IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-D9EA
PRODUCT NAME:	MAINTENANCE LOADER
DATE CREATED:	1Ø DECEMBER 197Ø
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	RICK FADDEN, ANDY VEROSTIC

1. ABSTRACT

THE PURPOSE OF THIS LOADER IS TO PROVIDE AN ALTERNATE METHOD OF LOADING DIAGNOSTICS WHICH MAY FUNCTION WHEN THE ABSOLUTE LOADER FAILS TO WORK DUE TO A HARDWARE FAILURE. A DESCRIPTION OF THE BOOTSTRAP LOADER LOADING THE MAINTENANCE LOADER IS ALSO PROVIDED TO AID IN ISOLATION OF TROUBLE SHOULD IT BE IMPOSSIBLE TO LOAD EVEN THE MAINTENANCE LOADER. THIS LOADER IS NOT INTENDED TO REPLACE THE ABSOLUTE LOADER AND SHOULD ONLY BE USED FOR LOADING OF DIAGNOSTIC PROGRAMS IF THE ABSOLUTE LOADER WILL NOT FUNCTION.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11/20 STANDARD COMPUTER

2.2 STORAGE

THE PROGRAM USES MEMORY LOCATIONS 17476 THRU 17776.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BOOTSTRAP TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

NONE

4.2 STARTING ADDRESS OR ADDRESSES

17500

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY. (BOTTOM 4K)
SET SWITCH REGISTER TO STARTING ADDRESS (17500)
LOAD ADDRESS.
PRESS START.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCH SETTINGS

NO SWITCHES USED

5.2 SUBROUTINE ABSTRACTS

THE ONLY SUBROUTINE USED IN THE LOADER IS FOR READER OPERATION TO FETCH DATA. IT IS ENTERED WITH A BRANCH INSTRUCTION TO ITS STARTING LOCATION. BEFORE BRANCHING THE PC IS SAVED IN R5. (THIS PC IS POINTING TO THE BRANCH INSTRUCTION). EXIT IS DONE BY AUTO INCREMENTING OF THE SAVED PC (R5) AND MOVING IT BACK TO THE PC (R7). THIS ELIMINATES THE USE OF OPERATIONS USING THE STACK (JSR&RTS).

5.3 PROGRAM AND/OR OPERATOR ACTION

OPERATING INSTRUCTIONS:

1. USING THE BOOT LOADER, LOAD THE MAINT LOADER TAPE.
2. THE DEVICE ADDRESS USED WILL BE CONTAINED IN 17776. IF THIS DEVICE ADDRESS IS ZERO THE MAINT LOADER WILL NOT RUN.
3. PLACE THE MAINT LOADER STARTING ADDRESS (17500) IN THE SWITCH REGISTER AND PRESS LOAD ADDRESS.
4. PRESS START.
5. CHECKSUM IS CONTAINED IN THE LOWER BYTE OF R0. (IN THE DATA LIGHTS AFTER HALT.) THE LOWER BYTE OF THE DISPLAY SHOULD EQUAL ZERO UPON SUCCESSFUL LOADING OF THE TAPE.
6. IF THE LOADER HALTS WHEN DONE IT IS ONLY NECESSARY TO PRESS CONTINUE TO READ IN ADDITIONAL TAPES.
7. IF THE LOADER HALTS AT THE FAIL ADDRESS, A CHECKSUM ERROR HAS OCCURRED. MOVE THE PAPER TAPE BACK TO THE BEGINNING OF THE CURRENT BLOCK AND HIT CONTINUE TO TRY AGAIN.

6. ERRORS

6.1 IF AT THE END OF A BLOCK THE CHECKSUM IS NOT ZERO, A HALT OCCURS AT LOCATION 17612 AND THE LOWER BYTE OF THE DATA LIGHTS (R0) CONTAINS THE FAULTY CHECKSUM.

6.2 ERROR RECOVERY

RECOVERY FROM CHECKSUM ERRORS MAY BE ACCOMPLISHED BY MOVING THE TAPE BACK TO THE BEGINNING OF THE BLOCK AND PRESSING CONTINUE TO ATTEMPT TO READ THE BLOCK CORRECTLY.

