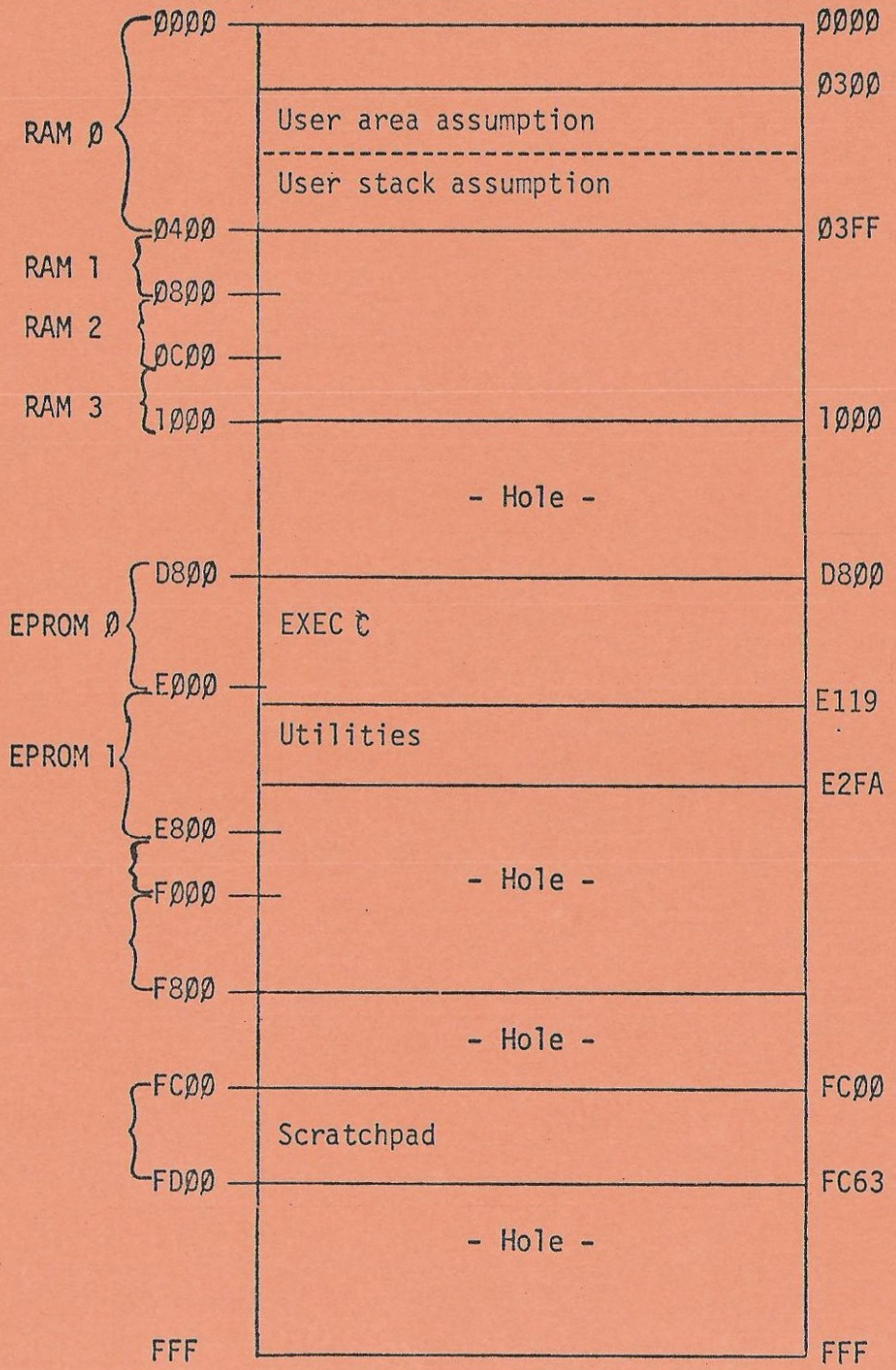
MMD-2

EXEC C Source Listing

801-0229
REV. A
4-81

E&L Instruments Incorporated
61 First Street , Derby, Connecticut 06418

GENERAL MEMORY MAP FOR EXEC C (SOFTWARE)



SOURCE LISTING

(Blank Page)

```
LCTCHNB330
>>ASML

 PASS #1
ORG330
INITB
EXECB1
EXECB2
EXECB3
EXECB4
CASS
PROM
TYIN
TYOUT
SCRATCHB

 PASS #2
ORG330
0000                30              ;FILE: ORG330
0000                31              ;
0000                32              ;
0000                33              ;
0000                50              ;THIS FILE, WHEN ASSEMBLED WITH THE REST OF EXEC-B,

0000                60              ;SETS THE ORIGIN OF EXEC-B TO 330 HI (PROM).  SCRAT
CHPAD IS 374.
```

```
0000                    70              ;
0000                    80              ;
0000                    900                             ;
0000                    950                             ;
0000                    1000                            ;FIELD DEFINITIONS FOLLOW
0000                    1050            FA0:    EQU     330000Q ;ORIGIN OF EXEC
0000                    1100            FSCR:   EQU     375000Q ;LAST ADD OF SCRATCHPAD+1
0000                    1150            FSCRHI:EQU      374Q ;HI SCRATCH PAD FIELD
0000                    1200            F0:     EQU     330Q ;RELATIVE FIELD 0 FOR EXEC
0000                    1250            F1:     EQU     F0+001Q ;RELATIVE FIELD 1 (AND SO
ON)
0000                    1300            F2:     EQU     F1+001Q
0000                    1350            F3:     EQU     F2+001Q
0000                    1400            F4:     EQU     F3+001Q
0000                    1450            F5:     EQU     F4+001Q
0000                    1500            F6:     EQU     F5+001Q
0000                    1550            F7:     EQU     F6+001Q
0000                    1600            F10:    EQU     F7+001Q
0000                    1650                            ;
0000                    1700                            ;
0000                    1750                            ;
0000                    1760            KE1:    EQU     013235Q
0000                    1770            KE2:    EQU     000006Q
INITB
0000                    10              ;THIS FILE CONTAINS NO INSTRUCTIONS - JUST EQU'S.
ALL
0000                    20              ;PROGRAM MASKS, CONSTANTS, ETC.  ARE DEFINED HERE
0000                    30              ;
0000                    40              ;
0000                    50              ;
0000                    100             ;FILE: INITB
0000                    140             ;
0000                    160             ;
0000                    180             ;VARIOUS DEFAULT CONDITIONS, CONSTANTS, MASKS, ETC.
 FOLLOW
0000                    200             ;
0000                    220             ;
0000                    240     DIPBASE:        EQU     00100000B ;MASK FOR HEX/OCT SWITCH

0000                    260     DIPLEDS:        EQU     00001000B ;MASK FOR BINARY/PORTS S
```

```
WITCH
0000                280    DIPPUPRES:    EQU       00010000B ;MASK FOR RESET/PUP SW: )
CH
0000                300    BITPUP:EQU    100Q      ;MASK FOR PUP BIT ON PRTPUP
0000                320    DEFLTADD:     EQU       0030000Q ;DEFAULT HL VAR'S ON POW( )
 UP (PUP)
0000                340    DEFLTUSP:     EQU       003376Q ;DEFUALT USER STACK POINTE
R
0000                360    LONG:  EQU    004011Q   ;TIMER CONSTANTS FOR CASSET UTILIT
Y
0000                380    MEDIUM:EQU    010010Q
0000                400    SHORT: EQU    010004Q
0000                420           ;
0000                440           ;
0000                460           ;PORT DEFINITIONS FOLLOW
0000                480           ;
0000                500           ;
0000                520           ;7-SEG CHAR. DEF'S FOLLOW:
0000                525           ;NOTE THAT 'W/DOT' MEANS CHARACTER INCLUDES A DOT (
DECIMAL POINT)
0000                540    TXT0:  EQU    077Q      ;0
0000                560    TXT1:  EQU    006Q      ;1
0000                580    TXT2:  EQU    133Q      ;2
0000                600    TXT3:  EQU    117Q      ;3
0000                620    TXT4:  EQU    146Q      ;4
0000                640    TXT5:  EQU    155Q      ;5
0000                660    TXT6:  EQU    175Q      ;6
0000                680    TXT7:  EQU    007Q      ;7
0000                700    TXT8:  EQU    177Q      ;8
0000                720    TXT9:  EQU    157Q      ;9
0000                740    TXTAD: EQU    367Q      ;A W/DOT
0000                760    TXTA:  EQU    167Q      ;A
0000                780    TXTBD: EQU    374Q      ;B W/DOT
0000                800    TXTB:  EQU    174Q      ;B
0000                820    TXTCD: EQU    330Q      ;C W/DOT
0000                840    TXTC:  EQU    130Q      ;C
0000                860    TXTDD: EQU    336Q      ;D W/DOT
0000                880    TXTD:  EQU    136Q      ;D
0000                900    TXTED: EQU    371Q      ;E W/DOT
0000                920    TXTE:  EQU    171Q      ;E
```

```
0000                940   TXTFD:  EQU     361Q        ;F W/DOT
0000                960   TXTF:   EQU     161Q        ;F
0000                980   TXTGD:  EQU     275Q        ;G W/DOT
0000                1000  TXTG:   EQU     075Q        ;G
0000                1020  TXTH:   EQU     164Q        ;H
0000                1040  TXTHD:  EQU     364Q        ;H W/DOT
0000                1060  TXTI:   EQU     004Q        ;I
0000                1080  TXTID:  EQU     204Q        ;I W/DOT
0000                1100  TXTJ:   EQU     016Q        ;
0000                1120  TXTJD:  EQU     216Q        ;J W/DOT
0000                1140  TXTK:   EQU     160Q        ;K
0000                1160  TXTKD:  EQU     360Q        ;K W/DOT
0000                1180  TXTLD:  EQU     270Q        ;L W/DOT
0000                1200  TXTL:   EQU     070Q        ;L
0000                1220  TXTM:   EQU     124Q        ;M
0000                1240  TXTMD:  EQU     324Q        ;M W/DOT
0000                1260  TXTN:   EQU     067Q        ;N
0000                1280  TXTND:  EQU     267Q        ;N W/DOT
0000                1300  TXTO:   EQU     134Q        ;O
0000                1320  TXTOD:  EQU     334Q        ;O W/DOT
0000                1340  TXTP:   EQU     163Q        ;P
0000                1360  TXTPD:  EQU     363Q        ;P W/DOT
0000                1380  TXTR:   EQU     120Q        ;R
0000                1400  TXTRD:  EQU     320Q        ;R W/DOT
0000                1420  TXTS:   EQU     155Q        ;S
0000                1440  TXTSD:  EQU     355Q        ;S W/DOT (LOOKS LIKE '5')
0000                1460  TXTT:   EQU     170Q        ;T
0000                1480  TXTTD:  EQU     370Q        ;T W/DOT
0000                1500  TXTU:   EQU     034Q        ;U
0000                1520  TXTUD:  EQU     234Q        ;U W/DOT
0000                1540  TXTV:   EQU     076Q        ;V
0000                1560  TXTVD:  EQU     276Q        ;V W/DOT
0000                1580  TXTW:   EQU     176Q        ;W???
0000                1600  TXTWD:  EQU     376Q
0000                1620  TXTX:   EQU     144Q        ;X
0000                1640  TXTXD:  EQU     344Q        ;X W/DOT
0000                1650  TXTY:   EQU     156Q        ;Y
0000                1655  TXTYD:  EQU     356Q        ;Y W/DOT
0000                1660  TXTZ:   EQU     133Q        ;Z (LOOKS LIKE '2')
```

```
0000                1680  TXTZD: EQU    333Q       ;Z W/DOT
0000                1700  TXTDSH:EQU    100Q       ;DASH (-)
0000                1720  TXTBLK:EQU    000Q       ;BLANK (NO SEG'S)
0000                1740  TXTDOT:EQU    200Q       ;DOT (.)
