```
                                835  ***     ERROR - COMMAND ERROR.
                                836  *
                                837  *       ERROR IS CALLED AS A 'BAIL-OUT' ROUTINE.
                                838  *
                                839  *       IT RESETS THE OPERATIONAL MODE, AND RESTORES THE STACKPOINTER.
                                840  *
                                841  *       ENTRY   NONE
                                842  *       EXIT    TO MTR LOOP
                                843  *               CTLFLG SET
                                844  *               .MFLAG CLEARED
                                845  *       USES    ALL
                                846
    000.001                     847          IF      .RAM.
                                848          ELSE
    000.000                     849          ERRNZ   *-322A
                                850          ENDIF
                                851
    000.322                     852  ERROR   EQU     *
    000.322  041 010 040        853          LXI     H,.MFLAG
    000.325  176                854          MOV     A,M             (A) = .MFLAG
    000.326  346 275            855          ANI     377Q-UO.DDU-UO.NFR      RE-ENABLE DISPLAYS
    000.330  167                856          MOV     M,A             REPLACE
    000.331  043                857          INX     H
    000.332  066 360            858          MVI     M,CB.SSI+CB.MTL+CB.CLI+CB.SPK   RESTORE *CTLFLG*
    000.000                     859          ERRNZ   CTLFLG-.MFLAG-1
    000.334  373                860          EI
    000.335  052 035 040        861          LHLD    REGPTR
    000.340  371                862          SPHL                    RESTORE STACK POINTER TO EMPTY STATE
    000.341  315 136 002        863          CALL    ALARM           ALARM FOR 200 MS


                                865  **      MTR - MONITOR LOOP.
                                866  *
                                867
    000.001                     868          IF      .RAM.
                                869          ELSE
    000.000                     870          ERRNZ   *-344A
                                871          ENDIF
                                872
    000.344                     873  MTR     EQU     *
    000.344  373                874          EI
                                875
    000.345                     876  MTR1    EQU     *
    000.345  041 345 000        877          LXI     H,MTR1
    000.350  345                878          PUSH    H               SET 'MTR1' AS RETURN ADDRESS
    000.351  303 113 002        879          JMP     CKAUTO          CHECK AUTO BOOT, IF NOT CONTROL BACK TO NEXT
    000.354  315 100 006        880  MTR.15  CALL    TYPMSG          PRINT 'H:'
                                881
    000.357  315 262 003        882  MTR.2   CALL    RCC             READ A CONSOLE CHARACTER
    000.362  346 137            883          ANI     01011111B       MAKE SURE ITS UPPER CASE TO MATCH TABLE
    000.364  041 025 001        884          LXI     H,MTRA          LOOK UP CHARACTER IN *MTRA*
    000.367  006 004            885          MVI     B,MTRAL         (B) = LENGTH OF TABLE
    000.371  276                886  MTR.3   CMP     M               SEE IF CHARACTER FROM CONSOLE = TABLE ENTRY
    000.372  312 014 001        887          JZ      MTR.4           IF EQUAL
```

```
                                888
                                889             INX     H               POINT TO NEXT TABLE ENTRY
                                890             INX     H
                                891             INX     H
                                892             DCR     B               SEE IF PAST END OF TABLE
                                893             JNZ     MTR.3           IF NOT PAST
                                894
                                895             MVI     A,A.BEL         ELSE, DING ERROR
                                896             CALL    WCC
                                897             JMP     MTR.2           TRY AGAIN
                                898
                                899     MTR.4   CALL    WCC             WRITE CHARACTER BACK TO CONSOLE
                                900             INX     H               GET ROUTINE ADDRESS LSB
                                901             MOV     A,M
                                902             INX     H               GET MSB
                                903             MOV     H,M
                                904             MOV     L,A             (H,L) = ROUTINE ADDRESS
                                905             PCHL                    GO TO ROUTINE
                                906
                                907
                                908
                                909     MTRA    EQU     *               JUMP TABLE
                                910             SET     *#256           ALL ROUTINES MUST START IN THIS PAGE
                                911             DB      'G'             GO TO USER ROUTINE
                                912             DW      GO88
                                913
                                914             DB      'S'             SUBSTITUTE MEMORY MODE
                                915             DW      SUBM
                                916
                                917             DB      'P'             PROGRAM COUNTER ALTER MODE
                                918             DW      PCA
                                919
                                920             DB      'B'             BOOT H-17 OR Z-47 DRIVE
                                921             DW      BOOT
                                922
                                923     MTRAL   EQU     *-MTRA/3        NUMBER OF TABLE ENTRYS          /JWT 790507/
                                925     **      BSMSG   = BOOT SECONDARY DEVICE MESSAGE
                                926
                                927     BSMSG   DB      ' SD',0         'SECONDARY DEVICE'
                                929     ERRMSG  DB      '?',0           ERROR MESSAGE
```

```
001,047                 931             ORG     1047A
                        932  **         MSG.ERR - ERROR MESSAGE FOR RAM TEST
                        933  *
                        934  *          'ERROR @ '
                        935
001,047  015 012 012    936  MSG.ERR    DB      A.CR,A.LF,A.LF
001,052  105 122 122    937             DB      'ERROR @ '
001,062  000            938             DB      0
```

```
                                941  **      SAE - STORE ABUSS AND EXIT.
                                942  *
                                943  *       ENTRY   (HL) = ABUSS VALUE
                                944  *       EXIT    TO (RET)
                                945  *       USES    NONE
                                946
  000.001                       947           IF      ,RAM,
                                948           ELSE
  000.000                       949           ERRNZ   *-1063A
                                950           ENDIF
                                951
  001.063  042 024 040          952  SAE      SHLD    ABUSS
  001.066  311                  953           RET



                                955  **      PIN     -       PORT IN
                                956  *
                                957  *       PIN  INPUTS A BYTE FROM DISK
                                958  *
                                959  *       ENTRY:  NONE
                                960  *
                                961  *       EXIT:   (A) = INPUT BYTE FROM Z47
                                962  *
                                963  *       USE:    AF
                                964
  001.067                       965  PIN      EQU     *
  001.067  315 170 006          966           CALL    IN.                     GET STATUS
  001.072  346 200              967           ANI     S.DTR                   CHECK FOR DATA TERMINAL REQUEST
  001.074  050 371              968           JR      Z,PIN                   IF READY, WAIT
  001.076  315 156 006          969           CALL    IN1.                    INPUT A BYTE FROM PORT
  001.101  311                  970           RET
```

```
001.103                     973             ORG     1103A
                            974  **         PCA - PROGRAM COUNTER ALTER
                            975  *
                            976  *          PCA INPUTS AND/OR DISPLAYS THE CURRENT USER PROGRAM VALUE AND ALLOWS
                            977  *          A NEW VALUE TO BE ENTERED OR RETAINS THE CURRENT VALUE IF
                            978  *          A CR IS TYPED
                            979  *
                            980  *          ENTRY   NONE
                            981  *          EXIT    NONE
                            982  *          USES    A,D,E,H,L,F
                            983
                            984
001.103  041 214 006       985  PCA        LXI     H,MSG.PC        COMPLETE PC MESSAGE
001.106  315 100 006       986             CALL    TYPMSG
001.111  076 012           987             MVI     A,10            GET LOCATION OF USER PC
001.113  315 052 003       988             CALL    LRA.
001.116  136               989             MOV     E,M             (D,E) = USER PC VALUE
001.117  043               990             INX     H
001.120  126               991             MOV     D,M
001.121  353               992             XCHG                    (H,L) = USER PC VALUE
                            993
001.122  315 150 005       994             CALL    IROC            INPUT NEXT CHARACTER
001.125  332 137 001       995             JC      PCA1            IF FIRST CHARACTER WAS OCTAL, INPUT NEW PC
                            996
001.130  315 313 005       997             CALL    TOA             ELSE, OUTPUT CURRENT VALUE
001.133  315 150 005       998             CALL    IROC            SEE IF USER WANTS TO CHANGE IT NOW
001.136  320               999             RNC                     IF NO CHANGE, EXIT
                           1000
                           1001  *          ENTER NEW USER PC VALUE
                           1002
001.137  353               1003 PCA1       XCHG                    (H,L) = ADDRESS OF USER PC VALUE
001.140  026 015           1004            MVI     D,A.CR          END BYTE WITH A RETURN
001.142  315 062 003       1005            CALL    IOA             INPUT NEW ADDRESS
001.145  311               1006            RET                     EXIT


                           1008  **         GO88 - GO TO USER ROUTINE FROM H88 MONITOR
                           1009  *
                           1010  *          GO88 WAITS FOR A CARRIAGE RETURN OR A NEW ADDRESS TERMINATED WITH
                           1011  *          A CARRIAGE RETURN.  IF NO ADDRESS IS ENTERED, GO88 TRANSFERS
                           1012  *          CONTROL TO THE ADDRESS SPECIFIED BY THE USER PC VALUE
                           1013
                           1014
001.146  041 165 006       1015 GO88       LXI     H,MSG.GO        COMPLETE GO MESSAGE
001.151  315 100 006       1016            CALL    TYPMSG
001.154  315 150 005       1017            CALL    IROC            INPUT A RETURN OR AN OCTAL CHARACTER
001.157  322 177 001       1018            JNC     GO88.1          IF RETURN, GO TO CURRENT USER PC
                           1019
001.162  365               1020            PUSH    PSW             ELSE SAVE OCTAL CHARACTER AND FLAGS
001.163  076 012           1021            MVI     A,10            GET ADDRESS OF USER PC
001.165  315 052 003       1022            CALL    LRA.
001.170  043               1023            INX     H               POINT TO MSB
001.171  361               1024            POP     PSW             GET FIRST CHARACTER BACK
001.172  026 015           1025            MVI     D,A.CR          END ADDRESS WITH A RETURN
```

```
MTR89 - H89 MONITOR  #09.01.00.                          Zenith Data Systems UNIX H8/H89 Cross Assembler  PA
GE   27
MONITOR TASK SUBROUTINES.                                G088          15:27:32  28-MAY-80

001.174  315 062 003  1026            CALL    IOA       INPUT NEW GO ADDRESS
001.177  315 302 003  1027  G088.1    CALL    WCC       ECHO RETURN
001.202  076 012      1028            MVI     A,A.LF    LINE FEED
001.204  315 302 003  1029            CALL    WCC
001.207  303 222 001  1030            JMP     GO        EXECUTE USER ROUTINE


                      1032  **        AUTOBO  - AUTO BOOT
                      1033  *
                      1034  *         ENTRY:  NONE
                      1035  *
                      1036  *         EXIT:   (SEE 'DEVICE' ROUTINE)
                      1037  *
                      1038  *         USE:    ALL
                      1039
001.212  257          1040  AUTOBO    XRA     A         SET TO PRIMARY FLAG
001.213  315 301 002  1041            CALL    DEVICE    CHECK DEVICE INFROMATION
001.216  303 336 001  1042            JMP     BOOTO     GOTO BOOT IT


001.222               1044            ORG     1222A
                      1045  **        GO - RETURN TO USER MODE
                      1046  *
                      1047  *         ENTRY   NONE
                      1048
000.001               1049            IF      .RAM.
