT(R1),12(SP)  ; Is command for failed connection?
                   73     12    0170  450          BNEQ    40$                   ; No, branch
                                0172  451
                                0172  452          SET_CMD_FLAG CONN_LOST        ; Mark this command as connection lost
              0000'CF   96      0178  453          INCB    CX$B_CMDERRCNT        ; One more command resulted in error
                                017C  454
                      50  B4    017C  455          CLRW    R0                    ; Assume NOT saving all response
                      01  E1    017E  456          BBC     #FLAG_V_SAVE_ALL,-    ; Branch if NOT saving all responses
```

```
                13 04 A3        0180   457                  CDC$W_FLAG(R3),10$
                 0000'CF   A5   0183   458          MULW3   CX_MSG_RSP_SIZ,-          ; Assume messages, get offset to
          50       06 A3        0187   459                  CDC$W_RSPCNT(R3),R0       ; appropriate response slot
                      02   E1   018A   460          BBC     #FLAG_V_TYP_PKT_REC,-     ; Messages ARE expected, branch
              07 04 A3         018C   461                  CDC$W_FLAG(R3),10$
                 0000'CF   A5   018F   462          MULW3   CX_DG_RSP_SIZ,-           ; Get offset to appropriate response
          50       06 A3        0193   463                  CDC$W_RSPCNT(R3),R0       ; slot for datagrams
          50       02 A3   A0   0196   464 10$:     ADDW2   CDC$W_RSPOFSET(R3),R0    ; Get offset for this command
             50    50      3C   019A   465          MOVZWL  R0,R0                     ; Make word into longword
       50    0000'CF      C0   019D   466          ADDL2   CX$L_RSPBUFBGN,R0         ; Get address of appropriate rsp slot
    0C A0      04 AE      7D   01A2   467          MOVQ    4(SP),CDC$L_ROSTATUS(R0); Copy R0,R1 status to response buffer
                          01A7   468
          55    0C A3      D0   01A7   469          MOVL    CDC$L_CDRP(R3),R5         ; Get CDRP for this command
                   38      13   01AB   470          BEQL    40$                       ; No CDRP, branch
                          01AD   471
                    FE50'  30   01AD   472          BSBW    CHECK_LINK_FQ             ; Is CDRP on SCS queue?
                 09 50     E9   01B0   473          BLBC    R0,20$                    ; No, branch
             50    65      0F   01B3   474          REMQUE  CDRP$L_FQFL(R5),R0        ; Remove CDRP from SCS queue
                          01B6   475          SET_CMD_FLAG REC_WAIT                     ; Show command was resource waited
                          01BC   476
                    FE41'  30   01BC   477 20$:     BSBW    CHECK_LINK_PQ             ; Is CDRP on CDB pending queue?
                 0A 50     E9   01BF   478          BLBC    R0,30$                    ; No, branch
          50    44 A5      0F   01C2   479          REMQUE  CDRP$L_PENDQFL(R5),R0     ; Remove CDRP from SCS queue
                          01C6   480          SET_CMD_FLAG REC_WAIT                     ; Show command was resource waited
                          01CC   481
                 2C A5      D5   01CC   482 30$:     TSTL    CDRP$L_LBUFH_AD(R5)       ; Buffer XFR type command?
                    0B      12   01CF   483          BNEQ    35$                       ; Yes branch
                  FE2C'     30   01D1   484          BSBW    CLEAN_CDRP                ; Release CDRP resources
                  FE29'     30   01D4   485          BSBW    DEALLOC_CDRP              ; Deallocate the CDRP
                 0C A3      D4   01D7   486          CLRL    CDC$L_CDRP(R3)            ; Show CDRP is gone
                    09      11   01DA   487          BRB     40$
                          01DC   488
              44 A5      0E   01DC   489 35$:     INSQUE  CDRP$L_PENDQFL(R5),-      ; Requeue mapped buffer CDRP to
    0000'CF               01DF   490                  CX$Q_MAPLST                   ; mapped buffer list
              0C A3      D4   01E2   491          CLRL    CDC$L_CDRP(R3)            ; Show CDRP is gone
                          01E5   492
                          01E5   493 40$:     RSB
                          01E6   494
                          01E6   495
                          01E6   496
                          01E6   497
                          01E6   498
```

```
                        01E6    500          .SBTTL  SET_SEQ_NUM,    Set Response Sequence Number
                        01E6    501 ;+
                        01E6    502 ;
                        01E6    503 ;  SET_SEQ_NUM
                        01E6    504 ;
                        01E6    505 ;  This routine writes the current value of the response sequence number
                        01E6    506 ;  to the response slot, then increments the response sequence number for
                        01E6    507 ;  the next response that may be received. This is used to by the control
                        01E6    508 ;  process to verify the responses are received in the correct order when
                        01E6    509 ;  sent over virtual circuits.
                        01E6    510 ;
                        01E6    511 ;  INPUTS:
                        01E6    512 ;
                        01E6    513 ;       R4              - Address of response slot for this response
                        01E6    514 ;
                        01E6    515 ;  OUTPUTS:
                        01E6    516 ;
                        01E6    517 ;       All registers preserved
                        01E6    518 ;
                        01E6    519 ;-
                        01E6    520
                        01E6    521 SET_SEQ_NUM::
                        01E6    522
0000'CF   D0            01E6    523          MOVL    CX$L_RSPSEQNUM,-        ; Move the sequence number
  01 A4                 01EA    524                  CDC$[_SEQNUM(R4)        ; to the response buffer slot
0000'CF   B6            01EC    525          INCW    CX$L_RSPSEQNUM          ; Create next incarnation
          05            01F0    526          RSB
                        01F1    527
                        01F1    528
                        01F1    529
```

B 1 Documentation
C 1 Documentation
D 1 Documentation
E 1 Map
F 1 Map
G 1 Map
H 1 Map
I 1 Map
J 1 Map
K 1 Map
L 1 Map
M 1 Map
N 1 Map
B 2 Map
C 2 Map
D 2 Map
E 2 Map
F 2 Map
G 2 Map
H 2 Map
I 2 Map
J 2 Map
K 2 Map
L 2 Map
M 2 Map
N 2 Map
B 3 MACRO DEFINITIONS
C 3 MACRO DEFINITIONS
D 3 $CXQIODEF, CXDRIVER QIO interf
E 3 $CXSTATEDEF, CXDRIVER internal
F 3 $CXCDRPDEF, CXDRIVER CDRP exte
G 3 $CXCDBDEF, Class driver data b
H 3 $CINTREGDEF, CINT register def
I 3 $CINTREGDEF, CINT register def
J 3 $CINTREGDEF, CINT register def
K 3 $CINTREGDEF, CINT register def
L 3 $CINTREGDEF, CINT register def
M 3 $CINTREGDEF, CINT register def
N 3 $CINTREGDEF, CINT register def
B 4 $CINTREGDEF, CINT register def
C 4 Psect synopsis
D 4 , Cancel I/O Module
E 4 , Cancel I/O Module
F 4 DEFINITIONS
G 4 CX_CANCELIO, Cancel I/O Routin
H 4 CX_CANCELIO, Cancel I/O Routin
I 4 CAN$INITCINT, Initialize any p
J 4 CLEAN_RST_STR, Clean possible
K 4 CLEAN_PORT_STATE,Return port t
L 4 CANCEL_CMD_BUF, Cancel all com
M 4 CANCEL_CMD_BUF, Cancel all com
N 4 FINISH_IO, Complete the origin
B 5 CANCEL_MAPPED_BUFFERS, Unmap a
C 5 CANCEL_CONNECTIONS, Clean up a
D 5 START_POLLER, Restart the conf
E 5 KILL_CDB, Clean Up A Connectio
F 5 CLEAN_CDB, Clean pending CDRP'
G 5 CHECK_LINK_FQ, Check for CDRP
H 5 CHECK_LINK_PQ, Check for CDRP
I 5 CLEAN_CDRP, Deallocate CDRP re
J 5 DEALLOC_CDRP, Deallocate a CDR
K 5 Symbol table
L 5 Symbol table
M 5 Symbol table
N 5 Psect synopsis
B 6 Cross reference
C 6 Cross reference
D 6 Cross reference
E 6 Cross reference
F 6 Command Start I/O
G 6 Command Start I/O
H 6 Command Start I/O
I 6 DEFINITIONS
J 6 CX_STARTIO, Start I/O routine
K 6 CX_STARTIO, Start I/O routine
L 6 CX_STARTIO, Start I/O routine
M 6 CX_STARTIO, Start I/O routine
N 6 CX_STARTIO, Start I/O routine
B 7 CX_STARTIO, Start I/O routine
C 7 SIO_CONFIG, Gather configurati
D 7 SIO_CONFIG, Gather configurati
E 7 SIO_CONNECT, Connect to remote
F 7 SIO_CONNECT, Connect to remote
G 7 SIO_DISC, Disconnect from remo
H 7 SIO_DISC, Disconnect from remo
I 7 SIO_MSG, Send message
J 7 SIO_MSG, Send message
K 7 SIO_DG, Send datagram
L 7 SIO_MAP, Map buffer
M 7 SIO_MAP, Map buffer
N 7 SIO_UNMAP, Unmap buffer
B 8 SIO_SNDDAT Send data
C 8 SIO_REQDAT, Request data
D 8 SIO_SNDMDAT, Send maintenance
E 8 SIO_READCNT, Read port counter
F 8 SIO_READCNT, Read port counter
G 8 SIO_RESET, Send reset
H 8 SIO_START, Send start
I 8 SIO_GARBAGE, Read the garbage
J 8 SIO_MOVCTP, Copy CTP data to M
K 8 SIO_SETUP_SCS, Setup SCS inter
L 8 MARK_CMD_DONE, Mark this comma
M 8 CHECK_VC, Check for virtual ci
N 8 XFR_SETUP, Setup data XFR para
B 9 XFR_SETUP, Setup data XFR para
C 9 MNT_XFR_SETUP, Setup maintenan
D 9 MNT_XFR_SETUP, Setup maintenan
E 9 FIND_CDRP_XFR, Locate XFR CDRP
F 9 FIND_CDRP_XFR, Locate XFR CDRP
G 9 ALLOC_CDRP_CTP, Allocate a CTP
H 9 ALLOC_CDRP_XFR, Allocate a CDR
I 9 ALLOC_CDRP_XFR, Allocate a CDR
J 9 ALLOC_CDB, Allocate connection
K 9 ALLOC_CDB, Allocate connection
L 9 SET_RSP_STATUS, Set response s
M 9 CHECK_RST_STR, Prepare for arr
N 9 CATCH_RST_STR, Catch reset or
B 10 CATCH_RST_STR, Catch reset or
C 10 Symbol table
D 10 Symbol table
E 10 Symbol table
F 10 Symbol table
G 10 Symbol table
H 10 Symbol table
I 10 Symbol table
J 10 Symbol table
K 10 Symbol table
L 10 Psect synopsis
M 10 Cross reference
N 10 Cross reference
B 11 Cross reference
C 11 Cross reference
D 11 Cross reference
E 11 Cross reference
F 11 Cross reference
G 11 Cross reference
H 11 Cross reference
I 11 , STARTDATA FDT Routines
J 11 , STARTDATA FDT Routines
K 11 , STARTDATA FDT Routines
L 11 CX_STARTDATA, Start command li
M 11 CX_STARTDATA, Start command li
N 11 CX_STARTDATA, Start command li
B 12 FDT_DISPATCH, Check validity o
C 12 FDT_DISPATCH, Check validity o
D 12 FDT_DISPATCH, Check validity o
E 12 FDT_ABORT, Abort the I/O for v
F 12 FDT_ABORT, Abort the I/O for v
G 12 FDT_CONFIG, Gather configurati
H 12 FDT_CONNECT, Establish connect
I 12 FDT_DISCONNECT, Disconnect rou
J 12 FDT_SENDMSG, Send message
K 12 FDT_SENDDG, Send datagram
L 12 FDT_MAP, Map buffer
M 12 FDT_MNTMAP, Map a maintenance
N 12 FDT_UNMAP, Unmap buffer
B 13 FDT_SNDDAT, Send data
C 13 FDT_SNDMDAT, Send maintenance
D 13 FDT_RDCNT, Read performance co
E 13 FDT_RESET, Send reset
F 13 FDT_POLLER, Start/Stop poller
G 13 CHECK_LOC_PORT, Check local po
H 13 ALLOC_RSP_SCS, Allocate SCS re
I 13 ALLOC_RSP_CTP, Allocate CTP re
J 13 FILL_CMD_REF, Build a command
K 13 CHECK_CONN, Check for connecti
L 13 CHECK_ACTV_CONN, Check for an
M 13 CLEAR_BUFFER, Clear buffer rou
N 13 ALLOC_SEQNUM, Create a sequenc
B 14 Symbol table
C 14 Symbol table
D 14 Symbol table
E 14 Symbol table
F 14 Cross reference
G 14 Cross reference
H 14 Cross reference
I 14 Cross reference
J 14 Cross reference
K 14 Cross reference
L 14 Initialization and Misc. Routi

M 14 Initialization and Misc. Routi
N 14 DEFINITIONS
B 15 DEFINITIONS
C 15 UNIT INITIALIZATION ROUTINE
D 15 UNIT INITIALIZATION ROUTINE
E 15 UNSOLICITED INTERRUPT ROUTINE
F 15 HIPL_ALLPOOL
G 15 CX_END, The END of CXDRIVER
H 15 Symbol table
I 15 Symbol table
J 15 Psect synopsis
K 15 Cross reference
L 15 Cross reference
M 15 Cross reference
N 15 , DRIVER INPUT routines
B 16 , DRIVER INPUT routines
C 16 DEFINITIONS
D 16 CX$MSG_IN, Message input routi
E 16 CX$DG_IN, Datagram input routi
F 16 CX$DG_IN, Datagram input routi
G 16 CX$DG_IN, Datagram input routi
H 16 CX$CONNECT_ERR, Connection err
I 16 CX$CONNECT_ERR, Connection err
J 16 CLEAN_CMD, Clean up command re
K 16 CLEAN_CMD, Clean up command re
L 16 SET_SEQ_NUM, Set Response Sequ

```
                        01F1    531
                        01F1    532          .SBTTL  COPY_RSP,       Copy CTP response to response buffer
                        01F1    533 ;+
                        01F1    534 ;
                        01F1    535 ;  COPY_RSP
                        01F1    536 ;
                        01F1    537 ;  This routine copies the data contained in a received message or datagram
                        01F1    538 ;  the the response buffer. The data copied is shown in the diagram below.
                        01F1    539 ;
                        01F1    540 ;  A Message/Datagram has the general form:
                        01F1    541 ;
                        01F1    542 ;                  +-------+-------+-------+-------+ <-----+
                        01F1    543 ;                  |      Command queue FLINK      |       |
                        01F1    544 ;                  +-------------------------------+       |
                        01F1    545 ;  DATA COPIED      |      Command queue BLINK      |       |
                        01F1    546 ;      to          +-------------------------------+       |
                        01F1    547 ;  RESPONSE BUF    |SW flag| Type  |  Struc size   |       | Port envelope
                        01F1    548 ;        +------>  +-------------------------------+       |
                        01F1    549 ;        |         | Flags |  OPC  |Status | Port #|       |
                        01F1    550 ;        |         +-------------------------------+       |
                        01F1    551 ;        |         |     unused    /  Msg length |       |
                        01F1    552 ;        +------>  +-------------------------------+ <-----+
                        01F1    553 ;                  |     Credit    | SCS msg type  |       |
                        01F1    554 ;                  +-------------------------------+       |
                        01F1    555 ;                  |     Receiver connection ID    |       | SCS envelope
                        01F1    556 ;                  +-------------------------------+       |
                        01F1    557 ;                  |     Sender connection ID      |       |
                        01F1    558 ;          +-----> +-------------------------------+ <-----+
                        01F1    559 ;          |       |                               |
                        01F1    560 ;          |       |      Application data         |
                        01F1    561 ;          |       |                               |
                        01F1    562 ;          +-----> +-------------------------------+
                        01F1    563 ;
                        01F1    564 ;
                        01F1    565 ;  INPUTS:
                        01F1    566 ;
                        01F1    567 ;       R1              - Length of appl data in message
                        01F1    568 ;       R2              - Address of appl data in message
                        01F1    569 ;       R4              - Starting address of response slot
                        01F1    570 ;
                        01F1    571 ;  OUTPUTS:
                        01F1    572 ;
                        01F1    573 ;       All registers preserved
                        01F1    574 ;
                        01F1    575 ;-
                        01F1    576
                        01F1    577 ASSUME  SCS$C_APPL_BASE  EQ 0          ;                          [5-001]
                        01F1    578 ASSUME  PPD$C_LENGTH     EQ 18
                        01F1    579 ASSUME  PPD$B_PORT       EQ 12
                        01F1    580 ASSUME  PPD$W_LENGTH     EQ 16
                        01F1    581
                        01F1    582 COPY_RSP::
                        01F1    583
                    3F  BB      01F1    584          PUSHR   #^M<R0,R1,R2,R3,R4,R5>  ; Save regs
                EC  A2  7D      01F3    585          MOVQ    -20(R2),-              ; Copy Port envelope part
                04  A4          01F6    586                  CDC$L_PORTHDR(R4)      ; (note assumptions)
          14 A4 62  51  28      01F8    587          MOVC3   R1,(R2),CDC$K_DATA(R4) ; Copy application data
```

```
       3F   BA  01FD   588        POPR    #^M<R0,R1,R2,R3,R4,R5>  ; Restore regs
            05  01FF   589        RSB
                0200   590
                0200   591
```

```
                        0200   593          .SBTTL  DEAL_MSG,       Deallocate an SCS Message Buffer
                        0200   594 ;+
                        0200   595 ;
                        0200   596 ;  DEAL_MSG
                        0200   597 ;
                        0200   598 ;  This routine performs the necessary setup, then deallocates a message
                        0200   599 ;  buffer to SCS.
                        0200   600 ;
                        0200   601 ;  INPUTS:
                        0200   602 ;
                        0200   603 ;       R2              - Address of appl data area of MSG buffer
                        0200   604 ;       R3              - CDT address
                        0200   605 ;
                        0200   606 ;  OUTPUTS:
                        0200   607 ;
                        0200   608 ;       R0-R2,R4        - Destroyed
                        0200   609 ;       All other registers preserved
                        0200   610 ;
                        0200   611 ;-
                        0200   612
                        0200   613 DEAL_MSG::
                        0200   614
    54    10 A3   D0    0200   615          MOVL    CDT$L_PDT(R3),R4        ; Get PDT address
                        0204   616          DEALLOC_MSG_BUF_REG             ; Deallocate the message buffer
                  05    0207   617          RSB
                        0208   618
                        0208   619
                        0208   620
```