0000                1760          ;
0000                1780          ;
0000                1800          ;PORT DEFINITIONS:
0000                1820  PORT0: EQU    000Q        ;LED PORTS
0000                1840  PORT1: EQU    001Q
0000                1860  PORT2: EQU    002Q
0000                1880  PORT8279C:    EQU        003Q ;8279 CONTROL I/O
0000                1900  PORT8279D:    EQU        004Q ;8279 DATA I/O
0000                1920  PORTPUP:      EQU        001Q ;PORT WHICH PICKS UP PUP SIGN
AL
0000                1940  PORTDIPS:     EQU        001Q ;PORT WHICH PICKS UP DIP SWIT
CHES
0000                1960          ;
0000                1980          ;
0000                2000          ;CODES FOR ALL OF THE KEYS
0000                2020          ;AS INPUT FROM THE 8279 FOLLOW:
0000                2040          ;THESE CAN BE FOUND ON PAGE 43
0000                2060          ;OF THE USER'S GUIDE TO THE MMD-2
0000                2080  KEYNEXT:      EQU        1FH ;NEXT
0000                2100  KEYSTORE:     EQU        1EH ;STORE
0000                2120  KEYPREV:      EQU        1DH ;PREV
0000                2140  KEYSTEP:      EQU        1CH ;STEP
0000                2160  KEYHIGH:      EQU        17H ;HIGH
0000                2180  KEYLOW:EQU    16H        ;LOW
0000                2200  KEYGO: EQU    15H        ;GO
0000                2220  KEYOPTION:    EQU        14H ;OPTION
0000                2240  KEYLOAD:      EQU        0FH ;LOAD
0000                2260  KEYDUMP:      EQU        0EH ;DUMP
0000                2280  KEYPROM:      EQU        0DH ;PROM
0000                2300  KEYCOPY:      EQU        0CH ;COPY
0000                2320  KEYMEM:EQU    07H        ;MEMORY
0000                2340  KEYREGS:      EQU        06H ;REGISTER
0000                2360  KEYAUX:EQU    05H        ;AUXILLIARY
0000                2380  KEYCANCEL:    EQU        04H ;CANCEL
EXECB1
0000                1000          ;FILE :EXECB1
0000                1050          ;*****************************************
```

```
0000                    1100            ;EXEC-C
0000                    1150            ;
0000                    1200            ;CURRENT VERSION. EXEC-C 03/11/81
0000                    1250            ;**********************************
0000                    1300            ;
0000                    1350            ORG     FA0         ;SET EXEC ORIGIN AS DEFINED IN INI
T
D800                    1400            ;
D800                    1450            ;
D800                    1500            ;UPON ENTERING EXECB, WE MUST ASSUME THAT
D800                    1550            ;THE USER REGISTERS MUST BE SAVED BECAUSE
D800                    1600            ;USER JUST DID A GO...
D800                    1650            ;
D800 22 16 FC           1700    EXECB:  SHLD    USERHL1     ;FREE UP HL AND AC
D803 C3 9D 0B           1750            JMP     KE1;THIS IS A PATCH TO FIX THE TTY INI.
D806 D3 05              1800            OUT     005Q;OUTPUT TO TTY
D808 D3 0A              1850            OUT     012Q        ;& TURN OF PROM PROG PULSE
D80A                    1900            ;
D80A                    1950            ;
D80A                    2000            ;THIS IS HOW TO SET MEM MAP TO MEM MAP 2:
D80A 21 DB 06           2050    MAP2:   LXI     H           006333Q ;SET HL=INSTR FOR:IN 006
D80D 22 00 FC           2100            SHLD    SIM1        ;SAVE THIS INSTR IN SCRATCHPAD
D810 3E E9              2150            MVI     A           351Q ;STORE A PCHL NEXT
D812 32 02 FC           2200            STA     SIM3
D815 21 1B D8           2250            LXI     H           SAVEGOREG ;HL=ADR TO VECTOR TO AF
ER MEM SET
D818 C3 00 FC           2300            JMP     SIM1        ;JMP TO ROUTINE IN SCRATCHPAD
D81B                    2350            ;
D81B                    2400            ;
D81B E1                 2450    SAVEGOREG:      POP         H ;RESTORE USER STACK CONTENTS
D81C 22 18 FC           2500            SHLD    USERSC
D81F 3A 03 FC           2550            LDA     SIM4        ;GET BACK AC
D822 F5                 2600            PUSH    PSW         ;SAVE FLAGS FROM 'CPI KEYGO' BELOW

D823 3A 1F FC           2650            LDA     LASTKEY     ;SEE IF LAST KEY STRUCK WAS GO
D826 FE 15              2700            CPI     KEYGO
D828 C2 40 D8           2750            JNZ     RESET       ;JMP IF NO...
D82B                    2800            ;
D82B                    2850            ;
D82B 21 37 D8           2900            LXI     H           PTA
```

```
D82E 22 0B FC     2950          SHLD    VECT2
D831 F1           3000          POP     PSW         ;RESTORE FLAGS AND AC FROM PUSH AB
OVE
D832 3B           3010          DCX     SP          ;OFF-SET POP H (AT SAVEGOREG) SO U
SER'S SP WILL BE RIGHT
D833 3B           3020          DCX     SP          ;WE DON'T CARE WHAT H&L ARE CAUS'
THE REAL H&L ARE STORED
D834              3025                              ;STORED AT USERHL1 (SEE PTA JUST B
ELOW...)
D834 C3 CC DE     3050          JMP     SAVEM
D837 2A 16 FC     3100  PTA:    LHLD    USERHL1
D83A 22 14 FC     3150          SHLD    USERHL
D83D              3200          ;
D83D              3250          ;
D83D              3300          ;THIS IS EXECUTED ONLY IF LAST
D83D              3350          ;KEY STRUCK WAS 'GO'
D83D CD 1F DE     3400          CALL    REMOVEBP    ;REMOVE BREAK-POINTS IF ACTIVE
D840              3450          ;
D840              3500          ;
D840              3550          ;NORMAL RESET SEQUENCE STARTS HERE:
D840 AF           3600  RESET:  XRA     A           ;CLEAR OUT PORTS
D841 D3 00        3650          OUT     PORT0
D843 D3 01        3700          OUT     PORT1
D845 D3 02        3750          OUT     PORT2
D847              3800          ;
D847              3850          ;
D847 31 00 FD     3900  SETSP:  LXI     SP          FSCR ;SET EXEC STACK POINTER (DE!
NED IN INIT)
D84A              3950          ;
D84A              4000          ;
D84A              4050          ;SET UP 8279 FOR EXEC USE
D84A 3E 08        4100  SET8279:        MVI         A 010Q ;16 CHAR,LEFT ENTRY,2-KEY R
OLL
D84C D3 03        4150          OUT     PORT8279C
D84E 3E 24        4200          MVI     A           044Q ;SET CLOCK DIVIDER
D850 D3 03        4250          OUT     PORT8279C
D852 3E C2        4300          MVI     A           302Q ;CLEAR INPUT FIFO
D854 D3 03        4350          OUT     PORT8279C
D856              4400          ;
D856              4450          ;
```

```
D856                    4500          ;COPY INITIAL VARIABLE SET
D856                    4550          ;FOR RESET FROM ROM TO RAM
D856 01 CD DF           4600  RESVAR:LXI    B         STRVAR ;START OF INIT VARS
D859 11 DD DF           4650          LXI    D         SAUX+001Q ;END OF INIT VARS
D85C 21 2D FC           4700          LXI    H         HLBUF ;DESTINATION IN SCRATCHPAD
D85F CD D5 DB           4750          CALL   COPYBTOH  ;DO THE TRANSFER
D862                    4800          ;
D862                    4850          ;
D862                    4900          ;DO A MEM MODE CANCEL TO SET UP DATA VARIABLE
D862                    4950          ;THAT'S WHAT A CANCEL DOES-SET THE DATA VARIABLE
D862 CD B8 D8           5000          CALL   CANCLEM
D865                    5050          ;
D865                    5100          ;
D865                    5150          ;NOW, DESTINGUISH BETWEEN A PUP & RESET
D865 DB 01              5200  PUPTST:IN     PORTDIPS  ;INPUT PUP SIGNALS
D867 E6 50              5250          ANI    120Q      ;AND OUT MMD-2 PUP SIGNAL AND
D869                    5300                           ;PUP/RESET DIP SWITCH (SAME PORT.
D869 CA 89 D8           5350          JZ     MAIN1     ;JMP IF RESET (NOT PUP)
D86C                    5400          ;
D86C                    5450          ;
D86C                    5500          ;PUP SEQUENCE STARTS HERE.  FIRST SET
D86C                    5550          ;ALL OF SCRATCHPAD TO 000
D86C 21 FF FC           5600  DOPUP: LXI    H         FSCR-001Q ;HL=LAST ADR OF SCRATCH..
AD
D86F AF                 5650  CLRSCR:XRA    A         ;CLEAR AC
D870 77                 5700          MOV    M         A ;CLEAR MEM LOC
D871 2D                 5750          DCR    L         ;DECREMENT POINTER (HL)
D872 C2 6F D8           5800          JNZ    CLRSCR    ;JMP IF NOT DONE
D875                    5850          ;
D875                    5900          ;
D875                    5950          ;COPY IN INITIAL VARIABLE SET FOR
D875                    6000          ;A PUP.  (A SUBSET OF THESE ARE USED IN RESET)
D875 01 CD DF           6050  PUPVAR:LXI    B         STRVAR ;START OF INIT VARS FOR PUP

D878 11 02 E0           6100          LXI    D         ENDVAR+001Q ;END OF INIT VARS
D87B 21 2D FC           6150          LXI    H         HLBUF ;DESTINATION IN SCRATCHPAD
D87E CD D5 DB           6200          CALL   COPYBTOH  ;COPY PUP VARIABLES
D881                    6250          ;
D881                    6300          ;
D881                    6350          ;NOW SHOW EXEC VERSION NUMBER
```

```
D881 21 08 E1    6400          LXI     H          MSGXECVER ;HL=ADR OF MESSAGE
D884 CD 23 DC    6450          CALL    MSG8
D887 D3 09       6500          OUT     011Q       ;TURN PUP SIGNAL OFF
D889             6550          ;
D889             6600          ;
D889             6650  ;*****************************************
D889             6700          ;
D889             6750          ;
D889             6800          ;
D889             6850          ;
D889             6900          ;TASK MANAGER IS HERE
D889 21 89 D8    6950  MAIN1:  LXI     H          MAIN1 ;SET TASK RET ADR
D88C E5          7000          PUSH    H          ;PUSH RET ADR ON STACK
D88D 2A 33 FC    7050          LHLD    TASK       ;GET CURRENT TASK POINTER
D890 5E          7100          MOV     E          M ;MOVE LO BYTE TO E
D891 23          7150          INX     H          ;INCREMENT TASK POINTER TO HI BYTE

D892 56          7200          MOV     D          M ;MOVE HI BYTE TO D
D893 23          7250          INX     H          ;INCREMENT TASK POINTER TO NEXT TA
SK
D894 22 33 FC    7300          SHLD    TASK       ;STORE NEW TASK POINTER
D897 EB          7350          XCHG               ;MOVE TASK ADRESS (IN DE) TO HL
D898 E9          7400          PCHL               ;BRANCH TO TASK (RETURN SET UP ALR
EADY)
D899             7450          ;
D899             7500          ;
D899             7550          ;COMMAND DISPATCHER ROUTINE IS HERE
D899             7600          ;THIS ROUTINE BRANCHES TO A COMMAND ROUTINE
D899             7650          ;BASED ON THE LAST FUNCTION KEY STRUCK.