7. RESTRICTIONS

7.1 THIS LOADER WILL OPERATE IN THE 1ST 4K ONLY (175ØØ), IF IT IS DESIROUS TO USE THIS IN AN EXPANDED SYSTEM AND PLACE ABOVE 175ØØ, THE FOLLOWING LOCATIONS MUST BE CHANGED ONCE THE MAINTENANCE LOADER IS LOADED INTO THE APPROPRIATE BANK,

```
X75Ø2=X747Ø      X = 1 FOR 4K
X751Ø=X7474          3 FOR 8K
X7542=X7475          5 FOR 12K
X7566=X7475          7 FOR 16K
X7624=X7776         11 FOR 2ØK
X7674=X7474         13 FOR 24K
                    15 FOR 28K
```

7.2 THE MAINTENANCE LOADER HAS NO CAPABILITIES FOR RELOCATING PROGRAMS WHILE LOADING,

8. MISCELLANEOUS

8.1 BOOTSTRAP LOADER SOURCE PROGRAM

THE BOOTSTRAP LOADER SOURCE PROGRAM IS SHOWN BELOW, THE STARTING ADDRESS IN THE EXAMPLE DENOTES THAT THE LOADER IS TO BE LOADED INTO MEMORY BANK ZERO (A 4K SYSTEM),

```
        ØØØØØ1             R1=%1                ;USED FOR THE DEVICE ADDRESS
        ØØØØØ2             R2=%2                ;USED FOR THE LOAD ADDRESS DISPLACEMENT
        Ø174ØØ             LOAD=174ØØ           ;DATA MAY BE LOADED NO LOWER THAN THIS
        Ø17744             ,=17744              ;STARTING ADDRESS OF THE BOOTSTRAP LOADER
Ø17744  Ø167Ø1  START!  MOV      DEVICE,R1      ;PICK UP DEVICE ADDRESS, PLACE IN R1
        ØØØØ26
Ø1775Ø  Ø127Ø2  LOOP:      MOV  #,-LOAD+2,R2    ;PICK UP ADDRESS DISPLACEMENT
        ØØØ352
Ø17754  ØØ5211  ENABLE!    INC  @R1             ;ENABLE THE PAPER TAPE READER
Ø17756  1Ø5711  WAIT:      TSTB @R1
Ø1776Ø  1ØØ376             BPL  WAIT            ;WAIT UNTIL FRAME IS AVAILABLE
Ø17762  116162             MOVB 2(R1),LOAD(R2)  ;STORE FRAME READ FROM TAPE IN MEMORY
        ØØØØØ2
        Ø174ØØ
Ø1777Ø  ØØ5267             INC  LOOP+2          ;INCREMENT LOAD ADDRESS DISPLACEMENT
        177756
Ø17774  ØØØ765  BRNCH!     BR   LOOP            ;GO BACK AND READ MORE DATA
Ø17776  ØØØØØØ  DEVICE!    Ø                    ;ADDRESS OF INPUT DEVICE
```

8.2 MAINTENANCE LOADER TAPE FORMAT

SPECIAL BOOT LEADER CODE COMES FIRST. THIS VALUE LEAVES THE LOADING OFFSET AS IS.

```
351
351
 .
 .
 .
351

 75     ;THIS IS THE ADDRESS DISPLACEMENT WHICH WILL CAUSE THE
        ;NEXT FRAME TO BE LOADED IN LOCATION 17476

MAINT   ;THIS CODE LOADS ADDRESSES 17476 THRU 17700 WITH THE
LOADER  ;MAINTENANCE LOADER CODE

000     ;22 FRAMES OF ZERO FILL IN THE REMAINING SPACE UP
 .      ;TO THE CODE WHICH RESTORES THE BOOT LOADER
 .
 .
000
```

THE FOLLOWING CODE IS LOADED STARTING AT 17724. WHEN EXECUTED IT WILL RESTORE LOCATIONS IN THE BOOT WHICH WERE MODIFIED AND THEN JUMP TO THE HALT AT LOCATION 17476.

```
367
025     ;012767
352
000     ;000352
020
000     ;000020
367
025     ;012767
365
001     ;000765
034
000     ;000034
167
000     ;000167
132
377     ;177532
```

THE FOLLOWING CODE IS LOADED OVER THE FIRST 4 LOCATIONS IN THE BOOT LOADER. THE FIRST 3 ARE LEFT UNCHANGED, AND THE 4TH MODIFIES THE BRANCH OFFSET TO CAUSE EXECUTION TO JUMP TO LOCATION 17724. AS NOTED ABOVE, 17724 IS THE START OF THE SECTION WHICH RESTORES THE BOOT LOADER.

```
301
035     ;016701
026
000     ;000026
302
025     ;012702
373     ;000373
```

8.3 STEP BY STEP DESCRIPTION OF BOOTSTRAP LOADER OPERATION (4K OPERATION)

THE BOOTSTRAP LOADER STARTS BY LOADING THE DEVICE STATUS REGISTER ADDRESS FROM 17776 INTO R1 AND 352 INTO R2. THE NEXT INSTRUCTION INITIATES A READ OPERATION IN THE DEVICE AND THE NEXT TWO INSTRUCTIONS FORM A LOOP TO WAIT FOR THE READ OPERATION TO BE COMPLETED. WHEN DATA IS ENCOUNTERED IT IS TRANSFERRED TO A LOCATION WHICH IS THE SUM OF THE INDEX WORD (174ØØ) AND THE CONTENTS OF R2.