```
                          0208   622          .SBTTL  RE_QUEUE_DG,    Queue DG buffer to free queue
                          0208   623 ;+
                          0208   624 ;
                          0208   625 ;  RE_QUEUE_DG
                          0208   626 ;
                          0208   627 ;  This routine queues a datagram buffer to the free queue. Used when a
                          0208   628 ;  datagram response is received, after the contents of the response have
                          0208   629 ;  been copied to the response buffer.
                          0208   630 ;
                          0208   631 ;  INPUTS:
                          0208   632 ;
                          0208   633 ;       R2              - Address of appl data area in DG buffer
                          0208   634 ;       R3              - CDT address
                          0208   635 ;
                          0208   636 ;  OUTPUTS:
                          0208   637 ;
                          0208   638 ;       R0-R2           - Destroyed
                          0208   639 ;
                          0208   640 ;-
                          0208   641
                          0208   642 RE_QUEUE_DG::
                          0208   643
  54    10 A3    D0       0208   644          MOVL    CDT$L_PDT(R3),R4        ; Recover PDT addr
52    0088 C4    C2       020C   645          SUBL2   PDT$L_DGOVRHD(R4),R2    ; Back up to top of buffer
                          0211   646          QUEUE_DG_BUF                    ; Return DG to free queue
                    05    0214   647          RSB
                          0215   648
                          0215   649
                          0215   650          .END
```

$$$CURSIZ	=	000001C4	CDC$K_CMDSPEC		00000014
$$$NEWSIZ	=	000001D0	CDC$K_DATA		00000014
BIT...	=	0000000D	CDC$LRGPACKET		00000019
CDB$B_CDRPCNT		00000028	CDC$L_BUFADR		00000024
CDB$B_RSTNADR		0000000E	CDC$L_BUFLENGTH		00000020
CDB$B_TYPE		0000000A	CDC$L_BUFLNAME		00000024
CDB$K_LEN		00000032	CDC$L_BUFLOFSET		00000028
CDB$L_CDBSLOT		00000024	CDC$L_BUFRNAME		0000002C
CDB$L_CDRPQBL		0000002E	CDC$L_BUFROFSET		00000030
CDB$L_CDRPQFL		0000002A	CDC$L_CDB_AD		0000000C
CDB$L_CDT		0000001C	CDC$L_CDRP		0000000C
CDB$L_PDT		00000020	CDC$L_CFGHWTYPE		00000022
CDB$L_QBL		00000004	CDC$L_CFGPRTMSK		00000026
CDB$L_QFL		00000000	CDC$L_CNTRDISCDG		00000038
CDB$T_LPRTNAM		00000016	CDC$L_CNTRPOACK		00000020
CDB$W_SIZE		00000008	CDC$L_CNTRPONAK		00000024
CDB$W_STATUS		0000000C	CDC$L_CNTRPONORSP		00000028
CDB_M_DISC	=	00000002	CDC$L_CNTRP1ACK		0000002C
CDC$ARMCINTPATH		0000001B	CDC$L_CNTRP1NAK		00000030
CDC$B_CFGCBLSTS		00000021	CDC$L_CNTRP1NORSP		00C00034
CDC$B_CFGPOLBSTS		0000002A	CDC$L_CTPREF		00000015
CDC$B_CFGPOSTS		0000001F	CDC$L_DISCONREAS		00000019
CDC$B_CFGP1LBSTS		0000002B	CDC$L_LPORT		00000010
CDC$B_CFGP1STS		00000020	CDC$L_PORTHDR		00000004
CDC$B_CMDCOUNT		00000000	CDC$L_R0STATUS		0000000C
CDC$B_CNTFLG		0000001F	CDC$L_R1STATUS		00000010
CDC$B_COUNT		00000015	CDC$L_RESERV04		00000004
CDC$B_CTPOPCODE		00000014	CDC$L_RESERV08		00000008
CDC$B_DELAY		00000019	CDC$L_RESERV12		0000000C
CDC$B_DONECNT		00000001	CDC$L_SEQNUM		00000001
CDC$B_DSTPORT		00000001	CDC$L_STARTADR		00000020
CDC$B_EXTEND		0000001C	CDC$L_SVAPTE		00000018
CDC$B_FDTCNT		00000002	CDC$MAINTSTATE		0000000F
CDC$B_GARBAGE_FLAG		00000014	CDC$MAPBUF		00000005
CDC$B_NOACTFLAG		0000001E	CDC$MAPMBUF		00000010
CDC$B_NODEADR		00000014	CDC$NAK		00000016
CDC$B_OPCODE		00000000	CDC$NOACK		00000017
CDC$B_OTHERNODE		0000001E	CDC$NOACT		0000000E
CDC$B_PATH		0000001C	CDC$NORSP		00000015
CDC$B_PKTMULT		0000001F	CDC$OVERSIZEPKT		0000001D
CDC$B_PKTSIZ		0000001D	CDC$POLLER		0000000E
CDC$B_POLLERFLAG		0000001E	CDC$RDCINTPATH		0000001C
CDC$B_PROTREV		00000036	CDC$READCNT		0000000A
CDC$B_PROTVER		00000035	CDC$REQDATA		00000004
CDC$B_PROTYPE		00000034	CDC$REQMDATA		00000013
CDC$B_QNUMBER		0000001F	CDC$RESET		0000000C
CDC$B_RST_PORT		00000015	CDC$SELADR		00C00014
CDC$B_SIOCNT		00000003	CDC$SENDDATA		00000003
CDC$B_TYPE		00000014	CDC$SENDDG		00000002
CDC$CARRIER		00000018	CDC$SENDMDATA		00000012
CDC$CONFIGDATA		00000007	CDC$SENDMSG		0000C001
CDC$CONNECT		00000008	CDC$START		0000000D
CDC$C_CMDSPEC		00000014	CDC$T_CONDAT		00000040
CDC$DISCONNECT		00000009	CDC$T_ENDMSG		00000072
CDC$GARBAGE		0000000B	CDC$T_GARBAGE		00000018
CDC$INITCINT		0000001A	CDC$T_LOCPROCNAM		00000014
CDC$K_CMDHDRSIZ		0000000C	CDC$T_PACKETDATA		00000080

```
CDC$T_REMPROCNAM                00000024
CDC$T_SYSTEMID                  00000034
CDC$UNMAPBUF                    00000006
CDC$UNMAPMBUF                   00000011
CDC$W_CMDOFSET                  00000015
CDC$W_CMDSEQNUM                 00000017
CDC$W_DGCREDIT                  0000003E
CDC$W_FLAG                      00000004
CDC$W_MAXDG                     00000014
CDC$W_MAXMSG                    00000018
CDC$W_MINSENDCR                 0000003C
CDC$W_MSGCREDIT                 0000003A
CDC$W_POLLINT                   0000001C
CDC$W_POLLNUM                   00000020
CDC$W_REPEAT                    0000001A
CDC$W_RESERV10                  0000000A
CDC$W_RSPBUFOFSET               00000004
CDC$W_RSPCNT                    00000006
CDC$W_RSPOFSET                  00000002
CDC$W_TXTLEN                    00000008
CDRP$B_FLAGS                    00000040
CDRP$C_BT_LEN                 = 00000040
CDRP$K_CTPLEN                   00000050
CDRP$K_CX_LEN                   00000068
CDRP$K_XFRLEN                   00000068
CDRP$L_FQFL                   = 00000000
CDRP$L_LBUFH_AD               = 0000002C
CDRP$L_PDT                      0000004C
CDRP$L_PENDQBL                  00000048
CDRP$L_PENDQFL                  00000044
CDRP$T_CX_LBUFHNDL              00000050
CDRP$T_CX_RBUFHNDL              0000005C
CDT$L_AUXSTRUC                = 00000000
CDT$L_PDT                     = 00000010
CHECK_CONN                      ********    X   01
CHECK_LINK_FQ                   ********    X   01
CHECK_LINK_PQ                   ********    X   01
CLEAN_CDRP                      ********    X   01
CLEAN_CMD                       0000014A  R     01
CLEAN_CMD_BUF                   000000F9  R     01
CLEAN_CONN                      00000113  R     01
COM$DRVDEALMEM                  ********    X   01
COPY_RSP                        000001F1  RG    01
CTP$ACTCOUNT                    00000006
CTP$BUFLENGTH                   0000000C
CTP$BUFLNAME                    00000010
CTP$BUFLOFSET                   00000014
CTP$BUFMAPREQ                   00000001
CTP$BUFMAPRSP                   00000041
CTP$BUFRNAME                    00000018
CTP$BUFROFSET                   0000001C
CTP$BUFTYPE                     00000005
CTP$BUFUNMREQ                   00000002
CTP$BUFUNMRSP                   00000042
CTP$CDATFREV                    00000002
CTP$CDATPTYPE                   00000000
CTP$CDATPVERS                   00000001
CTP$CFGPOSTS                    0000000B
CTP$CFGP1STS                    0000000C
CTP$CNTFLG                      0000000A
CTP$CNTRDISCDG                  00000024
CTP$CNTRP0ACK                   0000000C
CTP$CNTRP0NAK                   00000010
CTP$CNTRP0NRSP                  00000014
CTP$CNTRP1ACK                   00000018
CTP$CNTRP1NAK                   0000001C
CTP$CNTRP1NRSP                  00000020
CTP$CONFIGREQ                   00000009
CTP$CONFIGRSP                   00000049
CTP$CONNECTREQ                  0000000B
CTP$CONNECTRSP                  0000004B
CTP$COUNTSREQ                   0000000A
CTP$COUNTSRSP                   0000004A
CTP$DELAY                       00000005
CTP$EXTEND                      00000008
CTP$FINISHREQ                   0000000C
CTP$FINISHRSP                   0000004C
CTP$FMASK                       00000006
CTP$FUNCTREQ                    00000000
CTP$FUNCTRSP                    00000040
CTP$GENCONST                    00000009
CTP$GENDATA                     0000000E
CTP$GENDGRREQ                   00000005
CTP$GENDGRRSP                   00000045
CTP$GENFUNCT                    00000008
CTP$GENLENGTH                   0000000C
CTP$GENMSGREQ                   00000004
CTP$GENMSGRSP                   00000044
CTP$GENRSTREQ                   00000006
CTP$GENRSTRSP                   00000046
CTP$GENSTRREQ                   00000007
CTP$GENSTRRSP                   00000047
CTP$IMAGEDATA                   0000000E
CTP$MAXCMDOPC                   00000011
CTP$MBUFMAPREQ                  0000000D
CTP$MBUFMAPRSP                  0000004D
CTP$MBUFUNMREQ                  0000000E
CTP$MBUFUNMRSP                  0000004E
CTP$MOVBUFREQ                   00000003
CTP$MOVBUFRSP                   00000043
CTP$MOVETYPE                    00000008
CTP$MOVMBUFREQ                  0000000F
CTP$MOVMBUFRSP                  0000004F
CTP$MSTATEREQ                   00000010
CTP$MSTATERSP                   00000050
CTP$NOACTFLAG                   0000000A
CTP$NOACTREQ                    00000008
CTP$NOACTRSP                    00000048
CTP$OPCODE                      00000000
CTP$OPEXPAND                    000000FF
CTP$OTHERNODE                   00000009
CTP$PKTMULT                     0000000B
CTP$PKTSIZ                      0000000A
CTP$REFERENCE                   00000001
```

```
CTP$REPCOUNT                    00000006
CTP$RESERV10                    0000000A
CTP$RESERV11                    0000000B
CTP$RESERV12                    0000000C
CTP$RESERV20                    00000014
CTP$RESERV5                     00000005
CTP$RESERV6                     00000006
CTP$RESERV7                     00000007
CTP$RESERV9                     00000009
CTP$REVISION                =   00000000  G
CTP$STARTADR                    0000000C
CTP$STATUS                      00000005
CTP$VERSION                 =   00000003  G
CX$B_CMDERRCNT                  ********    X   01
CX$B_GARBAGE_FLAG               ********    X   01
CX$CONNECT_ERR                  000000E6 RG     01
CX$DG_IN                        00000000 RG     01
CX$L_CMDBUFBGN                  ********    X   01
CX$L_CMDBUFEND                  ********    X   01
CX$L_RSPBUFBGN                  ********    X   01
CX$L_RSPSEQNUM                  ********    X   01
CX$L_STATE                      ********    X   01
CX$MSG_IN                       00000000 RG     01
CX$Q_MAPLST                     ********    X   01
CX$T_GARBAGE_BUF                ********    X   01
CX_DG_RSP_SIZ                   ********    X   01
CX_MSG_RSP_SIZ                  ********    X   01
DEALLOC_CDRP                    ********    X   01
DEAL_MSG                        00000200 RG     01
DYN$C_CIDG                  =   0000003B
FINISH_IO                       0000012F R      01
FLAG_M_CMD_DONE             =   00008000
FLAG_M_CONN_LOST            =   00000800
FLAG_M_EXTRA_INPUT          =   00000200
FLAG_M_REC_WAIT             =   00000400
FLAG_M_SCS_ERR              =   00001000
FLAG_V_CMD_DONE             =   0000000F
FLAG_V_SAVE_ALL             =   00000001
FLAG_V_TYP_PKT_REC          =   00000002
PA_CNF                          00000000
PA_CQ0                          00000908
PA_CQ1                          0000090C
PA_CQ2                          00000910
PA_CQ3                          00000914
PA_DFQ                          00000928
PA_MADR                         00000014
PA_MDATR                        00000018
PA_MFQ                          0000092C
PA_MTC                          00000930
PA_MTEC                         00000934
PA_PDC                          00000920
PA_PEC                          0000091C
PA_PESR                         0000093C
PA_PFAR                         00000938
PA_PIC                          00000924
PA_PMC                          00000004
PA_PPR                          00000940
PA_PQBBR                        00000904
PA_PS                           00000900
PA_PSR                          00000918
PDT$B_DQIMAP                    00000154
PDT$B_HSHUT_DG                  000001E0
PDT$B_MAX_PORT                  0000017C
PDT$B_NXT_PORT                  0000017E
PDT$B_P0_LBSTS                  00000180
PDT$B_P1_LBSTS                  00000181
PDT$B_PLOGMAP                   00000134
PDT$B_PORTMAP                   00000114
PDT$B_PORT_NUM                  0000017D
PDT$B_REQIDPS                   0000017F
PDT$C_LENGTH                =   000000E4
PDT$C_PAREGBASE                 000000E4
PDT$C_PAREGEND                  00000110
PDT$C_PQB                   =   000001E0
PDT$L_CNF                       000000E4
PDT$L_CQ0                       000000F0
PDT$L_CQ1                       000000F4
PDT$L_DEALRGMSG             =   00000024
PDT$L_DFQ                       000000FC
PDT$L_DFQHDR                    00000208
PDT$L_DGHDRSZ                   00000190
PDT$L_DGNETHD                   00000194
PDT$L_DGOVRHD               =   000000B8
PDT$L_DQELOGOUT                 000002E0
PDT$L_GPTBASE                   0000022C
PDT$L_GPTLEN                    00000230
PDT$L_LBDG                      00000184
PDT$L_MFQ                       00000100
PDT$L_MFQHDR                    0000020C
PDT$L_MQELOGOUT                 00000320
PDT$L_MTC                       00000104
PDT$L_PFAR                      00000108
PDT$L_PMC                       000000E8
PDT$L_POLLERDUE                 0000018C
PDT$L_POOLDUE                   00000188
PDT$L_PPR                       0000010C
PDT$L_PS                        000000EC
PDT$L_PSR                       000000F8
PDT$L_QUEUEDG               =   00000C3C
PDT$L_SPTBASE                   00000224
PDT$L_SPTLEN                    00000228
PDT$L_VBDT                      0000021C
PDT$L_VPQB                      00000218
PDT$Q_COMQ2                     00C001F0
PDT$Q_COMQ3                     000001F8
PDT$Q_COMQBASE                  000001E0
PDT$Q_COMQH                     000001E8
PDT$Q_COMQL                     000001E0
PDT$Q_DFREEQ                    000001D0
PDT$Q_FORMPB                    00000174
PDT$Q_MFREEQ                    000001D8
PDT$Q_RSPQ                      00000200
PDT$Q_TEMP_RSPQ                 00000198
PDT$W_BDTLEN                    00000220
```

PDT$W_DQELEN	00000210	PPD$W_MASK		00000010		
PDT$W_LPORT_STS	00000110	PPD$W_MAXDG		0000001C		
PDT$W_MQELEN	00000214	PPD$W_MAXMSG		0000001E		
PDT$W_PBCOUNT	00000112	PPD$W_MTYPE		00000012		
PPD$B_DEF_ST	0000001C	PPD$W_M_VAL		00000014		
PPD$B_FLAGS	0000000F	PPD$W_SIZE		00000008		
PPD$B_HWVERS	00000034	RE_QUEUE_DG		00000208	RG	01
PPD$B_LBDATA	00000012	SCS$B_PPD	=	FFFFFFE0		
PPD$B_LCB_0	00000012	SCS$C_APPL_BASE	=	00000000		
PPD$B_LCB_LPORT	00000010	SCS$DISCONNECT		********	X	01
PPD$B_LCB_NPORT	0000000F	SCS$K_APPL_BASE	=	00000000		
PPD$B_LCB_OPC	00000011	SET_SEQ_NUM		000001E6	RG	01
PPD$B_LCB_PORT	0000000E	SIO_COMPLETE		********	X	01
PPD$B_OPC	0000000E	SIZ...	=	00000001		
PPD$B_PORT	0000000C	STATE_M_ALTGRBG	=	00000004		
PPD$B_PROTOCOL	0000001A	STATE_M_CANCEL	=	00000008		
PPD$B_RSTATE	00000025	STATE_M_CMAP	=	00000002		
PPD$B_RST_PORT	00000024	STATE_M_FDT	=	00000400		
PPD$B_STATUS	0000000D	STATE_M_NOTPROC	=	00001000		
PPD$B_SWFLAG	0000000B	STATE_M_RMAP	=	00000001		
PPD$B_SYSTEMID	00000014	STATE_M_SIO	=	00000800		
PPD$B_TYPE	0000000A	STATE_M_STOP	=	00000200		
PPD$C_LB_LENGTH	00000046	STATE_V_ALTGRBG	=	00000002		
PPD$C_LCB_DATA	00000013	STATE_V_CANCEL	=	00000003		
PPD$C_LENGTH	00000012	STATE_V_CMAP	=	00000001		
PPD$C_MIN_DGSIZ	00000050	STATE_V_FDT	=	0000000A		
PPD$K_LB_LENGTH	00000046	STATE_V_NOTPROC	=	0000000C		
PPD$K_LENGTH	00000012	STATE_V_RMAP	=	00000000		
PPD$L_BLINK	00000004	STATE_V_SIO	=	0000000B		
PPD$L_DG_DISC	00000028	STATE_V_STOP	=	00000009		
PPD$L_FLINK	00000000					
PPD$L_IN_VCD	00000018					
PPD$L_LBRC	00000042					
PPD$L_PO_ACK	00000010					
PPD$L_PO_NAK	00000014					
PPD$L_PO_NRSP	00000018					
PPD$L_P1_ACK	0000001C					
PPD$L_P1_NAK	00000020					
PPD$L_P1_NRSP	00000024					
PPD$L_REC_BOFF	00000028					
PPD$L_REC_NAME	00000024					
PPD$L_RPORT_FCN	00000020					
PPD$L_RPORT_REV	0000001C					
PPD$L_RPORT_TYP	00000018					
PPD$L_SND_BOFF	00000020					
PPD$L_SND_NAME	0000001C					
PPD$L_ST_ADDR	00000018					
PPD$L_XCT_LEN	00000018					
PPD$Q_CURTIME	00000048					
PPD$Q_NODENAME	00000040					
PPD$Q_SWINCARN	00000028					
PPD$Q_XCT_ID	00000010					
PPD$T_HWTYPE	00000030					
PPD$T_SWTYPE	00000020					
PPD$T_SWVERS	00000024					
PPD$W_LCB_LEN?	0000000C					
PPD$W_LENGTH	00000010					