                      1050            ELSE
000.000               1051            ERRNZ   *-1222A
                      1052            ENDIF
                      1053
001.222  303 063 000  1054  GO        JMP     GO.


                      1056  **        SSTEP - SINGLE STEP INSTRUCTION.
                      1057  *
                      1058  *         ENTRY   NONE
                      1059
000.001               1060            IF      .RAM.
                      1061            ELSE
000.000               1062            ERRNZ   *-1225A
                      1063            ENDIF
                      1064
001.225               1065  SSTEP     EQU     *         SINGLE STEP
001.225  363          1066            DI                DISABLE INTERRUPTS UNTIL THE RIGHT TIME
001.226  072 011 040  1067            LDA     CTLFLG
001.231  356 020      1068            XRI     CB.SSI    CLEAR SINGLE STEP INHIBIT
001.233  323 360      1069            OUT     OP.CTL    PRIME SINGLE STEP INTERRUPT
001.235  062 011 040  1070  SST1      STA     CTLFLG    SET NEW FLAG VALUES
001.240  341          1071            POP     H         CLEAN STACK
001.241  303 172 000  1072            JMP     INTXIT    RETURN TO USER ROUTINE FOR STEP
```

```
                                1074  **      STPRTN - SINGLE STEP RETURN
                                1075
    000.001                     1076          IF      .RAM.
                                1077          ELSE
    000.000                     1078          ERRNZ   *-1244A
                                1079          ENDIF
                                1080
    001.244                     1081  STPRTN  EQU     *
    001.244  366 020            1082          ORI     CB.SSI          DISABLE SINGLE STEP INTERRUPTION
    001.246  323 360            1083          OUT     OP.CTL          TURN OFF SINGLE STEP ENABLE
    040.011                     1084          SET     CTLFLG
    001.250  022                1085          STAX    D
    001.251  346 040            1086          ANI     CB.MTL          SEE IF IN MONITOR MODE
    001.253  302 344 000        1087          JNZ     MTR
    001.256  303 042 040        1088          JMP     UIVEC+3         TRANSFER TO USER'S ROUTINE
```

```
                        1091  **      NBOOT   -       NORMAL BOOT
                        1092  *
                        1093  *       NBOOT IS ENTERED WHEN USER TYPE 'BOOT' COMMAND FROM MONITOR,
                        1094  *       IT WILL ACCEPT THE BOOT DEVICE AS WELL AS THE UNIT NUMBER FROM
                        1095  *       CONSOLE AND GO TO THE BOOT CODE.
                        1096  *
                        1097  *       ENTRY:  NONE
                        1098  *
                        1099  *       EXIT:   (AIO.UNI) = UNIT NUMBER TO BOOT
                        1100  *               (PRIM)  = PORT ADDRESS OF THE BOOT DEVICE
                        1101  *               (TMFG)  = DEVICE TYPE, =1 IS Z47; =0 IS H17
                        1102  *
                        1103  *       USED:   ALL
                        1104
001.261  257            1105  NBOOT   XRA     A                       SET Z FLAG TO PRIMARY DEVICE
001.262  315 301 002    1106  NBOOT0  CALL    DEVICE                  READ SWITCH TO DETERMINE BOOT DEVICE
001.265  315 262 003    1107  START1  CALL    RCC                     INPUT FROM KB
001.270  376 015        1108          CPI     A,CR                    IF INPUT IS CR
001.272  050 042        1109          JR      Z,BOOT0                   THEN TAKE IT AS DRIVE 0
001.274  376 060        1110          CPI     '0'                     CHECK INPUT IS WITHIN DRIVE 0 - (B)
001.276  070 007        1111          JR      C,WRONG                   IF LESS THEN 0, WRONG INPUT
001.300  270            1112          CMP     B
001.301  070 036        1113          JR      C,BOOT5                   IF WITHIN THE RANGE, BOOT IT!
001.303  010            1114          DB      MI.EXAF                 SAVE INPUT, CHECK PRIM OR SEC?
001.304  050 010        1115          JR      Z,NB7                   IF PRIMARY, CHECK 'S'
001.306  010            1116          DB      MI.EXAF                 RESTORE (Z) FLAG
001.307                  1117  WRONG   EQU     *
001.307  076 007        1118          MVI     A,A.BEL                 NOT THE CASES, BEEP!
001.311  315 302 003    1119          CALL    WCC
001.314  030 347        1120          JR      START1                  AND TRY AGAIN
                        1121
001.316  010            1122  NB7     DB      MI.EXAF                 RESTORE INPUT & PRIM, SEC FLAG
001.317  346 137        1123          ANI     01011111B               MASK TO UPPER CASE LETTER
001.321  376 123        1124          CPI     'S'                     CHECK THE USER LIKE TO BOOT FROM
001.323  040 362        1125          JR      NZ,WRONG                BOOT SECONDARY DEVICE
                        1126
                        1127  *       USER WISHES TO BOOT FROM SECONDARY DEVICE
                        1128
001.325                 1129  BSEC    EQU     *
001.325  041 041 001    1130          LXI     H,BSMSG                 PRINT BOOT SECONDARY MESSAGE
001.330  315 100 006    1131          CALL    TYPMSG
001.333  074            1132          INR     A                       SET (Z)=0 FOR SECONDARY DEVICE
001.334  030 324        1133          JR      NBOOT0
                        1134
                        1135  *  SAVE THE AIO.UNI, CHECK IF THERE IS THE BOOT DEVICE AND GO!
                        1136
001.336  257            1137  BOOT0   XRA     A                       TAKE CR OR AUTO BOOT AS DRIVE 0
001.337  030 012        1138          JR      BOOT6
                        1139
001.341  315 302 003    1140  BOOT5   CALL    WCC                     PRINT UNIT NUMBER
001.344  326 060        1141          SUI     '0'                     MAKE IT BINARY
001.346  107            1142          MOV     B,A                     SAVE THE UNIT #
001.347  315 003 006    1143          CALL    WCR                     WAIT FOR A CR
001.352  170            1144          MOV     A,B                     GET UNIT NUMBER BACK
001.353  062 061 041    1145  BOOT6   STA     AIO.UNI                 STORE THE UNIT #
001.356  174            1146          MOV     A,H                     CHECK IF NO DEVICE AT ADDR. PORT
```

```
001.357  247          1147        ANA     A
001.360  312 171 002  1148        JZ      NODEV                NO DEVICE
001.363  351          1149        PCHL                         JMP TO THE EXECUTION ROUTINE
```

```
                        1152  **      Z47     -       BOOT FROM Z47 DISK DRIVE
                        1153  *
                        1154  *       Z47 WILL LOAD DATA FROM DISK TRACK 0 SECTOR 1 AND 2 TO
                        1155  *       USER FIRST AVALIABLE RAM LOCATION. IF THE BOOT IS SUCCED,
                        1156  *       CONTROL PASS TO THAT LOCATION.
                        1157  *
                        1158  *       ENTRY:  (AIO,UNI) = UNIT NUMBER TO BOOT
                        1159  *
                        1160  *       EXIT:   NONE
                        1161  *
                        1162  *       USE:    ALL
                        1163
001,364                 1164  Z47     EQU     *
                        1165  *       LD      (STK),SP                SAVE STACK POINTER FOR RE-BOOT
001,364 355 163         1166          DB      355Q,163Q
001.366 124 041         1167          DW      STK
                        1168
001.370                 1169  Z47A    EQU     *
001,370 373             1170          EI                              LET THE TIMER FLY
001.371 072 061 041     1171          LDA     AIO.UNI                 GET UNIT NUMBER
001,374 007             1172          RLC                             SET TO SIDE/UNIT/SECTOR FORMAT
001.375 007             1173          RLC
001,376 007             1174          RLC
001.377 007             1175          RLC
002,000 007             1176          RLC
002.001 074             1177          INR     A                       SET TO SECTOR 1
002,002 117             1178          MOV     C,A                     SAVE SIDE/UNIT/SECTOR (SIDE=0)
002.003 076 002         1179  RESET   MVI     A,W.RES                 RESET Z47
002,005 315 063 006     1180          CALL    OUT.
