

DSC IDENT H19 3/7/68  
\*\* CHANGED 11/25/68 DD. LOC DTE+1  
\* ENTRY POINTS

ENTRY DRMTRY,DTXS1,IDCL  
ENTRY AWD,ARD,INTNOP,DTEB  
ENTRY DSCFLG,DSCINT,DSCMSK,DTE  
ENTRY NDCL,DTH,DTS,DTC,DTW,DTP  
ENTRY SSMF,SWSF,DRMSI,DRMSO  
ENTRY DRQ,DRQU,EDCL,DTXS2  
ENTRY DFDL,DFER,DFRX,DFCD

IF -1

ENTRY DREW,DWND,DBSR,DFSR  
ENTRY DIEOR,DOEOR,DISR,DDLRL,DOEOF

ENDF

ENTRY DRMOPN,IDM,DCWBIT

\* ENTRIES (FROM MDBG)

ENTRY FBWRD  
ENTRY BXO,BBP,BFP,BIN,BIC,BDN,BDC,BIP,BIA

DCWBIT DATA 40000000B

DRQ BSS NDRQ\*4

DRQU DATA DCBR+DRQ

\$DSCMSK DATA 777774B

\$TABLE RPT WINLEN; DATA -1; ENDR

ETABLE EQU \*

LTABLE EXT \*-TABLE

CDBA ZRO; SUB =FAVDS; ETR DSCMSK  
LRSH 23; DIV =NWINBLKS\*4; CBA  
LRSH 3; COPY BX,B; LRSH 4  
XXB; LSH 1; CXB; LRSH 19  
DIV =24; BRR CDBA

CDDA ZRO; STB CDDA1; MUL =24; LSH 23  
ADD CDDA1; LRSH 4; COPY BX,B  
LRSH 1; XXB; LSH 4; CXB; LSH 3  
ADD =FAVDS; BRR CDDA

CDDA1 ZRO

DTA ZRO; BRM CDBA; STA DTA3

DTA2 AXC; SKE TABLE,2; BRU DTA1  
BRX \*+1; CXA; SKE =LTABLE; BRU \*+2  
CLA; SKE DTA3; BRU DTA2; BRR DTA

DTA1 STX DTA3; CLA; LDB TABLE,2; LDX =23  
NOD 46; EOR X2; LCY 25,2; CXA; CNA  
LDX DTA3; STB TABLE,2  
COPY XA,AB; BRM CDDA  
SKR\* DBAJOB; NOP; LDX BUFF  
MIN DTA; BRR DTA

