



```

DDDDDDDD  RRRRRRR  7777777  8888888  000000
DDDDDDDD  RRRRRRR  7777777  8888888  000000
DD      DD  RR      RR      77      88      88  00      00
DD      DD  RR      RR      77      88      88  00      00
DD      DD  RR      RR      77      88      88  00      0000
DD      DD  RR      RR      77      88      88  00      0000
DD      DD  RRRRRRR  77      8888888  00  00  00
DD      DD  RRRRRRR  77      8888888  00  00  00
DD      DD  RR  RR  77      88      88  0000  00
DD      DD  RR  RR  77      88      88  0000  00
DD      DD  RR  RR  77      88      88  00      00
DD      DD  RR  RR  77      88      88  00      00
DDDDDDDD  RR      RR  77      8888888  000000
DDDDDDDD  RR      RR  77      8888888  000000

```

```

....
....
....
....

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL IIIIII  SSSSSSS
LLLLLLLLLL IIIIII  SSSSSSS

```

```
0001          SUBROUTINE DR780 (LUN)
0002          C
0003          C Version:      'V04-000'
0004          C
0005          C*****
0006          C*
0007          C* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0008          C* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0009          C* ALL RIGHTS RESERVED.
0010          C*
0011          C* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0012          C* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0013          C* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0014          C* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0015          C* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0016          C* TRANSFERRED.
0017          C*
0018          C* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0019          C* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0020          C* CORPORATION.
0021          C*
0022          C* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0023          C* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0024          C*
0025          C*
0026          C*****
0027          C
0028          C
0029          C
0030          C      AUTHOR  BRIAN PORTER          CREATION DATE  8-MAY-1979
0031          C
0032          C
0033          C++
0034          C      Functional description:
0035          C
0036          C      This module display entries for the DR780.
0037          C
0038          C      Modified by:
0039          C
0040          C      V03-004 SAR0220          Sharon A. Reynolds,      28-Mar-1984
0041          C      Changed the call to UCBSL_OWNUIC to ORBSL_OWNER.
0042          C
0043          C      V03-003 SAR0071          Sharon A. Reynolds,      20-Jun-1983
0044          C      Changed the carriage control in the 'format' statements
0045          C      for use with ERF.
0046          C
0047          C      V03-002 SAR0038          Sharon A. Reynolds,      8-Jun-1983
0048          C      Removed brief/cryptic support.
0049          C
0050          C      v03-001 BP0003          Brian Porter,          20-AUG-1982
0051          C      Minor edit.
0052          C**
0053          C--
0054          C
0055          C      Include 'SRC$:DR32COM.FOR /NOLIST'
0106          C      INCLUDE 'SRC$:MSGHDR.FOR /NOLIST'
0165          C      INCLUDE 'SRC$:DEVERR.FOR /NOLIST'
```