                        1181
                        1182  * DETERMINE THE DISK IS SINGLE OR DOUBLE DENSITY
                        1183
002,010 076 002         1184          MVI     A,DC,RAS                SEND READ AUX. STATUS COMMAND
002.012 315 027 006     1185          CALL    COM
002,015 171             1186          MOV     A,C                     GET SIDE/UNIT/SECTOR
002.016 315 023 006     1187          CALL    DAT                     SEND SECOND COMMAND BYTE
002,021 315 067 001     1188          CALL    PIN                     GET AUX. STATUS
002.024 346 100         1189          ANI     AS.ODD                  CHECK IT IS SINGLE OR DOUBLE DENSITY
002,026 007             1190          RLC
002.027 356 200         1191          XRI     10000000B               REVERSE THEN 7TH BIT, MAKE THE SECTOR
002,031 107             1192          MOV     B,A                     # TO 128 OR 256(B=0) BYTES
                        1193
                        1194  *       READ BOOT CODE FROM Z47
                        1195
002,032 041 200 042     1196          LXI     H,USERFWA               BOOT DESTINATION
002.035 305             1197          PUSH    B                       SAVE SECTOR SIZE & SIDE/UNIT/SECTOR
002,036 315 121 006     1198          CALL    RDBLCK                  READ A SECTOR FROM DISK
002.041 301             1199          POP     B                       GET SECTOR SIZE & SUS BACK
002,042 014             1200          INR     C                       SET TO NEXT SECTOR
002.043 315 121 006     1201          CALL    RDBLCK                  READ ANOTHER SECTOR
                        1202
                        1203  *       CHECK ANY ERROR DURING BOOT
                        1204
002.046 315 170 006     1205          CALL    IN.                     GET INTERFACE STATUS
002,051 346 001         1206          ANI     S.ERR                   IS THERE ANY ERROR WHEN BOOT
002.053 040 114         1207          JR      NZ,NODEV                      THEN ABORT
```

```
002.055  062 010 040  1208           STA     .MFLAG                  STOP TIMER
002.060  303 200 042  1209           JMP     USERFWA

                      1211  **       RETRY   -       RE-BOOT Z47
                      1212  *
                      1213  *       RETRY IS ENTERED WHEN 3.5 SECONDS TIME OUT & BOOT Z47
                      1214  *       STILL NOT SUCCEED. IT RESTORE STACK & JUMP TO BOOT Z47 ROUTINE
                      1215  *
                      1216  *       ENTRY:  NONE
                      1217  *
                      1218  *       EXIT:   (HL) = (SP)
                      1219  *
                      1220  *       USE:    HL, SP
                      1221
002.063  052 124 041  1222  RETRY    LHLD    STK                     GET OLD STACK ADDRESS
002.066  371         1223            SPHL                            SET TO STACK POINTER
002.067  030 277      1224           JR      Z47A                    RE-BOOT
```

```
                                1227  **      R.SDP   - SET DEVICE PARAMETER, ALLOW TO SET DRIVE 0, 1, AND 2,
                                1228  *       (MORE INFORMATION CAN BE FOUND IN H17 ROM CODE 36062A)
                                1229
     002.071                    1230  R.SDP   EQU     *
     002.071  076 012           1231          MVI     A,ERPTCNT
     002.073  062 264 040       1232          STA     D.OECNT                 SET MAX ERROR COUNT FOR OPERATION
     002.076  072 061 041       1233          LDA     AIO.UNI                 LOAD DRIVE NUMBER
     002.101  365               1234          PUSH    PSW                     SAVE IT
     002.102  376 002           1235          CPI     2                       IS IT DRIVE 2?
     002.104  070 002           1236          JR      C,R.SDP1                IF NOT JMP TO H17 ROM ROUTINE
     002.106  076 003           1237          MVI     A,3
     002.110  303 073 036       1238  R.SDP1  JMP     SDP3



                                1240  **      CKAUTO  -       CHECK IF IT IS AUTO BOOT
                                1241  *
                                1242  *       CKAUTO IS ENTERED FROM MONITOR LOOP, IT WILL CHECK IF AUTO BOOT
                                1243  *       CONDITION IS TRUE.IF NOT, BACK TO MONITOR LOOP
                                1244  *               IF AUTO BOOT, JUMP TO BOOT DEVICE ROUTINE
                                1245  *
                                1246  *       ENTRY:  NONE
                                1247  *
                                1248  *       EXIT:   NONE
                                1249  *
                                1250  *       USE:    ALL
                                1251
     002.113                    1252  CKAUTO  EQU     *
     002.113  333 362           1253          IN      H88.SW          GET SWITCH DATA
     002.115  346 200           1254          ANI     H88S.AT         CHECK AUTO BOOT SWITCH BIT SET
     002.117  050 007           1255          JR.     Z,CHAT2         NOT SET
     002.121  041 123 041       1256          LXI     H,AUTOB         SET AUTO BOOT FLAG ADDR,
     002.124  276               1257          CMP     M               CHECK AUTO BOOT BEFORE?
     002.125  302 243 004       1258          JNZ     ATB             YES, AUTO BOOT
     002.130  041 111 006       1259  CHAT2   LXI     H,MSG.PR        LOAD 'H:' ADDR.
     002.133  303 354 000       1260          JMP     MTR.15          BACK TO MONITOR LOOP
```

```
002.136                 1263            ORG     2136A
                        1264  **        HORN - MAKE NOISE.
                        1265  *
                        1266  *         ENTRY   (A) = (MILLISECOND COUNT)/2
                        1267  *         EXIT    NONE
                        1268  *         USES    A,F
                        1269
000.001                 1270            IF      .RAM.
                        1271            ELSE
000.000                 1272            ERRNZ   *-2136A
                        1273            ENDIF
                        1274
002.136                 1275  ALARM     EQU     *
002.136  030 026        1276            JR      ALARMB          BRANCH TO A JUMP TO NOISE TO DING BELL
                        1277
000.001                 1278            IF      .RAM.
                        1279            ELSE
000.000                 1280            ERRNZ   *-2140A
                        1281            ENDIF
                        1282
002.140  365            1283  HORN      PUSH    PSW
002.141  076 200        1284            MVI     A,CB.SPK        TURN ON SPEAKER
                        1285
002.143  343            1286  HRN0      XTHL                    SAVE (HL), (H)  = COUNT
002.144  325            1287            PUSH    D               SAVE (DE)
002.145  353            1288            XCHG                    (D) = LOOP COUNT
002.146  041 011 040     1289            LXI     H,CTLFLG
002.151  256            1290            XRA     M
002.152  136            1291            MOV     E,M             (E) = OLD CTLFLG VALUE
002.153  167            1292            MOV     M,A             TURN ON HORN
002.154  056 033        1293            MVI     L,#TICCNT
                        1294
002.156  172            1295            MOV     A,D             (A) = CYCLE COUNT
002.157  206            1296            ADD     M
002.160  276            1297  HRN2      CMP     M               WAIT REQUIRED TICCOUNTS
002.161  040 375        1298            JR      NZ,HRN2
                        1299
002.163  303 045 006    1300            JMP     HRNX            JUMP TO AN EXTENSION OF HORN SO ROOM
                        1301  *                                       CAN BE MADE FOR A JUMP TO NIOSE
                        1302
                        1303
002.166  303 053 006    1304  ALARMB    JMP     NOISE           SEND A BELL TO THE CONSOLE
```

```
                        1307  **      NODEV   - NO DEVICE AT THE UNIT USER INDICATE
                        1308  *
                        1309  *       NODEV IS ENTERED WHEN:  1. 15 SECONDS TIME OUT
                        1310  *                                OR 2. NO DEVICE IS INDICATED ON SWITCH
                        1311  *                                OR 3. USER HIT <DELETE> TO ABORT BOOT
                        1312  *                                OR 4. BOOT ERROR
                        1313  *       IT WILL EXIT TO 'ERROR' ROUTINE AND MONITOR LOOP
                        1314  *
                        1315  *       ENTRY:  NONE
                        1316  *
                        1317  *       EXIT:   (A) = 0
                        1318  *
                        1319  *       USE:    AF, HL
                        1320
002,171                 1321  NODEV   EQU     *
002,171  041 045 001    1322          LXI     H,ERRMSG        PRINT ERROR MESSAGE
002,174  315 100 006    1323          CALL    TYPMSG
002,177  062 010 040    1324          STA     .MFLAG          STOP TIMER
002,202  323 177        1325          OUT     DP.DC           OFF DISK
002,204  303 322 000    1326          JMP     ERROR           BACK TO MONITOR LOOP
```

```
                        1329  **      H17     -   BOOT FROM H17 DISK SYSTEM
                        1330  *                   (THIS IS THE MODIFICATION OF THE H17 BOOT ROUTINE,
                        1331  *                   MORE INFORMATION CAN BE FOUND ON H17 BOOT ROM 30000A)
                        1332  *
                        1333  *       ENTRY:  (AIO.UNI) = THE UNIT TO BOOT
                        1334  *
                        1335  *       EXIT:   NONE
                        1336  *
                        1337  *       USE:    ALL
                        1338
002.207                 1339  H17     EQU     *
002.207  001 130 000    1340          LXI     B,BOOTAL        SET THE COUNT TO MOVE IN CONSTANTS AND VECTORS
002.212  021 132 037    1341          LXI     D,BOOTA         SET THE SOURCE ADDRESS
002.215  041 110 040    1342          LXI     H,D.CON         SET THE DESTINATION ADDRESS
002.220  315 252 030    1343          CALL    *MOVE           MOVE IT
                        1344
                        1345  **      SET ADDRESS FOR 'SET DEVICE PARAMETER' ROUTINE
                        1346  *       TO HANDLE DISK DRIVE 0, 1, AND 2.
