BOOM MOVING SUBROUTINES (BOOM,11) 10/2/73

```
4RD,        0
            DAP 4RDX
4RD5,       LAC ONTRACK     /ARE WE ON A HALF TRACK
            SPA
            JMP 4RD1        /NO
            XOR 4RD         /YES-ARE WE ON SAME TRACK
            AND (177774
            SZA I
            JMP 4RDX1       /YES-EXIT
            LAC TRACKD      /NO-ARE REWRITE BITS IN CORE
            SMA
            JMP 4RD2        /YES
4RD3,       LAC 4RD         /NO-GO MOVE BOOM
            JDA FGO
4RDX1,      LAC 4RD
            DAC ONTRACK
4RDX,       JMP .

4RD1,       LAC TRACKD      /NOT ON TRACK-ARE REWRITE BITS IN CORE
            SPA
            JMP 4RD3        /NO-JUST MOVE BOOM
            JDA FGO
            JMP 4RD5        /TRY AGAIN

4RD2,       JSP METH1       /GO TO CHANNEL TO HANDLE BITS
            IDX 4RDX
            LAC 4RD
            JDA READ
            JMP 4RDX
```

```
/MOVE BOOM ON CHANNEL 1 ROUTINE
READ,     0                /TABLE POINTER IN AC
          DAC READX
          LAC READ         /DO WE WANT SAME TRACK
          XOR ONTRACK      /IF NOT ON A TRACK WILL NOT COMPARE(CORE 15)
          AND (177774
          SZA I
          JMP READX1       /SAME TRACK
          LAC READ         /SET TO MOVE THE BOOM
          SUB (FRELST      /16 BIT CONSTANT
          SAR 2S
          IOR C400
          DAP BMC0+12
          LAW BMC0 11
          DAP READP
          LAC TRACKD       /ARE BIT TABLES IN CORE
          SPA
          JMP READ2        /NO : JUST MOVE THE BOOM
          LIO ONTRACK      /ARE WE ON TRACK
          SPI
          JMP READ1        /NO
          LAW BMC0+3       /RESET DEVICE START SEQUENCE
READ3,    DAP READP
          LAC TRACKD
          RAR 8S           /WHICH QUARTER
          AND (6000
          DAP BMC0+4
READ2,    JSP .GO          /GO MOVE EVERYTHING
READP,    30000 .
          CLC              /NO BITS IN CORE
          DAC TRACKD
READX1,   LAC READ
          DAC ONTRACK
          JMP I READX
READX,    0

READ1,    SUB (FRELST)
          SAR 2S
          IOR C400
          DAP BMC0+1
          LAW BMC0
          JMP READ3

/DATA CHANNEL PROGRAM FOR READ
BMC0,     560001           /IF NOT ON TRACK
BMC0+1,   300000 .
          140000
BMC0+3,   560001
BMC0+4,   260000 .
          610000 BITTB4
          040100
          630000 CNT4
          100001
BMC0 11,  560001
BMC0 12,  300000 .
          140000
          000000
```

```
/REWRITE FOR CHANNEL 4; ENTER WITH QUARTER TRACK POINTER IN AC
/TRANSPARENT TO AC

4RW,        0
            DAP 4RWX
            LAC 4RW
            JDA REWT        /FIND IF ALREADY IN CORE, SET POINTERS
            JMP 4RW1        /ALREADY IN CORE=JUST MOVE BOOM
            LAC TRACKD      /MOVE BOOM TO EITHER OLD OR NEW BITS
            SPA
            LAC 4RW
            JDA 4RD
            JSP METH1
            LAC 4RW         /GO GET BIT TABLE
            JDA MANIP
            3
            JMP 4RWX+1      /GIVE RETURN 2 TO JDA (CHANNEL 1)

4RW1,       JDA 4RD         /BITS ALREADY IN CORE = MOVE THERE
4RWX,       JMP .
4RWX+1,     IDX 4RWX
            LAC 4RW
            JMP 4RWX
/REWRITE FOR CHANNEL 1; ENTER WITH HALF TRACK POINTER IN AC
/TRANSPARENT TO AC

REWRITE,    0
            DAP RWTX
            LAC REWRITE
            JDA REWT        /FIND IF ALREADY IN CORE
            JMP RWT1        /ALREADY IN CORE
            JDA MANIP
THREE,      3
            LAC REWRITE
RWTX,       JMP .