XF

A

AR

LA

FU

CO

CO

```

0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322

```

BYTE	LUN
INTEGER*4	FIELD
INTEGER*4	FIELD1
INTEGER*4	COMPRESSC
REAL	FLOAT
INTEGER*4	COMPRESS4
INTEGER*4	COMPRESSF
INTEGER*4	DR_CR
INTEGER*4	DR_UTL
INTEGER*4	DR_WCS(1:3)
INTEGER*4	DR_BA
INTEGER*4	DR_BC(1:2)
INTEGER*4	DR_CA(1:2)
INTEGER*4	DR_WD(1:5)
INTEGER*4	DR_SL
INTEGER*4	DR_SBR
INTEGER*4	DR_GBR
INTEGER*4	DR_CB(1:3)
INTEGER*4	DR_BB(1:3)
INTEGER*4	DR_GPR(0:15)
EQUIVALENCE	(DR_CR,EMBSL_DV_REGSAV(0))
EQUIVALENCE	(DR_UTL,EMBSL_DV_REGSAV(1))
EQUIVALENCE	(DR_WCS,EMBSL_DV_REGSAV(2))
EQUIVALENCE	(DR_BA,EMBSL_DV_REGSAV(5))
EQUIVALENCE	(DR_BC,EMBSL_DV_REGSAV(6))
EQUIVALENCE	(DR_CA,EMBSL_DV_REGSAV(9))
EQUIVALENCE	(DR_WD,EMBSL_DV_REGSAV(11))
EQUIVALENCE	(DR_SL,EMBSL_DV_REGSAV(16))
EQUIVALENCE	(DR_SBR,EMBSL_DV_REGSAV(17))
EQUIVALENCE	(DR_GBR,EMBSL_DV_REGSAV(18))
EQUIVALENCE	(DR_CB,EMBSL_DV_REGSAV(19))
EQUIVALENCE	(DR_BB,EMBSL_DV_REGSAV(22))
EQUIVALENCE	(DR_GPR,EMBSL_DV_REGSAV(25))
PARAMETER	TIMEOUT = 96
PARAMETER	DR_780 = 48
PARAMETER	SBI_FUNC1 = 1
PARAMETER	SBI_FUNC2 = 2
PARAMETER	SBI_FUNC7 = 7
PARAMETER	SBI_FUNC11 = 11
CHARACTER*10	V1DR_UTL(11:11)
DATA	V1DR_UTL(11) = /*WCS VALID*/
CHARACTER*31	V2DR_UTL(25:31)
DATA	V2DR_UTL(25) = /*FORCE DDI CONTROL PARITY ERROR*/
DATA	V2DR_UTL(26) = /*FORCE DDI DATA PARITY ERROR*/
DATA	V2DR_UTL(27) = /*ENABLE DDI DATA PE ABORT*/



```
0380 C
0381 C DR780 UTILITY REGISTER
0382 C
0383
0384 CALL LINCHK (LUN,1)
0385
0386 WRITE(LUN,25) DR UTL
0387 25 FORMAT(' ',T8,'DR UTL',T24,Z8.8)
0388
0389 FIELD = LIB$EXTZV(0,8,DR_UTL)
0390
0391 IF (FIELD .LE. 'FB'X) THEN
0392
0393 CALL LINCHK (LUN,2)
0394
0395
0396 c 2's complement value and get data rate in Mbytes.
0397 c The value 5 (maximum) = 8 Mbytes.
0398 c
0399
0400 field = jidim(256,field)
0401
0402
0403 c Use Table 3-9 dr780 users guide formulae to calculate data rate
0404 c
0405
0406 float = floatj(field)
0407
0408 float = 40/float
0409
0410
0411 c Calculate clock period. Since 32 bits are transfered at a time
0412 c then the number of micro-seconds in one second are divided by
0413 c the number of bytes transfered in one second to obtain the clock
0414 c period.
0415 c
0416
0417 float = (1024*1024) / ((float*1024*1024)/4)
0418
0419 WRITE(LUN,26) FLOAT
0420 26 FORMAT(' ',T40,'PERIOD = ',F<COMPRESSF (FLOAT,1)>.1,'. MICRO/SEC')
0421
0422 c
0423 c Calculate the Kbyte data rate.
0424 c
0425
0426 float = floatj(field)
0427
0428 float = (40/float)*1024
0429
0430 WRITE(LUN,27) FLOAT
0431 27 FORMAT(' ',T40,'TRANSFER RATE = '