                        1347
002.223  041 071 002    1348          LXI     H,R.SDP         SET THIS ROM ROUTINE ADDRESS
002.226  042 206 040    1349          SHLD    D,SDP           SET INTO RAM JUMP VECTOR
002.231  373            1350          EI                      RESTORE INTERRUPT
                        1351
                        1352  *       WAIT TILL USER INSERT THE DISK AND CLOSE THE DOOR
                        1353  *       (TIMER INTERRUPT IS AFFECTED NOW)
                        1354
002.232  006 012        1355          MVI     B,10            LOOK FOR SOME HOLE AND NO HOLE
002.234  315 071 002    1356          CALL    R,SDP           SELECT UNIT & MOTOR ON
002.237  315 271 036    1357  H17A    CALL    WNH             WAIT FOR NO HOLE
002.242  315 235 036    1358          CALL    WHD             WAIT FOR HOLE
002.245  020 370        1359          DJNZ    H17A
                        1360
                        1361  *       READ BOOT CODE
                        1362
002.247  315 366 033    1363          CALL    R,ABORT         RESET DISK DRIVE
002.252  021 200 042    1364          LXI     D,USERFWA       SET THE LOAD LOCATION
002.255  001 000 011    1365          LXI     B,9*256         LOAD 9 SECTORS
002.260  041 000 000    1366          LXI     H,0             LOAD FROM TRACK 0 SECTOR 1
002.263  315 077 034    1367          CALL    R,READ          READ DISK BOOT CODE
002.266  070 301        1368          JR      C,NODEV           ERROR ON BOOT, BACK TO "H:"
                        1369
                        1370  **      SETUP CLOCK INTERRUPT  FOR H17 ONLY
                        1371
002.270  041 031 034    1372          LXI     H,CLOCK17       LOAD CLOCK ROUTINE ADDRESS
002.273  042 040 040    1373          SHLD    .UIVEC+1        SET IT INTO VECTOR LOCATION
002.276  303 200 042    1374          JMP     USERFWA         GOTO BOOT CODE
```

```
                        1377  ***     DEVICE  -       DETERMINE BOOT WHICH DEVICE AT WHICH PORT
                        1378  *
                        1379  *       ENTRY:  Z FLAG  ( Z=1 FOR PRIMARY, Z=0 FOR SECONDARY)
                        1380  *
                        1381  *       EXIT:   HL = DEVICE BOOT EXECUTION ADDRESS
                        1382  *                    IF H=0 THEN NO DEVICE THERE
                        1383  *                    (I.E. THE EXEC. ADDR. MUST RESIDENT > 1000A)
                        1384  *               REG B = PRIMARY MAXI. DRIVE NUMBER
                        1385  *                    IF Z47, #='4'; IF H17, #='3'
                        1386  *               (PRIM) = PRIMARY DEVICE PORT ADDRESS
                        1387  *                    IF Z47 THEN THE PORT IS EITHER 170Q OR 174Q
                        1388  *                    IF H17 THEN DON'T CARE (H17 BOOT ROM TAKE CARE IT)
                        1389  *               (TMFG) = 1 IF BOOT FROM Z47, = 0 IF FROM H17
                        1390  *
                        1391  *       USE:    ALL
                        1392
    002.301             1393  DEVICE  EQU     *
    002.301  010        1394          DB      MI.EXAF         SAVE Z FLAG
                        1395
                        1396  *       INITIAL VARIABLES
                        1397
    002.302  363        1398          DI                      NO INTERRUPT
    002.303  041 240 040 1399          LXI     H,D.RAM         CLEAR H17 WORK RAM AREA
    002.306  006 037    1400          MVI     B,D.RAML        LENGTH TO CLEAR
    002.310  315 212 031 1401          CALL    $ZERO
    002.313  323 177    1402          OUT     DP.DC           OFF DISK
    002.315  062 033 040 1403          STA     TICCNT          0 TIMER COUNTER
    002.320  062 122 041 1404          STA     MYCNT           0.5 SECOND TIMER = 0
                        1405
    002.323  074        1406          INR     A               (A)=1
    002.324  062 121 041 1407          STA     TMFG            SET TIMER TO Z47 FLAG
    000.000             1408          ERRNZ   UO.CLK-1        TIMER INTERRPT MUST = 1
    002.327  062 010 040 1409          STA     .MFLAG          ALLOW TIMER INTERRUPT
    002.332  041 037 040 1410          LXI     H,.UIVEC        SET ALL VECTOR TO EI/RET PROCESS
    002.335  066 303    1411  BOOT2   MVI     M,MI.JMP
    002.337  043        1412          INX     H
    002.340  066 027    1413          MVI     M,$EIXIT        STORE LS BYTE
    002.342  043        1414          INX     H
    002.343  066 034    1415          MVI     M,EIXIT/256     STORE MS BYTE
    002.345  043        1416          INX     H
    002.346  207        1417          ADD     A
    002.347  362 335 002 1418          JP      BOOT2
                        1419
    002.352  041 302 004 1420          LXI     H,TMOUT         SET TIMER INTERRUPT VECTOR
    002.355  042 040 040 1421          SHLD    .UIVEC+1
                        1422
    002.360  076 170    1423          MVI     A,D.STA         ASSUME ALL DEVICE ARE Z47 & BOOT AT 170Q
    002.362  062 120 041 1424          STA     PRIM            SINCE H17 BOOT ROM WILL TAKE CARE OF ITS MATTER
    002.365  041 364 001 1425          LXI     H,Z47           SET Z47 BOOT ADDR.
    002.370  006 064    1426          MVI     B,'4'           SET MAX. UNIT TO 4
                        1427
                        1428  *       DETERMINE BOOT DEVICE AND ITS INFORMATION
                        1429
    002.372  333 362    1430          IN      H88.SW          READ SWITCH DATA
    002.374  365        1431          PUSH    PSW             SAVE IN STACK
    002.375  346 020    1432          ANI     H88S.DV         CHECK PRIMARY DEVICE ADDRESS
```

```
003,377  010          1433           DB      MI,EXAF         SAVE Z FLAG & GET Z' FOR PRIM. SEC. FLAG
003,000  040 005      1434           JR      NZ,SECOND       IT SECONDARY
003,002  010          1435           DB      MI,EXAF
003,003  040 033      1436           JR      NZ,B170
003,005  030 003      1437           JR      B174
003,007  010          1438  SECOND   DB      MI,EXAF
003,010  050 026      1439           JR      Z,B170          BOOT PRIMARY AT 170Q
003,012  076 174      1440  B174     MVI     A,UP.DP         PRIMARY DEVICE IS AT 174Q
003,014  062 120 041  1441           STA     PRIM
003,017  361          1442           POP     PSW             GET SWITCH DATA BACK
003,020  346 003      1443           ANI     H88S,4          CHECK THIS IS Z47 OR H17
003,022  050 004      1444           JR      Z,BH17          IT H17
003,024  075          1445           DCR     A
003,025  310          1446  DEV2     RZ                      IT IS Z47
003,026  045          1447           DCR     H               NO DEVICE THERE, Z47 LOCATION MUST ON 1***A
003,027  311          1448           RET
000,000               1449           ERRNZ   Z47/256-1
                      1450
                      1451  *        PRIMARY DEVICE IS H17
                      1452
003,030  041 207 002  1453  BH17     LXI     H,H17           SET TO H17 EXECUTION LOACTION
003,033  005          1454           DCR     B               SET TO MAX 3 DRIVE
003,034  062 121 041  1455           STA     TMFG            SET TIMER INTERRUPT = 0 FOR H17
003,037  311          1456           RET
                      1457
                      1458  **       PRIMARY DEVICE IS AT PORT 170Q
                      1459
003,040               1460  B170     EQU     *
003,040  361          1461           POP     PSW             GET SWITCH DATA
003,041  346 014      1462           ANI     H88S.0          CHECK ANY DEVICE IN 170Q
003,043  376 004      1463           CPI     00000100B       CHECK IF IT IS Z47
003,045  030 356      1464           JR      DEV2
```

```
003.047                  1467          ORG     3047A
                         1468  **      LRA - LOCATE REGISTER ADDRESS.
                         1469  *
                         1470  *       ENTRY   NONE.
                         1471  *       EXIT    (A) = REGISTER INDEX
                         1472  *               (H,L) = STORAGE ADDRESS
                         1473  *               (D,E) = (0,A)
                         1474  *       USES    A,D,E,H,L,F
                         1475
                         1476
000.001                  1477          IF      .RAM.
                         1478          ELSE
000.000                  1479          ERRNZ   *-3047A
                         1480          ENDIF
                         1481
003.047  072 005 040    1482  LRA     LDA     REGI
003.052  137            1483  LRA.    MOV     E,A
003.053  026 000        1484          MVI     D,0
003.055  052 035 040    1485          LHLD    REGPTR
003.060  031            1486          DAD     D               (DE) = (REGPTR)+(REGI)
003.061  311            1487          RET


                         1489  **      IOA - INPUT OCTAL ADDRESS.
                         1490  *
                         1491  *       ENTRY   (H,L) = ADDRESS OF RECEPTION DOUBLE BYTE.
                         1492  *               (D) = TERMINATING CHARACTER
                         1493  *       EXIT    NONE
                         1494  *       USES    A,D,E,H,L,F
                         1495
                         1496
000.001                  1497          IF      .RAM.
                         1498          ELSE
000.000                  1499          ERRNZ   *-3062A
                         1500          ENDIF
                         1501
003.062  303 176 005    1502  IOA     JMP     IOA1
003.065  000            1503          NOP                     RETAIN H8 ORG


                         1505  **      IOB - INPUT OCTAL BYTE.
                         1506  *
                         1507  *       READ ONE OCTAL BYTE FROM THE KEYSET.
                         1508  *
                         1509  *       ENTRY   (H,L) = ADDRESS OF BYTE TO HOLD VALUE
                         1510  *               'C' SET IF FIRST DIGIT IN (A)
                         1511  *       EXIT    NONE
                         1512  *       USES    A,D,E,H,L,F
                         1513
                         1514
000.001                  1515          IF      .RAM.
                         1516          ELSE
```

```
000.000                 1517            ERRNZ   *-3066A
                        1518            ENDIF
                        1519
003.066  066 000        1520  IOB       MVI     M,0             ZERO OUT OLD VALUE
003.070  324 262 003    1521  IOB1      CNC     RCC             READ CONSOLE CHARACTER
                        1522
                        1523  *         SEE IF CHARACTER IS A VALID OCTAL VALUE
                        1524  *
003.073  376 060        1525            CPI     '0'             LESS THAN ZERO?
003.075  332 135 003    1526            JC      IOB2            IF (A) < 0, SEE IF A TERMINATING CHARACTER
003.100  376 070        1527            CPI     '8'             GREATER THAN 7?
003.102  322 070 003    1528            JNC     IOB1            IF TOO LARGE, TRY AGAIN
                        1529
                        1530  *         HAVE AN OCTAL DIGIT
                        1531  *
003.105  315 302 003    1532            CALL    WCC             ECHO CHARACTER
003.110  346 007        1533            ANI     00000111B       MASK FOR BINARY VALUE
003.112  137            1534            MOV     E,A             (E) = VALUE
003.113  176            1535            MOV     A,M             GET OLD VALUE
003.114  007            1536            RLC                     SHIFT 3
003.115  007            1537            RLC
003.116  007            1538            RLC
003.117  303 126 003    1539            JMP     IOB1.5          JUMP AROUND AN H88/H89 TO H8 FAKE ROUTINE
                        1540
                        1541  **        FAKE OUT ROUTINE FOR CALLERS OF *DOD* FROM THE H8 FRONT PANEL
                        1542
                        1543
000.001                 1544            IF      .RAM.
                        1545            ELSE
000.000                 1546            ERRNZ   *-3122A
                        1547            ENDIF
                        1548
003.122  043            1549  DOD       INX     H
003.123  043            1550            INX     H
003.124  043            1551            INX     H
003.125  311            1552            RET
                        1553
                        1554
                        1555  *         CONTINUE
                        1556
003.126  346 370        1557  IOB1.5    ANI     11111000B       TOSS OLD LSB DIGIT
003.130  263            1558            ORA     E               REPLACE WITH NEW VALUE
003.131  167            1559            MOV     M,A
003.132  303 070 003    1560            JMP     IOB1            INPUT ANOTHER CHARACTER
                        1561
                        1562  *         CHECK FOR A CARRIAGE RETURN TO TERMINATE BYTE
                        1563  *
003.135  376 015        1564  IOB2      CPI     A.CR            CARRIAGE RETURN?