D899 CD 15 D9    7700  COMDIS: CALL    SETTASKA
D89C 2A 31 FC    7750          LHLD    TABLE      ;LOOKUP KEY IN TABLE(X)
D89F 16 00       7800          MVI     D          000Q ;SET HI BYTE FOR ADD TO 000
D8A1 3A 20 FC    7850          LDA     LASTKEYCO  ;GET COMPRESSED KEY CODE
D8A4 87          7900          ADD     A          ;DOUBLE IT (2 BYTES/ENTRY IN TABLE
)
D8A5 5F          7950          MOV     E          A ;MOVE OFFSET TO DE
D8A6 19          8000          DAD     D          ;ADD OFFSET IN DE TO TABLE BASE IN
 HL
D8A7 46          8050          MOV     B          M ;GET TABLE DATA INTO HL
D8A8 23          8100          INX     H
```

```
D8A9 66          8150          MOV    H          M
D8AA 68          8200          MOV    L          B
D8AB E9          8250          PCHL   ;BRANCH TO FUNCTION WHOSE ADR IS IN HL
D8AC             8300          ;
D8AC             8350          ;
D8AC             8400          ;
D8AC             8450          ;THIS ROUTINE SETS EXEC UP FOR MEM MODE
D8AC 21 85 D9    8500  MEM:    LXI    H          HLDOUT ;SET DSP UPDATE TO MEM
D8AF 22 2F FC    8550          SHLD   DSPUP      ;HLDOUT IS MEM MODE DPS UPDATE
D8B2 21 F5 DE    8600          LXI    H          TABLEM ;SET COMMAND TABLE TO MEM M
ODE
D8B5 22 31 FC    8650          SHLD   TABLE
D8B8             8700                           ;FALLS THRU TO CANCLEM
D8B8             8750          ;
D8B8             8800          ;
D8B8 CD C2 DE    8850  CANCLEM:        CALL       SETMEMMAP ;SET ROM/RAM MEMORY MAP
D8BB 2A 2D FC    8900          LHLD   HLBUF      ;SET UP DATBUF FROM ADR VARS
D8BE 22 2D FC    8950          SHLD   HLBUF
D8C1 7E          9000          MOV    A          M
D8C2 32 2C FC    9050          STA    DATBUF
D8C5             9100                           ;FALLS THRU TO UPDATE
D8C5             9150          ;
D8C5             9200          ;
D8C5             9250          ;THIS ROUTINE DIPATCHES TO THE PROPER
D8C5             9300          ;ROUTINE TO UPDATE ALL DISPLAYS FOR ANY
D8C5             9350          ;MODE DETERMINED BY SETTING OF DSPUP
D8C5 2A 2F FC    9400  UPDATE:LHLD    DSPUP      ;GET CURRENT DSP UPDATE ADR
D8C8 E9          9450          PCHL   ;JMP TO THAT ROUTINE (RETURN ALREADY
D8C9             9500                           ;SET UP BY MAIN1...)
D8C9             9550          ;
D8C9             9600          ;
D8C9             9650          ;THIS ROUTINE WAITS FOR INPUT ON EITHER OF
D8C9             9700          ;THE KEYPADS, AND RETURNS THE DATA IN  THE
D8C9             9750          ;A REG.  IT IS ALSO STORED IN LASTKEY, FOR
D8C9             9800          ;ANY USER THAT WANTS TO USE IT LATER ON OR
D8C9             9850          ;USE THIS SUBROUTINE.
D8C9 DB 03       9900  KEYRD:  IN     PORT8279C ;INPUT FIFO STATUS
D8CB E6 0F       9950          ANI    017Q       ;CLEAR ALL BUT BITS WHICH INDICATE

D8CD             10000                          ;# WORDS IN FIFO
```

```
D8CD CA C9 D8    10050         JZ      KEYRD      ;JMP IF ZERO WORDS (NO KEY STRUCK)

D8D0 3E 40       10100         MVI     A          100Q ;8279 COMMAND TO READ FIFO RA
M LOC 000
D8D2 D3 03       10150         OUT     PORT8279C
D8D4 DB 04       10200         IN      PORT8279D ;INPUT KEY CODE
D8D6 E6 3F       10250         ANI     077Q       ;MASK OUT UNWANTED BITS
D8D8 32 1F FC    10300         STA     LASTKEY    ;STUF THIS CODE AWAY SOMEWHERE
D8DB C9          10350         RET
D8DC             10400         ;
D8DC             10450         ;
D8DC             10500         ;THIS ROUTINE ACCUMULATES THE VALUE IN THE
D8DC             10550         ;B REG TO DATBUF (IN THE SCRATCHPAD) IN OCTAL
D8DC             10600         ;OR HEX, DEPENDING ON DIP SWITCH.   RESULT
D8DC             10650         ;IS PLACED IN DATBUF.
D8DC 47          10700 ADD1:   MOV     B          A ;SAVE NEW DIGIT IN B
D8DD DB 01       10750         IN      001Q       ;CHECK HEX/OCT DIP SWITCH
D8DF E6 20       10800         ANI     DIPBASE    ;MASK OUT BASE BIT
D8E1 CA F2 D8    10850         JZ      OCT1
D8E4 3A 2C FC    10900 HEX1:   LDA     DATBUF     ;GET OLD ACCUMULATION
D8E7 E6 0F       10950         ANI     017Q       ;SET MSD HEX TO 0
D8E9 07          11000         RLC                ;FOR HEX, ROTATE 4 TIMES TO
D8EA             11050                            ;MOVE LSD TO MSD
D8EA 07          11100 ROTATE: RLC                ;IF OCT, JUST 3 TIMES
D8EB 07          11150         RLC
D8EC 07          11200         RLC
D8ED B0          11250         ORA     B          ;COMBINE NEW DIGIT IN B WITH
D8EE             11300                            ;ROTATED ACCUMULATION IN A
D8EE 32 2C FC    11350         STA     DATBUF     ;STORE NEW ACCUMULATION
D8F1 C9          11400         RET
D8F2 78          11450 OCT1:   MOV     A          B ;CHECK IF NEW DIGIT > 8
D8F3 FE 08       11500         CPI     010Q
D8F5 D0          11550         RNC     ;RET IF NOT OCTAL
D8F6 3A 2C FC    11600         LDA     DATBUF     ;GET OLD ACCUMULATION
D8F9 E6 1F       11650         ANI     037Q       ;SET MSD (OCT) TO 0
D8FB C3 EA D8    11700         JMP     ROTATE
D8FE             11750         ;
D8FE             11800         ;
D8FE 3A 1F FC    11850 ACCUM:  LDA     LASTKEY    ;GET KEY CODE
D901 B7          11900         ORA     A          ;CLEAR CARRY
D902 1F          11950         RAR                ;COMPRESS CODE, SAVING
```

```
D903 1F          12000         RAR              ;KEYPAD I.D. IN CARRY
D904 1F          12050         RAR
D905 F5          12100         PUSH   PSW       ;SAVE KEYPAD I.D.
D906 07          12150         RLC
D907 07          12200         RLC
D908 32 20 FC    12250         STA    LASTKEYCO ;SAVE COMPRESSED CODE
D90B 47          12300         MOV    B         A ;SAVE COMPRESSED CODE
D90C F1          12350         POP    PSW       ;RETRIEVE KEYPAD I.D.
D90D 78          12400         MOV    A         B ;RETRIEVE COMPRESSED CODE
D90E D8          12450         RC               ;RET IF FUNCTION KEYPAD
D90F             12500                          ;IF NOT, ACCUMULATE DATA
D90F 2F          12550         CMA              ;COMPLIMENT 4 LOW ORDER BITS
D910 E6 0F       12600         ANI    017Q      ;BECAUSE PADS ARE INVERTED
D912 CD DC D8    12650         CALL   ADD1      ;UPDATE ACCUMULATION
D915             12700         ;FALLS THRU TO SETE1...
D915             12750         ;
D915             12800         ;
D915 21 E7 DE    12850 SETTASKA: LXI    H TASKA ;SET EVENT POINTER TO EVEN
T 1
D918 22 33 FC    12900         SHLD   TASK
D91B C9          12950         RET
D91C             13000         ;
D91C             13050         ;
D91C AF          13100 CLR7SG: XRA    A         ;SET AC=0 (BLANK CODE)
D91D F5          13150 DSPOUT: PUSH   PSW       ;SAVE AC
D91E DB 03       13200         IN     PORT8279C ;INPUT FIFO STATUS
D920 B7          13250         ORA    A         ;SET FLAGS
D921 FA 1E D9    13300         JM     DSPOUT+1  ;JMP IF DISPLAY NOT READY
D924 F1          13350         POP    PSW       ;RESTORE AC
D925 D3 04       13400         OUT    PORT8279D
D927 C9          13450 RETURN: RET              ;RET FOR UNDEFINED KEYS
D928             13500         ;
D928             13550         ;
D928 F5          13600 SSEG:   PUSH   PSW       ;SAVE SOME STUF
D929 C5          13650         PUSH   B
D92A E5          13700         PUSH   H
D92B 47          13750         MOV    B         A ;SAVE OUTPUT DIGIT
D92C 21 2E E0    13800         LXI    H         SSEGTB ;HL=BASE OF 7-SEG CODE TA
E
D92F E6 0F       13850         ANI    017Q      ;CLEAR BIT 7 IF SET
```

```
D931 16 00       13900         MVI    D          000Q ;SET HI ORDER BYTE FOR ADD )
 000
D933 5F          13950         MOV    E          A
D934 19          14000         DAD    D
D935 78          14050         MOV    A          B
D936 E6 80       14100         ANI    200Q
D938 B6          14150         ORA    M
D939 CD 1D D9    14200         CALL   DSPOUT     ;SEND BYTE TO DISPLAY
D93C E1          14250         POP    H
D93D C1          14300         POP    B
D93E F1          14350         POP    PSW
D93F C9          14400         RET
D940             14450         ;
D940             14500         ;
D940 06 96       14550 DATOUT: MVI    B          226Q
D942 AF          14600         XRA    A          ;CLEAR CARRY (RD FLAG)
D943 F5          14650 DSPLY:  PUSH   PSW
D944 78          14700         MOV    A          B
D945 D3 03       14750         OUT    PORT8279C
D947 F1          14800         POP    PSW
D948 D5          14850 DSPLY1: PUSH   D
D949 F5          14900         PUSH   PSW
D94A DB 01       14950         IN     PORTDIPS   ;FETCH DIP SWITCHES
D94C E6 20       15000         ANI    DIPBASE    ;HEX/OCT MASK
D94E CA 6E D9    15050         JZ     OCTPR      ;JMP IF OCTAL MODE
D951 CD 1C D9    15100 HEXPR:  CALL   CLR7SG     ;CLEAR LEADING DISPLAY
D954 79          15150         MOV    A          C ;GET OUTPUT DIGIT
D955 07          15200         RLC               ;GET JUST LSD
D956 07          15250         RLC
D957 07          15300         RLC
D958 07          15350         RLC
D959 E6 0F       15400         ANI    017Q       ;CLEAR OUT OTHER DIGIT
D95B CD 28 D9    15450         CALL   SSEG       ;SEND 1ST DIGIT
D95E 79          15500         MOV    A          C
D95F E6 0F       15550         ANI    017Q
D961 57          15600 LSTDIG: MOV    D          A
D962 F1          15650         POP    PSW        ;RESTORE FLAGS
D963 F5          15700         PUSH   PSW
D964 1F          15750         RAR    ;MOVE CARRY TO BIT 7
D965 E6 80       15800         ANI    200Q       ;CLEAR BITS 0-6
D967 B2          15850         ORA    D
```

```
D968 CD 28 D9    15900         CALL   SSEG       ;SEND LAST DIGIT
D96B F1          15950         POP    PSW
D96C D1          16000         POP    D
D96D C9          16050         RET
D96E 79          16100 OCTPR:  MOV    A          C
D96F 07          16150         RLC
D970 07          16200         RLC
D971 E6 03       16250         ANI    003Q       ;MASK FOR MSD
D973 CD 28 D9    16300         CALL   SSEG       ;SEND MSD
D976 79          16350         MOV    A          C
D977 0F          16400         RRC
D978 0F          16450         RRC
D979 0F          16500         RRC
D97A E6 07       16550         ANI    007Q       ;MSK FOR MIDDLE DIGIT
D97C CD 28 D9    16600         CALL   SSEG       ;SEND MIDDLE DIGIT
D97F 79          16650         MOV    A          C
D980 E6 07       16700         ANI    007Q       ;MSK FOR LSD
D982 C3 61 D9    16750         JMP    LSTDIG
D985             16800         ;
D985             16850         ;
D985             16900         ;THIS IS THE MEMORY MODE DISPLAY
D985             16950         ;UPDATE ROUTINE.  IT OUTPUTS THE
D985             17000         ;HI&LO&DATA VARIABLES IN HEX/OCT
D985             17050         ;AND BINARY/PORTS.
D985 C5          17100 HLDOUT: PUSH   B          ;SAVE REGIES
D986 E5          17150         PUSH   H
D987 F5          17200         PUSH   PSW
D988 2A 2D FC    17250         LHLD   HLBUF      ;HL=ADR VARS
D98B 4C          17300         MOV    C          H
D98C 37          17350         STC               ;TURN RD ON
D98D 06 D0       17400         MVI    B          320Q ;CODE TO CLEAR 8279 DISPLAY R
AM
D98F CD 43 D9    17450         CALL   DSPLY      ;SEND HI ADDRESS VARIABLE
D992 4D          17500         MOV    C          L
D993 CD 48 D9    17550         CALL   DSPLY1
D996 CD 6C DA    17600         CALL   PRTDAT     ;SUBROUTINE TO OUTPUT DATA VARIABL
E (INCLUDES LED'S TEST)
D999 DB 01       17650         IN     PORTDIPS   ;TEST FOR BINARY/PORT MODE
D99B E6 08       17700         ANI    DIPLEDS    ;BINARY/PORT MASK
D99D CA A6 D9    17750         JZ     ENDHLD     ;JMP IF LEDS OFF FOR EXEC
```

```
 ,A0 7D          17800          MOV     A          L ;ELSE OUTPUT H&L&A TO LEDS
D9A1 D3 00       17850          OUT     PORT0
D9A3 7C          17900          MOV     A          H
D9A4 D3 01       17950          OUT     PORT1
D9A6             18000          ;NOTE:WE DON'T NEED TO OUTPUT TO DATA LED'S BECAUSE
 CALL
D9A6             18050          ;TO PRTDAT TOOK CARE OF THAT FOR US.