```
                        0074   197          .SBTTL  DRIVER INTERNAL STORAGE DECLARATIONS
                        0074   198
                        0074   199
00000078                0074   200 CX$L_STATE::            .BLKL           ; Class driver internal status
                        0078   201
00000090                0078   202 CX$X_IRPSAV::           .BLKL   6       ; Saved R3 - R8. Used during
                        0090   203                                         ; command processing
                        0090   204
000000AC                0090   205 CX$X_CNCLSAV::          .BLKL   7       ; Saved R3 - R9. Used in cancel
                        00AC   206                                         ; code
                        00AC   207
000000C4                00AC   208 CX$X_STOPSAV::          .BLKL   6       ; Saved R3 - R8. Used in stop
                        00C4   209                                         ; code
                        00C4   210
000000C5                00C4   211 CX$B_CMDERRCNT::        .BLKB   1       ; STARTIO error count. Used to
                        00C5   212                                         ; determine if command is done
                        00C5   213
000000C9                00C5   214 CX$L_TEMP::             .BLKL   1       ; Temporary scratch longword
                        00C9   215
000000CD                00C9   216 CX$L_CETEMP::           .BLKL   1       ; Scratch for CONNECT_ERR code
000000D1                00CD   217 CX$L_CECC::             .BLKL   1       ; CONNECT_ERR current command
                        00D1   218
000000D5                00D1   219 CX$L_STOPCMD::          .BLKL   1       ; Current cmd for STOP_PROC
                        00D5   220
000000D9                00D5   221 CX$L_SEQNUM::           .BLKL   1       ; Command sequence number
                        00D9   222
000000DD                00D9   223 CX$L_CMDBUFBGN::        .BLKL   1       ; Start of command buffer
                        00DD   224 CX$L_CMDBUFEND::                        ; End of command buffer
000000E1                00DD   225 CX$L_RSPBUFBGN::        .BLKL   1       ; Start of response buffer
000000E5                00E1   226 CX$L_RSPBUFEND::        .BLKL   1       ; End of response buffer
000000E9                00E5   227 CX$L_CUR_CMD::          .BLKL   1       ; Current command in cmd buffer
000000ED                00E9   228 CX$L_NXT_RSP::          .BLKL   1       ; Addr of next response area
000000F1                00ED   229 CX$L_RSPCNTREM::        .BLKL   1       ; Space left in response buffer
000000F5                00F1   230 CX$L_RSPSEQNUM::        .BLKL   1       ; Response sequence number
                        00F5   231
                        00F5   232
000000FD                00F5   233 CX$Q_MAPLST::           .BLKQ   1       ; Mapped buffer CDRP list
                        00FD   234
                        00FD   235
                        00FD   236 ;  CXDRIVER maintains information about connections on a local port / remote
                        00FD   237 ;  port basis. Thus, for each combination of local and remote port, the
                        00FD   238 ;  controller can have a connection. The connection state is stored in a block
                        00FD   239 ;  called a CDB. The structures look as follows::
                        00FD   240 ;
                        00FD   241 ;
                        00FD   242 ;       PAA0 pointer -------> Remote port 0 ---------> CDB
                        00FD   243 ;                            Remote port 1
                        00FD   244 ;       PAB0 pointer                .
                        00FD   245 ;                                   .
                        00FD   246 ;            .                      .
                        00FD   247 ;            .                 Remote port n ---------> CDB
                        00FD   248 ;            .
                        00FD   249 ;
                        00FD   250 ;       PAx0 pointer -------> Remote port 0 ---------> CDB
                        00FD   251 ;                            Remote port 1
                        00FD   252 ;                                   .
                        00FD   253 ;                                   .
```

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
BIT...	=0000000D	88 (1)	88 (1)
CDB$L_CDBSLOT	00000024		#-379 (3)
CDB$L_CDT	0000001C		#-449 (4)
CDB$W_STATUS	0000000C		#-377 (3)
CDB_M_DISC	=00000002		#-376 (3)
CDC$B_CMDCOUNT	00000000		#-237 (2) 349 (3)
CDC$B_CTPOPCODE	00000014		#-218 (2) #-221 (2)
CDC$B_DONECNT	00000001		#-235 (2) #-236 (2) #-355 (3) #-389 (3)
CDC$B_OPCODE	00000000		#-439 (4) #-442 (4)
CDC$CONNECT	00000008		#-440 (4)
CDC$DISCONNECT	00000009		#-443 (4)
CDC$K_CMDHDRSIZ	0000000C		#-362 (3)
CDC$K_DATA	00000014		587 (7)
CDC$L_CDRP	0000000C		#-230 (2) #-231 (2) #-469 (4) #-486 (4)
			#-491 (4)
CDC$L_CTPREF	0000 0015		#-144 (2)
CDC$L_PORTHDR	0000 0004		#-586 (7)
CDC$L_ROSTATUS	0000000C		#-467 (4)
CDC$L_SEQNUM	00000001		#-524 (5)
CDC$W_FLAG	00000004		151 (2) 156 (2) 173 (2) #-233 (2)
			#-284 (2) #-436 (4) #-452 (4) 457 (4)
			461 (4) #-475 (4) #-480 (4)
CDC$W_REPEAT	0000001A		#-223 (2)
CDC$W_RSPCNT	00000006		#-171 (2) #-175 (2) #-210 (2) #-224 (2)
			#-459 (4) #-463 (4)
CDC$W_RSPOFSET	00000002		#-163 (2) #-182 (2) #-464 (4)
CDC$W_TXTLEN	00000008		#-367 (3)
CDRP$L_FQFL	=00000000		474 (4)
CDRP$L_LBUFH_AD	=0000002C		#-482 (4)
CDRP$L_PENDQFL	00000044		479 (4) 489 (4)
CDT$L_AUXSTRUC	=0000005C		#-375 (3)
CDT$L_PDT	=00000010		#-615 (8) #-644 (9)
CHECK_CONN	00000000-XR		#-446 (4)
CHECK_LINK_FQ	00000000-XR		#-472 (4)
CHECK_LINK_PQ	00000000-XR		#-477 (4)
CLEAN_CDRP	00000000-XR		#-484 (4)
CLEAN_CMD	00000148-R	431 (4)	#-363 (3)
CLEAN_CMD_BUF	000000F9-R	359 (3)	
CLEAN_CONN	00000113-R	372 (3)	#-353 (3) #-356 (3) #-366 (3)
COM$DRVDEALMEM	00000000-XR		232 (2) 381 (3)
COPY_RSP	000001F1-R	582 (7)	#-195 (2)
CTP$GENDGRREQ	00000005		#-220 (2)
CTP$GENMSGREQ	00000004		#-217 (2)
CTP$REFERENCE	00000001		#-137 (2) #-143 (2)
CTP$REVISION	=00000000	85 (1)	
CTP$VERSION	=00000003	85 (1)	
CX$B_CMDERRCNT	00000000-XR		#-390 (3) #-453 (4)
CX$B_GARBAGE_FLAG	00000000-XR		262 (2)
CX$CONNECT_ERR	000000E6-R	334 (3)	
CX$DG_IN	00000000-R	125 (2)	

```
CX$L_CMDBUFBGN       00000000-XR                #-138  (2)   #-141  (2)   #-234  (2)   #-354  (3)
                                                #-365  (3)   #-388  (3)
CX$L_CMDBUFEND       00000000-XR                #-139  (2)
CX$L_RSPBUFBGN       00000000-XR                #-189  (2)   #-466  (4)
CX$L_RSPSEQNUM       00000000-XR                #-523  (5)   #-525  (5)
CX$L_STATE           00000000-XR                  131  (2)     353  (3)     387  (3)
CX$MSG_IN            00000000-R    124   (2)
CX$Q_MAPLST          00000000-XR                  490  (4)
CX$T_GARBAGE_BUF     00000000-XR                  266  (2)
CX_DG_RSP_SIZ        00000000-XR                #-174  (2)   #-462  (4)
CX_MSG_RSP_SIZ       00000000-XR                #-170  (2)   #-458  (4)
DEALLOC_CDRP         00000000-XR                #-485  (4)
DEAL_MSG             00000200-R    613   (8)    #-206  (2)   #-272  (2)
DYN$C_CIDG          =0000003B                   #-203  (2)   #-269  (2)
FINISH_IO            0000012F-R    384   (3)
FLAG_M_CMD_DONE     =00008000                   #-233  (2)   #-434  (4)
FLAG_M_CONN_LOST    =00000800                   #-435  (4)   #-452  (4)
FLAG_M_EXTRA_INPUT  =00000200                   #-284  (2)
FLAG_M_REC_WAIT     =00000400                   #-475  (4)   #-480  (4)
FLAG_M_SCS_ERR      =00001000                   #-435  (4)
FLAG_V_CMD_DONE     =0000000F                   #-151  (2)
FLAG_V_SAVE_ALL     =00000001                   #-155  (2)   #-456  (4)
FLAG_V_TYP_PKT_REC  =00000002                   #-172  (2)   #-460  (4)
PDT$L_DEALRGMSG     =00000024                     616  (8)
PDT$L_DGOVRHD       =000000B8                   #-645  (9)
PDT$L_QUEUEDG       =0000003C                     646  (9)
PPD$B_PORT           0000000C                     579  (7)
PPD$B_TYPE           0000000A                   #-204  (2)   #-270  (2)
PPD$C_LENGTH         00000012                     578  (7)
PPD$W_LENGTH         00000010                     580  (7)
RE_QUEUE_DG          00000208-R    642   (9)    #-208  (2)   #-275  (2)
SCS$B_PPD           =FFFFFFE0                   #-201  (2)   #-260  (2)
SCS$C_APPL_BASE     =00000000                     577  (7)
SCS$DISCONNECT       00000000-XR                  378  (3)
SCS$K_APPL_BASE     =00000000                   #-201  (2)   #-260  (2)
SET_SEQ_NUM          000001E6-R    521   (5)    #-194  (2)
SIO_COMPLETE         00000000-XR                #-238  (2)   #-393  (3)
SIZ...              =00000001      88    (1)       88  (1)
STATE_M_ALTGRBG     =00000004      88    (1)
STATE_M_CANCEL      =00000008      88    (1)
STATE_M_CMAP        =00000002      88    (1)
STATE_M_FDT         =00000400      88    (1)
STATE_M_NOTPROC     =00001000      88    (1)
STATE_M_RMAP        =00000001      88    (1)
STATE_M_SIO         =00000800      88    (1)
STATE_M_STOP        =00000200      88    (1)
STATE_V_ALTGRBG     =00000002      88    (1)
STATE_V_CANCEL      =00000003      88    (1)
STATE_V_CMAP        =00000001      88    (1)    #-131  (2)   #-352  (3)   #-386  (3)
STATE_V_FDT         =0000000A      88    (1)
STATE_V_NOTPROC     =0000000C      88    (1)
STATE_V_RMAP        =00000000      88    (1)
STATE_V_SIO         =0000000B      88    (1)
STATE_V_STOP        =00000009      88    (1)
```

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$CDCDEF	8	83 (1)	83 (1)
$CDRPDEF	6	65 (1)	65 (1)
$CDTDEF	4	66 (1)	66 (1)
$CINTREGDEF	3	84 (1)	84 (1)
$CTPDEF	7	85 (1)	85 (1)
$CXCDBDEF	2	86 (1)	86 (1)
$CXCDRPDEF	1	87 (1)	87 (1)
$CXSTATEDEF	1	88 (1)	88 (1)
$DEFINI	1	65 (1)	65 (1) 66 (1) 67 (1) 68 (1)
			69 (1) 70 (1) 71 (1) 76 (1)
			77 (1) 78 (1) 83 (1) 84 (1)
			85 (1) 86 (1) 87 (1)
$DYNDEF	7	67 (1)	67 (1)
$IRPDEF	7	68 (1)	68 (1)
$PAPDTDEF	5	76 (1)	76 (1)
$PAREGDEF	5	77 (1)	77 (1)
$PDTDEF	4	69 (1)	69 (1)
$PPDDEF	10	78 (1)	78 (1)
$SCSDEF	4	70 (1)	70 (1)
$SSDEF	21	71 (1)	71 (1)
$VIELD1	1		88 (1)
ASSUME	1		349 (3) 577 (7) 578 (7) 579 (7)
			580 (7)
BBCW	1	130 (2)	130 (2)
BBSW	1	150 (2)	150 (2)
BEQLW	1	238 (2)	238 (2) 441 (4) 444 (4)
BGEQUW	1	140 (2)	140 (2)
BLEQUW	1	142 (2)	142 (2)
BNEQW	1	145 (2)	145 (2) 437 (4)
DEALLOC_MSG_BUF_REG	1	616 (8)	616 (8)
DISCONNECT	1	378 (3)	378 (3)
QUEUE_DG_BUF	1	646 (9)	646 (9)
SET_CMD_FLAG	1	233 (2)	233 (2) 284 (2) 452 (4) 475 (4)
			480 (4)
_VIELD	1		88 (1)

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	91	00:00:00.41	00:00:01.41
Command processing	134	00:00:00.59	00:00:01.39
Pass 1	628	00:00:24.03	00:00:29.93
Symbol table sort	0	00:00:02.93	00:00:03.76
Pass 2	118	00:00:03.60	00:00:03.93
Symbol table output	50	00:00:00.34	00:00:00.34
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	26	00:00:00.50	00:00:00.61

Assembler run totals 1053 00:00:32.44 00:00:41.39

The working set limit was 2250 pages.
111140 bytes (218 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1839 non-local and 30 local symbols.
650 source lines were read in Pass 1, producing 17 object records in Pass 2.
128 pages of virtual memory were used to define 43 macros.

+----------------------------+
! Macro library statistics !
+----------------------------+

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	3
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	6
DRB2:[SHULL.EVXCI.CXDRIVER]CXLIB.MLB;3	13
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	6
TOTALS (all libraries)	31

2273 GETS were required to define 31 macros.

There were no errors, warnings or information messages.

MACRO/LIS/CROSS CXINPUT+CXLIB/LIB+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB

Table of contents

(1) 47 DEFINITIONS
(2) 92 SIO_POLLER, Start/Stop configuration poller
(3) 130 SIO_INITCINT, Initialize CINT logic
(4) 170 SIO_ARMPATH, Arm CINT received path logic
(5) 205 SIO_READPATH, Read CINT received path status
(6) 249 SIO_SELADR, Change CINT port address
(7) 397 SIO_NORSP, Arm CINT NORSP logic
(8) 438 SIO_NAK, Arm CINT NAK logic
(9) 479 SIO_NOACK, Arm CINT NOACK logic
(10) 511 SIO_OVERSIZEPKT, Arm CINT oversized packet logic
(11) 543 SIO_LRGPACKET, Arm CINT large packet logic
(12) 567 SIO_CARRIER, Place carrier on specified path
(13) 623 SIO_MAINT_STATE, Set port to maintenance state
(14) 663 CLEAR_STS_REG Clear CINT status register
(15) 691 WRITE_COUNT_REG, Write CINT count register
(16) 725 WRITE_SRC_REG, Write CINT source register
(17) 755 SET_GO_BIT, Set the CINT GO bit
(18) 784 CLEAR_GO_BIT, Clear the CINT GO bit
(19) 811 SET_GOU_BIT, Set the CINT GO unconditional bit
(20) 840 CLEAR_GOU_BIT, Clear the CINT GO unconditional bit
(21) 868 CLEAR_CLER_REG, Clear the CINT CLER register
(22) 896 ARM_CINT_NORSP, Arm the CINT for NORSP's
(23) 923 ARM_CINT_NAK, Arm the CINT for NAK's
(24) 951 ARM_CINT_IGACK, Arm the CINT for ignoring ACK's
(25) 979 ARM_CINT_PATH, Arm the CINT for saving received path status
(26) 1008 SAVE_CINT_PDT, Show CINT logic has been used

```
                        1          .TITLE  CXMAINT
                        2          .IDENT  'V6-000'
                        3
                        4  ;******************************************************************************
                        5  ;
                        6  ;                          COPYRIGHT (c) 1981, 1984 BY
                        7  ;                 DIGITAL EQUIPMENT CORPORATION, MAYNARD,
                        8  ;                  MASSACHUSETTS.  ALL RIGHTS RESERVED.
                        9  ;
                       10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
                       11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
                       12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
                       13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
                       14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
                       15  ;
                       16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
                       17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
                       18  ;
                       19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
                       20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
                       21  ;
                       22  ;******************************************************************************
                       23  ;
                       24  ;+
                       25  ;
                       26  ; FACILITY:      VAX/VMS DIAGNOSTIC CLASS DRIVER
                       27  ;
                       28  ;
                       29  ; ABSTRACT:      This module contains the maintenance command execution
                       30  ;                routines for the CI controller class driver.
                       31  ;
                       32  ; AUTHOR:        Richard Hennessy  27-Jul-1982
                       33  ;                Base Systems Diagnostic Engineering
                       34  ;
                       35  ; MODIFIED BY:   Jim Klumpp  25 May-1983
                       36  ;
                       37  ;       6-000    Dave Shull       07-July-1984
                       38  ;                VMS V4 Modifications/Release
                       39  ;
                       40  ;       5-001    Dave Shull       22-March-1984
                       41  ;                Removed $PAPBDEF
                       42  ;
                       43  ;--
                       44
                       45          .DEFAULT DISPLACEMENT,WORD
                       46          .ENABLE  SUPPRESSION
                       47          .SBTTL   DEFINITIONS
                       48
                       49  ;
                       50  ; Set PSECT to driver code
                       51  ;
                       52
00000000               53          .PSECT   $$$115_DRIVER,LONG
                       54
                       55
                       56  ;  System Definitions (LIB.MLB):
                       57
```

```
0000     58          $CDRPDEF                        ; Define CDRP offsets
0000     59          $CDTDEF                         ; Define CDT offsets
0000     60          $CIBHANDEF                      ; Define CI BUFFER HANDLE offsets
0000     61          $DYNDEF                         ; Define DYN offsets
0000     62          $IODEF                          ; Define I/O FUNCTION codes
0000     63          $IRPDEF                         ; Define IRP offsets
0000     64          $PBDEF                          ; Define PB ofsets
0000     65          $PDTDEF                         ; Define PDT offsets
0000     66          $SBDEF                          ; Define SB offsets
0000     67          $SCSDEF                         ; Define SCS packet offsets
0000     68          $SSDEF                          ; Define system status symbols
0000     69          $SYSAPDEF                       ; Define SYSAP offsets
0000     70          $UCBDEF                         ; Define UCB offsets
0000     71
0000     72
0000     73 ;  PADRIVER Definitions (PALIB.MLB):
0000     74
0000     75          $PAPDTDEF                       ; Define PAPDT offsets
0000     76          $PAREGDEF                       ; Define PA registers
0000     77          $PPDDEF                         ; Define PPD offsets
0000     78          $PAMAINTDEF                     ; Define PA maint definitions
0000     79
0000     80
0000     81 ;  CXDRIVER Definitions (CXLIB.MLB):
0000     82
0000     83          $CDCDEF                         ; Define CDC interface symbols
0000     84          $CINTREGDEF                     ; Define CINT registers
0000     85          $CXCDBDEF                       ; Define CXCDB offsets
0000     86          $CXCDRPDEF                      ; Define CX CDRP extension
0000     87          $CXSTATEDEF                     ; Define CXDRIVER states
0000     88
0000     89
0000     90
```

```
                        0000     92          .SBTTL  SIO_POLLER,     Start/Stop configuration poller
                        0000     93 ;+
                        0000     94 ;
                        0000     95 ;  SIO_POLLER
                        0000     96 ;
                        0000     97 ;  Routine to turn the configuration poller on and off. The configuration
                        0000     98 ;  poller contains a flag which is set/cleared to enable/diable polling.
                        0000     99 ;
                        0000    100 ;  In order to make the poller call, the PDT address must be known. To get
                        0000    101 ;  this address, perform a CONFIG_PTH of the local port, and pull the PDT
                        0000    102 ;  address from the path block.
                        0000    103 ;
                        0000    104 ;  INPUTS:
                        0000    105 ;
                        0000    106 ;       R3                          - Address of current command
                        0000    107 ;
                        0000    108 ;-
                        0000    109
                        0000    110 SIO_POLLER::
                        0000    111
                FFFD'  30 0000  112          BSBW    CHECK_VC                ; Get PDT address for local port
                          0003  113          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                          0009  114
                          0009  115          ASSUME  CDC$_POLLERON EQ 0
                          0009  116
              1E A3   95  0009  117          TSTB    CDC$B_POLLERFLAG(R3)    ; Turn poller on or off?
                 0D   12  000C  118          BNEQ    10$                     ; Off, branch
                          000E  119          ENABLE_POLLER                   ; Turn the poller on
                 0B   11  0019  120          BRB     20$                     ; Branch
                          001B  121
                          001B  122 10$:      DISABLE_POLLER                  ; Turn the poller off
                          0026  123
              FFD7'  30   0026  124 20$:      BSBW    MARK_CMD_DONE           ; Finish up this command
              FFD4'  31   0029  125          BRW     SIO_NXT_CMD             ; Try next command
                          002C  126
                          002C  127
                          002C  128
```

```
                        002C    130          .SBTTL  SIO_INITCINT,   Initialize CINT logic
                        002C    131 ;+
                        002C    132 ;
                        002C    133 ;  SIO_INITCINT
                        002C    134 ;
                        002C    135 ;  Routine to initialize the node tester logic. The following steps are
                        002C    136 ;  taken:
                        002C    137 ;
                        002C    138 ;       - Clear GO bit in CINT GOR register
                        002C    139 ;       - Clear GOU bit in CINT GOUR register
                        002C    140 ;       - Clear CINT CLER register
                        002C    141 ;       - Clear CINT status register
                        002C    142 ;
                        002C    143 ;  INPUTS:
                        002C    144 ;
                        002C    145 ;       R3                      - Address of current command
                        002C    146 ;
                        002C    147 ;  OUTPUTS:
                        002C    148 ;
                        002C    149 ;       R0-R5                   - Destroyed
                        002C    150 ;       All other registers     - Preserved
                        002C    151 ;
                        002C    152 ;-
                        002C    153
                        002C    154 SIO_INITCINT::
                        002C    155
FFD1'  30               002C    156          BSBW    CHECK_VC                ; Get PDT address for local port
                        002F    157          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                        0035    158
02D0   30               0035    159          BSBW    CLEAR_GO_BIT            ; Clear the GO bit in the GOR reg
02E0   30               0038    160          BSBW    CLEAR_GOU_BIT           ; Clear the GOU bit in the GOUR reg
02F7   30               003B    161          BSBW    CLEAR_CLER_REG          ; Clear the CLER register
0283   30               003E    162          BSBW    CLEAR_STS_REG           ; Clear the CINT status register
                        0041    163
FFBC'  30               0041    164          BSBW    MARK_CMD_DONE           ; Finish up this command
FFB9'  31               0044    165          BRW     SIO_NXT_CMD             ; Try next command
                        0047    166
                        0047    167
                        0047    168
```