DTA3 ZRO

DTAR ZRO; LDA INT31; LSH 9; EOR CLINT; MUL REAL; LSH 32  
ETR =(NOT)BIT0; MUL =NAVDB  
LSH 2; ADD =FAVDS

BRM DTA; BRR DTAR; MIN DTAR; BRR DTAR

DTEC ZRO; ETR DSCMSK; SKG =FAVDS-4  
BRR DTEC; MIN DTEC; BRM CDBA; CAX  
LDA TABLE,2; XXB; BRR DTEC

DTEB ZRO; BRM DTEC; BRR DTEB; LCY 0,2

```

    ETR =(NOT)BIT0; RCY 0,2; XXB
    SKE TABLE,2; MIN DTEB; STA TABLE,2; BRR DTEB
DTE   ZRO
    ETR DSCMSK DD 11/25/68 - KILL HIGH BIT IN LAST DB ADDR
**   IF DB ADDRESS OVER TOP OF WINDOW, DON'T UNMAP, ELSE DO
    SKG =TOP; BRU *+2; BRU OVER DC 7/29/68
    BRM DTEC; BRR DTE; LCY 0,2
        SKA =BIT0; MIN DTECTR; MRG =BIT0; RGY 0,2; XXB
        STA TABLE,2
OVER MIN* DBAJOB; LDX BUFF; BRR DTE
DTECTR ZRO 0 COUNT OF ERR: UNMAP AN ALREADY UNMAPPED DB
*
* 'DRMOPN', 'DRMCLS' 11/13/65
*
* THESE BELONG TO 'MONOPN' AND 'MONCLS', AND OPEN AND
* CLOSE DRUM FILES
*
DRMOPN ZRO
    LDA DEV
DOPN2 MIN DSCTOP; SKA OUTBIT; MIN DSCWOP
    ABC; STA BIN,2; LDA =-1
    STA BIC,2
    STA BDC,2
    STA BDN,2
    CXA
    ADD =BX0-1
    SKB OUTBIT; ADD =1
    STA BIP,2
    BRM BSET
    LDA SS01
    ETR DSCMSK
    SKG =0
    BRU DOPN4
    STA BIA,2
*   BRM DTEC;*BRU DOPN12;*LCY 0,2 ASTERISKS BY DC 7/29/68
*   SKA =BIT0;*BRU DOPN1
*   LDX BUFF;*LDA BIA,2; LDB DSCMSK; LRSH 2; LDX =-NFILE+3
DOPN10 SKN EFA,2; SKM EFA,2; BRU DOPN11
    LDA EFD,2; CXB; LDX FILE
    SKA DRMBIT; BRU *+2; BRU DOPN13
    MRG FD,2
    SKA OUTBIT; BRU DOPN12
DOPN13 LDX BUFF; LDA BIA,2; CBX; LDB DSCMSK; LRSH 2
DOPN11 BRX DOPN10
    LDX BUFF; LDA BIA,2; LDX FILE; LRSH 2; ADM FA,2; LSH 2
    ABC; MIN 0
    LDA =BX0; ADD T; LDX =NDXW; BRM DTC
    LDB =IODMS; LDX PACPTR; BRM DPU
    LDX FILE; SKN FD,2; BRU DOPN5
    LDA =-4; BRU DOPN15
DOPN4 LDA DEV; SKA OUTBIT; BRU *+2
    BRU DOPN14; SKN* DBAJOB; BRU DOPN3
DOPN14 CLA
DOPN9 MIN DRMOPN

```

```

BRR      DRMOPN
DOPN12  LDA  =-1; BRU DOPN9
DOPN3   BRM      DTZ
        STB      BFP,2
        STB      BBP,2
        BRM      DTAR
        BRU DOPN14
        MIN 0
        STA      SS01
        STA      BIA,2
        LDB =IODMS; STB DOPN20; LDB PACPTR; STB DOPN21
DOPN6   LDX FILE
        LRSR 2
        EOR FA,2; ETR =177777B; EOR FA,2; STA FA,2
        LSH 2; ETR DSCMSK; ABC
        LDA T; ADD =BX0; LDX =NDXW; BRM DTW
        LDB DOPN20; LDX DOPN21; BRM DPU
        LDX FILE; SKN FD,2; BRU DOPN5; MIN* DBAJOB
        LDA XN4; ADM FD,2
        BRM DTAR; BRU DOPN7; LDX PUPAC; STA PA,2
        LDX BUFF; STA BIA,2; BRM BSET
        LDB =PUG0; STB DOPN20; LDB PUPAC; STB DOPN21; BRU DOPN6
DOPN7   CLA
DOPN15  LDX PUPAC; STA PA,2
DOPN16  LDA BUFF
        BRM BPUT; LDX FILE
        BRM FREL; BRU PUG0
DOPN5   LDX PUPAC; MIN PL,2; LDA FILE
        XMA PA,2; STA PX,2; BRU PUG0
DOPN20  ZRO
DOPN21  ZRO
*
$DSCLS  ZRO; LDX BUFF
        SKN BIC,2; BRU RDSB; BRR DSCLS
RDSB    COPY XA,XB; ADD =BX0; SKE BIP,2; BRU **2; BRU RDSB5
        LDX BIP,2; LDA X1; ADM -1,2; CBX
RDSB5   CXA; ADD =2; STA 0,2; STA 1,2
RDSB1   CLA; XMA* BIP,2; STA* 1,2; MIN 1,2; MIN BIP,2; SKE =0; BRU RDSB1
        BRM BSET; LDB BIA,2; LDA T; ADD =BX0; LDX =NDXW
        BRM DTW; LDB =IODMS; LDX PACPTR; BRM DPU; LDX FILE
        SKN FD,2; BRU RDSB2
        LDX PUPAC; LDA =TRAP; STA PL,2; BRU DOPN16
RDSB2   LDX BUFF
RDSB4   LDA* 0,2; SKE =0; BRU RDSB3
        LDA BUFF; ADD =2; STA 0,2; ADD =NDDW; STA 1,2
        LDA =-1; STA BIC,2; BRU PUG0
RDSB3   BRM DTE; MIN 0,2; BRU RDSB4
LAS     POPD      14600000B,1,1,0,1
LSM     BRU      TRAP
SAS     POPD      14700000B,1,1,0,1
SSM     BRU      TRAP