0432 1 F<COMPRESSF (FLOAT,2)>.2,'. KB/SEC')
0433 ENDIF
0434
0435 CALL OUTPUT (LUN,DR_UTL,V1DR_UTL,11,11,11,'0')
0436
```

```
0437 CALL OUTPUT (LUN,DR_UTL,V2DR_UTL,25,25,31,'0')
0438
0439 C
0440 C DR780 WCS REGISTERS
0441 C
0442
0443 DO 40 I = 1,3
0444
0445 FIELD = LIB$EXTZV(0,31,DR_WCS(I))
0446
0447 IF (FIELD .NE. 0) THEN
0448
0449 CALL LINCHK (LUN,1)
0450
0451 WRITE(LUN,30) I,DR_WCS(I)
0452 30 FORMAT(' ',T8,'DR WCS',I1,T24,Z8.8)
0453 ENDIF
0454
0455 40 CONTINUE
0456
0457 C
0458 C DR780 SBI ADDRESS REGISTER
0459 C
0460
0461 CALL LINCHK (LUN,1)
0462
0463 WRITE(LUN,50) DR_BA
0464 50 FORMAT(' ',T8,'DR BA',T24,Z8.8)
0465
0466 FIELD = LIB$EXTZV(28,2,DR_BA)
0467
0468 CALL LINCHK (LUN,1)
0469
0470 IF (FIELD .EQ. 0) THEN
0471
0472 WRITE(LUN,65) SILO_DIR
0473 65 FORMAT(' ',T40,A15,'= SBI READ')
0474
0475 ELSE IF (FIELD .EQ. 3) THEN
0476
0477 WRITE(LUN,70) SILO_DIR
0478 70 FORMAT(' ',T40,A15,'= SBI WRITE')
0479 ENDIF
0480
0481 C
0482 C DR780 BYTE COUNT REGISTERS
0483 C
0484
0485 DO 100 I = 1,2
0486
0487 CALL LINCHK (LUN,2)
0488
0489 WRITE(LUN,80) I,DR_BC(I),BYTE_COUNT(I)
0490 80 FORMAT(' ',T8,'DR BC',I1,T24,Z8.8,/,
0491 1 T40,A3,' BYTE COUNT')
0492
0493 100 CONTINUE
```





```

0551
0552      CALL LINCHK (LUN,1)
0553
0554      WRITE(LUN,180) DR_SL
0555 180   FORMAT(' ',T8,'DR-SL',T24,Z8.8)
0556
0557      CALL OUTPUT (LUN,DR_SL,V1DR_SL,0,0,14,'0')
0558
0559      FIELD = LIB$EXTZV(16,8,DR_SL)
0560
0561      IF (FIELD .NE. 0) THEN
0562
0563      CALL LINCHK (LUN,3)
0564
0565      WRITE(LUN,182) ('*',I = 0,32)
0566 182   FORMAT(' ',T40,33A1,/,T47,'FAR-END STATUS BYTE',/)
0567
0568      DO 186,I = 0,4
0569
0570      FIELD1 = LIB$EXTZV(I,1,FIELD)
0571
0572      IF (FIELD1 .NE. 0) THEN
0573
0574      CALL LINCHK (LUN,1)
0575
0576      WRITE(LUN,184) I
0577 184   FORMAT(' ',T40,'FAR-END DEVICE STATUS BIT ',I1,'. SET')
0578      ENDIF
0579
0580 186   CONTINUE
0581
0582      CALL OUTPUT (LUN,DR_SL,V2DR_SL,21,21,23,'0')
0583
0584      CALL LINCHK (LUN,1)
0585
0586      WRITE(LUN,188) ('*',I = 0,32)
0587 188   FORMAT(' ',T40,33A1)
0588      ENDIF
0589
0590      C
0591      C      DR780 SYSTEM BASE REGISTER
0592      C
0593
0594      CALL LINCHK (LUN,2)
0595
0596      WRITE(LUN,190) DR_SBR
0597 190   FORMAT(' ',T8,'DR-SBR',T24,Z8.8,/,
0598          1 T40,'DR SYSTEM BASE REG (PHY ADDR)')
0599
0600      C
0601      C      DR780 GLOBAL BASE REGISTER
0602      C
0603
0604      CALL LINCHK (LUN,2)
0605
0606      WRITE(LUN,200) DR_GBR
0607 200   FORMAT(' ',T8,'DR-GBR',T24,Z8.8,/.