```
                        0047   170          .SBTTL  SIO_ARMPATH,    Arm CINT received path logic
                        0047   171 ;+
                        0047   172 ;
                        0047   173 ;  SIO_ARMPATH
                        0047   174 ;
                        0047   175 ;  Routine to arm the CINT logic so that the path over which the last packet
                        0047   176 ;  was recieved is saved.
                        0047   177 ;
                        0047   178 ;  INPUTS:
                        0047   179 ;
                        0047   180 ;       R3                      - Address of current command
                        0047   181 ;
                        0047   182 ;  OUTPUTS:
                        0047   183 ;
                        0047   184 ;       R0-R5                   - Destroyed
                        0047   185 ;       All other registers     - Preserved
                        0047   186 ;
                        0047   187 ;-
                        0047   188
                        0047   189 SIO_ARMPATH::
                        0047   190
FFB6'  30               0047   191          @SBW    CHECK_VC                ; Get PDT address for local port
                        004A   192          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                        0050   193
0295   30               0050   194          BSBW    WRITE_SRC_REG           ; Write the CINT source register
0316   30               0053   195          BSBW    ARM_CINT_PATH           ; Arm the CINT for saving received path
029E   30               0056   196          BSBW    SET_GO_BIT              ; Set the CINT GO bit
                        0059   197
0322   30               0059   198          BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
FFA1'  30               005C   199          BSBW    MARK_CMD_DONE           ; Finish up this command
FF9E'  31               005F   200          BRW     SIO_NXT_CMD             ; Try next command
                        0062   201
                        0062   202
                        0062   203
```

```
                        0062   205          .SBTTL  SIO_READPATH,   Read CINT received path status
                        0062   206 ;+
                        0062   207 ;
                        0062   208 ;  SIO_READPATH
                        0062   209 ;
                        0062   210 ;  Routine to read the CINT status register to determine the path over which
                        0062   211 ;  the last packet was recieved.
                        0062   212 ;
                        0062   213 ;  INPUTS:
                        0062   214 ;
                        0062   215 ;       R3                      - Address of current command
                        0062   216 ;
                        0062   217 ;  OUTPUTS:
                        0062   218 ;
                        0062   219 ;       R0-R5                   - Destroyed
                        0062   220 ;       All other registers     - Preserved
                        0062   221 ;
                        0062   222 ;-
                        0062   223
                        0062   224 SIO_READPATH::
                        0062   225
                        0062   226          ASSUME  CDC$_PATH0 EQ 0
                        0062   227
                FF9B'   30      0062   228          BSBW    CHECK_VC                ; Get PDT address for local port
                                0065   229          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                                0068   230
            00E4 C4     C1      0068   231          ADDL3   PDT$L_CNF(R4),-         ; Get the CINT CLER address
                                006F   232                  #PA_CLER,R0
50  00001018 8F                 0075   233          BISL2   #CLER_M_LDSTA,(R0)      ; Load CINT internal status into CLSR
                                007C   234
                                007C   235          CALC_RSP DST=R1                 ; Get response slot address
              1C A1     94      0085   236          CLRB    CDC$B_PATH(R1)          ; Assume packet received on path A
            00E4 C4     C1      0088   237          ADDL3   PDT$L_CNF(R4),-         ; Get the CINT status register address
50  00001000 8F                 008C   238                  #PA_CLSR,R0
     04 60      05      E1      0092   239          BBC     #CLSR_V_PATHB,(R0),10$  ; If received on path A, branch
                01      90      0096   240          MOVB    #CDC$_PATH1,-           ; Show packet received on path B
              1C A1             0098   241                  CDC$B_PATH(R1)
                                009A   242
                02E1    30      009A   243 10$:     BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
                FF60'   30      009D   244          BSBW    MARK_CMD_DONE           ; Finish up this command
                FF5D'   31      00A0   245          BRW     SIO_NXT_CMD             ; Try next command
                                00A3   246
                                00A3   247
```

```
                                00A3   249          .SBTTL  SIO_SELADR,     Change CINT port address
                                00A3   250 ;+
                                00A3   251 ;
                                00A3   252 ;  SIO_SELADR
                                00A3   253 ;
                                00A3   254 ;  Routine to change the port address of the CINT as seen on the CI bus.
                                00A3   255 ;  The following steps are taken to change the CINT's node address:
                                00A3   256 ;
                                00A3   257 ;       - Get a datagram buffer (used later)
                                00A3   258 ;       - Write the new address to the CAR (CINT address register)
                                00A3   259 ;       - Send out a loopback datagram
                                00A3   260 ;
                                00A3   261 ;  The CINT port address does not actually get changed until the loopback
                                00A3   262 ;  datagram is received.
                                00A3   263 ;
                                00A3   264 ;  INPUTS:
                                00A3   265 ;
                                00A3   266 ;       R3                          - Address of current command
                                00A3   267 ;
                                00A3   268 ;  OUTPUTS:
                                00A3   269 ;
                                00A3   270 ;-
                                00A3   271
                                00A3   272 ;
                                00A3   273 ; Polynomial table used to calculate CRC for loopback datagram:
                                00A3   274 ;
                                00A3   275
                                00A3   276 CRC_TABLE:
                                00A3   277
                   00000000     00A3   278          .LONG   0
                   1DB71064     00A7   279          .LONG   ^X1DB71064
                   3B6E20C8     00AB   280          .LONG   ^X3B6E20C8
                   26D930AC     00AF   281          .LONG   ^X26D930AC
                   76DC4190     00B3   282          .LONG   ^X76DC4190
                   6B6B51F4     00B7   283          .LONG   ^X6B6B51F4
                   4DB26158     00BB   284          .LONG   ^X4DB26158
                   5005713C     00BF   285          .LONG   ^X5005713C
                   EDB88320     00C3   286          .LONG   ^XEDB88320
                   F00F9344     00C7   287          .LONG   ^XF00F9344
                   D6D6A3E8     00CB   288          .LONG   ^XD6D6A3E8
                   CB61B38C     00CF   289          .LONG   ^XCB61B38C
                   9B64C2B0     00D3   290          .LONG   ^X9B64C2B0
                   86D3D2D4     00D7   291          .LONG   ^X86D3D2D4
                   A00AE278     00DB   292          .LONG   ^XA00AE278
                   BDBDF21C     00DF   293          .LONG   ^XBDBDF21C
                                00E3   294
                                00E3   295 SIO_SELADR::
                                00E3   296
                     FF1A'  30  00E3   297          BSBW    CHECK_VC                ; Get PDT address for local port
                                00E6   298          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                                00EC   299
                                00EC   300          ALLOC_DG_BUF                    ; Get a datagram buffer
                                00EF   301          BLBCW   R0,ALLOC_DG_ERR         ; Error, branch
                0190  C4    C2  00F5   302          SUBL    PDT$L_DGADRS2(R4),R2    ; Back up to PPD header
                                00FA   303
                0184  C4    D0  00FA   304          MOVL    PDT$L_LBDG(R4),R0       ; Get template LB DG addr
                          3F  BB  00FF   305          PUSHR   #^M<R2,R3,R4,R5>        ; Save regs
```

```
                                3A   28  0101   306          MOVC3   #<PPD$C_LB_LENGTH-PPD$B_PORT>,- ; Copy template LB DG to
                             0C A0       0103   307                  PPD$B_PORT(R0),-          ; current LB DG
                             0C A2       0105   308                  PPD$B_PORT(R2)
                                3C   BA  0107   309          POPR    #^M<R2,R3,R4,R5>          ; Restore regs
                          0F A2   02   88  0109   310          BISB2   #^X02,PPD$B_FLAGS(R2)     ; Set path select to path A
                                           010D   311
    51    00001028 8F    00E4 C4   C1  010D   312          ADDL3   PDT$L_CNF(R4),#PA_CAR,R1; Get address of CAR reg
                       61    14 A3   9A  0117   313          MOVZBL  CDC$B_NODEADR(R3),(R1)    ; Write the new node address
                                           011B   314
                                           011B   315          $INS_COMQLOW ERRADDR=200$         ; Send the LB datagram off
                                           0136   316
                                           0136   317 ;  Wait for the node number to change in the port parameter register (PPR).
                                           0136   318 ;  Use loop count to prevent this check from hanging system.
                                           0136   319
                 51    00000064 8F    D0  0136   320          MOVL    #100,R1                   ; Initialize loop count
                    50    010C D4    D0  013D   321 10$:      MOVL    @PDT$L_PPR(R4),R0         ; Read the PPR
                       50    14 A3   91  0142   322          CMPB    CDC$B_NODEADR(R3),R0      ; Has node number changed?
                                03   13  0146   323          BEQL    20$                       ; Yes, branch
                             F2 51   F5  0148   324          SOBGTR  R1,10$                    ; Loop until loop count expires
                                           0148   325
                                           0148   326 ;****************************************************************************
                                           0148   327 ;
                                           0148   328 ;  SOFTWARE DEBUG CODE
                                           0148   329 ;
                                           0148   330 ;  Node number change failed - breakpoint.
                                           0148   331 ;
                                           0148   332 ;       jsb     g^ini$brk
                                           0148   333 ;
                                           0148   334 ;  END SOFTWARE DEBUG CODE
                                           0148   335 ;
                                           0148   336 ;****************************************************************************
                                           0148   337
                                           0148   338 ;  Now change the template loopback datagram to reflect the change in the
                                           0148   339 ;  local port number.
                                           0148   340
                 53    0184 C4    D0  0148   341 20$:      MOVL    PDT$L_LBDG(R4),R3         ; Get addr of LB template
                    0100 D4    90  0150   342          MOVB    @PDT$L_PPR(R4),-          ; Save local port number
                          0C A3       0154   343                  PPD$B_PORT(R3)            ; in LB dg template
                 51    0043 8F    3C  0156   344          MOVZWL  #<PPD$C_LCB_DATA + -      ; Set up length of buffer
                                           0158   345                  PPD$C_LBDAT_LEN>,R1       ; to allocate
                  00000000'GF    16  0158   346          JSB     G^EXE$ALONONPAGED         ; Get a temporary buffer
                                           015B   347          BLBC    R0,150$                   ; Failure, branch
                                           0161   348
08 A2    51    00600000 8F    C9  0161   349          BISL3   #DYN$C_SCS@16,R1,-        ; Set structure type
                                           0164   350                  PPD$W_SIZE(R2)            ; and size in temporary buffer
                                           0164   351                                    ; Set up data to CRC:
                          0C A2   94  0164   352          CLRB    PPD$W_LCB_LEN7(R2)        ;  H.o. lb data length + 7
                    07    10 A3   81  0167   353          ADDB3   PPD$W_LENGTH(R3),#7,-     ;  L.o. lb data length + 7
                          0D A2       016B   354                  PPD$W_LCB_LEN7+1(R2)      ;
                    50    0F A3   90  016D   355          MOVB    PPD$B_PORT(R3),R0         ;
                 0E A2       50   90  0171   356          MOVB    R0,PPD$B_LCB_PORT(R2)     ;  own port number,
                 0F A2       50   92  0175   357          MCOMB   R0,PPD$B_LCB_NPORT(R2)    ;  NOT(own port),
                 10 A2       50   90  0179   358          MOVB    R0,PPD$B_LCB_LPORT(R2)    ;  local port,
                                0D   90  017D   359          MOVB    #PPD$C_SNDLB,-            ;  SNDLB opcode,
                             11 A2       0181   360                  PPD$B_LCB_OPC(R2)
                             12 A2   94  0181   361          CLRB    PPD$B_LCB_0(R2)           ;  and 0 (packing fmt = 0)
                                3C   BB  0184   362          PUSHR   #^M<R2,R3,R4,R5>          ; Save registers
```

```
                    30   28  0192   363          MOVC    #PPD$C_LBDAT_LEN,-      ; Copy LB data from
                 12 A3        0194   364                  PPD$B_LBDATA(R3),-      ;  template to temporary
                 13 A2        0196   365                  PPD$C_LCB_DATA(R2)      ;  buffer
              52    6E   7D   0198   366          MOVQ    (SP),R2                 ; Get R2 and R3 again
           50    0D A2   9A   019B   367          MOVZBL  PPD$W_LCB_LEN7+1(R2),R0 ; Get # bytes to crc
      FFFFFFFF 8F    FF00 CF   0B   019F   368          CRC     CRC_TABLE,#-1,-         ; Compute CRC from temporary
            0C A2    50        01A8   369                  R0,PPD$W_LCB_LEN7(R2)   ;  buffer
                     3C   BA   01AB   370          POPR    #^M<R2,R3,R4,R5>        ; Retreive registers
           42 A3    50   D2   01AD   371          MCOML   R0,PPD$L_LBCRC(R3)      ; Put CRC comp into template
              50    52   D0   01B1   372          MOVL    R2,R0                   ; Copy temporary buffer addr
        00000000'GF    16   01B4   373          JSB     G^COM$DRVDEALMEM        ;  and return it to pool
                             01BA   374
                 FE43'   30   01BA   375          BSBW    MARK_CMD_DONE           ; Finish up this command
                 FE40'   31   01BD   376          BRW     SIO_NXT_CMD             ; Try next command
                             01C0   377
                             01C0   378
                             01C0   379
                             01C0   380 ;  UNABLE TO SEND LB DB
                             01C0   381 ;
                             01C0   382 ;  A failure occured in attemting to send the loopback datagram (actually, in
                             01C0   383 ;  trying to queue the command to the port's command queue). Set the appropriate
                             01C0   384 ;  status in the response buffer.
                             01C0   385
                             01C0   386 150$:
                             01C0   387 200$:
                             01C0   388
                             01C0   389 ;        SET_RSP_STATUS_2 STS_1=R0,STS_2=#0      ; Indicate failure
                             01C0   390 ;        SET_CMD_FLAG SCS_ERR                    ; Show it as an SCS error
                             01C0   391 ;        BRW     SIO_NXT_CMD_ERR                 ; Go execute the next command
                             01C0   392
                             01C0   393
                             01C0   394
                             01C0   395
```

```
                                01C0   397          .SBTTL  SIO_NORSP,      Arm CINT NORSP logic
                                01C0   398 ;+
                                01C0   399 ;
                                01C0   400 ;  SIO_NORSP
                                01C0   401 ;
                                01C0   402 ;  Routine to arm the CINT logic so that the specified number of packets
                                01C0   403 ;  received off the wire are NORSP'd.
                                01C0   404 ;
                                01C0   405 ;  INPUTS:
                                01C0   406 ;
                                01C0   407 ;       R3                          - Address of current command
                                01C0   408 ;
                                01C0   409 ;  OUTPUTS:
                                01C0   410 ;
                                01C0   411 ;-
                                01C0   412
                                01C0   413 SIO_NORSP::
                                01C0   414
                    FE3D'  30   01C0   415          BSBW    CHECK_VC                ; Get PDT address for local port
                                01C3   416          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                                01C9   417
                    0106   30   01C9   418          BSBW    WRITE_COUNT_REG         ; Write the CINT count register
                    0119   30   01CC   419          BSBW    WRITE_SRC_REG           ; Write the CINT source register
                    0170   30   01CF   420          BSBW    ARM_CINT_NORSP          ; Arm the CINT for NORSP's
                                01D2   421
              01    14 A3  91   01D2   422          CMPB    CDC$B_TYPE(R3),#CDC$_ALL; Ignore count?
                 05  13        01D6   423          BEQL    10$                     ; Yes, branch
                                01D8   424
                    011C   30   01D8   425          BSBW    SET_GO_BIT              ; Set the CINT GO bit
                 05  11        01DB   426          BRB     20$                     ; Branch
                                01DD   427
                    0136   30   01DD   428 10$:     BSBW    SET_GOU_BIT             ; Set the CINT GO unconditional
                                01E0   429
                    019B   30   01E0   430 20$:     BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
                    FE1A'  30   01E3   431          BSBW    MARK_CMD_DONE           ; Finish up this command
                    FE17'  31   01E6   432          BRW     SIO_NXT_CMD             ; Try next command
                                01E9   433
                                01E9   434
                                01E9   435
                                01E9   436
```

```
                        01E9   438          .SBTTL  SIO_NAK,         Arm CINT NAK logic
                        01E9   439 ;+
                        01E9   440 ;
                        01E9   441 ;  SIO_NAK
                        01E9   442 ;
                        01E9   443 ;  Routine to arm the CINT logic so that the specified number of packets
                        01E9   444 ;  received off the wire are NAK'd.
                        01E9   445 ;
                        01E9   446 ;  INPUTS:
                        01E9   447 ;
                        01E9   448 ;       R3                          - Address of current command
                        01E9   449 ;
                        01E9   450 ;  OUTPUTS:
                        01E9   451 ;
                        01E9   452 ;-
                        01E9   453
                        01E9   454 SIO_NAK::
                        01E9   455
                FE14'   30      01E9   456          BSBW    CHECK_VC                ; Get PDT address for local port
                        01EC   457          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                        01F2   458
                00DD    30      01F2   459          BSBW    WRITE_COUNT_REG         ; Write the CINT count register
                00F0    30      01F5   460          BSBW    WRITE_SRC_REG           ; Write the CINT source register
                0155    30      01F8   461          BSBW    ARM_CINT_NAK            ; Arm the CINT for NAK's
                        01FB   462
          01    14 A3   91      01FB   463          CMPB    CDC$B_TYPE(R3),#CDC$_ALL; Ignore count?
                  05    13      01FF   464          BEQL    10$                     ; Yes, branch
                        0201   465
                00F3    30      0201   466          BSBW    SET_GO_BIT              ; Set the CINT GO bit
                  03    11      0204   467          BRB     20$                     ; Branch
                        0206   468
                0100    30      0206   469 10$:     BSBW    SET_GOU_BIT             ; Set the CINT GO unconditional
                        0209   470
                0172    30      0209   471 20$:     BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
                FDF3'   30      020C   472          BSBW    MARK_CMD_DONE           ; Finish up this command
                FDEE'   31      020F   473          BRW     SIO_NXT_CMD             ; Try next command
                        0212   474
                        0212   475
                        0212   476
                        0212   477
```

```
                     0212   479          .SBTTL  SIO_NOACK,      Arm CINT NOACK logic
                     0212   480 ;+
                     0212   481 ;
                     0212   482 ;  SIO_NOACK
                     0212   483 ;
                     0212   484 ;  Routine to arm the CINT logic so that the next ACK received is ignored.
                     0212   485 ;  This causes the local port to send a duplicate packe on the wire.
                     0212   486 ;
                     0212   487 ;  INPUTS:
                     0212   488 ;
                     0212   489 ;       R3                          - Address current command
                     0212   490 ;
                     0212   491 ;  OUTPUTS:
                     0212   492 ;
                     0212   493 ;-
                     0212   494
                     0212   495 SIO_NOACK::
                     0212   496
                FDEB' 30     0212   497          BSBW    CHECK_VC                ; Get PDT address for local port
                               0215   498          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                               021B   499
                00CA   30     021B   500          BSBW    WRITE_SRC_REG           ; Write the CINT source register
                013D   30     021E   501          BSBW    ARM_CINT_IGACK          ; Arm the CINT for ignoring ACK's
                00D3   30     0221   502          BSBW    SET_GO_BIT              ; Set the CINT GO bit
                               0224   503
                0157   30     0224   504          BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
                FDD6'  30     0227   505          BSBW    MARK_CMD_DONE           ; Finish up this command
                FDD3'  31     022A   506          BRW     SIO_NXT_CMD             ; Try next command
                               022D   507
                               022D   508
                               022D   509
```

```
                                022F    511          .SBTTL  SIO_OVERSIZEPKT,Arm CINT oversized packet logic
                                022F    512 ;+
                                022D    513 ;
                                022D    514 ;  SIO_OVERSIZEPKT
                                022D    515 ;
                                022D    516 ;  Routine to arm the CINT logic so that the next packet sent out has
                                022D    517 ;  an extra byte appended to it.
                                022D    518 ;
                                022D    519 ;  INPUTS:
                                022D    520 ;
                                022D    521 ;       R3                          - Address of current command
                                022D    522 ;
                                022D    523 ;  OUTPUTS:
                                022D    524 ;
                                022D    525 ;-
                                022D    526
                                022D    527 SIO_OVERSIZEPKT::
                                022D    528
                      FDD0'     30    022D    529          BSBW    CHECK_VC                ; Get PDT address for local port
                                0230    530          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                                0236    531
                   00E4 C4      C1    0236    532          ADDL3   PDT$L_CNF(R4),-         ; Get address of CLER register
          50    00001018 8F           023A    533                  #PA_CLER,R0
          60    00000040 8F     C8    0240    534          BISL    #CLER_M_OVPAC,(R0)      ; Arm CINT for oversized packets
                                0247    535
                      0134      30    0247    536          BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
                      FDB3'     30    024A    537          BSBW    MARK_CMD_DONE           ; Finish up this command
                      FDB0'     31    024D    538          BRW     SIO_NXT_CMD             ; Try next command
                                0250    539
                                0250    540
                                0250    541
```

```
                     0250   543          .SBTTL  SIO_LRGPACKET,  Arm CINT large packet logic
                     0250   544 ;+
                     0250   545 ;
                     0250   546 ;  SIO_LRGPACKET
                     0250   547 ;
                     0250   548 ;  Routine to arm the CINT logic to send out "large" packets on the wire.
                     0250   549 ;  Currently, this routine does nothing.
                     0250   550 ;
                     0250   551 ;  INPUTS:
                     0250   552 ;
                     0250   553 ;       R3                          - Address of current command
                     0250   554 ;
                     0250   555 ;  OUTPUTS:
                     0250   556 ;
                     0250   557 ;-
                     0250   558
                     0250   559 SIO_LRGPACKET::
                     0250   560
FDAD'  30            0250   561          BSBW    MARK_CMD_DONE          ; Finish up this command
FDAA'  31            0253   562          BRW     SIO_NXT_CMD            ; Try next command
                     0256   563
                     0256   564
                     0256   565
```