D9A6 F1          18100 ENDHLD:  POP     PSW
D9A7 E1          18150          POP     H
D9A8 C1          18200          POP     B
D9A9 C9          18250          RET
D9AA             18300          ;
D9AA             18350          ;
D9AA             18400          ;
D9AA             18450          ;
D9AA CD F3 DC    18500 STPERR:  CALL    SETTASKC
D9AD 21 73 E0    18550          LXI     H          MSGERR ;SET HL='ERR' TEXT
D9B0             18600                             ;FALLS THRU TO MSGAT6
D9B0             18650          ;
D9B0             18700          ;
D9B0             18750          ;8279 MESSAGE DRIVERS FOLLOW:
D9B0 3E 96       18800 MSGAT6:  MVI     A          226Q ;8279 INSTR TO SET DISPLAY
D9B2             18850                             ;RAM POINTER TO #6
D9B2 06 03       18900          MVI     B          003Q ;# CHARS IN MESSAGE
D9B4             18950                             ;FALLS THRU TO MSGGEN
D9B4 D3 03       19000 MSGGEN:  OUT     PORT8279C ;OUTPUT 8279 INSTR
D9B6 7E          19050          MOV     A          M ;GET A CHAR
D9B7 CD 1D D9    19100          CALL    DSPOUT     ;OUTPUT THIS CHAR
D9BA 23          19150          INX     H          ;INR CHAR POINTER
D9BB 05          19200          DCR     B          ;DCR # CHARS REG
D9BC C2 B6 D9    19250          JNZ     MSGGEN+002Q ;JMP BACK IF NOT DONE YET
D9BF C9          19300          RET
D9C0 06 06       19350 MSGAT0:  MVI     B          006Q ;6 CHARS IN MESSAGE
D9C2 3E D0       19400          MVI     A          320Q ;COMMAND TO CLEAR 8279
D9C4 C3 B4 D9    19450          JMP     MSGGEN     ;DO GENERAL MESSAGE ROUTINE
D9C7             19500          ;
D9C7             19550          ;
D9C7             19600          ;MEMORY MODE PREVIOS FUNCTION:
D9C7 2A 2D FC    19650 PREVM:   LHLD    HLBUF      ;SET HL=ADR VARS
D9CA 2B          19700          DCX     H          ;DCR H&L
```

```
D9CB 22 2D FC    19750 SAVEHL:SHLD   HLBUF      ;ENTRY POINT FOR OTHER ROUTINES
D9CE C3 B8 D8    19800        JMP    CANCLEM    ;DO A MEM MODE CANCLE
D9D1             19850                           ;WHICH WILL UPDATE EVERYTHING
D9D1             19900        ;
D9D1             19950        ;
D9D1             20000        ;MEMORY MODE STORE FUNCTION:
D9D1 2A 2D FC    20050 STOREM:LHLD   HLBUF      ;SET HL=ADR VARS
D9D4 3A 2C FC    20100        LDA    DATBUF     ;A=DATA VAR
D9D7 77          20150        MOV    M          A
D9D8 BE          20200        CMP    M           ;TEST FOR BAD WRITE
D9D9 C2 20 DC    20250        JNZ    NOMEM      ;IF BAD WRITE, SHOW 'NOT.STORED'
D9DC 21 7A E0    20300 DODASH:LXI    H          MSGDSH ;HL=ADR OF '---' TEXT
D9DF CD B0 D9    20350        CALL   MSGAT6     ;OUTPUT DASHES
D9E2 21 00 20    20400 TIMEOUT:       LXI        H 040000Q ;TIME DELAY
D9E5 2B          20450        DCX    H
D9E6 7C          20500        MOV    A          H
D9E7 B7          20550        ORA    A
D9E8 C2 E5 D9    20600        JNZ    TIMEOUT+3
D9EB C9          20650        RET
D9EC             20700        ;
D9EC             20750        ;
D9EC             20800        ;MEMORY MODE NEXT FUNCTION
D9EC 2A 2D FC    20850 NEXTM: LHLD   HLBUF      ;SET HL=ADR VARS
D9EF 23          20900        INX    H          ;IN HLBUF
D9F0 C3 CB D9    20950        JMP    SAVEHL
D9F3             21000        ;
D9F3             21050        ;
D9F3             21100        ;SOURCE CONTINUES IN EXECB2...
D9F3             21150        ;
D9F3             21200        ;
D9F3             21250        ;
D9F3             21300        ;
D9F3             21350        ;
D9F3             21400        ;
D9F3             21450        ;
EXECB2
D9F3             10           ;FILE: EXECB2
D9F3             100          ;EXEC B SOURCE CONTINUED FROM EXECB1...
D9F3             150          ;
D9F3             200          ;
```

```
D9F3                    250             ;MEMORY MODE HI FUNCTION:
D9F3 2A 2D FC   300     HIM:    LHLD    HLBUF       ;SET HL=ADR VARS
D9F6 3A 2C FC   350             LDA     DATBUF      ;A=DATA BAR
D9F9 67         400             MOV     H           A
D9FA C3 CB D9   450             JMP     SAVEHL
D9FD            500             ;
D9FD            550             ;
D9FD            600             ;MEMORY MODE LO FUNCTION
D9FD 2A 2D FC   650     LOM:    LHLD    HLBUF       ;HL=ADR VARS
DA00 3A 2C FC   700             LDA     DATBUF      ;A=DATA VAR
DA03 6F         750             MOV     L           A
DA04 C3 CB D9   800             JMP     SAVEHL
DA07            850             ;
DA07            900             ;
DA07            950             ;MEMORY MODE GO FUNCTION
DA07 21 70 E0   1000    GOM:    LXI     H           MSGGO ;OUTPUT 'GO'
DA0A CD B0 D9   1010            CALL    MSGAT6
DA0D DB 01      1020            IN      PORTDIPS    ;FETCH DIP SWITCHES
DA0F E6 08      1030            ANI     DIPLEDS     ;MASK PORT/BINARY SWITCH
DA11 CA 17 DA   1040            JZ      SETUPGO     ;JMP IF LEDS OFF FOR EXEC
DA14 AF         1050            XRA     A           ;ELSE SET PORT 2 TO 000
DA15 D3 02      1060            OUT     PORT2
DA17 CD A2 DE   1070    SETUPGO:        CALL        BPADRCHK ;CHECK IF CURRENT ADR=BP
ADR
DA1A C3 32 DA   1080            JMP     NOBRONGO    ;RETURN HERE IF NO (ELSE RETURN .I
LOW)
DA1D AF         1090            XRA     A           ;YES, BP ON GO.  FAKE OUT LASTKEY
DA1E 32 1F FC   1100            STA     LASTKEY     ;IN CASE WE DO AN ERR RESET COMB.

DA21 CD AE DA   1110            CALL    STEP        ;DO 3 STEPS TO MAKE SURE WE'RE PAS
T JMP
DA24 CD AE DA   1112            CALL    STEP
DA27 CD AE DA   1114            CALL    STEP
DA2A DA AA D9   1120            JC      STPERR      ;JMP IF UNDEF. OPCODE
DA2D 3E 15      1130            MVI     A           KEYGO ;RESTORE LASTKEY BECAUSE WE
DA2F 32 1F FC   1140            STA     LASTKEY     ;GOT THRU SECCESSFULLY
DA32 CD 1F DE   1160    NOBRONGO:       CALL        SETBP ;SET UP JMP VECTOR IN USER .P
ROG
DA35 2A 2D FC   1165            LHLD    HLBUF       ;STORE ADR FOR GO
DA38 22 01 FC   1170            SHLD    SIM2
DA3B 3E C3      1180            MVI     A           303Q ;STORE JMP OP-CODE
```

```
DA3D 32 00 FC     1190          STA     SIM1
DA40 C3 38 DB     1200          JMP     DOINST    ;LOAD REAL REG.S WITH USER VALUES
AND GO
DA43              2350          ;
DA43              2400          ;
DA43              2450          ;REGISTER MODE PREVIOUS FUNCTION
DA43 2A 38 FC     2500  PREVR:  LHLD    DSPREG    ;GET CURRENT REGISTER POINTER
DA46 2B           2550          DCX     H
DA47 2B           2600          DCX     H
DA48 2B           2650          DCX     H
DA49 3E 00        2700          MVI     A         MSGFL-3 ;WAS LAST REG DISPLAYED
DA4B BD           2750          CMP     L         ;1ST IN LIST (FL)?
DA4C C2 52 DA     2800          JNZ     SHOWRG    ;JMP IF NOT
DA4F 21 1E E0     2850          LXI     H         MSGSL ;RESET POINTER IF YES
DA52 22 38 FC     2900  SHOWRG: SHLD    DSPREG    ;STORE NEW REG POINTER
DA55 6E           2950          MOV     L         M
DA56 26 FC        3000          MVI     H         FSCRHI ;H=HI ADR OF SCRATCHPAD
DA58 7E           3050          MOV     A         M
DA59 32 2C FC     3100          STA     DATBUF    ;MOVE NEW REG CNTNTS TO DATA VAR
DA5C              3150          ;
DA5C              3200          ;
DA5C              3250          ;GENERAL REG MODE DISPLAY UPDATE ROUTINE:
DA5C 06 04        3350  REGOUT: MVI     B         004Q ;# CHARS IN MESSAGE
DA5E 21 76 E0     3400          LXI     H         MSGREG ;HL=ADR OF 'REG ' TEXT
DA61 CD C2 D9     3450          CALL    MSGAT0+002Q ;OUTPUT 'REG '
DA64 2A 38 FC     3500          LHLD    DSPREG    ;SEND REG NAME
DA67 06 03        3550          MVI     B         003Q ;# CHARS IN MESSAGE
DA69 CD BA D9     3600          CALL    MSGGEN+006Q ;CALL MESSAGE ROUTINE
DA6C              3650                      ;FALLS THRU TO PRTDAT
DA6C              3700          ;
DA6C              3750          ;
DA6C 3A 2C FC     3800  PRTDAT: LDA     DATBUF    ;GENERAL ROUTINE TO DISPLAY
DA6F 4F           3850          MOV     C         A ;DATA VARIABLE
DA70 CD 40 D9     3900          CALL    DATOUT
DA73 DB 01        3950  PRT2TST:        IN        PORTDIPS ;CHECK TO SEE IF WE OUT
T
DA75 E6 08        4000          ANI     DIPLEDS   ;INFO ON LEDS
DA77 C8           4050          RZ      ;RET IF NO
DA78 AF           4100          XRA     A         ;ELSE CLEAR HI&LO
DA79 D3 01        4150          OUT     PORT1
DA7B D3 00        4200          OUT     PORT0
```

```
DA7D 79          4250          MOV    A          C ;MOVE DATA VAR TO AC
DA7E D3 02       4300          OUT    PORT2
DA80 C9          4350          RET
DA81             4400          ;
DA81             4450          ;
DA81             4500          ;REGISTER MODE STORE FUNCTION:
DA81 2A 38 FC    4550  STORER:LHLD   DSPREG     ;HL=ADR OF CURRENT REG
DA84 6E          4600          MOV    L          M ;L=LO ADR OF REG CONTENTS
DA85 26 FC       4650          MVI    H          FSCRHI ;H=HI ADR OF REG CONTENTS
DA87 C3 D4 D9    4700          JMP    STOREM+3
DA8A             4750          ;
DA8A             4800          ;
DA8A             4850          ;REGISTER MODE NEXT FUNCTION:
DA8A 2A 38 FC    4900  NEXTR: LHLD   DSPREG     ;GET ADR OF LAST REG DISPLAYED
DA8D 23          4950          INX    H
DA8E 23          5000          INX    H
DA8F 23          5050          INX    H
DA90 3E 21       5100          MVI    A          MSGSL+3 ;WAS IT LAST REG IN LIST
DA92 BD          5150          CMP    L          ;COMPARE CURRENT WITH LAST REG
DA93 C2 52 DA    5200          JNZ    SHOWRG     ;IF NO JMP AND SHOW REG INFO
DA96 21 03 E0    5250          LXI    H          MSGFL ;YES, SET POINTER TO 1ST REG

DA99 C3 52 DA    5300          JMP    SHOWRG
DA9C             5350          ;
DA9C             5400          ;
DA9C             5600          ;
DA9C             5650          ;
DA9C             5700          ;MEMORY/AUX MODE REG FUNCTION
DA9C 21 5C DA    5750  REG:    LXI    H          REGOUT ;SET REG UPDATE ROUTINE
DA9F 22 2F FC    5800          SHLD   DSPUP
DAA2 21 15 DF    5850          LXI    H          TABLER ;SET COMDIS TABLE TO REG
DAA5 22 31 FC    5900          SHLD   TABLE
DAA8             5950                             ;FALLS THRU TO CANCLR...