```

```

*
* 'IDM' 4/16/66
*

```

```

* THIS IS THE DRUM INTERRUPT ROUTINE FOR ALL DRUM I/O
*
NDCL DATA -1
* COUNT OF DRUM COMMANDS IN LIST
EDCL DATA DRQ
* CURRENT END OF LIST
IDCL DATA DRQ
* CURRENT INTERRUPT POINTER
DRMTRY ZRO
* TRY-AGAIN COUNTDOWN
*
DSCFLG DATA -1
DSCINT DATA 0
INTNOP ZRO
BRR INTNOP
IDT DATA INTNOP, IDR, IDBR, IDP, IDF, IDA, IDWD
IDM ZRO
DET; BRU ID3; DRT; BRU *-3
BETW
BRU ID1
ID2 LDX IDCL PREVIOUSLY PATCHED.
LDA 1,2
RSH 9
EOR 2,2
ETR =37B
EOR 2,2
RSH 2
ETR =37B
XXA
SKR RMC,2
NOP
COPY AX, XB
LDA 3,2
ETR =77B
COPY BA, AX
BRM* IDT,2
MIN ACTR
LDA IDCL
ADD =4
SKE =DRQU
BRU *+2
LDA =DRQ
STA IDCL
SKR NDCL
BRU ID5
RDC FOR FAST AND EFFICIENT AND LONG-LIVED DISC ARMS
* (J. MASTBROOK MEMORIAL DISC PATCH)
CLA
STA BLK31
STA DSCINT
LDA =-1
STA DSCFLG
BRR IDM
ID3 MIN DSCERR; SKS 12026B; MIN DSCSER

```

```

LDX IDCL; LDA 0,2; LRSB 13; ETR =37B
CAX; MIN ERDSC,2
ID1  EOM      0
      IF -1
      LDA*    IDCL
      EOR     =200B
      STA     IDR
      ADA
      POT     IDR
      ENDF
      MIN     DRMERR
      SKR     DRMTRY
      BRU     ID4
      MIN     DUERR
      BRU     ID2
ID5   LDA     =NDTRY-1
      STA     DRMTRY
ID4   LDA     =IDM
      STA     BLK31
      LDX     IDCL
      SKS 14000B; BRU *-1
      ADA
      POT     0,2
      LDA 2,2; STA *+2
      EOM* 10000B; EXU 2,2; POT 1,2; SKN 3,2; BRU *+3
      EOM 3666B; BRR IDM
      EOM 2626B
      BRR     IDM
DSCERR ZRO
DSCSER ZRO
ERDSC  BSS 32
* MAKE FILE AVAILABLE
IDRE  ZRO; LDA FD,2; ETR =(NOT)BIT1; STA FD,2; BRR IDRE
IDR   ZRO; BRM IDF; BRM IDRE; BRR IDR
* CLEAR PROTECTED FILE BUSY BIT
IDP   ZRO      0
      BRM     IDF
      LDA     FD,2
      ETR     =77377777B
      STA     FD,2
      BRR     IDP
* MARK FILE NOT BUSY AND SET SECOND POINTER FROM WORD COUNT
IDBR  ZRO      0
      EOR     RRL3
      ETR     =77B
      EOR     RRL3
      STA     IDR
      LRR3
      POT     IDR
      BRM     IDR
      LDX     FC,2
      LDA     NDDW+2,2
ETR =255
      ADM     1,2