```
                     0256   567          .SBTTL  SIO_CARRIER,    Place carrier on specified path
                     0256   568 ;*
                     0256   569 ;
                     0256   570 ;  SIO_CARRIER
                     0256   571 ;
                     0256   572 ;  Routine to put carrier on the specified path of the CI.
                     0256   573 ;
                     0256   574 ;  INPUTS:
                     0256   575 ;
                     0256   576 ;       R3                          - Address of current command
                     0256   577 ;
                     0256   578 ;  OUTPUTS:
                     0256   579 ;
                     0256   580 ;-
                     0256   581
                     0256   582 SIO_CARRIER::
                     0256   583
                FDA7' 30  0256   584          BSBW    CHECK_VC                ; Get PDT address for local port
                          0259   585          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
                          025F   586
              00E4 C4   C1  025F   587          ADDL3   PDT$L_CNF(R4),-         ; Get CLER register address
          51  00001018 8F       0263   588                  #PA_CLER,R1
                50   D4  0269   589          CLRL    R0                      ; Initialize carrier mask
                          026B   590
                          026B   591          ASSUME  CDC$_PATH0 EQ 0
                          026B   592
             1C A3    95  026B   593          TSTB    CDC$B_PATH(R3)          ; Use path A?
                05    12  026E   594          BNEQ    10$                     ; No, branch
          50    02    C8  0270   595          BISL2   #CLER_M_CARRA,R0        ; Set up carrier mask for path A
                0E    11  0273   596          BRB     30$
                          0275   597
             1C A3    91  0275   598 10$:     CMPB    CDC$B_PATH(R3),-        ; Use path B?
                01        0278   599                  #CDC$_PATH1
                05    12  0279   600          BNEQ    20$                     ; No, branch
          50    01    C8  027B   601          BISL2   #CLER_M_CARRB,R0        ; Set up carrier mask for path B
                03    11  027E   602          BRB     30$
                          0280   603
          50    03    C8  0280   604 20$:     BISL2   #<CLER_M_CARRA!-        ; Set up carrier mask for both paths
                                                     CLER_M_CARRB>,R0
                          0283   605
                          0283   606
                          0283   607          ASSUME  CDC$_OFF EQ 0
                          0283   608
             14 A3    95  0283   609 30$:     TSTB    CDC$B_TYPE(R3)          ; Turn carrier on or off?
                05    13  0286   610          BEQL    40$                     ; Off, branch
          61    50    C8  0288   611          BISL2   R0,(R1)                 ; Set carrier on specified paths
                03    11  028B   612          BRB     50$
                          028D   613
          61    50    CA  028D   614 40$:     BICL2   R0,(R1)                 ; Clear carrier on specified paths
                          0290   615
             00EB   30    0290   616 50$:     BSBW    SAVE_CINT_PDT           ; Show that CINT port has been used
             FD6A'  30    0293   617          BSBW    MARK_CMD_DONE           ; Finish up this command
             FD67'  31    0296   618          BRW     SIO_NXT_CMD             ; Try next command
                          0299   619
                          0299   620
                          0299   621
```

```
                                0299    623          .SBTTL  SIO_MAINT_STATE,Set port to maintenance state
                                0299    624 ;+
                                0299    625 ;
                                0299    626 ;  SIO_MAINT_STATE
                                0299    627 ;
                                0299    628 ;  Routine to set the local port into the enabled/maintenance state. This
                                0299    629 ;  causes all virtual circuits to go down, as well as all connections through
                                0299    630 ;  the local port to crash. The SYSAP's are notified that connections are lost
                                0299    631 ;  in the standard manner.
                                0299    632 ;
                                0299    633 ;  INPUTS:
                                0299    634 ;
                                0299    635 ;       R3                          - Address of current command
                                0299    636 ;
                                0299    637 ;  OUTPUTS:
                                0299    638 ;
                                0299    639 ;-
                                0299    640
                                0299    641 SIO_MAINT_STATE::
                                0299    642
                     FD64'  30  0299    643          BSBW    CHECK_VC                ; Get PDT address for local port
                                029C    644          BLBCW   R0,CHECK_VC_ERR         ; Error, branch
          0000'CF    54   D0  02A2    645          MOVL    R4,CX$L_PORT_STATE_PDT  ; Save PDT address (used by CANCEL)
                                02A7    646
                                02A7    647          ASSUME  CDC$_MSTATE EQ 0
                                02A7    648
                  14 A3    95  02A7    649          TSTB    CDC$B_TYPE(R3)          ; Set port to enabled/maint state
                     08    12  02AA    650          BNEQ    10$                     ; No, branch
                                02AC    651          SET_MAINT_STATE             ; Put local port in enabled/maint state
                     0A    11  02B2    652          BRB     20$                     ; Branch to common code
                                02B4    653
     52             15 A3    90  02B4    654 10$:     MOVB    CDC$B_RST_PORT(R3),R2   ; Get new reset port number
                                02B8    655          SET_UNINIT_STATE            ; Put port in uninit/maint state
                                02BE    656
                     FD3E'  30  02BE    657 20$:     BSBW    MARK_CMD_DONE           ; Finish up this command
                     FD3C'  31  02C1    658          BRW     SIO_NXT_CMD             ; Try next command
                                02C4    659
                                02C4    660
                                02C4    661
```

```
                                02C4    663          .SBTTL  CLEAR_STS_REG,  Clear CINT status register
                                02C4    664 ;+
                                02C4    665 ;
                                02C4    666 ;  CLEAR_STS_REG
                                02C4    667 ;
                                02C4    668 ;  Routine to clear the CINT status register. This should be done before
                                02C4    669 ;  arming the CINT for any of its special functions.
                                02C4    670 ;
                                02C4    671 ;  INPUTS:
                                02C4    672 ;
                                02C4    673 ;       R4                          - PDT address
                                02C4    674 ;
                                02C4    675 ;  OUTPUTS:
                                02C4    676 ;
                                02C4    677 ;       R0                          - Destroyed
                                02C4    678 ;       All other registers         - Preserved
                                02C4    679 ;
                                02C4    680 ;-
                                02C4    681
                                02C4    682 CLEAR_STS_REG::
                                02C4    683
                00E4 C4   C1    02C4    684          ADDL3   PDT$L_CNF(R4),-        ; Get CINT STATUS register address
            50  0000'000 8F       02C8    685                  #PA_CISR,R0
                   60   00   D2   02CE    686          MCOML   #0,(R0)               ; Clear the status regsiter
                             05    02D1    687          RSB
                                   02D2    688
                                   02D2    689
```

```
                                02D2    691          .SBTTL  WRITE_COUNT_REG, Write CINT count register
                                02D2    692 ;+
                                02D2    693 ;
                                02D2    694 ;  WRITE_COUNT_REG
                                02D2    695 ;
                                02D2    696 ;  Routine the write the count, specified in the command buffer, the the
                                02D2    697 ;  CINT count register. The count in the command buffer must be changed
                                02D2    698 ;  to two's complement form before being written.
                                02D2    699 ;
                                02D2    700 ;  INPUTS:
                                02D2    701 ;
                                02D2    702 ;       R3                          - Address of current command
                                02D2    703 ;       R4                          - PDT address
                                02D2    704 ;       CDC$B_COUNT(R3)             - Count to write
                                02D2    705 ;
                                02D2    706 ;  OUTPUTS:
                                02D2    707 ;
                                02D2    708 ;       R0,R1                       - Destroyed
                                02D2    709 ;       All other registers         - Preserved
                                02D2    710 ;
                                02D2    711 ;-
                                02D2    712
                                02D2    713 WRITE_COUNT_REG:
                                02D2    714
          50    15 A3    9A     02D2    715          MOVZBL  CDC$B_COUNT(R3),R0      ; Get NORSP count
          50    FF 8F    8C     02D6    716          XORB2   #^XFF,R0                ; Convert to count to ONE's complement
       00E4 C4    C1     02DA    717          ADDL3   PDT$L_CNF(R4),-                 ; Get addr of CINT NNCR (count) reg
51  00001020 8F               02DE    718                  #PA_NNCR,R1
          61    50    9A     02E4    719          MOVZBL  R0,(R1)                 ; Write the count register
                      05     02E7    720          RSB
                             02E8    721
                             02E8    722
                             02E8    723
```

```
                                02E8    725          .SBTTL  WRITE_SRC_REG,  Write CINT source register
                                02E8    726 ;+
                                02E8    727 ;
                                02E8    728 ;  WRITE_SRC_REG
                                02E8    729 ;
                                02E8    730 ;  Routine to write the UUT port number, specified in the command buffer, to
                                02E8    731 ;  the CINT source register.
                                02E8    732 ;
                                02E8    733 ;  INPUTS:
                                02E8    734 ;
                                02E8    735 ;       R3                          - Address of current command
                                02E8    736 ;       R4                          - PDT address
                                02E8    737 ;       CDC$B_OTHERNODE(R3)         - UUT port address
                                02E8    738 ;
                                02E8    739 ;  OUTUTS:
                                02E8    740 ;
                                02E8    741 ;       R0                          - Destroyed
                                02E8    742 ;       All other registers         - Preserved
                                02E8    743 ;
                                02E8    744 ;-
                                02E8    745
                                02E8    746 WRITE_SRC_REG:
                                02E8    747
                      00F4 C4   C1  02E8    748          ADDL3   PDT$L_CNF(R4),-          ; Get CINT SOURCE regsiter address
                   50 00001030 8F   02EC    749                  #PA_SR,R0
                      60   1E A3   9A  02F2    750          MOVZBL  CDC$B_OTHERNODE(R3),(R0); Write UUT port number to SOURCE reg
                                   05  02F6    751          RSB
                                       02F7    752
                                       02F7    753
```

```
                        02F7   755          .SBTTL  SET_GO_BIT,     Set the CINT GO bit
                        02F7   756 ;+
                        02F7   757 ;
                        02F7   758 ;  SET_GO_BIT
                        02F7   759 ;
                        02F7   760 ;  Routine to set the GO bit in the CINT GOR register. First, however, the
                        02F7   761 ;  GO unconditional bit in the GOU register is cleared.
                        02F7   762 ;
                        02F7   763 ;  INPUTS:
                        02F7   764 ;
                        02F7   765 ;       R4                          - PDT address
                        02F7   766 ;
                        02F7   767 ;  OUTUTS:
                        02F7   768 ;
                        02F7   769 ;       R0                          - Destroyed
                        02F7   770 ;       All other registers         - Preserved
                        02F7   771 ;
                        02F7   772 ;-
                        02F7   773
                        02F7   774 SET_GO_BIT:
                        02F7   775
                  002D  30      02F7   776          BSBW    CLEAR_GOU_BIT           ; Clear the GO unconditional bit
                00E4 C4     C1  02FA   777          ADDL3   PDT$L_CNF(R4),-         ; Get GO register address
        50    00001008 8F       02FE   778                  #PA_GOR,R0
              60    01    C8    0304   779          BISL2   #GOR_M_GO,(R0)          ; Set GO bit
                          05    0307   780          RSB
                                0308   781
                                0308   782
```

```
                                0308   784          .SBTTL  CLEAR_GO_BIT,   Clear the CINT GO bit
                                0308   785 ;+
                                0308   786 ;
                                0308   787 ;  CLEAR_GO_BIT
                                0308   788 ;
                                0308   789 ;  Routine to clear the GO bit in the CINT GOR register.
                                0308   790 ;
                                0308   791 ;  INPUTS:
                                0308   792 ;
                                0308   793 ;       R4                      - PDT address
                                0308   794 ;
                                0308   795 ;  OUTUTS:
                                0308   796 ;
                                0308   797 ;       R0                      - Destroyed
                                0308   798 ;       All other registers     - Preserved
                                0308   799 ;
                                0308   800 ;-
                                0308   801
                                0308   802 CLEAR_GO_BIT::
                                0308   803
                        00E4 C4 C1  0308   804          ADDL3   PDT$L_CNF(R4),-         ; Get GO register address
                    50  00001008 8F       0309   805                  #PA_GOR,R0
                           60  01 CA 0312   806          BICL2   #GOR_M_GO,(R0)          ; Clear GO bit
                               05   0315   807          RSB
                                0316   808
                                0316   809
```

```
                                0316   811          .SBTTL  SET_GOU_BIT,    Set the CINT GO unconditional bit
                                0316   812 ;+
                                0316   813 ;
                                0316   814 ;  SET_GOU_BIT
                                0316   815 ;
                                0316   816 ;  Routine to set the GO unconditional bit in the CINT GOU register.
                                0316   817 ;  First, however, the GO bit in the GOR register is cleared.
                                0316   818 ;
                                0316   819 ;  INPUTS:
                                0316   820 ;
                                0316   821 ;       R4                          - PDT address
                                0316   822 ;
                                0316   823 ;  OUTUTS:
                                0316   824 ;
                                0316   825 ;       R0                          - Destroyed
                                0316   826 ;       All other registers         - Preserved
                                0316   827 ;
                                0316   828 ;-
                                0316   829
                                0316   830 SET_GOU_BIT:
                                0316   831
                      FFFF   30 0316   832          BSBW    CLEAR_GO_BIT            ; Clear the GO bit
                     00E4 C4 C1 0319   833          ADDL3   PDT$L_CNF(R4),-         ; Get GOU register address
                                0319   834                  #PA_GOUR,R0
                  50 00001010 8F 031D   835          BISL2   #GOUR_M_GOU,(R0)        ; Set GO unconditional bit
                        60   01 C8 0323   836          RSB
                                05 0326   837
                                0327   838
```

```
                     0327   840          .SBTTL  CLEAR_GOU_BIT,  Clear the CINT GO unconditional bit
                     0327   841 ;+
                     0327   842 ;
                     0327   843 ;  CLEAR_GOU_BIT
                     0327   844 ;
                     0327   845 ;  Routine to clear the GO unconditional bit in the CINT GOU register.
                     0327   846 ;
                     0327   847 ;  INPUTS:
                     0327   848 ;
                     0327   849 ;       R4                          - PDT address
                     0327   850 ;
                     0327   851 ;  OUTUTS:
                     0327   852 ;
                     0327   853 ;       R0                          - Destroyed
                     0327   854 ;       All other registers         - Preserved
                     0327   855 ;
                     0327   856 ;-
                     0327   857
                     0327   858 CLEAR_GOU_BIT::
                     0327   859
            00E4 C4  C1  0327   860          ADDL3   PDT$L_CNF(R4),-         ; Get GOU register address
 50  00001010 8F      032B   861                  #PA_GOUR,R0
        60    01  CA  0331   862          BICL2   #GOUR_M_GOU,(R0)        ; Clear GO unconditional bit
              05      0334   863          RSB
                      0335   864
                      0335   865
                      0335   866
```

11
(8)

```
                          0335   868         .SBTTL  CLEAR_CLER_REG, Clear the CINT CLER register
                          0335   869 ;+
                          0335   870 ;
                          0335   871 ;  CLEAR_CLER_REG
                          0335   872 ;
                          0335   873 ;  Routine to clear all bits in the CINT control logic enable register.
                          0335   874 ;
                          0335   875 ;  INPUTS:
                          0335   876 ;
                          0335   877 ;       R3                      - Address of current command
                          0335   878 ;
                          0335   879 ;  OUTPUTS:
                          0335   880 ;
                          0335   881 ;       R0                      - Destroyed
                          0335   882 ;       All other registers     - Preserved
                          0335   883 ;
                          0335   884 ;-
                          0335   885
                          0335   886 CLEAR_CLER_REG::
                          0335   887
          00E4 C4    C1   0335   888         ADDL3   PDT$L_CNF(R4),-         ; Get CLER register address
50    00001018 8F         0339   889                 #PA_CLER,R0
                60   D4   033F   890         CLRL    (R0)                    ; Clear the CLER register
                     05   0341   891         RSB
                          0342   892
                          0342   893
                          0342   894
```

```
                              0342          .SBTTL  ARM_CINT_NORSP, Arm the CINT for NORSP's
                              0342   ;+
                              0342   ;
                              0342   ;    ARM_CINT_NORSP
                              0342   ;
                              0342   ;    Routine to arm the CINT for NORSP's.
                              0342   ;
                              0342   ;    INPUTS:
                              0342   ;
                              0342   ;         R4                         - PDT address
                              0342   ;
                              0342   ;    OUTUTS:
                              0342   ;
                              0342   ;         R0                         - Destroyed
                              0342   ;         All other registers        - Preserved
                              0342   ;
                              0342   ;-
                              0342
                              0342   ARM_CINT_NORSP:
                              0342
           00E4 C4      C1    0342          ADDL3   PDT$L_CNF(R4),-          ; Get CLER register address
  50   00001018 8F            0346                  #PA_CLER,R0
           60   10      C8    034C          BISL2   #CLER_M_NO_RSP,(R0)      ; Arm the CINT for NORSP's
                        05    034F          RSB
                              0350
                              0350
```

```
                                0350   923          .SBTTL  ARM_CINT_NAK,    Arm the CINT for NAK's
                                0350   924 ;+
                                0350   925 ;
                                0350   926 ;  ARM_CINT_NAK
                                0350   927 ;
                                0350   928 ;  Routine to arm the CINT for NAK's
                                0350   929 ;
                                0350   930 ;  INPUTS:
                                0350   931 ;
                                0350   932 ;       R4                      - PDT address
                                0350   933 ;
                                0350   934 ;  OUTUTS:
                                0350   935 ;
                                0350   936 ;       R0                      - Destroyed
                                0350   937 ;       All other registers     - Preserved
                                0350   938 ;
                                0350   939 ;-
                                0350   940
                                0350   941 ARM_CINT_NAK:
                                0350   942
                    00E4 C4     0350   943          ADDL3   PDT$L_CNF(R4),-         ; Get CLER register address
          50    00001018 8F     0354   944                  #PA_CLER,R0
                      60  08    035A   945          BISL2   #CLER_M_NAK,(R0)        ; Arm the CINT for NAK's
                          05    035D   946          RSB
                                035E   947
                                035E   948
                                035E   949
```

```
                                035E    951          .SBTTL  ARM_CINT_IGACK,  Arm the CINT for ignoring ACK's
                                035E    952 ;+
                                035E    953 ;
                                035E    954 ;  ARM_CINT_IGACK
                                035E    955 ;
                                035E    956 ;  Routine to arm the CINT for ignoring ACK's. The causes the CINT port to
                                035E    957 ;  send duplicate packets to the UUT.
                                035E    958 ;
                                035E    959 ;  INPUTS:
                                035E    960 ;
                                035E    961 ;       R4                          - PDT address
                                035E    962 ;
                                035E    963 ;  OUTUTS:
                                035E    964 ;
                                035E    965 ;       R0                          - Destroyed
                                035E    966 ;       All other registers         - Preserved
                                035E    967 ;
                                035E    968 ;-
                                035E    969
                                035E    970 ARM_CINT_IGACK:
                                035E    971
                    00E4 C4     C1  035E    972          ADDL3   PDT$L_CNF(R4),-         ; Get CLER register address
          50    00001018 8F         0362    973                  #PA_CLER,R0
                      60 20     C8  0368    974          BISL2   #CLER_M_IGACK,(R0)      ; Arm the CINT ignore ACK logic
                                05  0368    975          RSB
                                    036C    976
                                    036C    977
```

```
                                036C   979          .SBTTL  ARM_CINT_PATH,  Arm the CINT for saving received path status
                                036C   980 ;+
                                036C   981 ;
                                036C   982 ;  ARM_CINT_PATH
                                036C   983 ;
                                036C   984 ;  Routine to write the CINT control logic enable register to arm the CINT
                                036C   985 ;  for saving received path status.
                                036C   986 ;
                                036C   987 ;  INPUTS:
                                036C   988 ;
                                036C   989 ;       R4                      - PDT address
                                036C   990 ;
                                036C   991 ;  OUTPUTS:
                                036C   992 ;
                                036C   993 ;       R0                      - Destroyed
                                036C   994 ;       All other registers     - Preserved
                                036C   995 ;
                                036C   996 ;-
                                036C   997
                                036C   998 ARM_CINT_PATH:
                                036C   999
                    00E4 C4     036C  1000          ADDL3   PDT$L_CNF(R4),-         ; Get the CINT CLER register address
          50    00001018 8F     0370  1001                  #PA_CLER,R0
          60    00008000 8F  C8 0376  1002          BISL2   #CLER_M_STTST,(R0)      ; Set the status test bit
                               05 037D  1003          RSB
                                  037E  1004
                                  037E  1005
                                  037E  1006
```