DAA8             6000          ;
DAA8             6050          ;
DAA8             6100          ;REGISTER MODE CANCLE FUNCTION:
DAA8 2A 38 FC    6150  CANCLR:LHLD   DSPREG     ;HL=CURRENT REG POINTER
DAAB C3 55 DA    6200          JMP    SHOWRG+3
DAAE             6250          ;
DAAE             6300          ;
```

```
DAAE                    6350          ;COMPLETE, SELF CONTAINED 8080 SINGLE-STEPPER
DAAE                    6400          ;FOLLOWS.  CAN BE CALLED AS SUBROUTINE ANYTIME (SEE

DAAE                    6450          ;DOCUMENTATION FOR MORE INFO.)
DAAE 21 00 00           6500  STEP:   LXI     H         000000Q ;SAVE PROG STACK POINTER
DAB1 39                 6550          DAD     SP
DAB2 22 1A FC           6600          SHLD    EXECSP
DAB5 21 00 00           6650          LXI     H         000000Q ;SAVE NOP'S AT SIMS 1&2
DAB8 22 01 FC           6700          SHLD    SIM2
DABB 21 AD DB           6750          LXI     H         ENDSTP ;STORE LO&HI ADR OF 'ENDSTP
' ROUTINE
DABE 22 04 FC           6760          SHLD    SIM5
DAC1 3E C3              6770          MVI     A         303Q ;SAVE JMP OPCODE FOR ABOVE
DAC3 32 03 FC           6780          STA     SIM4
DAC6 2A 5B FC           7200          LHLD    USERSP    ;GET USER SP TO REAL SP
DAC9 F9                 7250          SPHL
DACA 2A 2D FC           7300          LHLD    HLBUF     ;GET ADR VARS TO HL
DACD 46                 7350          MOV     B         M ;MOVE SINGLE-STEP OP CODE TO B
DACE 11 21 E0           7400  UNDEF:  LXI     D         UNDTBL ;DE=BASE OF UNDEFINED OP-C
DE TABLE
DAD1 1A                 7500          LDAX    D         ;GET AN UNDEF OP-CODE IN AC
DAD2 B7                 7550          ORA     A         ;SET FLAGS
DAD3 CA DE DA           7600          JZ      EXPLIC    ;000 INDICATES END OF TABLE
DAD6 B8                 7650          CMP     B         ;COMPARE AGAINST USER'S OP-CODE
DAD7 CA B9 DB           7700          JZ      ERRERR    ;JMP IF MATCH (UNDEF ERROR)
DADA 13                 7750          INX     D
DADB C3 D1 DA           7800          JMP     UNDEF+3
DADE                    7850          ;CHECK EXPLICITLY FOR CERTAIN OP-CODES
DADE 78                 7900  EXPLIC:MOV     A         B ;OPCODE TO AC
DADF 11 00 FC           7950          LXI     D         SIM1 ;DE=ADR OF SIMULATION BYTE #1

DAE2 FE C9              8000          CPI     311Q      ;RET?
DAE4 CA 48 DB           8050          JZ      EXRET
DAE7 FE E9              8100          CPI     351Q      ;PCHL?
DAE9 CA 59 DB           8150          JZ      EXPCHL
DAEC FE C3              8200          CPI     303Q      ;JMP?
DAEE CA 74 DB           8250          JZ      EXJMP
DAF1 FE D3              8300          CPI     323Q      ;OUT?
DAF3 CA 2E DB           8350          JZ      BYTES2
DAF6 FE DB              8400          CPI     333Q      ;IN?
```

```
DAF8 CA 2E DB     8450          JZ      BYTES2
DAFB FE CD        8500          CPI     315Q        ;CALL?
DAFD CA 6D DB     8550          JZ      EXCALL
DB00 E6 CF        8600  GROUPS: ANI     317Q        ;TRY TO CLASSIFY OP-CODE
DB02 FE 01        8650          CPI     001Q
DB04 CA 2A DB     8700          JZ      BYTES3      ;0 EVEN 1
DB07 78           8750          MOV     A           B ;OP CODE IN B
DB08 E6 E7        8800          ANI     347Q
DB0A FE 22        8850          CPI     042Q
DB0C CA 2A DB     8900          JZ      BYTES3      ;0 >3 2
DB0F E6 C7        8950          ANI     307Q
DB11 FE 06        9000          CPI     006Q
DB13 CA 2E DB     9050          JZ      BYTES2      ;0 X 6
DB16 FE C6        9100          CPI     306Q
DB18 CA 2E DB     9150          JZ      BYTES2      ;3 X 6
DB1B FE C7        9200          CPI     307Q
DB1D CA 7C DB     9250          JZ      EXRST       ;3 X 7
DB20 E6 C1        9300          ANI     301Q
DB22 FE C0        9350          CPI     300Q
DB24 CA 87 DB     9400          JZ      EXCOND      ;3 X EVEN
DB27 C3 32 DB     9450          JMP     BYTE1       ;DEFAULT 1 BYTE NORMALS
DB2A 7E           9500  BYTES3: MOV     A           M ;STORE 3 BYTES
DB2B 12           9550          STAX    D
DB2C 13           9600          INX     D
DB2D 23           9650          INX     H
DB2E 7E           9700  BYTES2: MOV     A           M ;STORE 2 BYTES
DB2F 12           9750          STAX    D
DB30 13           9800          INX     D
DB31 23           9850          INX     H
DB32 7E           9900  BYTE1:  MOV     A           M ;STORE 1 BYTE
DB33 12           9950          STAX    D
DB34 23           10000         INX     H
DB35 22 2D FC     10050         SHLD    HLBUF       ;SAVE NEW ADR VARS
DB38 31 0D FC     10100 DOINST: LXI     SP          REGFL ;PUT USER REGS IN
DB3B F1           10150         POP     PSW         ;REAL REGS FOR SINGLE-STEP
DB3C C1           10200         POP     B
DB3D D1           10250         POP     D
DB3E 2A 5B FC     10300         LHLD    USERSP      ;HL=USER STACK POINTER
DB41 F9           10350         SPHL
DB42 2A 14 FC     10400         LHLD    USERHL
```

```
DB45 C3 00 FC    10450          JMP     SIM1        ;HERE'S WERE THE ACTUAL
DB48             10500                              ;SINGLE-STEP IS DONE
DB48 E1          10550 EXRET:  POP     H           ;POP USER RET ADR TO HL
DB49 22 2D FC    10600 FINISH:SHLD     HLBUF       ;SAVE NEW POINTER
DB4C 21 00 00    10650          LXI     H           000000Q ;SAVE USER SP
DB4F 39          10700          DAD     SP
DB50 22 5B FC    10750          SHLD    USERSP
DB53 B7          10800 FINISH1:         ORA         A ;CLEAR ERR FLAG
DB54 2A 1A FC    10850 FINISH2:         LHLD        EXECSP ;RETREIVE EXEC STACK POINTE
R
DB57 F9          10900          SPHL
DB58 C9          10950          RET
DB59 2A 14 FC    11000 EXPCHL:LHLD     USERHL      ;GET USER HL
DB5C C3 49 DB    11050          JMP     FINISH
DB5F 2A 2D FC    11100 MET:    LHLD     HLBUF       ;GET USER ADR
DB62 7E          11150          MOV     A           M ;MOVE OP CODE TO A
DB63 E6 07       11200          ANI     007Q        ;MASK LAST DIGIT
DB65 CA 48 DB    11250          JZ      EXRET       ;JMP IF COND RET
DB68 FE 02       11300          CPI     002Q
DB6A CA 74 DB    11350          JZ      EXJMP       ;JMP IF COND JMP
DB6D             11400                              ;MUST BE COND CALL BY DEFAULT
DB6D 23          11450 EXCALL:INX      H           ;PUSH CURRENT ADR+3 ON USER STACK
DB6E 23          11500          INX     H
DB6F 23          11550          INX     H
DB70 E5          11600          PUSH    H
DB71 2B          11650          DCX     H
DB72 2B          11700          DCX     H
DB73 2B          11750          DCX     H
DB74 23          11800 EXJMP:  INX     H           ;MOVE <B2> AND <B3> TO HL
DB75 5E          11850          MOV     E           M
DB76 23          11900          INX     H
DB77 66          11950          MOV     H           M
DB78 6B          12000          MOV     L           E
DB79 C3 49 DB    12050          JMP     FINISH
DB7C 23          12100 EXRST:  INX     H           ;SIMULATE A RESTART
DB7D E5          12150          PUSH    H
DB7E 78          12200          MOV     A           B
DB7F E6 38       12250          ANI     070Q
DB81 6F          12300          MOV     L           A
DB82 26 00       12350          MVI     H           000Q
DB84 C3 49 DB    12400          JMP     FINISH
```

```
DB87 78          12450 EXCOND:MOV   A        B ;GET OP CODE
DB88 E6 F8       12500        ANI   370Q
DB8A F6 02       12550        ORI   002Q
DB8C 32 00 FC    12600        STA   SIM1
DB8F 21 5F DB    12650        LXI   H        MET ;SAVE HI&LO ADR OF 'MET' ROUTI
NE
DB92 22 01 FC    12660        SHLD  SIM2
DB95 21 9E DB    12850        LXI   H        NOTMET ;SAVE LO/HI OF NOTMET
DB98 22 04 FC    12900        SHLD  SIM5
DB9B C3 38 DB    12950        JMP   DOINST   ;DO THE CONDITIONAL
DB9E 2A 2D FC    13000 NOTMET:LHLD  HLBUF    ;HL=ADR VARS
DBA1 7E          13050        MOV   A        M
DBA2 23          13100        INX   H
DBA3 E6 07       13150        ANI   007Q     ;MASK LAST DIGIT
DBA5 CA 49 DB    13200        JZ    FINISH   ;JMP IF COND RET
DBA8 23          13250        INX   H
DBA9 23          13300        INX   H
DBAA C3 49 DB    13350        JMP   FINISH
DBAD 22 14 FC    13400 ENDSTP:SHLD  USERHL   ;SAVE USER HL
DBB0 21 53 DB    13450        LXI   H        FINISH1
DBB3 22 0B FC    13500        SHLD  VECT2
DBB6 C3 CC DE    13550        JMP   SAVEM
DBB9 37          14150 ERRERR:STC             ;SET ERR FLAG
DBBA C3 54 DB    14200        JMP   FINISH2
DBBD             14250        ;
DBBD             14300        ;
DBBD             17550        ;
DBBD             17600        ;
DBBD D5          17650 SWAP1: PUSH  D        ;SWAP [BC] WITH [HL] 3 TIMES
DBBE 11 03 00    17700        LXI   D        000003Q
DBC1 CD C6 DB    17750        CALL  SWAP2
DBC4 D1          17800        POP   D
DBC5 C9          17850        RET
DBC6 7E          17900 SWAP2: MOV   A        M ;SWAOP [BC] WITH [HL] DE TIMES
DBC7 F5          17950        PUSH  PSW
DBC8 0A          18000        LDAX  B
DBC9 77          18050        MOV   M        A
DBCA F1          18100        POP   PSW
DBCB 02          18150        STAX  B
```

```
DBCC 23          18200          INX    H
DBCD 03          18250          INX    B
DBCE 1B          18300          DCX    D
DBCF 7B          18350          MOV    A          E
DBD0 B7          18400          ORA    A
DBD1 C2 C6 DB    18450          JNZ    SWAP2
DBD4 C9          18500          RET
DBD5             18550          ;
DBD5             18600          ;
DBD5 0A          18650 COPYBTOH: LDAX   B ;COPY [BC] TO [HL] UNTIL BC=DE I
NCLUSIVE
DBD6 77          18700          MOV    M          A
DBD7 79          18750          MOV    A          C
DBD8 BB          18800          CMP    E
DBD9 CA E1 DB    18850          JZ     STUCK
DBDC 03          18900          INX    B