```



```
0665
0666      CALL LINCHK (LUN,2)
0667
0668      WRITE(LUN,290) DR_GPR(9)
0669 290   FORMAT(' ',T8,'DR-R9',T24,Z8.8,/,
0670        1 T40,'PAGE COUNT')
0671
0672      CALL LINCHK (LUN,2)
0673
0674      WRITE(LUN,300) DR_GPR(10)
0675 300   FORMAT(' ',T8,'DR-R11',T24,Z8.8,/,
0676        1 T40,'DDI CONTROL FLAG REGISTER')
0677
0678      CALL LINCHK (LUN,2)
0679
0680      WRITE(LUN,310) DR_GPR(11)
0681 310   FORMAT(' ',T8,'DR-R12',T24,Z8.8,/,
0682        1 T40,'COPY OF STATE REGISTER')
0683
0684      CALL LINCHK (LUN,2)
0685
0686      WRITE(LUN,320) DR_GPR(12)
0687 320   FORMAT(' ',T8,'DR-R13',T24,Z8.8,/,
0688        1 T40,'COPY OF FLAG REGISTER 0')
0689
0690      CALL LINCHK (LUN,2)
0691
0692      WRITE(LUN,330) DR_GPR(13)
0693 330   FORMAT(' ',T8,'DR-R14',T24,Z8.8,/,
0694        1 T40,'COPY OF LS.SAV')
0695
0696      CALL LINCHK (LUN,2)
0697
0698      WRITE(LUN,340) DR_GPR(14)
0699 340   FORMAT(' ',T8,'DR-R15',T24,Z8.8,/,
0700        1 T40,'MICRO STATUS REGISTER')
0701
0702      call linchk (lun,1)
0703
0704      write(lun,345)
0705 345   format(' ',.)
0706
0707      call orb$l_owner (lun,emb$l_dv_ownuic)
0708
0709      call ucb$l_char (lun,emb$l_dv_char)
0710
0711      call ucb$w_sts (lun,emb$w_dv_sts)
0712
0713      call ucb$l_opcnt (lun,emb$l_dv_opcnt)
0714
0715      call ucb$w_errcnt (lun,emb$w_dv_errcnt)
0716
0717      if (emb$w_hd_entry .ne. 98) then
0718
0719      call linchk (lun,1)
0720
0721      write(lun,345)
```

```

0722
0723      call xfdriver_qio (lun,emb$w_dv_func)
0724
0725      call irp$w_bcmt (lun,emb$w_dv_bcmt)
0726
0727      call irp$w_boff (lun,emb$w_dv_boff)
0728
0729      call irp$l_pid (lun,emb$l_dv_rapid)
0730
0731      call irp$q_iosb (lun,emb$l_dv_iosb1)
0732      endif
0733
0734      RETURN
0735      END
    
```

PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	2486	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	1061	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	956	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 DR32	600	PIC OVR REL GBL SHR NOEXE RD WRT LONG
4 EMB	512	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated		5615

ENTRY POINTS

Address	Type	Name
0-00000000		DR780

VARIABLES

Address	Type	Name	Address	Type	Name
2-00000110	I*4	COMPRESS4	4-00000066	I*4	DR_BA
4-00000052	I*4	DR_CR	4-0000009A	I*4	DR_GBR
4-00000096	I*4	DR_SBR	4-00000092	I*4	DR_SL
4-00000056	I*4	DR_UTL	4-0000001C	L*1	EMB\$B_DV_CLASS
4-00000010	L*1	EMB\$B_DV_ERTCNT	4-00000011	L*1	EMB\$B_DV_ERTMAX
4-0000003E	L*1	EMB\$B_DV_NAMLANG	4-0000003A	L*1	EMB\$B_DV_SLAVE
4-0000001D	L*1	EMB\$B_DV_TYPE	4-00000036	I*4	EMB\$l_DV_CHAR
4-00000012	I*4	EMB\$l_DV_IOSB1	4-00000016	I*4	EMB\$l_DV_IOSB2
4-00000026	I*4	EMB\$l_DV_MEDIA	4-0000004E	I*4	EMB\$l_DV_NUMREG
4-0000002E	I*4	EMB\$l_DV_OPCNT	4-00000032	I*4	EMB\$l_DV_OWNUIC
4-0000001E	I*4	EMB\$l_DV_RQPID	4-00000000	I*4	EMB\$l_HD_SID
4-0000003F	CHAR	EMB\$T_DV_NAME	4-00000024	I*2	EMB\$W_DV_BCNT
4-00000022	I*2	EMB\$W_DV_BOFF	4-0000002C	I*2	EMB\$W_DV_ERRCNT
4-0000003C	I*2	EMB\$W_DV_FUNC	4-0000001A	I*2	EMB\$W_DV_STS
4-0000002A	I*2	EMB\$W_DV_UNIT	4-00000004	I*2	EMB\$W_HD_ENTRY



```
0001  
0002  
0003  
0004 Subroutine XFDRIVER_QIO (lun,emb$w_dv_func)  
0005  
0006  
0007 include 'src$:qiocommon.for /nolist'  
0271  
0272  
0273 byte lun  
0274  
0275 integer*2 emb$w_dv_func  
0276  
0277 integer*4 qiocode(0:1,0:63)  
0278  
0279  
0280 if (qiocode(0,0) .eq. 0) then  
0281  
0282 qiocode(1,01) = %loc(io$_loadmcode)  
0283  
0284 qiocode(1,06) = %loc(io$_startdatap)  
0285  
0286 qiocode(1,56) = %loc(io$_startdata)  
0287  
0288 do 10,i = 0,63  
0289  
0290 qiocode(0,i) = 33  
0291  
0292 if (qiocode(1,i) .eq. 0) then  
0293  
0294 qiocode(1,i) = %loc(qio_string)  
0295 endif  
0296  
0297 10 continue  
0298 endif  
0299  
0300 call irp$w_func (lun,emb$w_dv_func,  
0301 1 qiocode(0,lib$extzv(0,6,emb$w_dv_func)))  
0302  
0303 return  
0304  
0305 end
```