BECAUSE R2 IS INITIALLY 352(8), THE FIRST WORD IS MOVED TO LOCATION 17752, AND IT BECOMES THE IMMEDIATE DATA TO BE PLACED IN R2 DURING THE NEXT EXECUTION OF THE LOOP. THIS IMMEDIATE DATA IS THEN INCREMENTED BY ONE AND THE PROGRAM BRANCHES TO THE BEGINNING OF THE LOOP.

THE LEADER CODE (351), PLUS THE INCREMENT, IS EQUAL IN VALUE (352) TO THE DATA PLACED IN R2 DURING THE INITIALIZATION. THEREFORE, LEADER CODE HAS NO EFFECT ON THE LOADER PROGRAM. EACH TIME LEADER CODE IS READ THE PROCESSOR EXECUTES THE SAME LOOP AND THE PROGRAM REMAINS UNMODIFIED. THE FIRST CODE AFTER THE LEADER CODE (75) REPLACES THE DATA TO BE LOADED INTO R2. THIS IS ADDED TO THE INDEX VALUE (174ØØ) AND THE RESULT POINTS TO THE FIRST LOCATION OF THE PROGRAM TO BE LOADED (17476). THE INC INSTRUCTION WHICH OPERATES ON THE DATA FOR R2 PUTS DATA BYTES IN SEQUENTIAL LOCATIONS, AND REQUIRES THAT THE VALUE OF THE LEADER CODE AND THE OFFSET BE ONE LESS THAN THE VALUE DESIRED IN R2 SINCE THE INCREMENT OCCURS BEFORE THE DATA IS LOADED.

THE BOOT OVERLAY IS ACCOMPLISHED BY LOADING SEQUENTIAL LOCATIONS FROM THE START OF THE PROGRAM UP TO THE BEGINNING OF THE BOOT LOADER (I.E. FROM 17476 TO 17744) WHERE IT BEGINS THE OVERLAY. THE SECTION LOADED JUST BEFORE THE BOOT LOADER IN CORE IS A SUBROUTINE WHICH WILL RESTORE THE LOADER TO ITS INITIAL FORM AND JUMP TO THE BEGINNING OF THE LOADED PROGRAM. THIS PORTION OF THE PROGRAM IS ENTERED BY THE OVERLAY'S CHANGING OF THE BRANCH INSTRUCTION AT THE END OF THE BOOT LOADER (17774) ONCE THE END OF THE TAPE IS REACHED. THIS CAUSES A BRANCH TO THE BEGINNING OF THE RESTORE ROUTINE, RATHER THAN THE BEGINNING OF LOADER. SINCE THE MAINTENANCE LOADER IS SHORTER THAN THE SPACE AVAILABLE, THE SECTION BETWEEN THE END OF THE MAINTENANCE LOADER AND THE BEGINNING OF THE RESTORE ROUTINE IS LOADED WITH ZEROES. THIS LOADING OF EXTRA FRAMES HOLDS TRUE FOR OTHER BOOT TAPES WHICH DO NOT USE ENOUGH LOCATIONS TO ALLOW CONTINUOUS LOADING. THE DATA LOADED IS NOT NECESSARILY ZEROES, HOWEVER.

ONCE THE LAST BYTE OF THE RESTORE ROUTINE IS PLACED IN THE CORE LOCATION IMMEDIATELY PRECEDING THE LOADER, THE BOOT OVERLAY CODE WILL OVERLAY THE FIRST TWO INSTRUCTIONS OF THE

BOOTSTRAP LOADER. THE 1ST INSTRUCTION IS UNCHANGED BY THE OVERLAY, BUT THE SECOND INSTRUCTION IS CHANGED TO 373 TO MAKE THE ADDRESS DISPLACEMENT, ONCE INCREMENTED, POINT TO LOCATION 17774. THE NEXT BYTE READ IS THEREFORE PLACED INTO THE LOWER BYTE OF THE BRANCH INSTRUCTION. BY CHANGING THE OFFSET IN THIS BRANCH INSTRUCTION, THE LOADER WILL BRANCH TO THE START OF THE RESTORE CODE WHICH WILL RESTORE THE BOOTSTRAP LOADER TO ITS ORIGINAL CONDITION BY RESTORING THE CONTENTS OF LOCATIONS 17752 AND 17774 TO 351 AND 765 RESPECTIVELY AND THEN JUMP TO 17476 AND HALT (AT THE BEGINNING OF THE MAINT LOADER).