RWT1,       JDA READ
            JMP RWTX
```

```
/SUBROUTINE FOR REWRITE ROUTINES
REWT,     0
          DAP REWTX
          LAC REWT
          SAD TRACKA      /ARE BITS IN FIRST BIT TABLE
          JMP REWT1       /YES
          SAD TRACKB      /ARE BITS IN SECOND BIT TABLE
          JMP REWT2       /YES
          SAD TRACKC
          JMP REWT3       /BITS ARE IN THIRD BIT TABLE
          SAS TRACKD
          IDX REWTX       /RETURN 2, NOT IN CORE
          LAW BITTB4
          DAP BITPTR
          LAW CNT4
REWT4,    DAP CNTPTR
          LAC REWT
REWTX,    JMP .

REWT1,    LAW BITTB1
          DAP BITPTR
          LAW CNT1
          JMP REWT4

REWT2,    LAW BITTB2
          DAP BITPTR
          LAW CNT2
          JMP REWT4

REWT3,    LAW BITTB3
          DAP BITPTR
          LAW CNT3
          JMP REWT4

CNTPTR,   JDA .
BITPTR,   DCH I .
```

```
/PROGRAM TO MANIPULATE BIT TABLES
/ENTER WITH DRUM POINTERS IN AC
/WORD AFTER IS 0-3 FOR BIT TABLE

MANIP,    0
          DAC MANX
          LAW 4BTTB1      /RESET DATA CHANNEL PROG,
          ADD I MANX
          SAL 6S
          DAP MANC3
          DAP MANC7
          LAW CNT1
          ADD I MANX
          DAP MANC9
          DAP MANC10
          LAW TRACKA      /GET POINTER TO RIGHT TABLE
          ADD I MANX
          DAP MAN1        /LAC WHICH WE NEED
          DAP MAN2        /DAC WHICH WE NEED
          IDX MANX        /RESET RETURN

MAN1,     LAC .           /GET TRACK TO MOVE TO
          SPA             /IS THERE ANYTHING THERE
          JMP MAN3        /NOTHING THERE
          JDA READ        /MOVE BOOM TO THAT POSITION
          SAD MANIP       /IS IT THE ONE WE WANT
          JMP I MANX      /YES
          RAR 8S
          AND (6000
          DAP MANC2
          JSP GO          /WRITE IT OUT
          30000 MANC1
          CLC
MAN2,     DAC .           /RESET POINTER

MAN3,     LAC MANIP       /MOVE TO TRACK
          SAD TRACKD      /CHECK SPECIAL CASE OF IN ANOTHER TABLE
          JMP MAN4
          JDA READ
          RAR 8S
          AND (6000
          DAP MANC6
          JSP GO
          30000 MANC5
          LAC MANIP       /RESET POINTER
          XCT MAN2
          JMP I MANX
MANX,     0
```

```
MAN4,       RAR 8S
            AND (6000
            DAP MANC2
            LAW BITTB4
            DAP MANC3
            LAW CNT4
            DAP MANC9
            JSP GO
            30000 MANC1
            CLC
            DAC TRACKD
            JMP MAN3

/DATA CHANNEL PROGRAM

MANC1,      560001
MANC2,      260000+.
MANC3,      610000 .
            040100
MANC9,      630000 .
            100001
            000000

MANC5,      560001
MANC6,      200000+.
MANC7,      610000 .
            040100
MANC10,     630000 .
            100001
            000000

/COUNT REGISTERS
CNT1,       0
CNT2,       0
CNT3,       0
CNT4,       0
```

```
/BIT TABLE MANIPULATION ROUTINES
/BIT 0=1 UNUSEABLE
/BIT 1=1 HELD
/BIT 2=1 HELD BY SWAPPER
/BIT 3=1 HELD BY INDEX
/BIT 4=1 NO BLOCKS LEFT
/BIT 5=1 NOT CLEAN HALF TRACK

/GET BLOCKS ROUTINE; C(FBLOCK)=-NO. OF BLOCKS WANTED
/C(AC)=SECTOR TO START AT; ASSUME BITPTR,CNTPTR,HTRACK,HTR SET

GBLKS,    0
          DAP GBLKSX
          LAW BUFF        /SET POINTER TO STORAGE AREA
          DAP GBLK1
          LAC BITPTR      /SET STARTING POINTER TO BIT TABLE
          ADD GBLKS
          DAC GBLK3
          LAW 100         /SET TEST FOR END OF RING BUFFER
          ADD BITPTR
          DAC GBLK4
          LAC FBLOCK      /CHECK FOR CASE OF ZERO
          SAS MZERO
          SZA I
          JMP GBLK14
          LAC HTRACK      /SET UP ALL BITS EXCEPT HEAD BITS
          IOR GBLKS
GBLK13,   DAC GBLKS
GBLK6,    LAC I GBLK3     /PICK UP BITS THIS SECTOR
          SAD MZERO       /ARE THERE ANY HEAD POSITIONS LEFT
          JMP GBLK5       /NO-INCREMENT TO NEXT HEAD POSITION
          LIA             /YES
          FBC
          DIO I GBLK3     /RESTORE BIT TABLE
          RAL 6S
          IOR GBLKS       /CODE AS DRUM ADDRESS
GBLK1,    DAC .