0001  
0002  
0003  
0004  
0005  
0006  
0007  
0008  
0009  
0010  
0011  
0012  
0013  
0014  
0015  
0016  
0017  
0018  
0019  
0020  
0021  
0022  
0023  
0024  
0025  
0026  
0027  
0028  
0029  
0030  
0031  
0032  
0033  
0034  
0035  
0036  
0037  
0038  
0039  
0040  
0041  
0042  
0043  
0044  
0045  
0046  
0047  
0048  
0049  
0050  
0051  
0052  
0053  
0054  
0055  
0056  
0057  
0058  
0059  
0060  
0061  
0062  
0063  
0064  
0065  
0066  
0067  
0068  
0069  
0070  
0071  
0072  
0073  
0074  
0075  
0076  
0077  
0078  
0079  
0080  
0081  
0082  
0083  
0084  
0085  
0086  
0087  
0088  
0089  
0090  
0091  
0092  
0093  
0094  
0095  
0096  
0097  
0098  
0099  
0100  
0101  
0102  
0103  
0104  
0105  
0106  
0107  
0108  
0109  
0110  
0111  
0112  
0113  
0114  
0115  
0116  
0117  
0118  
0119  
0120  
0121  
0122  
0123  
0124  
0125  
0126  
0127  
0128  
0129  
0130  
0131  
0132  
0133  
0134  
0135  
0136  
0137  
0138  
0139  
0140  
0141  
0142  
0143  
0144  
0145  
0146  
0147  
0148  
0149  
0150  
0151  
0152  
0153  
0154  
0155  
0156  
0157  
0158  
0159  
0160  
0161  
0162  
0163  
0164  
0165  
0166  
0167  
0168  
0169  
0170  
0171  
0172  
0173  
0174  
0175  
0176  
0177  
0178  
0179  
0180  
0181  
0182  
0183  
0184  
0185  
0186  
0187  
0188  
0189  
0190  
0191  
0192  
0193  
0194  
0195  
0196  
0197  
0198  
0199  
0200  
0201  
0202  
0203  
0204  
0205  
0206  
0207  
0208  
0209  
0210  
0211  
0212  
0213  
0214  
0215  
0216  
0217  
0218  
0219  
0220  
0221  
0222  
0223  
0224  
0225  
0226  
0227  
0228  
0229  
0230  
0231  
0232  
0233  
0234  
0235  
0236  
0237  
0238  
0239  
0240  
0241  
0242  
0243  
0244  
0245  
0246  
0247  
0248  
0249  
0250  
0251  
0252  
0253  
0254  
0255  
0256  
0257  
0258  
0259  
0260  
0261  
0262  
0263  
0264  
0265  
0266  
0267  
0268  
0269  
0270  
0271  
0272  
0273  
0274  
0275  
0276  
0277  
0278  
0279  
0280  
0281  
0282  
0283  
0284  
0285  
0286  
0287  
0288  
0289  
0290  
0291  
0292  
0293  
0294  
0295  
0296  
0297  
0298  
0299  
0300  
0301  
0302  
0303  
0304  
0305  
0306  
0307  
0308  
0309  
0310  
0311  
0312  
0313  
0314  
0315  
0316  
0317  
0318  
0319  
0320  
0321  
0322  
0323  
0324  
0325  
0326  
0327  
0328  
0329  
0330  
0331  
0332  
0333  
0334  
0335  
0336  
0337  
0338  
0339  
0340  
0341  
0342  
0343  
0344  
0345  
0346  
0347  
0348  
0349  
0350  
0351  
0352  
0353  
0354  
0355  
0356  
0357  
0358  
0359  
0360  
0361  
0362  
0363  
0364  
0365  
0366  
0367  
0368  
0369  
0370  
0371  
0372  
0373  
0374  
0375  
0376  
0377  
0378  
0379  
0380  
0381  
0382  
0383  
0384  
0385  
0386  
0387  
0388  
0389  
0390  
0391  
0392  
0393  
0394  
0395  
0396  
0397  
0398  
0399  
0400  
0401  
0402  
0403  
0404  
0405  
0406  
0407  
0408  
0409  
0410  
0411  
0412  
0413  
0414  
0415  
0416  
0417  
0418  
0419  
0420  
0421  
0422  
0423  
0424  
0425  
0426  
0427  
0428  
0429  
0430  
0431  
0432  
0433  
0434  
0435  
0436  
0437  
0438  
0439  
0440  
0441  
0442  
0443  
0444  
0445  
0446  
0447  
0448  
0449  
0450  
0451  
0452  
0453  
0454  
0455  
0456  
0457  
0458  
0459  
0460  
0461  
0462  
0463  
0464  
0465  
0466  
0467  
0468  
0469  
0470  
0471  
0472  
0473  
0474  
0475  
0476  
0477  
0478  
0479  
0480  
0481  
0482  
0483  
0484  
0485  
0486  
0487  
0488  
0489  
0490  
0491  
0492  
0493  
0494  
0495  
0496  
0497  
0498  
0499  
0500  