```

LRR3

POT RRL3

BRR IDBR

\* CHECK FILE ERROR

IDFI ZRO; LDX IDCL; LDA 3,2; RSH 6; ETR =377B

CAX; BRR IDFI

IDE ZRO; SKN DRMTRY; BRR IDE; LDA FD,2

MRG =BIT0; STA FD,2; BRR IDE

IDF ZRO; BRM IDFI; BRM IDE; BRR IDF

IDWD ZRO; STA IDR; LRR3; POT IDR

BRM IDFI; BRM IDRE; LDA FC,2; ETR ADMSK; COPY AX,XB

SKN DRMTRY; BRU IDWD1; CLA; STA BIP,2

CBX; LDA FA,2; LRSB 17; ETR =77B; CAX; MIN DBA,2

IDWD2 LRR3; POT RRL3; BRR IDWD

IDWD1 LDA =-1; STA BDC,2; MIN BIP,2

EAX 2,2; STX -2,2; BRU IDWD2

\*

\* 'DPU' 3/20/66

\*

\* DRUM I/O EXIT TO PHANTOM USER

\*

DPU

ZRO

STB DPU1; LDA PIM,2; MRG =BIT3; STA PIM,2

LDA FILE

RSH 12

CXA

RSH 12

LDA DPU

ETR ADMSK

DIR

BRM EPU

LDX FILE

LDA =DBB

ADM FD,2

LDX DTXS2

LDA =2

ADM 3,2

BRM DTFF

BRU\* DPU1

DPU1 ZRO

\*

\* 'DRMSI', 'DRMSO' 12/7/65

\*

\* SEQUENTIAL I/O DRIVER, CALLED FROM 'GPW'

\*

DRMSI ZRO

LDX BIP,2

LDA 1,2

SKG =0

BRU DSI1

LDX BUFF

MIN BIP,2

LDA\* BIP,2

ETR DSCMSK

```

CAB
LDA      T
ADD      =2
LDX      =NDDW+1
BRM      DTC
LDX      BUFF
CXA
ADD      =2
STA      0,2
STA      1,2
LDB*     BIP,2
CLA
SKB      X1
MRG      EORBIT
ADM      1,2
LDX      DTXS2
MIN      3,2
BRM      DTFF
BRR      DRMSI
DSI1     LDX      BUFF
          EAX      2,2
          STX      -2,2
          STX      -1,2
          LDA      EOFBIT
          ADM      -1,2
          BRR      DRMSI
*
DRMSO    ZRO; LDA BUFF; ADD =BXO+NDXW-2
          SKG BIP,2; BRU DSO4
DSO2     LDA* BIP,2; ETR DSCMSK; SKC =0; BRU DSO3
          LDX BIP,2; LDA -1,2; BRM DTA; BRU DSO4
DSO3     STA* BIP,2; BRM WDB; LDX DTXS2; LDA =5
          ADM 3,2; BRM DTFF; BRR DRMSO
DSO4     LDA =-1; STA BDC,2
DSO1     LDA 1,2; MRG EOFBIT; STA 1,2
          LDA BUFF; ADD =2; STA 0,2
          LDX FILE; LDA FC,2; MRG X7; EOR X5; STA FC,2; BRU TRAP
WDB      ZRO; CAB; LDA 0,2; SUB BUFF; SUB =2; STA NDDW+2,2
          LDA T; ADD =2; LDX =NDDW+1
          BRM DTW; LDX BUFF; BRR WDB
*
* 'DTP', 'DTH', 'DTF', 'DTC', 'DTW', 'DTX', 'DTS', 'DTM', 'DTD', 'DTT'
* 'DTA', 'DTO', 'DTE', 'DTN', 'DTU', 'DTZ', 'DTR', 'DTL'
* 1/28/66
*
* GENERALLY USEFUL DRUM ROUTINES
*
DTXS1    DATA      -1
* COMMAND COUNT
DTXS2    ZRO
* LAST COMMAND LOC
DTLS1    ZRO
* TEMPORARY BIP
*

```

\* RESET COMMAND COUNT

DTP ZRO  
LDA =-1  
STA DTXS1  
BRR DTP

\* INITIALIZE CORE ADDRESS

DTH ZRO  
RSH 11  
STB T  
MUL =3  
CBX  
LDA RRL2  
LDB RRL1  
LCY 6,2  
ETR =77B  
SKE =0  
BRU \*+2  
BRU TRAP  
LDB T  
LCY 11  
STA T  
BRR DTH