```
                          037E  1008          .SBTTL  SAVE_CINT_PDT,  Sho  CINT logic has been used
                          037E  1009 ;+
                          037E  1010 ;
                          037E  1011 ;  SAVE_CINT_PDT
                          037E  1012 ;
                          037E  1013 ;  Routine to save the PDT address for the local CINT port. This address
                          037E  1014 ;  is then used by CANCEL to initialize the CINT port.
                          037E  1015 ;
                          037E  1016 ;  INPUTS:
                          037E  1017 ;
                          037E  1018 ;       R4                          - PDT address for CINT port
                          037E  1019 ;
                          037E  1020 ;  OUTPUTS:
                          037E  1021 ;
                          037E  1022 ;       All registers               - Preserved
                          037E  1023 ;
                          037E  1024 ;-
                          037E  1025
                          037E  1026 SAVE_CINT_PDT:
                          037E  1027
          0000'CF    54   D0  037E  1028          MOVL    R4,CX$L_CINT_PDT        ; Save PDT address for CINT port
                        05  0383  1029          RSB
                          0384  1030
                          0384  1031          .END
```

```
$$$CURSIZ                       = 000001C4
$$$NEWSIZ                       = 000001D0
ALLOC_DG_ERR                      ********    X   01
ARM_CINT_IGACK                    0000035E    R   01
ARM_CINT_NAK                      00000350    R   01
ARM_CINT_NORSP                    00000342    R   01
ARM_CINT_PATH                     0000036C    R   01
BIT...                          = 0000000D
CDB$B_CDRPCNT                     00000028
CDB$B_RSTNADR                     0000000E
CDB$B_TYPE                        0000000A
CDB$K_LEN                         00000032
CDB$L_CDBSLOT                     00000024
CDB$L_CDRPQBL                     0000002E
CDB$L_CDRPQFL                     0000002A
CDB$L_CDT                         0000001C
CDB$L_PDT                         00000020
CDB$L_QBL                         00000004
CDB$L_QFL                         00000000
CDB$T_LPRTNAM                     00000016
CDB$W_SIZE                        00000008
CDB$W_STATUS                      0000000C
CDC$ARMCINTPATH                   0000001B
CDC$B_CFGCBLSTS                   00000021
CDC$B_CFGPOLBSTS                  0000002A
CDC$B_CFGPOSTS                    0000001F
CDC$B_CFGP1LBSTS                  0000002B
CDC$B_CFGP1STS                    00000020
CDC$B_CMDCOUNT                    00000000
CDC$B_CNTFLG                      0000001F
CDC$B_COUNT                       00000015
CDC$B_CTPOPCODE                   00000014
CDC$B_DELAY                       00000019
CDC$B_DONECNT                     00000001
CDC$B_DSTPORT                     00000001
CDC$B_EXTEND                      0000001C
CDC$B_FDTCNT                      00000002
CDC$B_GARBAGE_FLAG                00000014
CDC$B_NOACTFLAG                   0000001E
CDC$B_NODEADR                     00000014
CDC$B_OPCODE                      00000000
CDC$B_OTHERNODE                   0000001E
CDC$B_PATH                        0000001C
CDC$B_PKTMULT                     0000001F
CDC$B_PKTSIZ                      0000001D
CDC$B_POLLERFLAG                  0000001E
CDC$B_PROTREV                     00000036
CDC$B_PROTVER                     00000035
CDC$B_PROTYPE                     00000034
CDC$B_QNUMBER                     0000001F
CDC$B_RST_PORT                    00000015
CDC$B_SIOCNT                      00000003
CDC$B_TYPE                        00000014
CDC$CARRIER                       00000018
CDC$CONFIGDATA                    00000007
CDC$CONNECT                       00000008
CDC$C_CMDSPEC                     00000014
CDC$DISCONNECT                    00000009
CDC$GARBAGE                       0000000B
CDC$INITCINT                      0000001A
CDC$K_CMDHDRSIZ                   0000000C
CDC$K_CMDSPEC                     00000014
CDC$K_DATA                        00000014
CDC$LRGPACKET                     00000019
CDC$L_BUFADR                      00000024
CDC$L_BUFLENGTH                   00000020
CDC$L_BUFLNAME                    00000024
CDC$L_BUFLOFSET                   00000028
CDC$L_BUFRNAME                    0000002C
CDC$L_BUFROFSET                   00000030
CDC$L_CDB_AD                      0000000C
CDC$L_CDRP                        0000000C
CDC$L_CFGHWTYPE                   00000022
CDC$L_CFGPRTMSK                   00000026
CDC$L_CNTRDISCDG                  00000038
CDC$L_CNTRP0ACK                   00000020
CDC$L_CNTRP0NAK                   00000024
CDC$L_CNTRP0NORSP                 00000028
CDC$L_CNTRP1ACK                   0000002C
CDC$L_CNTRP1NAK                   00000030
CDC$L_CNTRP1NORSP                 00000034
CDC$L_CTPREF                      00000015
CDC$L_DISCONREAS                  00000019
CDC$L_LPORT                       00000010
CDC$L_PORTHDR                     00000004
CDC$L_R0STATUS                    0000000C
CDC$L_R1STATUS                    00000010
CDC$L_RESERV04                    00000004
CDC$L_RESERV08                    00000008
CDC$L_RESERV12                    0000000C
CDC$L_SEQNUM                      00000001
CDC$L_STARTADR                    00000020
CDC$L_SVAPTE                      00000018
CDC$MAINTSTATE                    0000000F
CDC$MAPBUF                        00000005
CDC$MAPMBUF                       00000010
CDC$NAK                           00000016
CDC$NOACK                         00000017
CDC$NOACT                         0000000E
CDC$NORSP                         00000015
CDC$OVERSIZEPKT                   0000001D
CDC$POLLER                        0000000E
CDC$RDCINTPATH                    0000001C
CDC$READCNT                       0000000A
CDC$REQDATA                       00000004
CDC$REQMDATA                      00000013
CDC$RESET                         0000000C
CDC$SELADR                        00000014
CDC$SENDDATA                      00000003
CDC$SENDDG                        00000002
CDC$SENDMDATA                     00000012
CDC$SENDMSG                       00000001
CDC$START                         0000000D
CDC$T_CONDAT                      00000040
```

```
CDC$T_ENDMSG                    00000072
CDC$T_GARBAGE                   00000018
CDC$T_LOCPROCNAM                00000014
CDC$T_PACKETDATA                00000080
CDC$T_REMPROCNAM                00000024
CDC$T_SYSTEMID                  00000034
CDC$UNMAPBUF                    00000006
CDC$UNMAPMBUF                   00000011
CDC$W_CMDOFSET                  00000015
CDC$W_CMDSEQNUM                 00000017
CDC$W_DGCREDIT                  0000003E
CDC$W_FLAG                      00000004
CDC$W_MAXDG                     00000014
CDC$W_MAXMSG                    00000018
CDC$W_MINSENDCR                 0000003C
CDC$W_MSGCREDIT                 0000003A
CDC$W_POLLINT                   0000001C
CDC$W_POLLNUM                   00000020
CDC$W_REPEAT                    0000001A
CDC$W_RESERV10                  0000000A
CDC$W_RSPBUFOFSET               00000004
CDC$W_RSPCNT                    00000006
CDC$W_RSPOFSET                  00000002
CDC$W_TXTLEN                    00000008
CDC$_ALL                    =   00000001  G
CDC$_MSTATE                 =   00000000  G
CDC$_OFF                    =   00000000  G
CDC$_PATH0                  =   00000000  G
CDC$_PATH1                  =   00000001  G
CDC$_POLLERON               =   00000000  G
CDRP$B_FLAGS                    00000040
CDRP$C_BT_LEN               =   00000040
CDRP$K_CTPLEN                   00000050
CDRP$K_CX_LEN                   00000068
CDRP$K_XFRLEN                   00000068
CDRP$L_PDT                      0000004C
CDRP$L_PENDQBL                  00000048
CDRP$L_PENDQFL                  00000044
CDRP$T_CX_LBUFHNDL              00000050
CDRP$T_CX_RBUFHNDL              0000005C
CHECK_VC                        ********    X   01
CHECK_VC_ERR                    ********    X   01
CLEAR_CLER_REG                  00000335 RG     01
CLEAR_GOU_BIT                   00000327 RG     01
CLEAR_GO_BIT                    00000308 RG     01
CLEAR_STS_REG                   000002C4 RG     01
CLER_M_CARRA                =   00000002
CLER_M_CARRB                =   00000001
CLER_M_IGACK                =   00000020
CLER_M_LDSTA                =   00000080
CLER_M_NAK                  =   00000008
CLER_M_NO_RSP               =   00000010
CLER_M_OVPAC                =   00000040
CLER_M_STTST                =   00008000
CLSR_V_PATHB                =   00000005
COM$DRVDEALMEM                  ********    X   01
CRC_TABLE                       000000A3 R      01
CX$L_CINT_PDT                   ********    X   01
CX$L_PORT_STATE_PDT             ********    X   01
CX$L_RSPBUFBGN                  ********    X   01
DYN$C_SCS                   =   00000060
EXE$ALONONPAGED                 ********    X   01
EXE$GL_LOCKRTRY                 ********    X   01
GOR_M_GO                    =   00000001
GOUR_M_GOU                  =   00000001
MARK_CMD_DONE                   ********    X   01
MNT$CATCH_RST               =   00000007  G
MNT$CATCH_STR               =   00000008  G
MNT$MAINT_READ              =   00000005  G
MNT$MAINT_STATE             =   00000002  G
MNT$MAINT_WRITE             =   00000006  G
MNT$NORMAL_STATE            =   00000004  G
MNT$POLLER_OFF              =   00000000  G
MNT$POLLER_ON               =   00000001  G
MNT$UNINIT_STATE            =   00000003  G
PA_CAR                      =   00001028
PA_CLER                     =   00001018
PA_CLSR                     =   00001000
PA_CNF                          00000000
PA_CQ0                          00000908
PA_CQ0_M_CQC                =   00000001
PA_CQ1                          0000090C
PA_CQ2                          00000910
PA_CQ3                          00000914
PA_DFQ                          00000928
PA_GOR                      =   00001008
PA_GOUR                     =   00001010
PA_MADR                         00000014
PA_MDATR                        00000018
PA_MFQ                          0000092C
PA_MTC                          00000930
PA_MTEC                         00000934
PA_NNCR                     =   00001020
PA_PDC                          00000920
PA_PEC                          0000091C
PA_PESR                         0000093C
PA_PFAR                         00000938
PA_PIC                          00000924
PA_PMC                          00000004
PA_PPR                          00000940
PA_PQBBR                        00000904
PA_PS                           00000900
PA_PSR                          00000918
PA_SR                       =   C0C01030
PDT$B_DQIMAP                    00000154
PDT$B_HSHUT_DG                  000001B0
PDT$B_MAX_PORT                  0000017C
PDT$B_NXT_PORT                  0000017E
PDT$B_P0_LBSTS                  00000180
PDT$B_P1_LBSTS                  00000181
PDT$B_PLOGMAP                   00000134
PDT$B_PORTMAP                   00000114
PDT$B_PORT_NUM                  0000017D
PDT$B_REQIDPS                   0000017F
```

```
PDT$C_LENGTH                    =  000000E4
PDT$C_PAREGBASE                    000000E4
PDT$C_PAREGEND                     00000110
PDT$C_PQB                       =  000001E0
PDT$L_ALLOCDG                   =  00000010
PDT$L_CNF                          000000E4
PDT$L_CQO                          000000F0
PDT$L_CQ1                          000000F4
PDT$L_DFQ                          000000FC
PDT$L_DFQHDR                       00000208
PDT$L_DGHDRSZ                      00000190
PDT$L_DGNETHD                      00000194
PDT$L_DQELOGOUT                    000002E0
PDT$L_GPTBASE                      0000022C
PDT$L_GPTLEN                       00000230
PDT$L_LBDG                         00000184
PDT$L_MAINTFCN                  =  00000078
PDT$L_MFQ                          00000100
PDT$L_MFQHDR                       0000020C
PDT$L_MQELOGOUT                    00000320
PDT$L_MTC                          00000104
PDT$L_PFAR                         00000108
PDT$L_PMC                          000000E8
PDT$L_POLLERDUE                    0000018C
PDT$L_POOLDUE                      00000188
PDT$L_PPR                          0000010C
PDT$L_PS                           000000EC
PDT$L_PSR                          000000F8
PDT$L_SPTBASE                      00000224
PDT$L_SPTLEN                       00000228
PDT$L_VBDT                         0000021C
PDT$L_VPQB                         00000218
PDT$Q_COMQ2                        000001F0
PDT$Q_COMQ3                        000001F8
PDT$Q_COMQBASE                     000001E0
PDT$Q_COMQH                        000001E8
PDT$Q_COMQL                        000001E0
PDT$Q_DFREEQ                       000001D0
PDT$Q_FORMPB                       00000174
PDT$Q_MFREEQ                       000001D8
PDT$Q_RSPQ                         00000200
PDT$Q_TEMP_RSPQ                    00000198
PDT$W_BDTLEN                       00000220
PDT$W_DQELEN                       00000210
PDT$W_LPORT_STS                    00000110
PDT$W_MQELEN                       00000214
PDT$W_PBCOUNT                      00000112
PPD$B_DEF_ST                       0000001C
PPD$B_FLAGS                        0000000F
PPD$B_HWVERS                       00000034
PPD$B_LBDATA                       00000012
PPD$B_LCB_0                        00000012
PPD$B_LCB_LPORT                    00000010
PPD$B_LCB_NPORT                    0000000F
PPD$B_LCB_OPC                      00000011
PPD$B_LCB_PORT                     0000000E
PPD$B_OPC                          0000000F
PPD$B_PORT                         0000000C
PPD$B_PROTOCOL                     0000001A
PPD$B_RSTATE                       00000025
PPD$B_RST_PORT                     00000024
PPD$B_STATUS                       0000000D
PPD$B_SWFLAG                       0000000B
PPD$B_SYSTEMID                     00000014
PPD$B_TYPE                         0000000A
PPD$C_LBDAT_LEN                 =  00000030
PPD$C_LB_LENGTH                    00000046
PPD$C_LCB_DATA                     00000013
PPD$C_LENGTH                       00000012
PPD$C_MIN_DGSIZ                    00000050
PPD$C_SNDLB                     =  0000000D
PPD$K_LB_LENGTH                    00000046
PPD$K_LENGTH                       00000012
PPD$L_BLINK                        00000004
PPD$L_DG_DISC                      00000028
PPD$L_FLINK                        00000000
PPD$L_IN_VCD                       00000018
PPD$L_LBCRC                        00000042
PPD$L_PO_ACK                       00000010
PPD$L_PO_NAK                       00000014
PPD$L_PO_NRSP                      00000018
PPD$L_P1_ACK                       0000001C
PPD$L_P1_NAK                       00000020
PPD$L_P1_NRSP                      00000024
PPD$L_REC_BOFF                     00000028
PPD$L_REC_NAME                     00000024
PPD$L_RPORT_FCN                    00000020
PPD$L_RPORT_REV                    0000001C
PPD$L_RPORT_TYP                    00000018
PPD$L_SND_BOFF                     00000020
PPD$L_SND_NAME                     0000001C
PPD$L_ST_ADDR                      00000018
PPD$L_XCT_LEN                      00000018
PPD$Q_CURTIME                      00000048
PPD$Q_NODENAME                     00000040
PPD$Q_SWINCARN                     00000028
PPD$Q_XCT_ID                       00000010
PPD$T_HWTYPE                       00000030
PPD$T_SWTYPE                       00000020
PPD$T_SWVERS                       00000024
PPD$W_LCB_LEN7                     0000000C
PPD$W_LENGTH                       00000010
PPD$W_MASK                         00000010
PPD$W_MAXDG                        0000001C
PPD$W_MAXMSG                       0000001E
PPD$W_MTYPE                        00000012
PPD$W_M_VAL                        00000014
PPD$W_SIZE                         00000008
SAVE_CINT_PDT                      0000037E  R      01
SET_GOU_BIT                        00000316  R      01
SET_GO_BIT                         000002F7  R      01
SIO_ARMPATH                        00000047  RG     01
SIO_CARRIER                        00000256  RG     01
SIO_INITCINT                       0000002C  RG     01
```

```
SIO_LRGPACKET                   00000250 RG    01
SIO_MAINT_STATE                 00000299 RG    01
SIO_NAK                         000001E9 RG    01
SIO_NOACK                       00000212 RG    01
SIO_NORSP                       000001C0 RG    01
SIO_NXT_CMD                     ******** X     01
SIO_OVERSIZEPKT                 0000022D RG    01
SIO_POLLER                      00000000 RG    01
SIO_READPATH                    00000062 RG    01
SIO_SELADR                      000000E3 RG    01
SIZ...                      =   00000001
STATE_M_ALTGRBG             =   00000004
STATE_M_CANCEL              =   00000008
STATE_M_CMAP                =   00000002
STATE_M_FDT                 =   00000400
STATE_M_NOTPROC             =   00001000
STATE_M_RMAP                =   00000001
STATE_M_SIO                 =   00000800
STATE_M_STOP                =   00000200
STATE_V_ALTGRBG             =   00000002
STATE_V_CANCEL              =   00000003
STATE_V_CMAP                =   0000C001
STATE_V_FDT                 =   0000000A
STATE_V_NOTPROC             =   0000000C
STATE_V_RMAP                =   00000000
STATE_V_SIO                 =   0000000B
STATE_V_STOP                =   00000009
WRITE_COUNT_REG                 00000202 R     01
WRITE_SRC_REG                   000002E8 R     01
```

+----------------+
! Psect synopsis !
+----------------+