0501  
0502  
0503  
0504  
0505  
0506  
0507  
0508  
0509  
0510  
0511  
0512  
0513  
0514  
0515  
0516  
0517  
0518  
0519  
0520  
0521  
0522  
0523  
0524  
0525  
0526  
0527  
0528  
0529  
0530  
0531  
0532  
0533  
0534  
0535  
0536  
0537  
0538  
0539  
0540  
0541  
0542  
0543  
0544  
0545  
0546  
0547  
0548  
0549  
0550  
0551  
0552  
0553  
0554  
0555  
0556  
0557  
0558  
0559  
0560  
0561  
0562  
0563  
0564  
0565  
0566  
0567  
0568  
0569  
0570  
0571  
0572  
0573  
0574  
0575  
0576  
0577  
0578  
0579  
0580  
0581  
0582  
0583  
0584  
0585  
0586  
0587  
0588  
0589  
0590  
0591  
0592  
0593  
0594  
0595  
0596  
0597  
0598  
0599  
0600  
0601  
0602  
0603  
0604  
0605  
0606  
0607  
0608  
0609  
0610  
0611  
0612  
0613  
0614  
0615  
0616  
0617  
0618  
0619  
0620  
0621  
0622  
0623  
0624  
0625  
0626  
0627  
0628  
0629  
0630  
0631  
0632  
0633  
0634  
0635  
0636  
0637  
0638  
0639  
0640  
0641  
0642  
0643  
0644  
0645  
0646  
0647  
0648  
0649  
0650  
0651  
0652  
0653  
0654  
0655  
0656  
0657  
0658  
0659  
0660  
0661  
0662  
0663  
0664  
0665  
0666  
0667  
0668  
0669  
0670  
0671  
0672  
0673  
0674  
0675  
0676  
0677  
0678  
0679  
0680  
0681  
0682  
0683  
0684  
0685  
0686  
0687  
0688  
0689  
0690  
0691  
0692  
0693  
0694  
0695  
0696  
0697  
0698  
0699  
0700  
0701  
0702  
0703  
0704  
0705  
0706  
0707  
0708  
0709  
0710  
0711  
0712  
0713  
0714  
0715  
0716  
0717  
0718  
0719  
0720  
0721  
0722  
0723  
0724  
0725  
0726  
0727  
0728  
0729  
0730  
0731  
0732  
0733  
0734  
0735  
0736  
0737  
0738  
0739  
0740  
0741  
0742  
0743  
0744  
0745  
0746  
0747  
0748  
0749  
0750  
0751  
0752  
0753  
0754  
0755  
0756  
0757  
0758  
0759  
0760  
0761  
0762  
0763  
0764  
0765  
0766  
0767  
0768  
0769  
0770  
0771  
0772  
0773  
0774  
0775  
0776  
0777  
0778  
0779  
0780  
0781  
0782  
0783  
0784  
0785  
0786  
0787  
0788  
0789  
0790  
0791  
0792  
0793  
0794  
0795  
0796  
0797  
0798  
0799  
0800  
0801  
0802  
0803  
0804  
0805  
0806  
0807  
0808  
0809  
0810  
0811  
0812  
0813  
0814  
0815  
0816  
0817  
0818  
0819  
0820  
0821  
0822  
0823  
0824  
0825  
0826  
0827  
0828  
0829  
0830  
0831  
0832  
0833  
0834  
0835  
0836  
0837  
0838  
0839  
0840  
0841  
0842  
0843  
0844  
0845  
0846  
0847  
0848  
0849  
0850  
0851  
0852  
0853  
0854  
0855  
0856  
0857  
0858  
0859  
0860  
0861  
0862  
0863  
0864  
0865  
0866  
0867  
0868  
0869  
0870  
0871  
0872  
0873  
0874  
0875  
0876  
0877  
0878  
0879  
0880  
0881  
0882  
0883  
0884  
0885  
0886  
0887  
0888  
0889  
0890  
0891  
0892  
0893  
0894  
0895  
0896  
0897  
0898  
0899  
0900  
0901  
0902  
0903  
0904  
0905  
0906  
0907  
0908  
0909  
0910  
0911  
0912  
0913  
0914  
0915  
0916  
0917  
0918  
0919  
0920  
0921  
0922  
0923  
0924  
0925  
0926  
0927  
0928  
0929  
0930  
0931  
0932  
0933  
0934  
0935  
0936  
0937  
0938  
0939  
0940  
0941  
0942  
0943  
0944  
0945  
0946  
0947  
0948  
0949  
0950  
0951  
0952  
0953  
0954  
0955  
0956  
0957  
0958  
0959  
0960  
0961  
0962  
0963  
0964  
0965  
0966  
0967  
0968  
0969  
0970  
0971  
0972  
0973  
0974  
0975  
0976  
0977  
0978  
0979  
0980  
0981  
0982  
0983  
0984  
0985  
0986  
0987  
0988  
0989  
0990  
0991  
0992  
0993  
0994  
0995  
0996  
0997  
0998  
0999  
1000