9.Ø PROGRAM DESCRIPTION

9.1 INPUT FORMAT

```
FRAME   -1      ØØ1
        -2      ØØØ
        -3      BYTE COUNT - LOWER ORDER
        -4      BYTE COUNT - HIGHER ORDER
        -5      LOAD ADDRESS - LOWER ORDER
        -6      LOAD ADDRESS - HIGHER ORDER
                    .    DATA
                    .      PLACED
                    .        HERE
        CKSM - LAST FRAME CONTAINS THE CHECKSUM
```

IF THE BYTE COUNT IS EQUAL TO 6, THE LOAD ADDRESS SPECIFIED WILL BE CONSIDERED TO BE THE DESIRED JUMP ADDRESS. IF THIS ADDRESS IS ODD, THE LOADER WILL HALT.

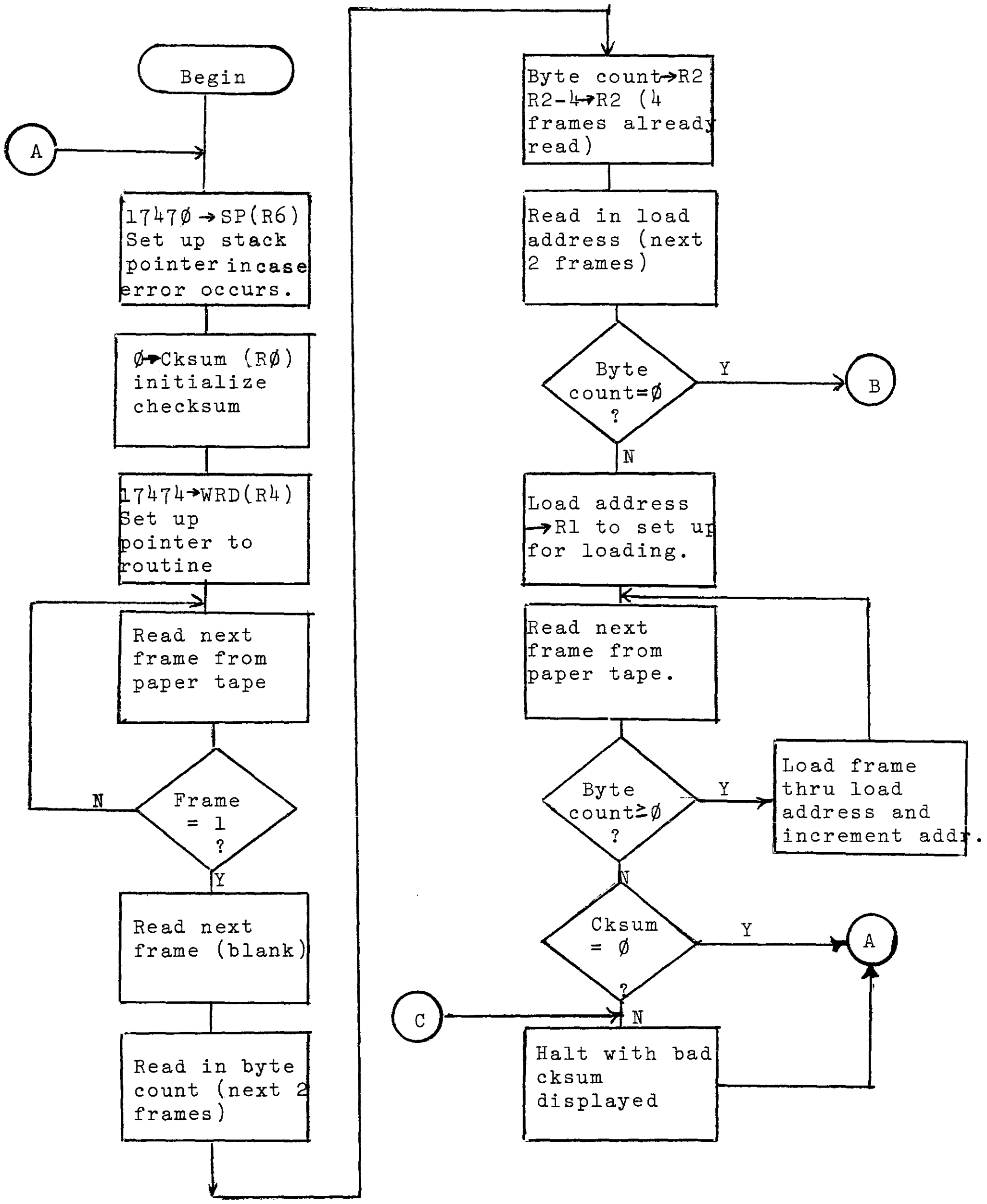
IF THE BYTE COUNT IS > 6, DATA WILL BE LOADED INTO MEMORY.