          IDX GBLK1
          ISP I CNTPTR    /RESET COUNT OF BLOCKS LEFT
          JMP GBLK9       /STILL BLOCKS LEFT
          IDX FBLOCK      /DID GET BLOCK THIS TIME
          LAC (30000)     /SET QTRK EMPTY, NOT CLEAN
GBLK10,   IOR I HTR                    FULL
          DAC I HTR
GBLK14,   CLC
          XCT GBLK1
GBLKSX,   JMP .

GBLK9,    ISP FBLOCK      /DO WE WANT MORE
          JMP GBLK12      /YES
GBLK7,    LAC B5          /NO-MARK AS NOT CLEAN
          JMP GBLK10
```

```
GBLK12,   LAC GBLK1
          SAD (DAC BUFF 77
          JMP GBLK7       /YES
          IDX GBLK5       /INCREMENT SECTOR BY 2 ENTRY
          IDX GBLK3
GBLK5,    IDX GBLK5       /INCREMENT SECTOR BY 1 ENTRY
          IDX GBLK3
          SUB GBLK4
          SPA
          JMP GBLK6       /NOT AROUND END
          LIA
          ADD BITPTR
          DAC GBLK3
          LAC HTRACK
          AAI
          JMP GBLK13

GBLK3,    0
GBLK4,    0
```

```
/CHANNEL 4 AND CHANNEL 1 CALLING OF GBLKS
HTR,      0              /CHANNEL 1 ENTRY
          DAP HTRX
          LAW GBLKS 1
          DAP CH15A      /SET CHANNEL 15 SERVICE ROUTINE
          DAP 15SET      /MARK ACTIVITY TO DO
4HTR1,    LAC HTR
          SUB THIRD
          RAR 8S
          DAC HTRACK     /SET HALF TRACK BITS
          RRI 1000       /GET PRESENT HEAD SECTOR
          LAW 15
          AAI
          AND (77
          DAC GBLKS      /SET UP SECTOR TO START AT
          LAC HTR
HTRX,     JMP .

4HTR,     0              /CHANNEL 4 ENTRY
          DAP HTRX
          LIO 4HTR
          DIO HTR
          LAW GBLKS 1
          DAP CH15B
          JMP 4HTR1

/RECALL ROUTINE TO GET MORE BLOCKS
/GETS MORE BLOCKS AND MOVES BOOM TO THAT POSITION

GBLOCK,   0              /SECTOR TO START AT
          DAP GBKX
          LAW 2          /YES, DROP TO 15 AND GET MORE BLOCKS
          ADD GBLOCK
          AND (77
          DAC GBLKS
          LAC GBLOCK
          AND (376000)
          DAC HTRACK
          RAL 8S
          ADD THIRD
          DAC HTR
          JDA REWRITE
          LAC I CNTPTR   /ARE THERE MORE BLOCKS LEFT
          SZA I
          JMP GBK1       /NO
          LAW GBLKS 1
          DAP CH15A
          DAP 15SET
          JSP 15DRP
GBKX,     JMP .
```

```
/CHANNEL 4 WRITE
4WT,        JSP WTS          /COMMON 4,1 SUBROUTINE
            DIO 4WT1
            DAP 4WT2

4WT2,       LAC .