DBDD 23          18950          INX    H
DBDE C3 D5 DB    19000          JMP    COPYBTOH
DBE1 78          19050 STUCK:   MOV    A          B
DBE2 BA          19100          CMP    D
DBE3 C2 D5 DB    19150          JNZ    COPYBTOH
DBE6 C9          19200          RET
DBE7             19250          ;
DBE7             19300          ;
DBE7             19350          ;AUX MODE COPY FUNCTION:
DBE7 CD 2D DE    19400 COPY:    CALL   SRCTST     ;TEST FOR SRC, DES, AND LEN STATES

DBEA CD 3F DE    19450          CALL   DESTST     ;WILL NOT RETURN IF STATE OFF
DBED CD 46 DE    19500          CALL   LENTST
DBF0 21 92 E0    19550          LXI    H          MSGBLO ;OUTPUT 'BLOCKS'
DBF3 CD C0 D9    19600          CALL   MSGAT0
DBF6 CD F3 DC    19650          CALL   SETTASKC   ;MAKE SURE MESSAGE IS NOT WIPED O
T
DBF9 2A 49 FC    19700          LHLD   SRC        ;MOVE SRC MEM ADR TO DE
DBFC 2E 00       19750          MVI    L          000Q
DBFE EB          19800          XCHG
DBFF 2A 4B FC    19850          LHLD   DES        ;MOVE DES MEM ADR TO HL
DC02 6B          19900          MOV    L          E ;SET LO ADR=0 (E ALREADY SET T
0)
DC03 3A 4E FC    19950          LDA    LEN+001Q   ;STORE HI LEN IN B
```

```
DC06 47          20000         MOV    B          A
DC07 4D          20050         MOV    C          L ;SET C=0 (L ALREADY=0 );C HOLDS B
LOCK COUNT FOR DISPLAY
DC08 1A          20100         LOOP2: LDAX       D ;DO THE COPY HERE:
DC09 77          20150         MOV    M          A
DC0A BE          20200         CMP    M
DC0B C2 20 DC    20250         JNZ    NOMEM      ;JMP IF BAD WRITE
DC0E 23          20300         INX    H
DC0F 13          20350         INX    D
DC10 7D          20400         MOV    A          L
DC11 B7          20450         ORA    A
DC12 C2 08 DC    20500         JNZ    LOOP2
DC15 0C          20550         INR    C
DC16 C5          20600         PUSH   B
DC17 CD 40 D9    20650         CALL   DATOUT
DC1A C1          20700         POP    B
DC1B 05          20750         DCR    B
DC1C C2 08 DC    20800         JNZ    LOOP2
DC1F C9          20850         RET
DC20 21 A4 E0    20900 NOMEM:  LXI    H          MSGNOT ;OUTPUT 'NOT.STORED'
DC23 06 09       21000 MSG8:   MVI    B          011Q ;9 CHARS IN MESSAGE
DC25 CD C2 D9    21010         CALL   MSGAT0+002Q ;OUTPUT MESSAGE
DC28 C3 F3 DC    21050         JMP    SETTASKC
DC2B             21100         ;
DC2B             21150         ;
DC2B             21200         ;MEM/REG MODE AUX FUNCTION
DC2B 21 44 DC    21250 AUX:    LXI    H          AUXOUTA ;STORE AUX UPDATE
DC2E 22 2F FC    21300         SHLD   DSPUP
DC31 21 35 DF    21350         LXI    H          TABLEAUX ;STORE AUX COMMAND TABLE
DC34 22 31 FC    21400         SHLD   TABLE
DC37             21450                          ;FALLS THRU TO CANCLEAUX
DC37             21500         ;
DC37             21550         ;
DC37 2A 3C FC    21600 CANCLEAUX:    LHLD       AUXSTATE ;GET LAST AUX STATE
DC3A E5          21650         PUSH   H          ;AND DATA
DC3B 6E          21700         MOV    L          M
DC3C 5E          21750         MOV    E          M
DC3D E1          21800         POP    H
DC3E 23          21850         INX    H          ;STORE IN AUX BUFFERS
```

```
DC3F 56          21900          MOV    D          M
DC40 EB          21950          XCHG              ;MOVE DE TO HL
DC41 22 2B FC    22000          SHLD   STATEBUF   ;FALLS THRU TO AUXOUTA
DC44             22050          ;
DC44             22100          ;
DC44 CD 73 DA    22150 AUXOUTA: CALL   PRT2TST ;OUTPUT DATA ON PORT 2
DC47 2A 3A FC    22200          LHLD   AUXMESPTR ;GET AUX MESSAGE POINTER
DC4A CD C0 D9    22250          CALL   MSGAT0    ;OUTPUT MESSAGE
DC4D EB          22300          XCHG
DC4E 2A 2B FC    22350          LHLD   STATEBUF   ;GET AUX STATE OFFSET BUFFER TO !
DC51 26 00       22450          MVI    H          000Q ;SET HI BYTE FOR 2-PREC. ADD
DC53 19          22500          DAD    D          ;ADD OFFSET TO START OF STATE TABL
E
DC54 5E          22550          MOV    E          M ;GET LO BYTE OF STATE ROUTINE AD
R
DC55 23          22600          INX    H          ;INR POINTER TO HI BYTE
DC56 66          22650          MOV    H          M ;GET HI BYTE
DC57 6B          22700          MOV    L          E ;MOVE LO BYTE TO L (H&L ARE NOW
SET UP)
DC58 E9          22750          PCHL   ;DO THE ROUTINE
DC59 2A 3C FC    22800 PREVAUX: LHLD   AUXSTATE ;CHECK FOR ROLL-OVER
DC5C 3E 43       22850          MVI    A          BR1H
DC5E BD          22900          CMP    L
DC5F C2 71 DC    22950          JNZ    NOTSTRAUX ;JMP IF NO
DC62 21 C3 DF    23000          LXI    H          MBAUD ;ELSE RESET POINTER
DC65 22 3A FC    23050          SHLD   AUXMESPTR
DC68 21 59 FC    23100          LXI    H          BAUD
DC6B 22 3C FC    23150 PREVAUX1: SHLD   AUXSTATE ;ENTRY POINT FOR BELOW
DC6E C3 37 DC    23200          JMP    CANCLEAUX
DC71 2A 3A FC    23250 NOTSTRAUX: LHLD  AUXMESPTR ;SUB 10 FROM MESSAGE PNT
ER
DC74 11 F6 FF    23300          LXI    D          377366Q ;THIS IS -12
DC77 19          23350          DAD    D          ;ACTUALLY ADDING -12 TO PNTER
DC78 22 3A FC    23400          SHLD   AUXMESPTR ;STORE NEW POINTER
DC7B 2A 3C FC    23450          LHLD   AUXSTATE  ;SUBTRACT 2 FROM STATE POINTER
DC7E 2B          23500          DCX    H
DC7F 2B          23550          DCX    H
DC80 C3 6B DC    23600          JMP    PREVAUX1
DC83             23700          ;
DC83             23750          ;
```

```
DC83                23800         ;SOURCE CONTINUES IN EXECB3
EXECB3
DC83                100           ;FILE: EXECB3
DC83                120           ;
DC83                140           ;
DC83 2A 2B FC       160   STOREAUX:     LHLD      STATEBUF ;GET AUX DATA & STATE BU
FER
DC86 EB             180           XCHG    ;MOVE DATA TO D, STATE TO E
DC87 2A 3C FC       200           LHLD    AUXSTATE  ;GET AUX DATA & STATE POINTER
DC8A E5             220           PUSH    H         ;SAVE FOR LATER
DC8B 6E             240           MOV     L         M
DC8C 73             260           MOV     M         E ;STORE STATE
DC8D E1             280           POP     H
DC8E 23             300           INX     H         ;MOVE (HL) TO DATA
DC8F 72             320           MOV     M         D ;STORE DATA
DC90 C3 DC D9       340           JMP     DODASH
DC93                360           ;
DC93                380           ;
DC93                400           ;AUX MODE NEXT FUNCTION:
DC93 2A 3C FC       420   NEXTAUX:       LHLD      AUXSTATE
DC96 3E 59          440           MVI     A         BAUD
DC98 BD             460           CMP     L
DC99 C2 A8 DC       480           JNZ     NOTENDAUX
DC9C 21 55 DF       500           LXI     H         MBR1H
DC9F 22 3A FC       520           SHLD    AUXMESPTR
DCA2 21 43 FC       540           LXI     H         BR1H
DCA5 C3 6B DC       560           JMP     PREVAUX1  ;JMP TO POINT TO: SHLD AUXSTATE, J
MP .CANCLEAUX
DCA8 2A 3A FC       600   NOTENDAUX:      LHLD      AUXMESPTR ;ADD 12 TO AUX MESSAGE P
NTER
DCAB 11 0A 00       620           LXI     D         000012Q
DCAE 19             640           DAD     D
DCAF 22 3A FC       660           SHLD    AUXMESPTR ;STORE NEW PNTER
DCB2 2A 3C FC       680           LHLD    AUXSTATE  ;ADD 2 TO AUX STATE PNTER
DCB5 23             700           INX     H
DCB6 23             720           INX     H
DCB7 C3 6B DC       740           JMP     PREVAUX1  ;JMP TO POINT TO:SHLD AUX STATE, J
MP CANCLEAUX
DCBA                780           ;
DCBA                800           ;
```

```
DCBA                    820          ;AUX MODE OPTION FUNCTION
DCBA 3A 2B FC   840  OPTIONAUX:  LDA      STATEBUF
DCBD 3C         860          INR     A
DCBE 3C         880          INR     A
DCBF E6 02      900          ANI     002Q
DCC1 32 2B FC   920          STA     STATEBUF
DCC4 2A 3C FC   940          LHLD    AUXSTATE
DCC7 23         950          INX     H
DCC8 7E         960          MOV     A          M
DCC9 32 2C FC   970          STA     DATBUF
DCCC C3 44 DC   980          JMP     AUXOUTA
DCCF            1021         ;
DCCF            1022         ;
DCCF            1023         ;FOLLOWING ARE VARIOUS ROUTINES USED IN
DCCF            1024         ;AUX MODE FOR OUTPUTING VARIOUS STATES
DCCF 21 7D E0   1040 PRTOFF:LXI     H          MSGOFF ;OUTPUT 'OFF'
DCD2 C3 B0 D9   1060         JMP     MSGAT6
DCD5 21 80 E0   1080 PRTON: LXI     H          MSGON ;OUTPUT 'ON'
DCD8 C3 B0 D9   1100         JMP     MSGAT6
DCDB 21 86 E0   1140 PRTRAM:LXI     H          MSGRAM ;OUTPUT 'RAM'
DCDE C3 B0 D9   1160         JMP     MSGAT6
DCE1 21 83 E0   1200 PRTROM:LXI     H          MSGROM ;OUTPUT 'ROM'
DCE4 C3 B0 D9   1220         JMP     MSGAT6
DCE7 21 89 E0   1240 PRT708:LXI     H          MSG708 ;OUTPUT '708'
DCEA C3 B0 D9   1260         JMP     MSGAT6
DCED 21 8C E0   1280 PRT716:LXI     H          MSG716 ;OUTPUT '716'
DCF0 C3 B0 D9   1300         JMP     MSGAT6
DCF3 21 ED DE   1320 SETTASKC:       LXI        H TASKC ;SET TASK POINTER=TASK C
DCF6 22 33 FC   1340         SHLD    TASK
DCF9 C9         1360         RET
DCFA            1380         ;
DCFA            1400         ;
DCFA            1420         ;AUX MODE LOAD FUNCTION:
DCFA 21 AD E0   1440 LOAD:   LXI     H          MSGLOADIN ;OUTPUT 'LOADING...'
DCFD CD 23 DC   1460         CALL    MSG8
DD00 CD 5C E1   1500         CALL    CLOAD
DD03 3A 63 FC   1520         LDA     SABACK     ;GET HI ORDER START ADR LOADED
DD06 4F         1525         MOV     C          A
DD07 32 4A FC   1540         STA     SRC+001Q   ;SAVE START ADR LOADED
DD0A C5         1550         PUSH    B          ;SAVE FOR COMPUTATION LATER
```

```
DD0B 06 D0       1560          MVI    B          320Q ;B=CODE TO CLEAR DISPLAYS
DD0D CD 43 D9    1570          CALL   DSPLY      ;SHOW STARTING ADR
DD10 21 8F E0    1590          LXI    H          MSGTO ;OUTPUT ' TO.'