9.2 REGISTERS

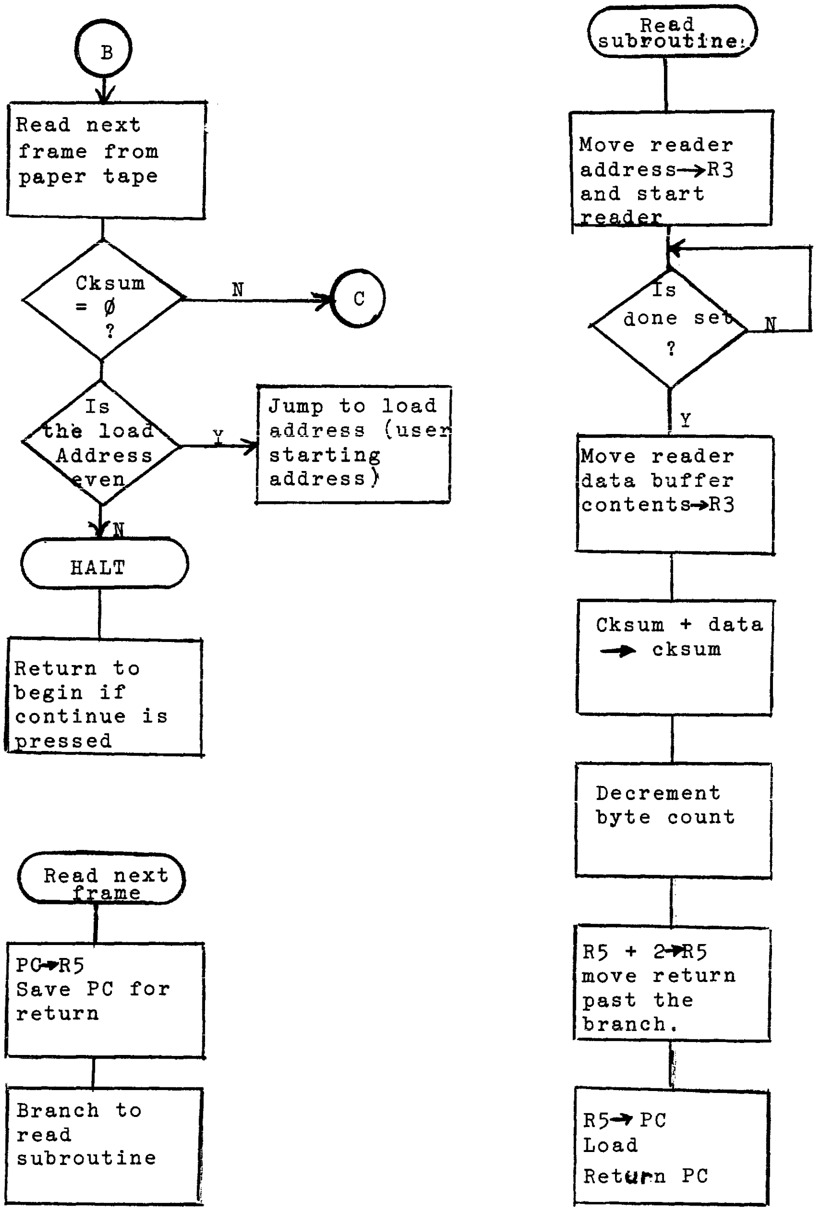
CKSM	= RØ	USED TO CALCULATE CHECKSUMS
ADR	= R1	LOAD ADDRESS
BC	= R2	BYTE COUNT
BYT	= R3	CONTENTS OF BYTE (ALSO USED FOR DEVICE ADRS)
WRD	= R4	FULL WORD ADDRESS (FOR ASSEMBLING BYTE COUNT AND LOAD ADDRESS)
RET	= R5	RETURN FROM SUBROUTINE POINTER
SP	= R6	STACK POINTER (NOT USED)
PC	= R7	PROGRAM COUNTER