```
PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
  ABS                           00000000  (     0.) 00 (  0.) NOPIC   USR  CON   ABS  LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$I$115_DRIVER                   00000384  (   900.) 01 (  1.) NOPIC   USR  CON   REL  LCL NOSHR   EXE   RD    WRT NOVEC LONG
$ABS$                           00000944  ( 2372.) 02 (  2.) NOPIC   USR  CON   ABS  LCL NOSHR   EXE   RD    WRT NOVEC BYTE
```

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION		REFERENCES...							
ALLOC_DG_ERR	00000000-XR			#-301	(6)						
ARM_CINT_GACK	0000035E-R	970	(24)	#-501	(9)						
ARM_CINT_NAK	00000350-R	941	(23)	#-461	(8)						
ARM_CINT_NORSP	00000342-R	914	(22)	#-420	(7)						
ARM_CINT_PATH	0000036C-R	998	(25)	#-195	(4)						
BIT...	=0000000D	87	(1)	87	(1)						
CDC$B_COUNT	00000015			#-715	(15)						
CDC$B_NODEADR	00000014			#-313	(6)	#-322	(6)				
CDC$B_OTHERNODE	0000001E			#-750	(16)						
CDC$B_PATH	0000001C			#-236	(5)	#-241	(5)	#-593	(12)	#-598	(12)
CDC$B_POLLERFLAG	0000001E			#-117	(2)						
CDC$B_RST_PORT	00000015			#-654	(13)						
CDC$B_TYPE	00000014			#-422	(7)	#-463	(8)	#-609	(12)	#-649	(13)
CDC$W_RSPOFSET	00000002			#-235	(5)						
CDC$_ALL	=00000001			#-422	(7)	#-463	(8)				
CDC$_MSTATE	=00000000			647	(13)						
CDC$_OFF	=00000000			607	(12)						
CDC$_PATH0	=00000000			226	(5)	591	(12)				
CDC$_PATH1	=00000001			#-240	(5)	#-599	(12)				
CDC$_POLLERON	=00000000			115	(2)						
CHECK_VC	00000000-XR			#-112	(2)	#-156	(3)	#-191	(4)	#-228	(5)
				#-297	(6)	#-415	(7)	#-456	(8)	#-497	(9)
				#-529	(10)	#-584	(12)	#-643	(13)		
CHECK_VC_ERR	00000000-XR			#-113	(2)	#-157	(3)	#-192	(4)	#-229	(5)
				#-298	(6)	#-416	(7)	#-457	(8)	#-498	(9)
				#-530	(10)	#-585	(12)	#-644	(13)		
CLEAR_CLER_REG	00000335-R	886	(21)	#-161	(3)						
CLEAR_GOU_BIT	00000327-R	858	(20)	#-160	(3)	#-776	(17)				
CLEAR_GO_BIT	00000308-R	802	(18)	#-159	(3)	#-832	(19)				
CLEAR_STS_REG	000002C4-R	682	(14)	#-162	(3)						
CLER_M_CARRA	=00000002			#-595	(12)	#-604	(12)				
CLER_M_CARRB	=00000001			#-601	(12)	#-605	(12)				
CLER_M_IGACK	=00000020			#-974	(24)						
CLER_M_LDSTA	=00000080			#-233	(5)						
CLER_M_NAK	=00000008			#-945	(23)						
CLER_M_NO_RSP	=00000010			#-918	(22)						
CLER_M_OVRAC	=00000040			#-534	(10)						
CLER_M_STTST	=00008000			#-1002	(25)						
CLSR_V_PATHB	=00000005			#-239	(5)						
COM$DRVDEALMEM	00000000-XR			373	(6)						
CRC_TABLE	000000A3-R	276	(6)	368	(6)						
CXSL_CIN_DT	00000000-XR			#-1028	(26)						
CXSL_PORT_STATE_PDT	00000C00-XR			#-645	(13)						
CXSL_RSPBL_BGN	00000000-XR			#-235	(5)						
DYN$C_SCS	=00000060			#-349	(6)						
EXE$ALONONPAGED	00000000-XR			346	(6)						
EXE$GL_LOCKRTRY	00000000-XR			#-315	(6)						
GOR_M_GO	=00000001			#-779	(17)	#-806	(18)				
GOUR_M_GOU	=00000001			#-835	(19)	#-862	(20)				
MARK_CMD_DONE	00000000-XR			#-124	(2)	#-164	(3)	#-199	(4)	#-244	(5)

Macros Cross Reference

MACRO	SIZE	DEFINITION	REFERENCES...
$CDCDEF	8	83 (1)	83 (1)
$CDRPDEF	6	58 (1)	58 (1)
$CDTDEF	4	59 (1)	59 (1)
$CIBHANDEF	1	60 (1)	60 (1)
$CINTREGDEF	3	84 (1)	84 (1)
$CXCDBDEF	2	85 (1)	85 (1)
$CXCDRPDEF	1	86 (1)	86 (1)
$CXSTATEDEF	1	87 (1)	87 (1)
$DEFINI	1	58 (1)	58 (1) 59 (1) 60 (1) 61 (1) 62 (1) 63 (1) 64 (1) 65 (1) 66 (1) 67 (1) 68 (1) 69 (1) 70 (1) 75 (1) 76 (1) 77 (1) 83 (1) 84 (1) 85 (1) 86 (1)
$DYNDEF	2	61 (1)	61 (1)
$INS_COMQLOW	1	315 (6)	315 (6)
$IODEF	17	62 (1)	62 (1)
$IRPDEF	7	63 (1)	63 (1)
$PAMAINTDEF	1	78 (1)	78 (1)
$PAPDTDEF	5	75 (1)	75 (1)
$PAREGDEF	5	76 (1)	76 (1)
$PBDEF	3	64 (1)	64 (1)
$PDTDEF	4	65 (1)	65 (1)
$PPDDEF	10	77 (1)	77 (1)
$QRETRY	1	315 (6)	315 (6)
$SBDEF	2	66 (1)	66 (1)
$SCSDEF	4	67 (1)	67 (1)
$SSDEF	2	68 (1)	68 (1)
$SYSAPDEF	2	69 (1)	69 (1)
$UCBDEF	16	70 (1)	70 (1)
$VIELD1	1		87 (1)
ALLOC_DG_BUF	1	300 (6)	300 (6)
ASSUME	1		115 (2) 226 (5) 591 (12) 607 (12) 647 (13)
BLBCW	1	113 (2)	113 (2) 157 (3) 192 (4) 229 (5) 298 (6) 301 (6) 347 (6) 416 (7) 457 (8) 498 (9) 530 (10) 585 (12) 644 (13)
CALC_RSP	1	235 (5)	235 (5)
DISABLE_POLLER	1	122 (2)	122 (2)
ENABLE_POLLER	1	119 (2)	119 (2)
SET_MAINT_STATE	1	651 (13)	651 (13)
SET_UNINIT_STATE	1	655 (13)	655 (13)
_VIELD	1		87 (1)

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	91	00:00:00.38	00:00:01.25
Command processing	101	00:00:00.63	00:00:01.42
Pass 1	751	00:00:31.38	00:00:42.50
Symbol table sort	0	00:00:03.96	00:00:04.47
Pass 2	165	00:00:04.91	00:00:09.23
Symbol table output	42	00:00:00.34	00:00:00.98
Psect synopsis output	2	00:00:00.02	00:00:00.03
Cross-reference output	35	00:00:00.61	00:00:00.65
Assembler run totals	1190	00:00:42.25	00:01:00.54

The working set limit was 2400 pages.
147103 bytes (288 pages) of virtual memory were used to buffer the intermediate code.
There were 140 pages of symbol table space allocated to hold 2528 non-local and 34 local symbols.
1031 source lines were read in Pass 1, producing 18 object records in Pass 2.
135 pages of virtual memory were used to define 44 macros.

Macro library statistics

Macro library name	Macros defined
DRB2:[SHULL.EVXCI.CXDRIVER]PALIB.MLB;1	5
SYS$SYSROOT:[SYSLIB]LIB.MLB;1	12
DRB2:[SHULL.EVXCI.CXDRIVER]XLIB.MLB;3	12
SYS$SYSROOT:[SYSLIB]STARLET.MLB;1	7
TOTALS (all libraries)	36

2983 GETS were required to define 36 macros.

There were no errors, warnings or information messages.

MACRO/LIS/ANA XMAIN+SYS$LIBRARY:LIB/LIB+CXDRIVER$DIR:PALIB/LIB

```
                        1          .TITLE  CXTABLES  Driver Tables
                        2          .IDENT  'V6-000'
                        3
                        4  ;****************************************************************************
                        5  ;
                        6  ;                          COPYRIGHT (c) 1981, 1984 BY
                        7  ;                DIGITAL EQUIPMENT CORPORATION, MAYNARD,
                        8  ;                 MASSACHUSETTS.  ALL RIGHTS RESERVED.
                        9  ;
                       10  ; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY  BE  USED  AND  COPIED
                       11  ; ONLY  IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION
                       12  ; OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES THEREOF
                       13  ; MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER  PERSON.  NO
                       14  ; TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.
                       15  ;
                       16  ; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE, AND
                       17  ; SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
                       18  ;
                       19  ; DIGITAL  ASSUMES  NO  RESPONSIBILITY  FOR  THE  USE OR RELIABILITY OF ITS
                       20  ; SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.
                       21  ;
                       22  ;****************************************************************************
                       23  ;
                       24  ;++
                       25  ;
                       26  ;
                       27  ; FACILITY:      CI DIAGNOSTIC CLASS DRIVER
                       28  ;
                       29  ;
                       30  ; ABSTRACT:      This module contains the DRIVER PROLOGUE TABLE,
                       31  ;                DRIVER DISPATCH TABLE, FUNCTION DECISION TABLE.
                       32  ;
                       33  ; AUTHOR:        Richard Hennessy  27-Jul-1982
                       34  ;                Base Systems Diagnostic Engineering
                       35  ;
                       36  ; MODIFIED BY:   Jim Klumpp  6-May-83
                       37  ;                Base Systems Diagnostic Engineering
                       38  ;
                       39  ; REVISION HISTORY:
                       40  ;
                       41  ;       6-000    Dave Shull      07-July-1984
                       42  ;                VMS V4 Modifications/Release
                       43  ;
                       44  ;       5-001    Dave Shull      7-March-1984
                       45  ;                Added CX$L_RST_STR_CDRP to store Restart/Start CDRP address.
                       46  ;--
                       47
                       48
                       49          .DEFAULT DISPLACEMENT,WORD
                       50          .ENABLE SUPPRESSION
                       51          .SBTTL  DEFINITIONS
                       52
                       53  ; Set PSECT to driver code
                       54
00000000               55          .PSECT  $$$115_DRIVER,LONG
                       56
                       57  ;  System definitions (LIB.MLB):
```

```
                     0000     58
                     0000     59          $CDRPDEF                        ; Define CDRP offsets
                     0000     60          $CRBDEF                         ; Define Channel Request Block
                     0000     61          $DDBDEF                         ; Define DDB symbols
                     0000     62          $DEVDEF                         ; Define DEVICE CHAR bits
                     0000     63          $DPTDEF                         ; Define DPT offsets
                     0000     64          $DYNDEF                         ; Define DYN offsets
                     0000     65          $IODEF                          ; Define I/O FUNCTION codes
                     0000     66          $SCSDEF                         ; Define SCS packet offsets
                     0000     67          $UCBDEF                         ; Define UCB offsets
                     0000     68          $VECDEF                         ; Define INTERRUPT DISPATCH VECTOR offsets
                     0000     69
                     0000     70
                     0000     71  ;  PADRIVER definitions (PALIB.MLB):
                     0000     72
                     0000     73          $PPDDEF                         ; Define PPD offsets
                     0000     74
                     0000     75
                     0000     76  ;  CXDRIVER definitions (CXLIB.MLB):
                     0000     77
                     0000     78          $CDCDEF                         ; Define CDC interface offsets
                     0000     79          $CXCDBDEF                       ; Define CXCDB offsets
                     0000     80          $CXCDRPDEF                      ; Define CX CDRP extension
                     0000     81          $CXSTATEDEF                     ; Define CXDRIVER states
                     0000     82
                     0000     83
```

```
                        85          .SBTTL  DRIVER PROLOGUE TABLE
                        86
                        87 ;+
                        88 ;
                        89 ;  DPTAB
                        90 ;
                        91 ;  This table defines some basic information about the driver in a standard
                        92 ;  form, so that VMS can load this driver.
                        93 ;
                        94 ;-
                        95
0000                    96          DPTAB   -                               ; DRIVER PROLOGUE TABLE
                        97                  END=CX_END,-                    ; End of driver
                        98                  ADAPTER=NULL,-                  ; The adapter is a CI port
                        99                  FLAGS=DPT$M_NOUNLOAD,-          ; CXDRIVER can not be unloaded
                       100                  UCBSIZE=UCB$K_LENGTH,-          ; Use standard UCB
                       101                  MAXUNITS=1,-                    ; One unit
                       102                  NAME=CXDRIVER                   ; Driver's name
0038                   103
0038                   104          DPT_STORE INIT                          ; Control block init values
0038                   105                                                  ;
0038                   106          DPT_STORE UCB,UCB$B_FIPL,B,IPL$_SCS     ; The driver's fork IPL
003C                   107          DPT_STORE UCB,UCB$L_DEVCHAR,L,-         ; DEVICE CHARACTERISTICS
003C                   108                  <DEV$M_AVL-                     ;       Available
003C                   109                  !DEV$M_IDV-                     ;       Input device
003C                   110                  !DEV$M_ODV>                     ;       Output device
0043                   111          DPT_STORE UCB,UCB$B_DIPL,B,20           ; Device IPL
0047                   112
0047                   113          DPT_STORE REINIT                        ; Control block RE-INIT values
0047                   114                                                  ;
0047                   115          DPT_STORE DDB,DDB$L_DDT,D,CX$DDT        ; DDT address
004C                   116          DPT_STORE CRB,-                         ; UNIT Init Routine
004C                   117                  CRB$L_INTD+VEC$L_UNITINIT,-
004C                   118                  D,CX_UNIT_INIT
0051                   119
0051                   120          DPT_STORE END                           ; End of init tables
0000                   121
```

```
                                .SBTTL  DRIVER DISPATCH TABLE

                        ;+
                        ;
                        ; This table lists entry points into the class driver
                        ;
                        ;-

0000                    DDTAB   -                       ; Driver dispatch table
0000                            DEVNAM=CX,-             ; Device name
0000                            START=CX_STARTIO,-      ; Start I/O routine
0000                            FUNCTB=CX_FUNCTABLE,-   ; Function decision table
0000                            CANCEL=CX_CANCELIO      ; Cancel I/O routine
0038
```

```
0038    138          .SBTTL  FUNCTION DECISION TABLE
0038    139
0038    140 ;+
0038    141 ;  CX_FUNCTABLE
0038    142 ;
0038    143 ;  The function decision table provides a list of the functions that are
0038    144 ;  supported by the driver and their associated action routines.
0038    145 ;
0038    146 ;-
0038    147
0038    148 CX_FUNCTABLE::
0038    149
0038    150          FUNCTAB ,-                                  ; LEGAL FUNCTIONS
0038    151                  <READVBLK,-
0038    152                   READPBLK,-
0038    153                   READLBLK,-
0038    154                  >
0040    155
0040    156          FUNCTAB ,-                                  ; BUFFERED I/O FUNCTIONS
0040    157                  <READVBLK,-
0040    158                   READPBLK,-
0040    159                   READLBLK,-
0040    160                   FORCE,-
0040    161                  >
0048    162
0048    163          FUNCTAB CX_STARTDATA,-                      ; The STARTDATA FDT routine
0048    164                  <READVBLK,-
0048    165                   READPBLK,-
0048    166                   READLBLK,-
0048    167                  >
0054    168
0054    169
```

```
                                        0054   171          .SBTTL  MISCELLANEOUS SYMBOL DEFINITIONS
                                        0054   172
                                        0054   173
                                        0054   174
                            00000010    0054   175 MAX_NODES        ==      16              ; Maximum remote ports in cluster
                            00000002    0054   176 MAX_LOC_PORTS    ==      2               ; Maximum local ports
                                        0054   177
                                        0054   178
4C 4C 4F 52 54 4E 4F 43 24 50 54 43     0054   179 CONTROLLER_NAME::       .ASCII /CTP$CONTROLLER  /
                      20 20 52 45       0060
45 44 4E 4F 50 53 45 52 24 50 54 43     0064   180 RESPONDER_NAME::        .ASCII /CTP$RESPONDER   /
                      20 20 20 52       0070
                                        0074   181
                                        0074   182
                                        0074   183 ;*****************************************************************************
                                        0074   184 ;
                                        0074   185 ;  GARBAGE BUFFER DEFINITIONS...Still requires work
                                        0074   186 ;
                            00000200    0074   187 GARBAGE_SIZE     ==      512             ; Total area for unsolicited inputs
                            00000080    0074   188 GARBCANSIZ       ==      128             ; Max size of each unsolicited input
                                        0074   189 ;
                                        0074   190 ;*****************************************************************************
                                        0074   191
                                        0074   192
                                        0074   193
                                        0074   194
                                        0074   195
```

```
                                0074    197          .SBTTL  DRIVER INTERNAL STORAGE DECLARATIONS
                                0074    198
                                0074    199
                    00000078    0074    200 CX$L_STATE::            .BLKL           ; Class driver internal status
                                0078    201
                    00000090    0078    202 CX$X_IRPSAV::           .BLKL   6       ; Saved R3 - R8. Used during
                                0090    203                                         ; command processing
                                0090    204
                    000000AC    0090    205 CX$X_CNCLSAV::          .BLKL   7       ; Saved R3 - R9. Used in cancel
                                00AC    206                                         ; code
                                00AC    207
                    000000C4    00AC    208 CX$X_STOPSAV::          .BLKL   6       ; Saved R3 - R8. Used in stop
                                00C4    209                                         ; code
                                00C4    210
                    000000C5    00C4    211 CX$B_CMDERRCNT::        .BLKB   1       ; STARTIO error count. Used to
                                00C5    212                                         ; determine if command is done
                                00C5    213
                    000000C9    00C5    214 CX$L_TEMP::             .BLKL   1       ; Temporary scratch longword
                                00C9    215
                    000000CD    00C9    216 CX$L_CETEMP::           .BLKL   1       ; Scratch for CONNECT_ERR code
                    000000D1    00CD    217 CX$L_CECC::             .BLKL   1       ; CONNECT_ERR current command
                                00D1    218
                    000000D5    00D1    219 CX$L_STOPCMD::          .BLKL   1       ; Current cmd for STOP_PROC
                                00D5    220
                    000000D9    00D5    221 CX$L_SEQNUM::           .BLKL   1       ; Command sequence number
                                00D9    222
                    000000DD    00D9    223 CX$L_CMDBUFBGN::        .BLKL   1       ; Start of command buffer
                                00DD    224 CX$L_CMDBUFEND::                        ; End of command buffer
                    000000E1    00DD    225 CX$L_RSPBUFBGN::        .BLKL   1       ; Start of response buffer
                    000000E5    00E1    226 CX$L_RSPBUFEND::        .BLKL   1       ; End of response buffer
                    000000E9    00E5    227 CX$L_CUR_CMD::          .BLKL   1       ; Current command in cmd buffer
                    000000ED    00E9    228 CX$L_NXT_RSP::          .BLKL   1       ; Addr of next response area
                    000000F1    00ED    229 CX$L_RSPCNTREM::        .BLKL   1       ; Space left in response buffer
                    000000F5    00F1    230 CX$L_RSPSEQNUM::        .BLKL   1       ; Response sequence number
                                00F5    231
                                00F5    232
                    000000FD    00F5    233 CX$Q_MAPLST::           .BLKQ   1       ; Mapped buffer CDRP list
                                00FD    234
                                00FD    235
                                00FD    236 ;  CXDRIVER maintains information about connections on a local port / remote
                                00FD    237 ;  port basis. Thus, for each combination of local and remote port, the
                                00FD    238 ;  controller can have a connection. The connection state is stored in a block
                                00FD    239 ;  called a CDB. The structures look as follows::
                                00FD    240 ;
                                00FD    241 ;
                                00FD    242 ;       PAA0 pointer -------> Remote port 0 ---------> CDB
                                00FD    243 ;                             Remote port 1
                                00FD    244 ;       PAB0 pointer                .
                                00FD    245 ;                                   .
                                00FD    246 ;            .                      .
                                00FD    247 ;            .                Remote port n ---------> CDB
                                00FD    248 ;            .
                                00FD    249 ;
                                00FD    250 ;       PAx0 pointer -------> Remote port 0 ---------> CDB
                                00FD    251 ;                             Remote port 1
                                00FD    252 ;                                   .
                                00FD    253 ;                                   .
```

```
$$$                             = 00000020 R      03
$$QP                            = 00000002
AT$_NULL                          ********    X   03
BIT...                          = 0000000D
CDB$B_CDRPCNT                     00000028
CDB$B_RSTNADR                     0000000E
CDB$B_TYPE                        0000000A
CDB$K_LEN                         00000032
CDB$L_CDBSLOT                     00000024
CDB$L_CDRPQBL                     0000002E
CDB$L_CDRPQFL                     0000002A
CDB$L_CDT                         0000001C
CDB$L_PDT                         00000020
CDB$L_QBL                         00000004
CDB$L_QFL                         00000000
CDB$T_LPRTNAM                     00000016
CDB$W_SIZE                        00000008
CDB$W_STATUS                      0000000C
CDC$ARMCINTPATH                   0000001B
CDC$B_CFGCBLSTS                   00000021
CDC$B_CFGPOLBSTS                  0000002A
CDC$B_CFGPOSTS                    0000001F
CDC$B_CFGP1LBSTS                  00000028
CDC$B_CFGP1STS                    00000020
CDC$B_CMDCOUNT                    00000000
CDC$B_CNTFLG                      0000001F
CDC$B_COUNT                       00000015
CDC$B_CTPOPCODE                   00000014
CDC$B_DELAY                       00000019
CDC$B_DONECNT                     00000001
CDC$B_DSTPORT                     00000001
CDC$B_EXTEND                      0000001C
CDC$B_FDTCNT                      00000002
CDC$B_GARBAGE_FLAG                00000014
CDC$B_NOACTFLAG                   0000001E
CDC$B_NODEADR                     00000014
CDC$B_OPCODE                      00000000
CDC$B_OTHERNODE                   0000001E
CDC$B_PATH                        0000001C
CDC$B_PKTMULT                     0000001F
CDC$B_PKTSIZ                      0000001D
CDC$B_POLLERFLAG                  0000001E
CDC$B_PROTREV                     00000036
CDC$B_PROTVER                     00000035
CDC$B_PROTYPE                     00000034
CDC$B_QNUMBER                     0000001F
CDC$B_RST_PORT                    00000015
CDC$B_SIOCNT                      00000003
CDC$B_TYPE                        00000014
CDC$CARRIER                       00000018
CDC$CONFIGDATA                    00000007
CDC$CONNECT                       00000008
CDC$C_CMDSPEC                     00000014
CDC$DISCONNECT                    00000009
CDC$GARBAGE                       00000008
CDC$INITCINT                      0000001A
CDC$K_CMDHDRSIZ                   0000000C
CDC$K_CMDSPEC                     00000014
CDC$K_DATA                        00000014
CDC$LRGPACKET                     00000019
CDC$L_BUFADR                      00000024
CDC$L_BUFLENGTH                   00000020
CDC$L_BUFLNAME                    00000024
CDC$L_BUFLOFSET                   00000028
CDC$L_BUFRNAME                    0000002C
CDC$L_BUFROFSET                   00000030
CDC$L_CDB_AD                      0000000C
CDC$L_CDRP                        0000000C
CDC$L_CFGHWTYPE                   00000022
CDC$L_CFGPRTMSK                   00000026
CDC$L_CNTRDISCDG                  00000038
CDC$L_CNTRP0ACK                   00000020
CDC$L_CNTRP0NAK                   00000024
CDC$L_CNTRP0NORSP                 00000028
CDC$L_CNTRP1ACK                   0000002C
CDC$L_CNTRP1NAK                   00000030
CDC$L_CNTRP1NORSP                 00000034
CDC$L_CTPREF                      00000015
CDC$L_DISCONREAS                  00000019
CDC$L_LPORT                       00000010
CDC$L_PORTHDR                     00000004
CDC$L_R0STATUS                    0000000C
CDC$L_R1STATUS                    00000010
CDC$L_RESERV04                    00000004
CDC$L_RESERV08                    00000008
CDC$L_RESERV12                    0000000C
CDC$L_SEQNUM                      00000001
CDC$L_STARTADR                    00000020
CDC$L_SVAPTE                      00000018
CDC$MAINTSTATE                    0000000F
CDC$MAPBUF                        00000005
CDC$MAPMBUF                       00000010
CDC$NAK                           00000016
CDC$NOACK                         00000017
CDC$NOACT                         0000000E
CDC$NORSP                         00000015
CDC$OVERSIZEPKT                   0000001D
CDC$POLLER                        0000000E
CDC$RDCINTPATH                    0000001C
CDC$READCNT                       0000000A
CDC$REQDATA                       00000004
CDC$REQMDATA                      00000013
CDC$RESET                         0000000C
CDC$SELADR                        00000014
CDC$SENDDATA                      00000003
CDC$SENDDG                        00000002
CDC$SENDMDATA                     00000012
CDC$SENDMSG                       00000001
CDC$START                         0000000D
CDC$T_CONDAT                      00000040
CDC$T_ENDMSG                      00000072
CDC$T_GARBAGE                     00000018
CDC$T_LOCPROCNAM                  00000014
CDC$T_PACKETDATA                  00000080
```