003.137  310            1565            RZ                      RETURN IF CARRIAGE RETURN       /JWT 790507/
003.140  257            1566            XRA     A               CLEAR CARRY                     /JWT 790507/
003.141  030 325        1567            JR      IOB1            GET A NEW CHARACTER             /JWT 790507/
```

```
                                1570  **      DYASC - DYNAMIC RAM ASCII OUTPUT TO CONSOLE
                                1571  *
                                1572  *       ENTRY   (A) = CHARACTER TO OUTPUT
                                1573  *               (IY) = RETURN ADDRESS
                                1574  *       EXIT    TO (IY)
                                1575  *       USES    A,C,F
                                1576
    003.143                     1577  DYASC   EQU     *
                                1578  *       EX      AF,AF'          SAVE CHARACTER TO OUTPUT
    003.143  010               1579          DB      MI,EXAF
    003.144  333 355           1580  DYASC1  IN      SC.ACE+UR,LSR   READ LINE STATUS REGISTER
    003.146  346 040           1581          ANI     UC.THE
    003.150  312 144 003       1582          JZ      DYASC1          WAIT IF UART CAN'T HOLD ANOTHER CHARACTER
                                1583
                                1584  *       EX      AF,AF'          GET CHARACTER TO OUTPUT
    003.153  010               1585          DB      MI,EXAF
    003.154  323 350           1586          OUT     SC.ACE+UR,THR   OUTPUT TO UART
                                1587  *       JP      (IY)            RETURN TO CALLER
    003.156  375 351           1588          DB      MI,JIYA,MI,JIYB


                                1590  **      DYBYT - DYNAMIC RAM BYTE OUTPUT
                                1591  *
                                1592  *       ENTRY   (A) = BYTE TO OUTPUT AS OCTAL
                                1593  *               (IX) = RETURN ADDRESS
                                1594  *       EXIT    TO (IX)
                                1595  *       USES    A,C,IY,F
                                1596
    003.160  117               1597  DYBYT   MOV     C,A             SAVE CHARACTER
    003.161  346 300           1598          ANI     11000000B       OUTPUT FIRST CHARACTER OF OCTAL VALUE
    003.163  017               1599          RRC
    003.164  017               1600          RRC
    003.165  017               1601          RRC
    003.166  017               1602          RRC
    003.167  017               1603          RRC
    003.170  017               1604          RRC
    003.171  366 060           1605          ORI     00110000B       MAKE INTO ASCII
                                1606
                                1607  *       LD      IY,DYBYT.2
    003.173  375 041           1608          DB      MI,LDYA,MI,LDYB
    003.175  202 003           1609          DW      DYBYT.2
                                1610
    003.177  303 143 003       1611          JMP     DYASC
                                1612
    003.202  171               1613  DYBYT.2 MOV     A,C             OUTPUT SECOND CHARACTER
    003.203  346 070           1614          ANI     00111000B
    003.205  017               1615          RRC
    003.206  017               1616          RRC
    003.207  017               1617          RRC
    003.210  366 060           1618          ORI     00110000B       MAKE INTO ASCII
                                1619
                                1620  *       LD      IY,DYBYT.4      RETURN ADDRESS
    003.212  375 041           1621          DB      MI,LDYA,MI,LDYB
    003.214  221 003           1622          DW      DYBYT.4
                                1623
```

```
003,216  303 143 003  1624            JMP     DYASC
                      1625
003,221  171          1626  DYBYT,4   MOV     A,C             OUTPUT LAST CHARACTER
003,222  346 007      1627            ANI     00000111B
003,224  366 060      1628            ORI     00110000B       MAKE ASCII
                      1629
                      1630  *         LD      IY,DYBYT.6      RETURN ADDRESS
003,226  375 041      1631            DB      MI.LDYA,MI.LDYB
003,230  235 003      1632            DW      DYBYT.6
                      1633
003,232  303 143 003  1634            JMP     DYASC
                      1635
003,235               1636  DYBYT.6   EQU     *
                      1637  *         JP      (IX)            RETURN TO CALLER
003,235  335 351      1638            DB      MI,JIXA,MI,JIXB


                      1640  **        MSG.PAS - PASS MESSAGE FOR DYNAMIC RAM TEST
                      1641  *
                      1642
003,237  015 012      1643  MSG.PAS   DB      A.CR,A.LF
003,241  040 040 040  1644            DB      '     Pass =    '
003,257  000          1645            DB      0
```

```
003,260                 1648            ORG     3260A
                        1649    **      RCK - READ CONSOLE KEYPAD
                        1650    *
                        1651    *       RCK IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE FRONT PANEL KEYPAD.
                        1652    *       SINCE THE H88/89 DOES NOT HAVE A FRONT PANEL, THIS ROUTINE IS PROVIDED
                        1653    *       ONLY TO MAINTAIN COMPATIBILITY WITH PAM-8.
                        1654    *       RCK WILL IMMEDIATELY RETURN WITH A VALUE OF 0 (ZERO) IN THE ACCUMULATOR.
                        1655    *
                        1656    *       ENTRY   NONE
                        1657    *       EXIT    (A) = 0
                        1658    *       USES    A,F
                        1659
                        1660    *       RCK MUST HAVE SAME ENTRY AS RCK IN PAM-8
000,000                 1661            ERRNZ   *-3260A
                        1662
003,260                 1663    RCK     EQU     *
                        1664
003,260  257           1665            XRA     A
003,261  311           1666            RET
                        1667
```

```
                        1671  **      RCC - READ CONSOLE CHARACTER.
                        1672  *
                        1673  *       RCC IS CALLED TO READ A KEYSTROKE FROM THE CONSOLE.
                        1674  *       IF A RUBOUT/DELETE IS RECEIVED, EXIT IS TO *ERROR*.
                        1675  *
                        1676  *       ENTRY   NONE
                        1677  *       EXIT    TO ERROR  - IF A DELETE OR RUBOUT IS ENCOUNTERED
                        1678  *               TO CALLER - WHEN A KEY IS HIT
                        1679  *               (A) =  ASCII KEY VALUE
                        1680  *       USES    A,F
                        1681
                        1682
                        1683
  003.262               1684  RCC     EQU     *
                        1685
  003.262  333 355      1686  RCC1    IN      SC.ACE+UR.LSR   INPUT ACE LINE STATUS REGISTER
  003.264  346 001      1687          ANI     UC.DR           SEE IF THERE IS A DATA READY
  003.266  050 372      1688          JR      Z,RCC1
                        1689
  003.270  333 350      1690  RCC2    IN      SC.ACE+UR.RBR   ELSE, INPUT CHARACTER
  003.272  346 177      1691          ANI     01111111B       TOSS ANY PARITY
  003.274  376 177      1692          CPI     A.DEL
  003.276  312 322 000  1693          JZ      ERROR           IF RUBOUT, EXIT TO ERROR
                        1694
  003.301  311          1695          RET                     ELSE, EXIT TO CALLER


                        1697  **      WCC - WRITE CONSOLE CHARACTER
                        1698  *
                        1699  *       WRITE A CHARACTER TO THE CONSOLE UART PORT
                        1700  *
                        1701  *       ENTRY   (A) = ASCII CHARACTER TO OUTPUT
                        1702  *       EXIT    NONE
                        1703  *       USES    NONE
                        1704
                        1705
  003.302  365          1706  WCC     PUSH    PSW             SAVE CHARACTER
  003.303  333 355      1707  WCC1    IN      SC.ACE+UR.LSR   INPUT ACE STATUS
  003.305  346 040      1708          ANI     UC.THE          SEE IF TRANSMITTER HOLDING REGISTER IS EMPTY
  003.307  050 372      1709          JR      Z,WCC1
                        1710
  003.311  361          1711          POP     PSW             GET CHARACTER
  003.312  323 350      1712          OUT     SC.ACE+UR.THR           OUTPUT TO CONSOLE
  003.314  311          1713          RET
```

```
                          1715  **      THE FOLLOWING IS ONLY A PORTION OF THE DYNAMIC RAM TEST!!!
                          1716  *
003.315  353             1717  DY9.3   XCHG
003.316  174             1718          MOV     A,H             OUTPUT MSB
                          1719
                          1720  *       LD      IX,DY9.4        RETURN ADDRESS
003.317  335 041         1721          DB      MI.LDXA,MI.LDXB
003.321  326 003         1722          DW      DY9.4
                          1723
003.323  303 160 003     1724          JMP     DYBYT
                          1725
003.326  175             1726  DY9.4   MOV     A,L             OUTPUT LSB
                          1727
                          1728  *       LD      IX,DY9.5        RETURN ADDRESS
003.327  335 041         1729          DB      MI.LDXA,MI.LDXB
003.331  335 003         1730          DW      DY9.5
                          1731
003.333  030 223         1732          JR      DYBYT
                          1733
003.335  353             1734  DY9.5   XCHG                    SAVE ERROR ADDRESS
003.336  041 362 007     1735          LXI     H,MSG.EQ        OUTPUT ' = '
                          1736
                          1737  *       LD      IX,DY9.8        RETURN ADDRESS
003.341  335 041         1738          DB      MI.LDXA,MI.LDXB
003.343  350 003         1739          DW      DY9.8
                          1740
003.345  303 306 007     1741          JMP     DYMSG           OUTPUT STRING
                          1742
003.350  032             1743  DY9.8   LDAX    D               OUTPUT RAM CONTENTS
                          1744
                          1745  *       LD      IX,DYMEM10      RETURN ADDRESS
003.351  335 041         1746          DB      MI.LDXA,MI.LDXB
003.353  360 003         1747          DW      DYMEM10
                          1748
003.355  303 160 003     1749          JMP     DYBYT
                          1750
003.360  076 007         1751  DYMEM10 MVI     A,A.BEL         DING BELL
                          1752
                          1753  *       LD      IY,DY10.5       RETURN ADDRESS
003.362  375 041         1754          DB      MI.LDYA,MI.LDYB
003.364  265 007         1755          DW      DY10.5
                          1756
003.366  303 143 003     1757          JMP     DYASC
```

```
                        1760  **      IO ROUTINES TO BE COPIED INTO AND USED IN RAM.
                        1761  *
                        1762  *       MUST CONTINUE TO 3777A FOR PROPER COPY.
                        1763  *       THE TABLE MUST ALSO BE BACKWARDS TO THE FINAL RAM
                        1764
    000.001             1765          IF      .RAM.
                        1767          ELSE
    000.000             1768          ERRNZ   4000A-7-*
                        1769          ENDIF
                        1770
    003.371             1771  PRSROM  EQU     *
    003.371  001       1772          DB      1               REFIND
    003.372  000       1773          DB      0               CTLFLG
    003.373  000       1774          DB      0               .MFLAG
    003.374  000       1775          DB      0               DSPMOD
    003.375  000       1776          DB      0               DSPROT
    003.376  012       1777          DB      10              REGI
    003.377  311       1778          DB      MI.RET
                        1779
    000.001             1780          IF      .RAM.