\* ENTER COMMAND IN LIST AND LOCK MEMORY BLOCK

\* A= ABS CORE ADDR

\* B= DRUM ADDR

\* X= WORD COUNT

DTC ZRO  
MIN DSCTC  
STA DTCS1  
SKN DTCS1; BRU DTC1; CXA; ADD =77B  
ETR =-100B; CAX; MIN DSCWC

DTC1 CBA  
ETR =777777B  
SKG =0; BRU TRAP  
MIN DTXS1  
STA\* EDCL  
LDA DTCS1  
ETR ADMSK  
RSH 14  
CXA  
ETR =77777B  
LCY 14  
LDX EDCL  
STX DTXS2  
STA 1,2  
LSH 19  
LDA DTCS1  
ETR =140000B  
CBX  
RSH 14  
CXB  
LSH 5  
MRG =IOSDW  
SKN DTCS1; SUB =200B



```

LDB      X4
SKN      DTCS1
CLB
LDX      EDCL
STA      2,2
STB      3,2
LDA      DTCS1
RSH      11
ETR      =37B
COPY     XB,AX
MIN      RMC,2
CBA
ADD      =4
SKE      =DRQU
BRU      *+2
LDA      =DRQ
STA      EDCL
LDA      =NDRQ-2
SUB      DTXS1
SKG      NDCL
BRU      *-1
BRR      DTC
DTCS1 ZRO 0
DTW
MRG      DCWBIT
BRM      DTC
BRR      DTW
* DRUM START FOR FILE OPERATIONS
DTFF ZRO
LDX      FILE
LDA      X2
MRG      FD,2
STA      FD,2
LDB      FILE
LSH      30
MRG      =1
LDX      DTXS2
ADM      3,2
BRM      DTS
BRR      DTFF
* DRUM START AND EXIT
* DRUM START
DTS ZRO
LDA      DTXS1
ADD      =1
SKG =0; BRR DTS
DIR
ADM      NDCL
SKN      DSCFLG
BRU      DTS1
CLA
SKE      BLK31
BRU      DTS2
EIR

```

```

BRM      DTSS
DTS1     EIR
        BRM      DTP
        BRR      DTS
DTS2     LDA      =DTSS
        STA      DSCINT
        BRU      DTS1
* REALLY START THE DISC
DTSS     ZRO
        MIN      DSCFLG
        LDA      =IDM
        STA      DSCINT
        STA      BLK31
        LDA      =NDTRY-1
        STA      DRMTRY
        LDX      IDCL
        DRT; BRU *-1; SKS 14000B; BRU *-1
        ADA
        POT      0,2
        LDA 2,2; STA *+2
        EOM* 10000B; EXU 2,2; POT 1,2; SKN 3,2; BRU *+3
        EOM 3666B; BRR DTSS
        EOM 2626B; BRR DTSS
* ZERO INDEX BLOCK
DTZ      ZRO
        LDA      BUFF
        ADD      =23600000B+BXO+NDXW-2
        STA      DTZ1
        CLB
        LDX      =-NDXW+2
DTZ1     STB      BXO+NDXW-2,2
        BRX      DTZ1
        LDX      BUFF
        MIN      BIC,2
        BRR      DTZ
DBI      POPD     14200000B,1,1,0,1
DBO      POPD     14300000B,1,1,0,1
DWI      POPD     14400000B,1,1,0,1
DWO      POPD     14500000B,1,1,0,1
        BRU      TRAP
* 'SSMF' 10/31/65
*
* THIS CAUSES A RANDOM FILE TO BE DECLARED AS SECONDARY MEMORY
SSMF     BRU      TRAP
*
* 'SWSF' 10/31/65
*
* THIS CHANGES THE MODE OF A SEQUENTIAL FILE TO INPUT
* OR OUTPUT
*
SWSF     BRU      TRAP
*
* 'DFER', 'DFRX' 11/11/65
*