```
CDC$T_REMPROCNAM                00000024
CDC$T_SYSTEMID                  00000034
CDC$UNMAPBUF                    00000006
CDC$UNMAPMBUF                   00000011
CDC$W_CMDOFSET                  00000015
CDC$W_CMDSEQNUM                 00000017
CDC$W_DGCREDIT                  0000003E
CDC$W_FLAG                      00000004
CDC$W_MAXDG                     00000014
CDC$W_MAXMSG                    00000018
CDC$W_MINSENDCR                 0000003C
CDC$W_MSGCREDIT                 0000003A
CDC$W_POLLINT                   0000001C
CDC$W_POLLNUM                   00000020
CDC$W_REPEAT                    0000001A
CDC$W_RESERV10                  0000000A
CDC$W_RSPBUFOFSET               00000004
CDC$W_RSPCNT                    00000006
CDC$W_RSPOFSET                  00000002
CDC$W_TXTLEN                    00000008
CDRP$B_FLAGS                    00000040
CDRP$C_BT_LEN                 = 00000040
CDRP$K_CTPLEN                   00000050
CDRP$K_CX_LEN                   00000068
CDRP$K_XFRLEN                   00000068
CDRP$L_PDT                      0000004C
CDRP$L_PENDQBL                  00000048
CDRP$L_PENDQFL                  00000044
CDRP$T_CX_LBUFHNDL              00000050
CDRP$T_CX_RBUFHNDL              0000005C
CONTROLLER_NAME                 00000054 RG     01
CRB$L_INTD                    = 00000024
CX$B_CMDERRCNT                  000000C4 RG     01
CX$B_GARBAGE_FLAG               00000145 RG     01
CX$B_PWRFAIL                    000001C6 RG     01
CX$DDT                          00000000 RG     01
CX$L_CECC                       000000CD RG     01
CX$L_CETEMP                     000000C9 RG     01
CX$L_CINT_PDT                   000001C7 RG     01
CX$L_CMDBUFBGN                  000000D9 RG     01
CX$L_CMDBUFEND                  000000DD RG     01
CX$L_CUR_CMD                    000000E5 RG     01
CX$L_LOCPRTLST                  000000FD RG     01
CX$L_NXT_RSP                    000000E9 RG     01
CX$L_PORT_STATE_PDT             000001D3 RG     01
CX$L_RSPBUFBGN                  000000DD RG     01
CX$L_RSPBUFEND                  000000E1 RG     01
CX$L_RSPCNTREM                  000000ED RG     01
CX$L_RSPSEQNUM                  000000F1 RG     01
CX$L_RST_STR_CDRP               000001CF RG     01
CX$L_RST_STR_PDT                000001CB RG     01
CX$L_SEQNUM                     000000D5 RG     01
CX$L_STATE                      00000074 RG     01
CX$L_STOPCMD                    000000D1 RG     01
CX$L_TEMP                       000000C5 RG     01
CX$Q_MAPLST                     000000F5 RG     01
CX$T_GARBAGE_BUF                00000146 RG     01
CX$X_CNCLSAV                    00000090 RG     01
CX$X_IRPSAV                     00000078 RG     01
CX$X_STOPSAV                    000000AC RG     01
CX_CANCELIO                     ******** X      01
CX_DG_RSP_SIZ                   00000141 RG     01
CX_END                          ******** X      03
CX_FUNCTABLE                    00000038 RG     01
CX_MSG_RSP_SIZ                  0000013D RG     01
CX_STARTDATA                    ******** X      01
CX_STARTIO                      ******** X      01
CX_UNIT_INIT                    ******** X      03
DDB$L_DDT                     = 0000000C
DEV$M_AVL                     = 00040000
DEV$M_IDV                     = 04000000
DEV$M_ODV                     = 08000000
DPT$C_LENGTH                  = 00000038
DPT$C_VERSION                 = 00000004
DPT$INITAB                      00000038 R      03
DPT$M_NOUNLOAD                = 00000004
DPT$REINITAB                    00000047 R      03
DPT$TAB                         00000000 R      03
DYN$C_CRB                     = 00000005
DYN$C_DDB                     = 00000006
DYN$C_DPT                     = 0000001E
DYN$C_UCB                     = 00000010
FUNCTAB_LEN                   = 0000001C
GARBAGE_SIZE                  = 00000200  G
GARBCANSIZ                    = 00000080  G
IO$_FORCE                     = 00000037
IO$_READLBLK                  = 00000021
IO$_READPBLK                  = 0000000C
IO$_READVBLK                  = 00000031
IO$_VIRTUAL                   = 0000003F
IOC$MNTVER                      ********   X    01
IOC$RETURN                      ********   X    01
IPL$_SCS                        ********   X    03
MASKH                         = 00020002
MASKL                         = 00001000
MAX_LOC_PORTS                 = 00000002  G
MAX_NODES                     = 00000010  G
PPD$B_DEF_ST                    0000001C
PPD$B_FLAGS                     0000000F
PPD$B_HWVERS                    00000034
PPD$B_LBDATA                    00000012
PPD$B_LCB_0                     00000012
PPD$B_LCB_LPORT                 00000010
PPD$B_LCB_NPORT                 00C0000F
PPD$B_LCB_OPC                   00000011
PPD$B_LCB_PORT                  0000000E
PPD$B_OPC                       0000000E
PPD$B_PORT                      0000000C
PPD$B_PROTOCOL                  0000001A
PPD$B_RSTATE                    00000025
PPD$B_RST_PORT                  00000024
PPD$B_STATUS                    0000000D
PPD$B_SWFLAG                    0000000B
PPD$B_SYSTEMID                  00000014
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
$$$105_DRIVER	000001D7 (471.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
ABS	00000080 (128.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
$$$105_PROLOGUE	00000052 (82.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE

Symbol Cross Reference

SYMBOL	VALUE	DEFINITION	REFERENCES...
$$$	=00000020-R	102 (2)	102 (2)
$$OP	=00000002	118 (2)	106 (2) 110 (2) 111 (2) 115 (2) 118 (2)
AT$_NULL	00000000-XR		102 (2)
BIT...	=0000000D	81 (1)	81 (1)
CONTROLLER_NAME	00000054-R	179 (5)	
CRB$L_INTD	=00000024		118 (2)
CX$B_CMDERRCNT	000000C4-R	211 (6)	
CX$B_GARBAGE_FLAG	00000145-R	266 (6)	
CX$B_PWRFAIL	000001C6-R	269 (6)	
CX$DDT	00000000-R	135 (3)	115 (2) 135 (3)
CX$L_CECC	000000CD-R	217 (6)	
CX$L_CETEMP	000000C9-R	216 (6)	
CX$L_CINT_PDT	000001C7-R	272 (6)	
CX$L_CMDBUFBGN	000000D9-R	223 (6)	
CX$L_CMDBUFEND	000000DD-R	224 (6)	
CX$L_CUR_CMD	000000E5-R	227 (6)	
CX$L_LOCPRTLST	000000FD-R	261 (6)	
CX$L_NXT_RSP	000000E9-R	228 (6)	
CX$L_PORT_STATE_PDT	00000103-R	282 (6)	
CX$L_RSPBUFBGN	000000DD-R	225 (6)	
CX$L_RSPBUFEND	000000E1-R	226 (6)	
CX$L_RSPCNTREM	000000ED-R	229 (6)	
CX$L_RSPSEQNUM	000000F1-R	230 (6)	
CX$L_RST_STR_CORR	000001CF-R	278 (6)	
CX$L_RST_STR_PDT	000001CB-R	275 (6)	
CX$L_SEQNUM	000000D5-R	221 (6)	
CX$L_STATE	00000074-R	200 (6)	
CX$L_STOPCMD	000000D1-R	219 (6)	
CX$L_TEMP	000000C5-R	214 (6)	
CX$Q_MAPLST	000000F5-R	233 (6)	
CX$T_GARBAGE_BUF	00000146-R	267 (6)	
CX$X_CNCLSAV	00000090-R	205 (6)	
CX$X_IRPSAV	00000078-R	202 (6)	
CX$X_STOPSAV	000000AC-R	208 (6)	
CX_CANCELIO	00000000-XR		135 (3)
CX_DG_RSP_SIZ	00000141-R	264 (6)	
CX_END	00000000-XR		102 (2)
CX_FUNCTABLE	00000038-R	148 (4)	135 (3)
CX_MSG_RSP_SIZ	0000013D-R	263 (6)	
CX_STARTDATA	00000000-XR		167 (4)
CX_STARTIO	00000000-XR		135 (3)
CX_UNIT_INIT	00000000-XR		118 (2)
DDB$L_DDT	=0000000C		115 (2)
DEV$M_AVL	=00040000		110 (2)
DEV$M_IDV	=04000000		110 (2)
DEV$M_ODV	=08000000		110 (2)
DPT$C_LENGTH	=00000038		102 (2)
DPT$C_VERSION	=00000004		102 (2)
DPT$INITAB	00000038-R	104 (2)	102 (2)

Fiche	Frame	Sequence		
1	M9	116	CHECK_RST_STR, Prepare for arrival of re	7-JUL-1984
1	N9	117	CATCH_RST_STR, Catch reset or start pack	7-JUL-1984
1	C10	119	Symbol table	7-JUL-1984
1	L10	128	Psect synopsis	7-JUL-1984
1	M10	129	Cross reference	7-JUL-1984
1	I11	138	, STARTDATA FDT Routines	7-JUL-1984
1	L11	141	CX_STARTDATA, Start command list process	7-JUL-1984
1	B12	144	FDT_DISPATCH, Check validity of individu	7-JUL-1984
1	E12	147	FDT_ABORT, Abort the I/O for various rea	7-JUL-1984
1	G12	149	FDT_CONFIG, Gather configuration data	7-JUL-1984
1	H12	150	FDT_CONNECT, Establish connection	7-JUL-1984
1	I12	151	FDT_DISCONNECT, Disconnect routine	7-JUL-1984
1	J12	152	FDT_SENDMSG, Send message	7-JUL-1984
1	K12	153	FDT_SENDDG, Send datagram	7-JUL-1984
1	L12	154	FDT_MAP, Map buffer	7-JUL-1984
1	M12	155	FDT_MNTMAP, Map a maintenance buffer	7-JUL-1984
1	N12	156	FDT_UNMAP, Unmap buffer	7-JUL-1984
1	B13	157	FDT_SNDDAT, Send data	7-JUL-1984
1	C13	158	FDT_SNDMDAT, Send maintenance data	7-JUL-1984
1	D13	159	FDT_RDCNT, Read performance counters	7-JUL-1984
1	E13	160	FDT_RESET, Send reset	7-JUL-1984
1	F13	161	FDT_POLLER, Start/Stop poller	7-JUL-1984
1	G13	162	CHECK_LOC_PORT, Check local port name	7-JUL-1984
1	H13	163	ALLOC_RSP_SCS, Allocate SCS response buf	7-JUL-1984
1	I13	164	ALLOC_RSP_CTP, Allocate CTP response buf	7-JUL-1984
1	J13	165	FILL_CMD_REF, Build a command reference	7-JUL-1984
1	K13	166	CHECK_CONN, Check for connection	7-JUL-1984
1	L13	167	CHECK_ACTV_CONN, Check for an active con	7-JUL-1984
1	M13	168	CLEAR_BUFFER, Clear buffer routine	7-JUL-1984
1	N13	169	ALLOC_SEQNUM, Create a sequence number	7-JUL-1984
1	B14	170	Symbol table	7-JUL-1984
1	F14	174	Cross reference	7-JUL-1984
1	L14	180	Initialization and Misc. Routines	7-JUL-1984
1	N14	182	DEFINITIONS	7-JUL-1984
1	C15	184	UNIT INITIALIZATION ROUTINE	7-JUL-1984
1	E15	186	UNSOLICITED INTERRUPT ROUTINE	7-JUL-1984
1	F15	187	HIPL_ALLPOOL	7-JUL-1984
1	G15	188	CX_END, The END of CXDRIVER	7-JUL-1984
1	H15	189	Symbol table	7-JUL-1984
1	J15	191	Psect synopsis	7-JUL-1984
1	K15	192	Cross reference	7-JUL-1984
1	N15	195	, DRIVER INPUT routines	7-JUL-1984
1	C16	197	DEFINITIONS	7-JUL-1984
1	D16	198	CX$MSG_IN, Message input routine	7-JUL-1984
1	E16	199	CX$DG_IN, Datagram input routine	7-JUL-1984
1	H16	202	CX$CONNECT_ERR, Connection error routine	7-JUL-1984
1	J16	204	CLEAN_CMD, Clean up command resources	7-JUL-1984
1	L16	206	SET_SEQ_NUM, Set Response Sequence Numbe	7-JUL-1984
2	C1	208	COPY_RSP, Copy CTP response to response	7-JUL-1984
2	D1	209	DEAL_MSG, Deallocate an SCS Message Buff	7-JUL-1984
2	E1	210	RE_QUEUE_DG, Queue DG buffer to free que	7-JUL-1984
2	F1	211	Symbol table	7-JUL-1984
2	J1	215	Psect synopsis	7-JUL-1984
2	K1	216	Cross reference	7-JUL-1984
2	B2	220		7-JUL-1984
2	C2	221	CXMAINT	7-JUL-1984
2	D2	222		7-JUL-1984

Fiche	Frame	Sequence		
2	E2	223	SIO_POLLER, Start/Stop configuration pol	7-JUL-1984
2	F2	224	SIO_INITCINT, Initialize CINT logic	7-JUL-1984
2	G2	225	SIO_ARMPATH, Arm CINT received path logi	7-JUL-1984
2	H2	226	SIO_READPATH, Read CINT received path st	7-JUL-1984
2	I2	227	SIO_SELADR, Change CINT port address	7-JUL-1984
2	L2	230	SIO_NORSP, Arm CINT NORSP logic	7-JUL-1984
2	M2	231	SIO_NAK, Arm CINT NAK logic	7-JUL-1984
2	N2	232	SIO_NOACK, Arm CINT NOACK logic	7-JUL-1984
2	B3	233	SIO_OVERSIZEPKT,Arm CINT oversized packe	7-JUL-1984
2	C3	234	SIO_LRGPACKET, Arm CINT large packet log	7-JUL-1984
2	D3	235	SIO_CARRIER, Place carrier on specified	7-JUL-1984
2	E3	236	SIO_MAINT_STATE,Set port to maintenance	7-JUL-1984
2	F3	237	CLEAR_STS_REG, Clear CINT status registe	7-JUL-1984
2	G3	238	WRITE_COUNT_REG, Write CINT count regist	7-JUL-1984
2	H3	239	WRITE_SRC_REG, Write CINT source registe	7-JUL-1984
2	I3	240	SET_GO_BIT, Set the CINT GO bit	7-JUL-1984
2	J3	241	CLEAR_GO_BIT, Clear the CINT GO bit	7-JUL-1984
2	K3	242	SET_GOU_BIT, Set the CINT GO uncondition	7-JUL-1984
2	L3	243	CLEAR_GOU_BIT, Clear the CINT GO uncondi	7-JUL-1984
2	M3	244	CLEAR_CLER_REG, Clear the CINT CLER regi	7-JUL-1984
2	N3	245	ARM_CINT_NORSP, Arm the CINT for NORSP'	7-JUL-1984
2	B4	246	ARM_CINT_NAK, Arm the CINT for NAK's	7-JUL-1984
2	C4	247	ARM_CINT_IGACK, Arm the CINT for ignori	7-JUL-1984
2	D4	248	ARM_CINT_PATH, Arm the CINT for saving r	7-JUL-1984
2	E4	249	SAVE_CINT_PDT, Show CINT logic has been	7-JUL-1984
2	F4	250	Symbol table	7-JUL-1984
2	J4	254	Cross reference	7-JUL-1984
2	B5	259	Driver Tables	7-JUL-1984
2	D5	261	DEFINITIONS	7-JUL-1984
2	E5	262	DRIVER PROLOGUE TABLE	7-JUL-1984
2	F5	263	DRIVER DISPATCH TABLE	7-JUL-1984
2	G5	264	FUNCTION DECISION TABLE	7-JUL-1984
2	H5	265	MISCELLANEOUS SYMBOL DEFINITIONS	7-JUL-1984
2	I5	266	DRIVER INTERNAL STORAGE DECLARATIONS	7-JUL-1984
2	K5	268	Symbol table	7-JUL-1984
2	N5	271	Psect synopsis	7-JUL-1984
2	B6	272	Cross reference	7-JUL-1984
2	F6	276	Directory	7-JUL-1984

SET_SEQ_NUM, Set Response Sequ 1
COPY_RSP, Copy CTP response to 1
DEAL_MSG, Deallocate an SCS Me 1
RE_QUEUE_DG, Queue DG buffer t 1
Symbol table 1
Symbol table 1
Symbol table 1
Symbol table 1
Psect synopsis 1
Cross reference 1
Cross reference 1
Cross reference 1
Cross reference 1

CXMAINT 2

SIO_POLLER, Start/Stop configu 2
SIO_INITCINT, Initialize CINT 2
SIO_ARMPATH, Arm CINT received 2
SIO_READPATH, Read CINT receiv 2
SIO_SELADR, Change CINT port a 2
SIO_SELADR, Change CINT port a 2
SIO_SELADR, Change CINT port a 2
SIO_NORSP, Arm CINT NORSP logi 2
SIO_NAK, Arm CINT NAK logic 2
SIO_NOACK, Arm CINT NOACK logi 2
SIO_OVERSIZEPKT,Arm CINT overs 3
SIO_LRGPACKET, Arm CINT large 3
SIO_CARRIER, Place carrier or 3
SIO_MAINT_STATE,Set port to ma 3
CLEAR_STS_REG, Clear CINT stat 3
WRITE_COUNT_REG, Write CINT co 3
WRITE_SRC_REG, Write CINT sou 3
SET_GO_BIT, Set the CINT GO bi 3
CLEAR_GO_BIT, Clear the CINT G 3
SET_GOU_BIT, Set the CINT GO , 3
CLEAR_GOU_BIT, Clear the CINT 3
CLEAR_CLER_REG, Clear the CINT 3
ARM_CINT_NORSP, Arm the CINT 3
ARM_CINT_NAK, Arm the CINT fo 4
ARM_CINT_NOACK, Arm the CINT 4
ARM_CINT_PATH, Arm the CINT fo 4
SAVE_CINT_PDT, Show CINT logic 4
Symbol table 4
Symbol table 4
Symbol table 4
Symbol table 4
Cross reference 4
Cross reference 4
Cross reference 4
Cross reference 4
Cross reference 4
Driver Tables 5
Driver Tables 5
DEFINITIONS 5
DRIVER PROLOGUE TABLE 5
DRIVER DISPATCH TABLE 5
FUNCTION DECISION TABLE 5
MISCELLANEOUS SYMBOL DEFINITI 5
DRIVER INTERNAL STORAGE DECLAR 5
DRIVER INTERNAL STORAGE DECLAR 5
Symbol table 5
Symbol table 5
Symbol table 5
Psect synopsis 5
Cross reference 6
Cross reference 6
Cross reference 6
Cross reference 6
Directory 6
Directory 6
Directory 6