                        1782          ELSE
    000.000             1783          ERRNZ *-4000A
                        1784          ENDIF
                        1785
```

```
                        1788  ***     INITOX          EXTENSION OF INITO TO SUPPORT H88
                        1789
004.000  076 002       1790  INITOX  MVI     A,H88B.CK               ENABLE CLOCK
004.002  323 362       1791          OUT     H88.CTL
                        1792
                        1793  *       SET UP ACE FOR CONSOLE COMMUNICATIONS
                        1794  *
004.004  076 200       1795          MVI     A,UC.DLA        SET DIVISOR LATCH ACCESS BIT
004.006  323 353       1796          OUT     SC.ACE+UR.LCR
004.010  041 101 004   1797          LXI     H,BRTAB         (H,L) = BEGINNING OF BAUD RATE TABLE
004.013  333 362       1798          IN      H88.SW          INPUT SWITCHES FOR DESIRED BAUD RATE
004.015  346 100       1799          ANI     H88S.BR         MASK FOR BAUD RATE SWITCHES ONLY
004.017  017           1800          RRC                     SHIFT FOR A *2 FOR TABLE
004.020  017           1801          RRC
004.021  017           1802          RRC
004.022  017           1803          RRC
004.023  017           1804          RRC
004.024  205           1805          ADD     L               ADD DISPLACEMENT FROM BEGINNING OF TABLE
004.025  157           1806          MOV     L,A
004.026  176           1807          MOV     A,M             GET MSB OF DIVISOR
004.027  323 351       1808          OUT     SC.ACE+UR.DLM
004.031  043           1809          INX     H               GET LSB
004.032  176           1810          MOV     A,M
004.033  323 350       1811          OUT     SC.ACE+UR.DLL
004.035  076 003       1812          MVI     A,UC.8BW        SET 8 BITS, 1 STOP BIT, NO PARITY
004.037  323 353       1813          OUT     SC.ACE+UR.LCR
004.041  076 000       1814          MVI     A,0             SET NO INTERRUPTS
004.043  323 351       1815          OUT     SC.ACE+UR.IER
                        1816
                        1817  *       WAIT A WHILE TO ALLOW THE CONSOLE RESET TO FINISH SO IT CAN
                        1818  *       ACCEPT THE FIRST PROMPT
                        1819  *
004.045  001 000 065   1820          LXI     B,65000A        APPROX. 100 MS
004.050  015           1821  INITOX1 DCR     C
004.051  040 375       1822          JR      NZ,INITOX1
                        1823
004.053  020 373       1824          DJNZ    INITOX1
                        1825
                        1826  *       INPUT SWITCH TO SEE IF TO BEGIN OPERATION OR MEMORY TEST
                        1827  *
004.055  333 362       1828          IN      H88.SW          GET SWITCHES
004.057  346 040       1829          ANI     H88S.M          MASK FOR MEMORY TEST ONLY
004.061  312 116 007   1830          JZ      DYMEM           IF TO PERFORM MEMORY TESTS
                        1831
                        1832  *       REPLACE WHAT WAS ORIGINALLY AT THE JUMP WHICH GOT US HERE
                        1833  *
004.064  021 371 003   1834          LXI     D,PRSROM        (DE) = ROM COPY OF PRS CODE
004.067  257           1835          XRA     A
004.070  062 123 041   1836          STA     AUTOB           INITIAL AUTO BOOT FLAG
004.073  062 066 040   1837          STA     DATA            INITIAL 362Q PORT DATA SAVE BYTE
004.076  303 003 000   1838          JMP     INITO.0         RETURN TO ORIGINAL CODE
```

```
                        1840  **      BRTAB - BAUD RATE DIVISOR TABLE
                        1841  *
    004.101             1842  BRTAB   EQU     *
                        1843
    004.101 000 014     1844  BR96    DB      0,12            9600 BAUD
    004.103 000 006     1845  BR19.2  DB      0,6           19,200 BAUD
                        1846  *BR38.4 DB       0,3          38,400 BAUD
                        1847  *BR56.0 DB       0,2          56,000 BAUD
                        1848
    000.004             1849  .       SET     */256
    000.000             1850          ERRNZ   BRTAB/256-.   TABLE MUST BE IN ONE PAGE


                        1852  ***     SAVALLX - SAVALL EXTENSION TO MAKE ROOM FOR A JUMP TO THE NMI HANDLER
                        1853
    004.105             1854  SAVALLX EQU     *             REPLACE OLD CODE
    004.105 345         1855          PUSH    H             SET ON STACK AS 'REGISTER'
    004.106 325         1856          PUSH    D             SET RETURN ADDRESS
    004.107 021 011 040 1857          LXI     D,CTLFLG
    004.112 032         1858          LDAX    D
    004.113 303 151 000 1859          JMP     SAVALLR       RETURN TO OLD CODE
```

```
                        1862  ****    NMI - NON MASKABLE INTERRUPT
                        1863  *
                        1864  *       NMI IS USED AS THE TRAP FOR ALL ILLEGAL PORT REQUESTS
                        1865  *
                        1866  *       PORT ADDRESSES TRAPPED ARE:
                        1867  *
                        1868  *                       IN 360Q FRONT PANEL KEYBOARD INPUT
                        1869  *                      OUT 360Q FRONT PANEL CONTROL
                        1870  *                      OUT 361Q FRONT PANEL DISPLAY CONTROL
                        1871  *                   IN/OUT 372Q CONSOLE DATA FOR AN 8251A
                        1872  *                      OUT 373Q CONSOLE CONTROL FOR AN 8251A
                        1873  *
                        1874  *
                        1875  *               THESE PORT REQUESTS ARE RESPONDED TO AS FOLLOWS:
                        1876  *
                        1877  *                       IN 360Q RETURNS WITH (A) = 377Q TO SHOW THAT
                        1878  *                                NO FRONT PANEL SWITCHES ARE PRESSED
                        1879  *
                        1880  *                       OUT 360Q MOVES BIT 6 (CB.CLI) TO BIT 1, AND
                        1881  *                                BIT 4 (CB.SSI) INVERTED, TO BIT 0, AND
                        1882  *                                OUTPUTS THESE BITS TO PORT 362Q TO
                        1883  *                                CONTROL THE CLOCK AND SINGLE STEP INTERRUPTS
                        1884  *
                        1885  *                       OUTPUTS TO 361Q, 372Q, AND 373Q JUST RETURN
                        1886  *
                        1887  *                       INPUTS FROM 361Q, 372Q, AND 373Q RETURN WITH (A) = 0
                        1888  *                                TO INDICATE AN EMPTY BUSS
                        1889  *
                        1890  *
                        1891  *               ENTRY   NONE
                        1892  *
                        1893  *               EXIT    NONE
                        1894  *
                        1895  *               USES    (A) ONLY IF 'FAKING' AN INPUT
                        1896
                        1897
004.116  343           1898  NMI     XTHL                    GET RETURN ADDRESS FROM STACK
004.117  042 064 040    1899          SHLD    NMIRET          SAVE FOR LATER USE
004.122  343           1900          XTHL                    PUT RETURN ADDRESS BACK ON STACK
                        1901
004.123  345           1902          PUSH    H               SAVE REGISTERS
004.124  305           1903          PUSH    B
004.125  365           1904          PUSH    PSW
004.126  107           1905          MOV     B,A             SAVE (A) PRIOR TO I/O
004.127  052 064 040    1906          LHLD    NMIRET          GET RETURN ADDRESS
004.132  053           1907          DCX     H               BACK UP TO PORT # WHICH GOT US HERE
004.133  176           1908          MOV     A,M             GET PORT #
                        1909
004.134  376 360       1910          CPI     360Q            PORT 360?
004.136  050 033       1911          JR      Z,NMI1            IF PORT WAS 360Q
                        1912
                        1913  *       PORT REFERENCED WAS 361Q, 372Q, OR 373Q
                        1914  *
004.140  376 361       1915          CPI     361Q            MAKE SURE PORT IS LEGAL
004.142  050 010       1916          JR      Z,NMI0.5          IF LEGAL
                        1917
```

```
004.144  376 372      1918           CPI     372Q
004.146  050 004      1919           JR      Z,NMI0.5
                      1920
004.150  376 373      1921           CPI     373Q
004.152  040 062      1922           JR      NZ,NMI2.5       IF NONE OF THE ABOVE, EXIT
                      1923
004.154  053          1924  NMI0.5   DCX     H               POINT TO IN/OUT INSTRUCTION
004.155  176          1925           MOV     A,M             SEE IF INPUT OR OUTPUT
004.156  376 323      1926           CPI     MI.OUT
004.160  050 054      1927           JR      Z,NMI2.5          IF OUTPUT, JUST EXIT
                      1928
004.162  376 333      1929           CPI     MI.IN
004.164  040 050      1930           JR      NZ,NMI2.5         IF NOT INPUT EITHER, ILLEGAL SO EXIT
                      1931
004.166  361          1932           POP     PSW             RESTORE FLAGS
004.167  076 000      1933           MVI     A,0             ELSE, RETURN LIKE AN EMPTY BUSS
004.171  030 044      1934           JR      NMI3            EXIT
                      1935
004.173  053          1936  NMI1     DCX     H               POINT TO IN/OUT INSTRUCTION
004.174  176          1937           MOV     A,M             GET I/O INSTRUCTION
004.175  376 333      1938           CPI     MI.IN           INPUT?