THE DEVICE ADDRESS IS HELD IN LOCATION 17776. THE DEVICE ADDRESS IS MOVED TO R3 AT THE BEGINNING OF THE READ SUBROUTINE. ONCE A FRAME HAS BEEN READ, THE DATA IS STORED IN THE LOWER BYTE OF R3. THE CHECKSUM IS CONTAINED IN THE LOWER BYTE OF RØ. AT COMPLETION OF LOADING, THE LOWER BYTE OF THE DATA LIGHTS SHOULD BE ZERO, SIGNIFYING SUCCESSFUL LOADING OF THE TAPE.

11. Maintenance Loader Flowchart


Begin
A
17470 → SP(R6) Set up stack pointer in case error occurs.
0 → Cksum (R0) initialize checksum
17474 → WRD(R4) Set up pointer to routine
Read next frame from paper tape
Frame = 1 ?
N
Y
Read next frame (blank)
Read in byte count (next 2 frames)
Byte count → R2 R2-4 → R2 (4 frames already read)
Read in load address (next 2 frames)
Byte count = 0 ?
Y
B
N
Load address → R1 to set up for loading.
Read next frame from paper tape.
Byte count ≥ 0 ?
Y
Load frame thru load address and increment addr.
N
Cksum = 0 ?
Y
A
N
C
Halt with bad cksum displayed