DD13 3E 93       1600          MVI    A          223Q ;8279 COMMAND
DD15 CD B2 D9    1610          CALL   MSGAT6+002Q ;OUTPUT CHARS
DD18 3A 22 FC    1620          LDA    EA         ;GET END ADR LOADED
DD1B 4F          1623          MOV    C          A ;SHOW ENDING ADR
DD1C CD 48 D9    1625          CALL   DSPLY1
DD1F C1          1626          POP    B          ;RESTORE HI ORDER ADR LOADED IN C
DD20 91          1630          SUB    C          ;COMPUTE LEN (START ADR IN C)
DD21 32 4E FC    1640          STA    LEN+001Q   ;STORE IN USER LEN REGISTER
DD24 3E 02       1670          MVI    A          002Q ;SET UP STATE OFFSETS
DD26             1680                           ;TO OPEN REGISETERS
DD26 32 66 FC    1690          STA    SRCSO      ;TURN ON DES REG
DD29 32 68 FC    1700          STA    LENSO      ;TURN ON LEN REG
DD2C C9          1710          RET
DD2D             1820          ;
DD2D             1840          ;
DD2D             1860          ;AUX MODE DUMP FUNCTION:
DD2D CD 2D DE    1880  DUMP:   CALL   SRCTST     ;TEST SRC AND LEN REGS
DD30 CD 46 DE    1900          CALL   LENTST
DD33 21 B6 E0    1920          LXI    H          MSGDUMPIN ;OUTPUT 'DUMPING...'
DD36 CD 23 DC    1940          CALL   MSG8
DD39 3A 4A FC    1980          LDA    SA         ;COMPUTE ENDING ADR
DD3C 47          2000          MOV    B          A
DD3D 3A 4E FC    2020          LDA    LEN+001Q
DD40 80          2040          ADD    B
DD41 32 22 FC    2060          STA    EA
DD44 CD 1A E1    2080          CALL   CDUMP
DD47 21 BF E0    2100          LXI    H          MSGDUDONE
DD4A C3 23 DC    2120          JMP    MSG8
DD4D             2140          ;
DD4D             2160          ;
DD4D             2180          ;AUX MODE PROM FUNCTION:
DD4D 3A 6C FC    2200  PROM:   LDA    PRMSO      ;PRMSO=000 FOR 2708, AND 002 FOR 2
716
DD50 0F          2240          RRC               ;MOVE PROM TYPE BIT (BIT 1) TO BIT
 7
DD51 0F          2260          RRC               ;TO HANDSHAKE W/UTILITY
DD52 32 23 FC    2280          STA    CONTROL    ;CONTROL IS USED BY PROM
```

```
DD55                2281                                ;UTILITY TO PASS PARAMETERS
DD55 3A 69 FC       2300        LDA     CLRSO           ;GET CLEAR STATE OFFSET (0=OFF,
ON)
DD58 B7             2320        ORA     A               ;SET FLAGS
DD59 C4 71 DD       2340        CNZ     DOCLR           ;DO A CLEAR IF CLEAR ON
DD5C 3A 6A FC       2360        LDA     POPSO           ;GET POP STATE OFFSET (0=OFF, 2=ON
)
DD5F B7             2380        ORA     A               ;SET FLAGS
DD60 C4 2D DE       2400        CNZ     SRCTST          ;IF POP ON, MAKE SURE SRC REG
DD63                2401                                ;IS NOT 'OFF' AND...
DD63 C4 96 DD       2420        CNZ     DOPOP           ;... IF POP ON, DO A POP!
DD66 3A 6B FC       2440        LDA     DUPSO           ;GET DUP STATE OFFSET (0=OFF, 2=ON
)
DD69 B7             2460        ORA     A               ;SET FLAGS
DD6A C4 2D DE       2480        CNZ     SRCTST          ;IF DUP ON, MAKE SURE SCR REG
DD6D                2481                                ;IS NOT OFF AND...
DD6D C4 A5 DD       2500        CNZ     DODUP           ;... DO A DUP
DD70 C9             2520        RET                     ;END OF PROM LINK
DD71                2521        ;CLEAR (BLANK) TEST HAND-SHAKE ROUTINE:
DD71 3A 23 FC       2540  DOCLR: LDA     CONTROL         ;BIT 7 OF CONTROL=0 FOR 2708 AND
 FOR 2716
DD74 F6 01          2560        ORI     001Q            ;IF BIT 0 OF CONTROL = 1, THEN PRO
M
DD76                2561                                ;UTILITY WILL DO A BLANK TEST
DD76 32 23 FC       2580        STA     CONTROL
DD79 21 98 E0       2600        LXI     H               MSGONES ;OUTPUT 'ONES...'
DD7C CD C0 D9       2620        CALL    MSGAT0
DD7F CD 56 E2       2640        CALL    VERIFY          ;CALL VERIFY ONES ROUTINE IN PROM
UTILITY
DD82 3A 2A FC       2660        LDA     STATUS          ;GET RESULT OF ONES TEST
DD85 B7             2680        ORA     A               ;ZERO=GOOD, NONZERO=BAD
DD86 C2 8F DD       2700        JNZ     CLRBAD          ;JMP IF BAD
DD89 21 C8 E0       2720  CLRGOOD:     LXI             H MSGONEGOD ;OUTPUT 'ONES GOOD'
DD8C C3 23 DC       2740              JMP     MSG8
DD8F D1             2760  CLRBAD:POP     D               ;POP RET ADR OFF STACK SO WE
DD90                2761                                ;WE WON'T DO REST OF PROM SEQUENCE
 (DUP TEST)
DD90 21 D1 E0       2780        LXI     H               MSGONEBAD ;OUTPUT 'ONES  BAD'
DD93 C3 23 DC       2800        JMP     MSG8
DD96 21 DA E0       2820  DOPOP: LXI     H               MSGPOPING ;OUTPUT 'POPPING...'
DD99 CD 23 DC       2840        CALL    MSG8
```

```
DD9C CD B4 E2     2860          CALL    POPPER     ;POP PROM
DD9F 21 E4 E0     2880          LXI     H          MSGPOPDON ;OUTPUT 'POP  DONE'
DDA2 C3 23 DC     2900          JMP     MSG8
DDA5 3A 6C FC     2920  DODUP:  LDA     PRMSO      ;PRMSO=000 FOR 2708 AND 002 FOR 27
16
DDA8 0F           2940          RRC                ;MOVE PROM TYPE BIT TO BIT 7
DDA9 0F           2960          RRC
DDAA 32 23 FC     2980          STA     CONTROL    ;IF BIT 0 OF CONTROL=0, DO A DATA
TEST
DDAD 21 9E E0     3000          LXI     H          MSGDATA ;OUTPUT 'DATA...'
DDB0 CD C0 D9     3020          CALL    MSGAT0
DDB3 CD 56 E2     3040          CALL    VERIFY     ;CALL VERIFY DATA ROUTINE IN PRO
UTILITY
DDB6 3A 2A FC     3060          LDA     STATUS     ;GET RESULT OF TESTING (NONZERO ME
ANS BAD)
DDB9 B7           3080          ORA     A          ;SET FLAGS
DDBA 21 ED E0     3100          LXI     H          MSGDATGOD ;SET UP 'DATA GOOD' MESS
AGE
DDBD CA 23 DC     3120          JZ      MSG8       ;JMP IF DATA GOOD (STATUS=0 FOR GO
OD)
DDC0 21 F6 E0     3130          LXI     H          MSGDATBAD ;ELSE SET UP 'DATA  BA.
 MESSAGE
DDC3 C3 23 DC     3140          JMP     MSG8       ;OUTPUT THIS MESSAGE
DDC6              3185          ;
DDC6              3186          ;
DDC6              3187          ;
DDC6              3188          ;ROUTINE TO HANDLE DISPLAYING PROPER BAUD
DDC6              3189          ;RATE AND INCREMENT BAUD POINTER WHEN 'OPTION'
DDC6              3190          ;KEY IS STRUCK FOLLOWS:
DDC6 3A 1F FC     3200  DOBAUD: LDA     LASTKEY    ;SEE IF WE 'RE DOING AN OPTION
DDC9 FE 14        3220          CPI     KEYOPTION
DDCB 3E 00        3222          MVI     A          000Q ;FAKE OUT LAST KEY
DDCD 32 1F FC     3224          STA     LASTKEY
DDD0 2A 3E FC     3240          LHLD    BAUDPTR    ;GET CURRENT BAUD RATE PNTER
DDD3 C2 E6 DD     3260  INRBAUD:        JNZ        DSPLYBAUD ;JMP IF NO OPTION (JUST
DISPLAY)
DDD6 3E 66        3280          MVI     A          BAUD1200 ;SEE IF WE'RE ALREADY DSP
LYING LAST BAUD
DDD8 BD           3300          CMP     L
```

```
DDD9 C2 DF DD    3320          JNZ     INRBAUD1  ;JMP IF NO
DDDC 21 34 E0    3340  RESBAUD:       LXI       H BAUD110-012Q ;RESET POINTER TO 1
ST BAUD RATE-12
DDDF 11 0A 00    3342  INRBAUD1:      LXI       D 000012Q ;ADD 12 TO BAUD POINTER
DDE2 19          3343          DAD     D
DDE3 22 3E FC    3344          SHLD    BAUDPTR   ;STORE NEW POINTER
DDE6 3E 95       3346  DSPLYBAUD:    MVI       A 225Q ;A=8279 COMMAND
DDE8 06 04       3380          MVI     B         004Q ;B=# CHARS
DDEA CD B4 D9    3400          CALL    MSGGEN    ;OUTPUT MESSAGE
DDED             3405          ;WE WANT BC=SOURCE FOR COPY, DE=END SOURCE FOR COPY

DDED             3406          ;AND HL=EQUAL DESTINATION FOR COPY
DDED 44          3420          MOV     B         H ;COPY NEW BAUD CONSTANTS TO SCRA
TCHPAD
DDEE 4D          3440          MOV     C         L
DDEF 11 05 00    3460          LXI     D         000005Q ;# TIMES FOR COPY
DDF2 19          3480          DAD     D         ;CALCULATE ENDING ADR FOR COPY
DDF3 EB          3500          XCHG
DDF4 21 5D FC    3520          LXI     H         TYHLFBD
DDF7 C3 D5 DB    3540          JMP     COPYBTOH
DDFA             3541          ;
DDFA             3542          ;
DDFA             3543          ;
DDFA             3544          ;THE FOLLOWING ROUTINE IS THE ENTRY POINT FOR
DDFA             3545          ;BREAK POINTS.  THAT MEANS, WHEN EVER BREAK POINT.1

DDFA             3546          ;IS ENCOUNTERED, PROGRAM CONTROL WILL VECTOR HERE B
ECAUSE
DDFA             3547          ;A JUMP WAS PLACED IN THE USER PROGRAM JUST PRIOR T
O A GO
DDFA             3548          ;(SEE 'GOM' ROUTINE).  BREAK POINTS 1 AND 2 ARE IDE
NTICAL
DDFA             3549          ;ACCEPT FOR DIFFERENT MESSAGES BUT CAN NOT BE COMBI
NED FOR
DDFA             3550          ;FOR COMPLEX REASONS.
DDFA 22 14 FC    3560  WEHITBR1:      SHLD      USERHL ;SAVE USER STUFF
DDFD 21 06 DE    3580          LXI     H         PTB
DE00 22 0B FC    3590          SHLD    VECT2
DE03 C3 CC DE    3600          JMP     SAVEM
DE04 31 00 FD    3840  PTB:    LXI     SP        FSCR ;GET EXEC SP SET UP
```

```
DE09 CD 1F DE    3860          CALL   REMOVEBP  ;REMOVE BREAK POINT FOR USER'S PRO
GRAM
DE0C CD 4D DE    3880          CALL   SETUPBR1  ;SET UP SOME VARIABLES FOR BREAK P
OINT 1
DE0F 22 2D FC    3900          SHLD   HLBUF
DE12 CD AC D8    3920          CALL   MEM
DE15 CD 9C DE    3940          CALL   STEPBP
DE18 AF          4000          XRA    A          ;SET LASTKEY LOCATIOJN SO WHEN
DE19             4020                             ;RESET IS STRUCK, BREAK-POINT ISN'
T SWAPPED INTO
DE19             4040                             ;USER PROG IF NO OTHER KEY STRUCK
FIRST.