```

\* THESE DELETE AND READ INDIVIDUAL INDEX BLOCKS

\*  
DFER SKN PQU,2  
BRU TRAP  
BRM DTE  
BRU POPX

\*  
DFRX BRU TRAP

\* 'DFDL' 12/7/65

\* THIS DELETES THE CONTENTS OF A DRUM FILE

\*  
DFDL BRM IOI; SKB DRMBIT; BRU \*\*2; BRU TRAP  
SKB OUTBIT; BRU \*\*2; BRU TRAP  
LDX BUFF; LDA =-1; STA BDC,2; MIN BIC,2  
LDA BUFF; ADD =BX0  
STA BIP,2  
LDA BUFF; ADD =2; STA 0,2; ADD =NDDW  
STA 1,2; BRU POPX

\*  
\* 'DFCD' 3/20/66

\* ADD THE NUMBER OF DATA WORDS IN FILE (A) TO X

\*  
DFCD BRM IOI  
SKB DRMBIT  
BRU \*\*2  
BRU TRAP  
LDX BUFF  
DFCD1 CXA  
ADD =27600000B+BX0+NDXW-2  
STA DFCD2  
LDX =-NDXW+2  
DFCD2 LDA BX0+NDXW-2,2  
SKG =0  
BRU \*\*3  
LDA =255  
ADM SS03  
BRX DFCD2  
BRU POPX

\*  
\* 'CBRF' 12/7/65

\* THIS DELETES A BLOCK OF INFORMATION FROM A RANDOM FILE

\*  
\* 'DISR', 'DDLRL' 12/7/65

\* THESE INSERT AND DELETE LOGICAL RECORDS IN A SEQUENTIAL FILE

\*  
IF -1  
DISR BRU TRAP  
DDLRL BRU TRAP

```

*
* 'DSF', 'DSB', 'DSS', 'DSZ' 12/13/65
**
* SPACE FORWARD OR BACKWARD 1 RECORD
*
*
* 'DIEOR', 'DOEOR' 12/7/65
*
* SPACE TO END OF RECORD (INPUT) OR WRITE END OF RECORD (OUTPUT)
* ON SEQUENTIAL FILE
*
DIEOR BRU TRAP
*
DOEOR BRU TRAP (NOT IMPLEMENTED)
*
* 'DOEOF' 3/21/66
*
* DELETE REMAINDER OF FILE
*
DOEOF BRU TRAP
*
* 'DFSR', 'DBSR' 12/7/65
*
* SPACE FORWARD OR BACKWARD (B) LOGICAL RECORDS
*
DFSR BRU TRAP
DBSR BRU TRAP
DREW BRU TRAP
DWND BRU TRAP
ENDF
* BRS 61 AND 62
* READ AND WRITE ON DISC
* USED TO GET AUD'S AND UFD'S
* (A) = CORE ADDRESS
* (B) = DISC ADDRESS
* (X) = WORD COUNT
AWD LDB =DTW
BRU ARD+1
ARD LDB =DTC
STB DTZ
SKN PQU,2
BRU TRAP
ETR ADMSK
STA T
ETR =3777B
ADD SS03
SKG =4000B
BRU *+2
BRU TRAP
LDA SS03
SKG X0
BRU ARWD1
LDA SS02
SKG X0

```

```

BRU      ARWD1
LDX      T
LDA      0,6
STA      0,6
LDA      T
BRM      DTH
LDB      SS02
SKB      =77000000B
BRU      TRAP
LDX      SS03
BRM*     DTZ
BRM DISA; ADD =5; LDX DTXS2; ADM 3,2; BRM DTS
$ARWD2 LDB PACDMB; LDX =QIO; MIN 0; BRU POPDMS
ARWD1 MIN 0; BRU POPX
IDA      ZRO; LDX IDCL; LDA 3,2; LRSB 6; MRG PLMSK; CAX
        SKN DRMTRY; MIN PL,2
        LDA PIM,2; ETR =(NOT)BIT3; STA PIM,2; BRR IDA
$DISA   ZRO
        LDX PACPTR; LDA PIM,2; MRG =BIT3; STA PIM,2
        COPY XA,B; ETR PRMSK; LSH 6; BRR DISA
        END

```