Maintenance loader flowchart continued


B
Read next frame from paper tape
Cksum = Ø ?
N
C
Is the load Address even
Y
Jump to load address (user starting address)
N
HALT
Return to begin if continue is pressed
Read next frame
PC→R5 Save PC for return
Branch to read subroutine
Read subroutine
Move reader address→R3 and start reader
Is done set ?
N
Y
Move reader data buffer contents→R3
Cksum + data → cksum
Decrement byte count
R5 + 2→R5 move return past the branch.
R5→ PC Load Return PC

```
                                        ;PDP-11 MAINTENANCE LOADER
                                        ;
                                        ;       COPYRIGHT 197Ø, DIGITAL EQUIPMENT CORPORATION
                                        ;
                                        ;INPUT FORMAT --
                                        ;       FRAME   -1      ØØ1
                                        ;               -2      ØØØ
                                        ;               -3      BYTE COUNT - LOWER ORDER
                                        ;               -4      BYTE COUNT - HIGHER ORDER
                                        ;               -5      LOAD ADDRESS - LOWER ORDER
                                        ;               -6      LOAD ADDRESS - HIGHER ORDER
                                        ;                           .    DATA
                                        ;                           .      PLACED
                                        ;                           .        HERE
                                        ;               CKSM - LAST FRAME CONTAINS THE CHECKSUM
                                        ;
                                        ;IF THE BYTE COUNT IS EQUAL TO 6, THE LOAD ADDRESS
                                        ; SPECIFIED WILL BE CONSIDERED TO BE THE DESIRED JUMP
                                        ; ADDRESS.  IF THIS ADDRESS IS ODD, THE LOADER WILL HALT.
                                        ;
                                        ;IF THE BYTE COUNT IS > 6, DATA WILL BE LOADED INTO MEMORY.
                                        ;
                                        ;
                                        ;STORAGE REQUIRED = 84 WORDS.  REGISTERS USED = R1,R2,R3,R4,R5,R6,R7.
                                        ;
                                        ;
                                        ;OPERATING INSTRUCTIONS:
                                        ;       1.  USING THE BOOT LOADER, LOAD THE MAINT LOADER TAPE.
                                        ;       2.  THE DEVICE ADDRESS USED WILL BE CONTAINED IN 17776.  IF THIS
                                        ;           DEVICE ADDRESS IS ZERO THE MAINT LOADER WILL NOT RUN.
                                        ;       3.  PLACE THE MAINT LOADER STARTING ADDRESS(175ØØ) IN THE SWITCH
                                        ;           REGISTER AND PRESS LOAD ADDRESS.
                                        ;       4.  PRESS START.
                                        ;       5.  CHECKSUM IS CONTAINED IN LOWER BYTE OF RØ.
                                        ;           LOWER BYTE OF DISPLAY SHOULD EQUAL ZERO UPON SUCCESSFUL
                                        ;           LOADING OF THE TAPE.
                                        ;       6.  IF THE LOADER HALTS WHEN DONE IT IS ONLY NECESSARY TO
                                        ;           PRESS CONTINUE TO READ IN ADDITIONAL TAPES.
                                        ;       7.  IF THE LOADER HALTS AT THE FAIL ADDRESS (17612), A CHECKSUM
                                        ;           ERROR HAS OCCURRED. MOVE THE PAPER TAPE BACK TO THE
                                        ;           BEGINNING OF THE CURRENT BLOCK AND HIT CONTINUE TO TRY AGAIN.
                                        ;
        ØØØØØØ                          CKSM  =         %Ø              ;CHECKSUM IS KEPT IN RØ
        ØØØØØ1                          ADR     =       %1              ;LOAD ADDRESS
        ØØØØØ2                          BC      =       %2              ;BYTE COUNT
        ØØØØØ3                          BYT     =       %3              ;CONTENTS OF BYTE
        ØØØØØ4                          WRD     =       %4              ;FULL WORD ADDRESS
        ØØØØØ5                          RET     =       %5              ;RETURN FROM SUBROUTINE POINTER
        ØØØØØ6                          SP      =       %6              ;STACK POINTER
        ØØØØØ7                          PC      =       %7              ;PROGRAM COUNTER
                                        ;

        Ø1775Ø                                  LOOP=1775Ø
        Ø174ØØ                                  LOAD=174ØØ
        Ø17774                                  BRNCH=17774
```

```
        Ø17476
        Ø17472                          .=17472
Ø17472  ØØØØØØ                          RETURN: Ø                       ;POINTER FOR NESTED RETURN
Ø17474  ØØØØØØ                          WRD1:   Ø                       ;LOCATION FOR ASSEMBLY OF FULL WORD
Ø17476  ØØØØØØ                          STRT1:  HALT                    ;HALT AT END OF LOADING MAINT LOADER
Ø175ØØ  Ø127Ø6  Ø1747Ø                  BEGIN:  MOV     #1747Ø,SP       ;SETUP STACK IN CASE
Ø175Ø4  ØØ5ØØØ                                  CLR     CKSM
Ø175Ø6  Ø127Ø4  Ø17474                          MOV     #WRD1,WRD       ;LOAD POINTER INTO @4
Ø17512  Ø1Ø7Ø5                          INIT:   MOV     PC,RET          ;SETUP RETURN
Ø17514  ØØØ442                                  BR      READ            ;READ NEXT FRAME
Ø17516  ØØ53Ø3                                  DEC     BYT             ;DECREMENT CHARACTER
Ø1752Ø  ØØ1374                                  BNE     INIT            ;IF NOT FIRST CHARACTER, LOOP