004.177  040 005      1939           JR      NZ,NMI1.5         IF NOT 'IN'
                      1940
004.201  361          1941           POP     PSW             RESTORE FLAGS
004.202  076 377      1942           MVI     A,11111111B     SHOW 'NO KEYS PRESSED'
004.204  030 031      1943           JR      NMI3            EXIT
                      1944
004.206  376 323      1945  NMI1.5   CPI     MI.OUT          MAKE SURE INTRUCTION IS AN 'OUT'
004.210  040 024      1946           JR      NZ,NMI2.5         IF NOT
                      1947
004.212  170          1948  NMI2     MOV     A,B             GET OUTPUT DATA AGAIN
004.213  346 120      1949           ANI     CB.CLI+CB.SSI   MOVE CLOCK INFO TO BIT 1
004.215  017          1950           RRC
004.216  017          1951           RRC
004.217  017          1952           RRC
004.220  017          1953           RRC
004.221  017          1954           RRC
004.222  070 001      1955           JR      C,NMI2.2
004.224  074          1956           INR     A
004.225  041 066 040  1957  NMI2.2   LXI     H,DATA          OR WITH THE BYTE IN RAM
004.230  266          1958           ORA     M               BEFORE OUTPUT IT
004.231  323 362      1959           OUT     H88.CTL         SET IN HARDWARE
004.233  346 374      1960           ANI     11111100B
004.235  167          1961           MOV     M,A
                      1962
004.236  361          1963  NMI2.5   POP     PSW             RESTORE (A,F)
                      1964
004.237  301          1965  NMI3     POP     B
004.240  341          1966           POP     H
                      1967  *        RETN                    Z80 RETURN FROM NMI
004.241  355 105      1968           DB      355Q,105Q
```

```
                        1971  **      ATB     - AUTO BOOT ROUTINE CONTINUE
                        1972
004,243  167           1973  ATB     MOV     M,A             SET AUTO BOOT FLAG
004,244  076 012       1974          MVI     A,10            SET TO AUTO BOOT ROUTINE
004,246  315 052 003   1975          CALL    LRA.
004,251  021 212 001   1976          LXI     D,AUTOBO        SET AUTO BOOT ROUTINE
004,254  030 016       1977          JR      BOOTX


004,256                 1979          ORG     4256A
                        1980  **      BOOT H-17 OR Z47 ENTRY POINT FOR H88
                        1981  *
                        1982  *       ENTRY   NONE
                        1983  *
                        1984  *       EXIT    (DE) = NORMAL BOOT ROUTINE ADDRESS
                        1985  *
                        1986  *       USES    ALL
                        1987
004,256  041 234 006   1988  BOOT    LXI     H,MSG.BT        COMPLETE BOOT MESSAGE
004,261  315 100 006   1989          CALL    TYPMSG
004,264  076 012       1990          MVI     A,10
004,266  315 052 003   1991          CALL    LRA.            GET LOCATION OF USER PC
004,271  021 261 001   1992          LXI     D,NBOOT         SET ITS VALUE TO THE NORMAL BOOT ROUTINE
004,274  163           1993  BOOTX   MOV     M,E
004,275  043           1994          INX     H
004,276  162           1995          MOV     M,D
                        1996
004,277  303 063 000   1997          JMP     GO.             DO IT
```

```
                        2000  **      TMOUT   -  BOOT CODE TIME OUT ROUTINE
                        2001  *
                        2002  *       TMOUT IS ENTERED FROM TIMER INTERRUPT EVER 100 MS. AND IT WILL
                        2003  *       EXIT:   IF BOOT SUCCESS THEN TIMER OFF.
                        2004  *               IF 15 SECONDS TIME OUT AND BOOT IS NOT SUCCESS YES
                        2005  *                  THEN ABORT BOOT Z47 & TO MONITOR LOOP
                        2006  *               IF < 15S & 3.5S THEN RE-BOOT
                        2007  *
                        2008  *       ENTRY:  (TMFG)  = 1 IF THE TIME OUT IS FOR Z47
                        2009  *                       = 0 IF THE TIME OUT IS FOR H17
                        2010  *       EXIT:   NONE
                        2011  *
                        2012  *       USE:    ALL (WHEN RETURN, ALL REGISTERS ARE RESTORED)
                        2013
004.302                 2014  TMOUT   EQU     *
004.302  333 355        2015          IN      SC.ACE+UR.LSR           INPUT ACE LINE STATUS REGISTER
004.304  346 001        2016          ANI     UC.DR                   SEE IF THERE IS A DATA READY
004.306  050 011        2017          JR      Z,TMOUT4                CHECK IF IT IS <DELETE>
                        2018
004.310  333 350        2019          IN      SC.ACE+UR.RBR           INPUT DATA FROM KB
004.312  346 177        2020          ANI     01111111B               IS IT <DEL>?
004.314  376 177        2021          CPI     A.DEL
004.316  312 171 002    2022          JZ      NODEV                   IF IT, ABORT THE BOOT
                        2023  *                                       ELSE IGNORE THE INPUT
004.321  041 121 041    2024  TMOUT4  LXI     H,TMFG
004.324  176            2025          MOV     A,M
004.325  247            2026          ANA     A
004.326  010            2027          DB      MI.EXAF                 SAVE Z FLAG
004.327  072 033 040    2028          LDA     TICCNT                  GET TIC
004.332  247            2029          ANA     A                       SET ZERO FLAG
004.333  040 024        2030          JR      NZ,TMOUT2               NOT IN 0.5 SECOND
004.335  043            2031          INX     H                       SET TO MYCNT
000.000                 2032          ERRNZ   MYCNT-TMFG-1            MYCNT MUST FOLLOW TMFG
004.336  064            2033          INR     M                       INCREASE THE COUNT FOR 0.5 SECOND
004.337  176            2034          MOV     A,M
004.340  376 036        2035          CPI     30                      CHECK IF MORE THAN 15 SECONDS
004.342  322 171 002    2036          JNC     NODEV                   NO DEVICE ?
004.345  336 007        2037  TMOUT1  SBI     7                       IS IT 3.5 SECONDS?
004.347  070 010        2038          JR      C,TMOUT2                IF NOT, WAIT
004.351  040 372        2039          JR      NZ,TMOUT1               CHECK MORE
004.353  010            2040          DB      MI.EXAF
004.354  302 063 002    2041          JNZ     RETRY                   IF IT IS Z47, THEN RE-BOOT
004.357  030 002        2042          JR      TMOUT3                  IT IS H-17, CONTINUE IT CLOCK ROUTINE
004.361  010            2043  TMOUT2  DB      MI.EXAF                 CHECK IT IS Z47 OR H17
004.362  300            2044          RNZ                             Z47, THEN RETURN
004.363  303 031 034    2045  TMOUT3  JMP     CLOCK17                 CONTINUE H17 CLOCK ROUTINE
```

```
004.370                 2048            ORG     4370A
                        2049    **      SUBM - SUBSTITUTE MEMORY
                        2050    *
                        2051    *       SUBM INPUTS A MEMORY ADDRESS FROM THE CONSOLE AND THEN DISPLAYS
                        2052    *       THAT ADDRESS AND ITS CONTENTS.  IF A CARRIAGE RETURN IS THEN TYPED,
                        2053    *       CONTROL RETURNS TO THE MONITOR.  IF A SPACE IS TYPED, THE NEXT
                        2054    *       MEMORY LOCATION AND CONTENTS ARE DISPLAYED.  IF A MINUS SIGN IS
                        2055    *       TYPED, THE PREVIOUS MEMORY LOCATION AND CONTENTS ARE DISPLAYED.
                        2056    *       IF AN OCTAL CHARACTER IS TYPED, A BYTE IS ENTERED AN PLACED AT THE
                        2057    *       CURRENT MEMORY LOCATION.
                        2058    *
                        2059    *
                        2060    *       ENTRY   NONE
                        2061    *       EXIT    NONE
                        2062    *       USES    A,E,H,L,F
                        2063
                        2064
004.370 041 201 006     2065    SUBM    LXI     H,MSG.SUB       COMPLETE SUBSTITUTE MESSAGE
004.373 315 100 006     2066            CALL    TYPMSG
004.376 315 150 005     2067            CALL    IROC            INPUT FIRST CHARACTER
005.001 320             2068            RNC                     IF A RETURN, EXIT
                        2069
005.002 041 003 040     2070            LXI     H,IOWRK+1       ELSE, INPUT STARTING ADDRESS
005.005 026 015         2071            MVI     D,A.CR          ENDING WITH A RETURN
005.007 315 062 003     2072            CALL    IOA
005.012 353             2073            XCHG                    (H,L) = INPUT ADDRESS
                        2074
005.013 315 313 005     2075    SUBM1   CALL    TOA             TYPE CRLF, ADDRESS, AND A SPACE
005.016 176             2076            MOV     A,M             GET MEMORY CONTENTS FOR DISPLAY
005.017 315 343 005     2077            CALL    TOB
005.022 076 040         2078            MVI     A,' '           SPACE
005.024 315 302 003     2079            CALL    WCC
                        2080
005.027 315 301 005     2081    SUBM2   CALL    IOC             INPUT FIRST CHARACTER
005.032 322 075 005     2082            JNC     SUBM7           IF FIRST CHARACTER IS OCTAL
                        2083
005.035 376 040         2084            CPI     ' '             SPACE?
005.037 302 046 005     2085            JNZ     SUBM4           IF NOT A SPACE
                        2086
005.042 043             2087    SUBM3   INX     H               POINT TO NEXT ADDRESS
005.043 303 013 005     2088            JMP     SUBM1           DISPLAY NEXT
                        2089
005.046 376 055         2090    SUBM4   CPI     '-'             MINUS?
005.050 302 062 005     2091            JNZ     SUBM6           IF NOT
                        2092
005.053 315 302 003     2093    SUBM5   CALL    WCC             ECHO HYPHEN
005.056 053             2094            DCX     H               POINT TO PREVIOUS ADDRESS
005.057 303 013 005     2095            JMP     SUBM1           DISPLAY PREVIOUS
                        2096
005.062 376 015         2097    SUBM6   CPI     A.CR            RETURN?
005.064 310             2098            RZ                      IF RETURN, EXIT
                        2099
005.065 076 007         2100            MVI     A,A.BEL         ELSE, DING BELL
005.067 315 302 003     2101            CALL    WCC
005.072 303 027 005     2102            JMP     SUBM2           TRY AGAIN
                        2103
```

```
005.075 066 000      2104  SUBM7  MVI   M,0             ZERO BYTE TO BE BUILT
                     2105
005.077 315 302 003  2106  SUBM8  CALL  WCC             ECHO OCTAL CHARACTER
005.102 346 007      2107         ANI   00000111B       GET BINARY VALUE
005.104 137          2108         MOV   E,A             SAVE PARTIAL
005.105 176          2109         MOV   A,M             GET CURRENT
005.106 007          2110         RLC                   MAKE ROOM FOR NEW CHARACTER
005.107 007          2111         RLC
005.110 007          2112         RLC
005.111 346 370      2113         ANI   11111000B       TOSS PREVIOUS LSB
005.113 263          2114         ORA   E               ADD NEW
005.114 167          2115         MOV   M,A             SAVE NEW TOTAL
005.115 315 301 005  2116  SUBM9  CALL  IOC             INPUT NEXT CHARACTER
005.120 322 077 005  2117         JNC   SUBM8           IF OCTAL
                     2118
005.123 376 040      2119         CPI   ' '             SPACE?