## PROGRAM SECTIONS

Name	Bytes	Attributes
0 \$CODE	119	PIC CON REL LCL SHR EXE RD NOWRT LONG
1 \$PDATA	8	PIC CON REL LCL SHR NOEXE RD NOWRT LONG
2 \$LOCAL	548	PIC CON REL LCL NOSHR NOEXE RD WRT LONG
3 \$QIOCOMMON	1247	PIC OVR REL GBL SHR NOEXE RD WRT LONG
Total Space Allocated	1922	

## ENTRY POINTS

Address	Type	Name
0-00000000		XFDRIVER_QIO

## VARIABLES

Address	Type	Name	Address	Type	Name
AP-00000008a	1*2	EMBSW DV FUNC	2-00000200	1*4	I
3-00000442	CHAR	IOS_ABORT	3-00000340	CHAR	IOS_ACCESS
3-000003C2	CHAR	IOS_ACPCONTROL	3-00000483	CHAR	IOS_AVAILABLE
3-00000297	CHAR	IOS_CLEAN	3-00000369	CHAR	IOS_CREATE
3-00000385	CHAR	IOS_DEACCESS	3-00000393	CHAR	IOS_DELETE
3-0000026D	CHAR	IOS_DIAGNOSE	3-00000065	CHAR	IOS_DRVCLR
3-000004CB	CHAR	IOS_DSE	3-000000A9	CHAR	IOS_ERASETAPE
3-00000276	CHAR	IOS_FORMAT	3-00000071	CHAR	IOS_INITIALIZE
3-00000014	CHAR	IOS_LOADMCODE	3-000003A1	CHAR	IOS_MODIFY
3-000003E2	CHAR	IOS_MOUNT	3-00000000	CHAR	IOS_NOP
3-0000009D	CHAR	IOS_OFFSET	3-000000EB	CHAR	IOS_PACKACK
3-000000E0	CHAR	IOS_QSTOP	3-000003EF	CHAR	IOS_RDSTATS
3-00000421	CHAR	IOS_READCSR	3-00000169	CHAR	IOS_READHEAD
3-00000286	CHAR	IOS_READLBLK	3-0000013F	CHAR	IOS_READPBLK
3-00000200	CHAR	IOS_READPRESET	3-00000195	CHAR	IOS_READTRACKD
3-0000033A	CHAR	IOS_READVBLK	3-0000045A	CHAR	IOS_READWTHBUF
3-00000484	CHAR	IOS_READWTHXBUF	3-0000004D	CHAR	IOS_RECAL
3-0000007C	CHAR	IOS_RELEASE	3-000001AB	CHAR	IOS_REREADN
3-000001B8	CHAR	IOS_REREADP	3-000000CA	CHAR	IOS_RETCENTER
3-000002E6	CHAR	IOS_REWIND	3-000002C9	CHAR	IOS_REWINDOFF
3-000000FC	CHAR	IOS_SEARCH	3-00000024	CHAR	IOS_SEEK
3-00000231	CHAR	IOS_SENSECHAR	3-00000309	CHAR	IOS_SENSEMODE
3-0000021D	CHAR	IOS_SETCHAR	3-000003B8	CHAR	IOS_SETCLOCK
3-00000088	CHAR	IOS_SETCLOCKP	3-000002DD	CHAR	IOS_SETMODE
3-000002ED	CHAR	IOS_SKIPFILE	3-000002FA	CHAR	IOS_SKIPRECORD
3-00000029	CHAR	IOS_SPACEFILE	3-0000010E	CHAR	IOS_SPACERECORD
3-000003D7	CHAR	IOS_STARTDATA	3-000000B4	CHAR	IOS_STARTDATAP
3-00000037	CHAR	IOS_STARTMPROC	3-0000020F	CHAR	IOS_STARTSPNDL
3-00000059	CHAR	IOS_STOP	3-0000000D	CHAR	IOS_UNLOAD
3-0000046B	CHAR	IOS_WRITEBUFNCRC	3-0000011E	CHAR	IOS_WRITECHECK
3-000001E4	CHAR	IOS_WRITECHECKH	3-000003FF	CHAR	IOS_WRITECSR
3-00000153	CHAR	IOS_WRITEHEAD	3-000002A2	CHAR	IOS_WRITELBLK
3-00000247	CHAR	IOS_WRITEMARK	3-00000314	CHAR	IOS_WRITEOF
3-0000012A	CHAR	IOS_WRITEPBLK	3-000001C9	CHAR	IOS_WRITERET





A large grid of 150 small, low-resolution terminal screens, each displaying a different system utility or program. The screens are arranged in approximately 10 rows and 15 columns. Many screens feature a title at the top, such as 'CLASSIFY LIS', 'DR250 LIS', 'COMPRESS LIS', 'CRYPTK LIS', and 'DUP327 LIS'. The content on the screens varies, including lists of files, data tables, and graphical elements like bar charts or histograms. The overall appearance is that of a dense collection of diagnostic or administrative tools from the VAX/VMS era.