Ø17522  Ø1Ø7Ø5                                  MOV     PC,     RET
Ø17524  ØØØ436                                  BR      READ            ;READ NEXT FRAME
Ø17526  Ø1Ø7Ø5                                  MOV     PC,RET          ;SETUP RETURN
Ø1753Ø  ØØØ434                                  BR      READ            ;READ IN BYTE COUNT
Ø17532  Ø1Ø314                                  MOV     BYT,@WRD        ;SAVE IN WRD1 (LOW BYTE)
Ø17534  Ø1Ø7Ø5                                  MOV     PC,RET          ;SETUP RETURN FROM READ SUB
Ø17536  ØØØ431                                  BR      READ            ;READ NEXT FRAME (HIGH BYTE)
Ø1754Ø  11Ø337  Ø17475                          MOVB    BYT,@#WRD1+1
Ø17544  Ø114Ø2                                  MOV     @WRD,BC         ;MOVE BYTE COUNT TO R2
Ø17546  1627Ø2  ØØØØØ4                          SUB     #4,BC           ;SUBTRACT 4 FROM BYTE COUNT
Ø17552  Ø1Ø7Ø5                                  MOV     PC,RET          ;SETUP RETURN
Ø17554  ØØØ422                                  BR      READ            ;READ IN LOAD ADDRESS
Ø17556  Ø1Ø314                                  MOV     BYT,@WRD        ;SAVE IN WRD1 (LOW BYTE)
Ø1756Ø  Ø1Ø7Ø5                                  MOV     PC,RET          ;SETUP RETURN FROM READ SUB
Ø17562  ØØØ417                                  BR      READ            ;READ NEXT FRAME (HIGH BYTE)
Ø17564  11Ø337  Ø17475                          MOVB    BYT,@#WRD1+1
Ø1757Ø  ØØ57Ø2                                  TST     BC              ;TEST BYTE COUNT
Ø17572  ØØ1426                                  BEQ     JUMP            ;BRANCH IF NO DATA IN BLOCK
Ø17574  Ø114Ø1                                  MOV     @WRD,ADR        ;STORE LOAD ADDRESS
Ø17576  Ø1Ø7Ø5                          A.LOAD: MOV     PC,RET          ;SETUP RETURN
Ø176ØØ  ØØØ41Ø                                  BR      READ            ;READ NEXT FRAME
Ø176Ø2  ØØ57Ø2                                  TST     BC
Ø176Ø4  ØØ2ØØ4                                  BGE     CONT            ;IF NOT CHECKSUM CONTINUE
Ø176Ø6  1057ØØ                                  TSTB    CKSM            ;TEST CKSUM TOTAL
Ø1761Ø  ØØ1733                                  BEQ     BEGIN           ;START ANOTHER BLOCK IF CKSUM OK
Ø17612  ØØØØØØ                          FAIL:   HALT                    ;CHECKSUM FAILED
Ø17614  ØØØ731                                  BR      BEGIN           ;RESTART
Ø17616  11Ø321                          CONT:   MOVB    BYT,(ADR)+      ;LOAD CHARACTER INTO LOAD ADDRESS
Ø1762Ø  ØØØ766                                  BR      A.LOAD          ;CONTINUE

Ø17622  Ø137Ø3  Ø17776                  READ:   MOV     @#17776,%3      ;MOVE DEVICE ADDRESS TO R3
Ø17626  ØØ5213                                  INC     @%3             ;START DEVICE
Ø1763Ø  1Ø5713                                  TSTB    @%3             ;WAIT FOR DONE
Ø17632  1ØØ376                                  BPL     .-2
Ø17634  Ø163Ø3  ØØØØØ2                          MOV     2(%3),BYT       ;MOVE BYTE TO R3
Ø1764Ø  Ø6Ø3ØØ                                  ADD     BYT,CKSM        ;ADD BYTE TO CHECKSUM
Ø17642  ØØ53Ø2                                  DEC     BC              ;DECREMENT BYTE COUNT
Ø17644  ØØ5725                                  TST     (RET)+          ;UPDATE NEW PC FOR RETURN
Ø17646  Ø1Ø5Ø7                                  MOV     RET,PC          ;RETURN
Ø1765Ø  Ø1Ø7Ø5                          JUMP:   MOV     PC,RET          ;SETUP RETURN
Ø17652  ØØØ763                                  BR      READ            ;READ NEXT FRAME
Ø17654  1Ø57ØØ                                  TSTB    CKSM            ;TEST CHECKSUM
Ø17656  ØØ1355                                  BNE     FAIL            ;BRANCH IF NOT ZERO
```

```
017660  006214                          ASR     @WRD            ;CHECK LOAD ADDRESS
017662  103002                          BCC     .+6             ;BRANCH IF LOAD ADDRESS EVEN
017664  000000                          HALT                    ;STOP IF LOAD ADDRESS ODD
017666  000704                          BR      BEGIN           ;RESTART LOADER
017670  006314                          ASL     @WRD            ;RESTORE LOAD ADDRESS
017672  013707  017474                  MOV     @#WRD1,PC       ;JUMP TO USER
        000013                          .REPT   13
                                        0
                                        .ENDR
017676  000000                          0
017700  000000                          0
017702  000000                          0
017704  000000                          0
017706  000000                          0
017710  000000                          0
017712  000000                          0
017714  000000                          0
017716  000000                          0
017720  000000                          0
017722  000000                          0
017724  012767  000352  000020          MOV     #LOOP-LOAD+2,LOOP+2
017732  012767  000765  000034          MOV     #765,BRNCH
017740  000167  177532                  JMP     STRT1
017744  016701                          016701                  ;LOAD OVER BOOTSTRAP TO CAUSE JUMP TO 17724
017746  000026                          000026
017750  012702                          012702
017752  000373                          000373
        000001                          .END
```

```
A.LOAD  017576
ADR     000001R
BC      000002R
BEGIN   017500
BRNCH   017774
BYT     000003R
CKSM    000000R
CONT    017616
FAIL    017612
INIT    017512
JUMP    017650
LOAD    017400
LOOP    017750
PC      000007R
READ    017622
RET     000005R
RETURN  017472
SP      000006R
STRT1   017476
WRD     000004R
WRD1    017474
```

```
ERRORS DETECTED: 0

RUN-TIME:  1 SECONDS

4K CORE USED
```