005.125 312 042 005  2120         JZ    SUBM3           IF SPACE, DISPLAY NEXT BYTE
                     2121
005.130 376 055      2122         CPI   '-'             MINUS?
005.132 312 053 005  2123         JZ    SUBM5           IF MINUS, DISPLAY PREVIOUS
                     2124
005.135 376 015      2125         CPI   A.CR            RETURN?
005.137 310          2126         RZ                    IF RETURN, EXIT
                     2127
005.140 076 007      2128         MVI   A,A.BEL         ELSE, DING BELL
005.142 315 302 003  2129         CALL  WCC
005.145 303 115 005  2130         JMP   SUBM9           TRY AGAIN
```

```
                     2133  **     IROC - INPUT A RETURN OR AN OCTAL CHARACTER
                     2134  *
                     2135  *      IROC INPUTS A CHARACTER FROM THE CONSOLE AND WAITS UNTIL IT
                     2136  *      RECEIVES EITHER A VALID OCTAL CAHRACTER OR A CARRIAGE RETURN
                     2137  *
                     2138  *      ENTRY  NONE
                     2139  *      EXIT   (A) = INPUT CHARACTER
                     2140  *             'C' = SET IF CHARACTER IS OCTAL
                     2141  *      USES   A,F
                     2142
                     2143
005.150 315 262 003  2144  IROC   CALL  RCC             INPUT CHARACTER
005.153 376 015      2145         CPI   A.CR            RETURN?
005.155 310          2146         RZ                    IF A CR
                     2147
005.156 376 060      2148         CPI   '0'             < 0?
005.160 332 166 005  2149         JC    IROC1           IF < OCTAL
                     2150
005.163 376 070      2151         CPI   '8'             > 8?
005.165 330          2152         RC                    IF OCTAL
                     2153
005.166 076 007      2154  IROC1  MVI   A,A.BEL         ELSE, RING BELL
005.170 315 302 003  2155         CALL  WCC
005.173 303 150 005  2156         JMP   IROC            TRY AGAIN
```

```
                        2158  **      IOA1 - INPUT OCTAL ADDRESS
                        2159  *
                        2160  *       IOA1 IS A CONTINUATION OF *IOA* AND INPUTS A SPLIT OCTAL ADDRESS
                        2161  *       WITHOUT REQUIRING LEADING ZEROS
                        2162  *
                        2163  *       ENTRY   (H,L) = ADDRESS + 1 WHERE INPUT ADDRESS IS TO BE PLACED
                        2164  *               (A) = FIRST OCTAL CHARACTER IF 'C' IS SET
                        2165  *       EXIT    (D,E) = INPUT ADDRESS
                        2166  *               (A) = LAST INPUT CHARACTER
                        2167  *       USES    A,D,E,H,L,F
                        2168
                        2169
005,176  305           2170  IOA1    PUSH    B               SAVE (B,C)
005,177  102           2171          MOV     B,D             (B) = TERMINATION CHARACTER
005,200  345           2172          PUSH    H               SAVE ADDRESS WHERE INPUT IS TO BE PLACED
005,201  041 000 000    2173          LXI     H,0             SET NEW VALUE TO ZERO
005,204  324 262 003    2174  IOA2    CNC     RCC             IF CARRY SET, FIRST CHARACTER IS IN ACC
005,207  376 060        2175          CPI     '0'             MAKE SURE CHARACTER IS OCTAL
005,211  332 242 005    2176          JC      IOA3            IF < OCTAL
                        2177
005,214  376 070        2178          CPI     '8'
005,216  322 242 005    2179          JNC     IOA3            IF > OCTAL
                        2180
005,221  315 302 003    2181          CALL    WCC             ECHO OCTAL CHARACTER
005,224  346 007        2182          ANI     00000111B       GET BINARY VALUE
005,226  365           2183          PUSH    PSW             SAVE NEW CHARACTER VALUE
005,227  051           2184          DAD     H               SHIFT THREE TO MAKE ROOM FOR NEW CHARACTER
005,230  051           2185          DAD     H
005,231  051           2186          DAD     H
005,232  365           2187          PUSH    PSW             SAVE CARRY FROM DAD
005,233  321           2188          POP     D               SAVE FLAG RESULT IN E
005,234  361           2189          POP     PSW             RETURN NEW CHARACTER VALUE TO (A)
005,235  205           2190          ADD     L
005,236  157           2191          MOV     L,A
005,237  303 204 005    2192          JMP     IOA2            SEE IF MORE CHARACTERS
                        2193
005,242  270           2194  IOA3    CMP     B               TERMINATING CHARACTER?
005,243  312 260 005    2195          JZ      IOA4            IF EQUAL
                        2196
005,246  076 007        2197          MVI     A,A.BEL         ELSE, DING BELL
005,250  315 302 003    2198          CALL    WCC
005,253  067           2199          STC                     TRY AGAIN
005,254  077           2200          CMC
005,255  303 204 005    2201          JMP     IOA2
                        2202
                        2203  *       END OF INPUT, PUT VALUE IN MEMORY AND EXIT
                        2204
005,260  315 302 003    2205  IOA4    CALL    WCC             ECHO CHARACTER
005,263  127           2206          MOV     D,A             LAST CHARACTER TO D
005,264  325           2207          PUSH    D
005,265  361           2208          POP     PSW             (PSW) = RESULT OF DAD
005,266  174           2209          MOV     A,H             MAKE (H) INTO SPLIT OCTAL
005,267  037           2210          RAR
005,270  147           2211          MOV     H,A
005,271  172           2212          MOV     A,D             RESTORE LAST INPUT CHARACTER
005,272  353           2213          XCHG                    (D,E) = INPUT ADDRESS
```

```
005.273  341          2214          POP     H               (H,L) = LOCATION TO PLACE THIS ADDRESS
005.274  162          2215          MOV     M,D
005.275  053          2216          DCX     H
005.276  163          2217          MOV     M,E
005.277  301          2218          POP     B               RESTORE (B,C)
005.300  311          2219          RET


                      2221  **      IOC - INPUT OCTAL CHARACTER
                      2222  *
                      2223  *
                      2224  *       ENTRY   NONE
                      2225  *       EXIT    (A) = INPUT CHARACTER
                      2226  *               'C' = SET IF CHARACTER NOT OCTAL
                      2227  *       USES    A,F
                      2228
                      2229
005.301  315 262 003  2230  IOC     CALL    RCC             INPUT CHARACTER
005.304  376 060      2231          CPI     '0'
005.306  330          2232          RC                      IF CHARACTER < OCTAL
                      2233
005.307  376 070      2234          CPI     '8'             CHARACTER > OCTAL?
005.311  077          2235          CMC                     'C' IF GREATER THAN
005.312  311          2236          RET


                      2238  **      TOA - TYPE OCTAL ADDRESS
                      2239  *
                      2240  *       TOA OUTPUTS TO THE CONSOLE A CRLF, THE SPCECIFIED ADDRESS AND A SPACE
                      2241  *
                      2242  *       ENTRY   (H,L) = ADDRESS TO BE DISPLAYED
                      2243  *       EXIT    NONE
                      2244  *       USES    A,B,C,F
                      2245
                      2246
005.313  076 015      2247  TOA     MVI     A,A.CR          CRLF
005.315  315 302 003  2248          CALL    WCC
005.320  076 012      2249          MVI     A,A.LF
005.322  315 302 003  2250          CALL    WCC
                      2251
005.325  174          2252  TOA.    MOV     A,H             ADDRESS
005.326  315 343 005  2253          CALL    TOB
005.331  175          2254          MOV     A,L
005.332  315 343 005  2255          CALL    TOB
                      2256
005.335  076 040      2257          MVI     A,' '           SPACE
005.337  315 302 003  2258          CALL    WCC
005.342  311          2259          RET
```

```
                                2261  **      TOB - TYPE OCTAL BYTE
                                2262  *
                                2263  *       TOB OUTPUTS TO THE CONSOLE IN OCTAL, THE BYTE IN A
                                2264  *
                                2265  *       ENTRY   (A) = BYTE TO BE OUTPUT
                                2266  *       EXIT    NONE
                                2267  *       USES    A,F
                                2268
                                2269
005.343  305                   2270  TOB     PUSH    B
005.344  006 002               2271          MVI     B,2             NUMBER OF CHARACTERS - 1
005.346  117                   2272          MOV     C,A             SAVE ORIGINAL BYTE
005.347  267                   2273          ORA     A               ASSURE 'C' = ZERO
005.350  037                   2274          RAR
005.351  037                   2275          RAR                     SHIFT TOP BYTE TO LSB
005.352  037                   2276          RAR
005.353  037                   2277  TOB1    RAR                     SHIFT MIDDLE BYTE TO LSB
005.354  037                   2278          RAR
005.355  037                   2279          RAR
005.356  346 007               2280          ANI     00000111B       MASK FOR HALF ASCII
005.360  366 060               2281          ORI     00110000B       MAKE WHOLE ASCII
005.362  315 302 003           2282          CALL    WCC             OUTPUT TO CONSOLE
005.365  171                   2283          MOV     A,C             GET ORIGINAL BYTE
005.366  005                   2284          DCR     B
005.367  302 353 005           2285          JNZ     TOB1            IF SECOND BYTE STILL NEEDS TO BE OUTPUT
                                2286
005.372  346 007               2287          ANI     00000111B       ELSE, OUTPUT LAST CHARACTER
005.374  366 060               2288          ORI     00110000B
005.376  315 302 003           2289          CALL    WCC
006.001  301                   2290          POP     B
006.002  311                   2291          RET
```

```
                                2293  **      WCR - WAIT FOR A CARRIAGE RETURN
                                2294  *
                                2295  *       WCR INPUTS CHARACTERS FROM THE CONSOLE UNTIL A CARRIAGE RETURN
                                2296  *       IS RECEIVED AND THEN ECHOS A CRLF
                                2297  *
                                2298  *
                                2299  *       ENTRY   NONE
                                2300  *       EXIT    NONE
                                2301  *       USES    A,F
                                2302
                                2303
006.003  315 262 003           2304  WCR     CALL    RCC             INPUT CHARACTER
006.006  376 015               2305          CPI     A,CR
006.010  040 371               2306          JR      NZ,WCR            IF NOT A CR
                                2307
006.012  315 302 003           2308          CALL    WCC             ELSE, ECHO CR
006.015  076 012               2309          MVI     A,A.LF          LINE FEED
006.017  315 302 003           2310          CALL    WCC
006.022  311                   2311          RET
```