            SPA
            JMP 4WT4         /NO BIT TABLE IN CORE
            LAC I CNTPTR
            SZA I
            JMP 4WT5         /NO BLOCKS LEFT
            XCT 4WT2
            XOR ONTRACK
            AND (177774
            SZA I
            JMP 4WT6         /SAME TRACK
            XCT 4WT2
            LIO TRACKD
            SPI I
            JMP 4WT7         /GO TO CHANNEL 1 TO WRITE OUT BIT TABLES
            JDA 4HTR
            JDA 4GO15
            JMP METH0

4WT7,       JSP METH1
            XCT 4WT2
            JDA HTR
            JDA READ
            JMP FSTRLS

4WT6,       XCT 4WT2
            DAC ONTRACK
            JDA 4HTR
            JSP 4DP15
            JMP METH0

4WT4,       LAC THIRD
            JMP 4WT5 1
4WT5,       XCT 4WT2
            JDA FIND
            DAC HTR
            JDA 4RD          /MOVE BOOM TO THIS HALF TRACK
            JSP METH1
            LAC HTR
            JDA MANIP
4WT1,       0
            LAW FSTRLS
            DAP WRITX        /JMP INTO WRITE CODING
            LAC 4WT2
            LIO 4WT1
            JMP 4WRT1
```

```
/CHANNEL 1 WRITE ROUTINE

GBK1,       LIO I ONTRACK  /CONTINUATION OF GBLOCK ROUTINE
            LAC GBKX
            RIL 1S
            SPI
            JMP 1HW
WRITE,      DAP WRITX
            JSP WTS
4WRT1,      DIO WRT1
            DAP WRT2
WRT2,       LAC .
            SPA
            JMP WRT3        /NO BIT TABLE
            LAC I CNTPTR
            SMA
            JMP WRT4        /NO BLOCKS LEFT
            XCT WRT2
            DAC READ
            JDA HTR
            XOR ONTRACK
            AND (177774
            SZA I
            JMP WRT5        /SAME TRACK
            JSP READ+1
WRITX,      JMP .

WRT5,       JSP 15DRP
            JMP WRITX

WRT3,       LAC THIRD
            JMP WRT4+1
WRT4,       XCT WRT2
WRT4+1,     JDA FIND
            JDA MANIP
WRT1,       0
            JMP WRT2

WTS,        DAP WTSX
            LAC THIRD
            SUB (FRELST
            SAR 7S
            LIA
            LAW 4BTTB1
            AAI
            SAL 6S
            DAP BITPTR
            LAW CNT1
            AAI
            DAP CNTPTR
            LAW TRACKA
            AAI
WTSX,       JMP .
```

USEABLE QTR-TRACK FOR FREE INTO

```
/FIND A ~~CLEAN HALF TRACK ROUTINE~~
FIND,     0                    /ONE TO START WITH
          DAP FINDX
          LAW 200
          ADD THIRD
          DAC FIND1
          LAC FIND
FIND4,    DAC FIND2
          LAC CNOP
          AND I FIND2
          SZA I
          JMP FIND3
          IDX FIND2
          SAD FIND1
          LAC THIRD
          SAS FIND
          JMP FIND4
          JMP 1EMPTY

FIND3,    LAC FIND2
FINDX,    JMP .

FIND1,    0
FIND2,    0
```

```
/SEARCH FOR A HELD TRACK ROUTINE

SEARCH,   DAP SRCHX
          LAC B1
          IOR FUSER
          DAC SRCH5
          LAW 200
          ADD THIRD
SRCH2,    SUB ONE
          DAC FIND1
          LAC (760177
          AND I FIND1
          SAD SRCH5
          JMP SRCH1       /FOUND ONE
          LAC FIND1
          SAS THIRD
          JMP SRCH2
          SAD (FRELST+400)            /THIRD 2?
          JMP SRCH6       /YES, GET LOWEST CLEAN Q.T.
          LAW 200
          ADD THIRD
SRCH3,    SUB ONE
          DAC FIND1
          LAC I FIND1
          SZA I
          JMP SRCH4
          LAC FIND1
          SAS THIRD
          JMP SRCH3
1EMPTY,   LAC DCONT
          SPQ             /SEE WHICH CHANNEL
          JSP METH1
          LAC B5
          JDA 1ERR

SRCH4,    LAC SRCH5
          DAC I FIND1
SRCH1,    LAC FIND1
          DAC HTR
SRCHX,    JMP .
SRCH5,    0

SRCH6,    LAC I FIND1
          SZA I
          JMP SRCH4       /FOUND A CLEAN ONE
          IDX FIND1
          SAS (FRELST+400+200)        /THIRD 2 IS FULL
          JMP SRCH6
          JMP 1EMPTY
```

```
/HELD WRITES
4HW,        JSP SEARCH
            JDA 4RW
            JMP 4HW1
            JSP HTR 1
            JSP 15DRP
            JMP FSTRLS
4HW1,       JDA 4HTR
            JSP 4DP15
            JMP METH0
1HW,        DAP 1HWX
            JSP SEARCH
            JDA REWRITE
            JSP HTR 1
            JSP 15DRP
1HWX,       JMP .

START XX-JMP
```