DE19 32 1F FC    4060          STA    LASTKEY
DE1C C3 89 D8    4080          JMP    MAIN1      ;JMP BACK TO MAIN LOOP
DE1F 3A 64 FC    4100  SETBP:  LDA    BR1SO
DE22 B7          4105          ORA    A
DE23 C8          4110          RZ
DE24             4120  REMOVEBP:   EQU        SETBP
DE24 CD 4D DE    4140          CALL   SETUPBR1
DE27 01 40 FC    4150          LXI    B          BR1USERD
DE2A C3 BD DB    4160          JMP    SWAP1
DE2D F5          4880  SRCTST:PUSH   PSW
DE2E 3A 66 FC    4900          LDA    SRCSO
DE31 B7          4920          ORA    A
DE32 CA 37 DE    4940          JZ     STAOFF
DE35 F1          4960          POP    PSW
DE36 C9          4980          RET
DE37 F1          5000  STAOFF:POP    PSW
DE38 D1          5040          POP    D          ;POP RET ADR
DE39 21 11 E1    5060          LXI    H          MSGSTAOFF
DE3C C3 23 DC    5080          JMP    MSG8
DE3F F5          5100  DESTST:PUSH   PSW
DE40 3A 67 FC    5120          LDA    DESSO
DE43 C3 31 DE    5140          JMP    SRCTST+004Q
DE46 F5          5160  LENTST:PUSH   PSW
DE47 3A 68 FC    5180          LDA    LENSO
DE4A C3 31 DE    5200          JMP    SRCTST+004Q
DE4D 3A 44 FC    5220  SETUPBR1:    LDA        BR1H+001Q
DE50 67          5240          MOV    H          A
```

```
DE51 3A 46 FC    5260          LDA    BR1L+001Q
DE54 6F          5280          MOV    L          A
DE55 C9          5300          RET
DE56 3A 65 FC    6000  STEPD:  LDA    STEPSO     ;STEPSO=0 FOR SINGLE, 2 FOR MULT
DE59 B7          6010          ORA    A          ;SET FLAGS
DE5A C2 6E DE    6020          JNZ    MULTISTEP  ;JMP IF MULTI MODE
DE5D CD AE DA    6030          CALL   STEP       ;DO ONE SINGLE STEP
DE60 DA AA D9    6040          JC     STPERR     ;JMP IF UNDEF. OPCODE
DE63 3A 31 FC    6050  CANCEL:LDA    TABLE      ;DO A CANCEL (MEM OR REG)
DE66 FE F5       6060          CPI    TABLEM     ;MEM MODE?
DE68 CA B8 D8    6070          JZ     CANCLEM    ;DO MEM MODE CANCEL IF YES
DE6B C3 A8 DA    6080          JMP    CANCLR     ;ELSE DO REG MODE CANCEL
DE6E 3E 0C       6090  MULTISTEP:    MVI        A 014Q ;SET 8279 FOR MATRIX MODE
DE70 D3 03       6100          OUT    PORT8279C
DE72 21 68 01    6105          LXI    H          001150Q ;TIME DELAY FOR QUIET DOWN

DE75 CD E5 D9    6110          CALL   TIMEOUT+003Q ;WAIT TO QUIET DOWN 8279 AFTER
SWITCHING MODES ON IT
DE78 DB 03       6120  MORESTEPS:    IN         PORT8279C ;GET FIFO STATUS
DE7A E6 40       6130          ANI    100Q       ;MASK SENSURE CLOSURE BIT
DE7C C8          6140          RZ                ;RET IF NO KEY DOWN ELSE...
DE7D 3E E0       6142          MVI    A          340Q ;SET SENSURE/CLOSURE BIT TO

DE7F D3 03       6144          OUT    PORT8279C ;SO >1 KEY DOESN'T LOCK KEYBOARD
DE81 CD AE DA    6150          CALL   STEP       ;... DO (ANOTHER) SINGLE STEP
DE84 DA AA D9    6160          JC     STPERR     ;JMP IF UNDEF. OPCODE
DE87 CD 63 DE    6170          CALL   CANCEL     ;UPDATE APPROPRIATE VARIABLES
DE8A CD C5 D8    6180          CALL   UPDATE     ;UPDATE APPROPRIATE DISPLAYS
DE8D 2A 47 FC    6190          LHLD   STEP1      ;GET USER TIME DELAY
DE90 23          6200  PT2:    INX    H          ;DO A TIME DELAY
DE91 7C          6210          MOV    A          H
DE92 B7          6220          ORA    A
DE93 C2 90 DE    6230          JNZ    PT2        ;JMP IF NOT DONE WAITING
DE96 CD A2 DE    6240          CALL   BPADRCHK   ;CHECK IF BP ADR=CURRENT HI&LO A

DE99 C3 78 DE    6250          JMP    MORESTEPS  ;JMP IF NO ELSE...
DE9C 21 FF E0    6260  STEPBP:LXI    H          MSGBREAK1 ;OUTPUT 'BREAK PT.1'
DE9F C3 23 DC    6270          JMP    MSG8
DEA2 3A 64 FC    6280  BPADRCHK:     LDA        BR1SO ;0=BP OFF, 2=BP ON
DEA5 B7          6290          ORA    A          ;SET FLAGS
```

```
DEA6 C8          6300          RZ      ;RETURN IF OFF
DEA7 2A 2D FC    6310          LHLD    HLBUF      ;CHECK IF BP ADR=CURRENT H&L ADR
DEAA 3A 44 FC    6320          LDA     BR1H+001Q ;CHEKC HI BYTE
DEAD BC          6330          CMP     H
DEAE C0          6340          RNZ     ;RET IF NOT EQUAL
DEAF 3A 46 FC    6350          LDA     BR1L+001Q ;CHECK LO BYTE
DEB2 BD          6360          CMP     L
DEB3 C0          6370          RNZ     ;RET IF NOT EQUAL
DEB4 D1          6380          POP     D          ;INCREMENT RET ADR BY 3
DEB5 13          6390          INX     D
DEB6 13          6400          INX     D
DEB7 13          6410          INX     D
DEB8 D5          6420          PUSH    D
DEB9 C9          6430          RET                ;DO A RETURN+3
DEBA CD 56 DE    6440  STEPC:  CALL    STEPD
DEBD 3E 08       6450          MVI     A          010Q
DEBF D3 03       6460          OUT     PORT8279C
DEC1 C9          6470          RET
DEC2             6590;
DEC2             6591;
DEC2             6592;
DEC2             6593  ;THE FOLLOWING ROUTINE SETS THE MMD-2 MEMORY MAP
DEC2             6594  ;TO RAM/ROM STATE DEPENDING UPON LAST  S T O R E D
DEC2             6595  ;VALUE IN MEMMAP (MEMORY MAP) AUX REGISTER
DEC2 DB 06       6600  SETMEMMAP:      IN         006Q ;ESTABLISH RAM MAP
DEC4 3A 6D FC    6610          LDA     MAPSO      ;GET MEM MAP STATE OFFSET (0=RAM,2
=ROM)
DEC7 B7          6620          ORA     A          ;SET FLAGS
DEC8 C8          6630          RZ                 ;RETURN IF RAM MEM MAP
DEC9 DB 07       6640          IN      007Q       ;ESTABLISH ROM MAP
DECB C9          6650          RET                ;ALL DONE
DECC             6670  ;
DECC             6680  ;
DECC             6690  ;
DECC             6700  ;
DECC             6710  ;
DECC E1          6740  SAVEM:  POP     H
DECD 22 18 FC    6750          SHLD    USERSC
DED0 F5          6760          PUSH    PSW
DED1 21 00 00    6770          LXI     H          000000Q
```

```
DED4 39          6780          DAD    SP
DED5 22 5B FC    6790          SHLD   USERSP
DED8 F1          6800          POP    PSW
DED9 2A 18 FC    6810          LHLD   USERSC
DEDC E5          6820          PUSH   H
DEDD 31 13 FC    6830          LXI    SP          PUSHRG
DEE0 D5          6840          PUSH   D
DEE1 C5          6850          PUSH   B
DEE2 F5          6860          PUSH   PSW
DEE3 2A 0B FC    6870          LHLD   VECT2
DEE6 E9          6880          PCHL
EXECB4
DEE7             1000          ;FILE: EXECB4
DEE7             1020          ;
DEE7             1040          ;
DEE7             1060          ;FOLLOWING IS THE TASK BLOCK
DEE7 27 D9       1080  TASKA:  DW     RETURN     ;USER DEFINED (HARD)
DEE9 C2 DE       1085  TASKG:  DW     SETMEMMAP  ;SET MEMORY MAP (ROM/RAM)
DEEB C5 D8       1100  TASKB:  DW     UPDATE     ;UPDATE ALL DISPLAYS (ANY MODE)
DEED C9 D8       1120  TASKC:  DW     KEYRD      ;WAIT FOR AN INPUT
DEEF 35 FC       1140  TASKD:  DW     USEREVENT  ;USER DEFINED (SOFT)
DEF1 FE D8       1160  TASKE:  DW     ACCUM      ;DATA RESPONDER
DEF3 99 D8       1180  TASKF:  DW     COMDIS     ;FUNCTION RESPONDER
DEF5             1200          ;
DEF5             1220          ;
DEF5             1240          ;MEMORY MODE KEYS DEFINED HERE:
DEF5 B8 D8       1260  TABLEM: DW     CANCLEM    ;CANCLE KEY
DEF7 2B DC       1280          DW     AUX        ;AUX
DEF9 9C DA       1300          DW     REG        ;REG
DEFB 27 D9       1320          DW     RETURN     ;MEM
DEFD 27 D9       1340          DW     RETURN     ;COPY
DEFF 27 D9       1360          DW     RETURN     ;PROM
DF01 27 D9       1380          DW     RETURN     ;DUMP
DF03 27 D9       1400          DW     RETURN     ;LOAD
DF05 27 D9       1420          DW     RETURN     ;OPTION
DF07 07 DA       1440          DW     GOM        ;GO
DF09 FD D9       1460          DW     LOM        ;LO
DF0B F3 D9       1480          DW     HIM        ;HI
DF0D BA DE       1500          DW     STEPC      ;SETP
DF0F EC D9       1520          DW     NEXTM      ;NEXT
```

```
DF11 D1 D9        1540          DW      STOREM     ;STORE
DF13 C7 D9        1560          DW      PREVM      ;PREV
DF15              1580          ;
DF15              1600          ;
DF15              1620          ;REGISTER MODE KEYS DEFINED:
DF15 A8 DA        1640  TABLER:DW       CANCLR     ;CANCEL
DF17 2B DC        1660          DW      AUX        ;AUX
DF19 27 D9        1680          DW      RETURN     ;REGS
DF1B AC D8        1700          DW      MEM        ;MEM
DF1D 27 D9        1720          DW      RETURN     ;COPY
DF1F 27 D9        1740          DW      RETURN     ;PROM
DF21 27 D9        1760          DW      RETURN     ;DUMP
DF23 27 D9        1780          DW      RETURN     ;LOAD
DF25 27 D9        1800          DW      RETURN     ;OPTION
DF27 07 DA        1820          DW      GOM        ;GO (SAME AS IN MEM MODE)
DF29 27 D9        1840          DW      RETURN     ;LO
DF2B 27 D9        1860          DW      RETURN     ;HI
DF2D BA DE        1880          DW      STEPC      ;STEP
DF2F 8A DA        1900          DW      NEXTR      ;NEXT
DF31 81 DA        1920          DW      STORER     ;STORE
DF33 43 DA        1940          DW      PREVR      ;PREV
DF35              1960          ;
DF35              1980          ;
DF35              2000          ;AUX MODE KEYS DEFINED:
DF35 37 DC        2020  TABLEAUX:       DW         CANCLEAUX ;CANCEL
DF37 27 D9        2040          DW      RETURN     ;AUX
DF39 9C DA        2060          DW      REG        ;REG
DF3B AC D8        2080          DW      MEM        ;MEM
DF3D E7 DB        2100          DW      COPY       ;COPY
DF3F 4D DD        2120          DW      PROM       ;PROM
DF41 2D DD        2140          DW      DUMP       ;DUMP
DF43 FA DC        2160          DW      LOAD       ;LOAD
DF45 BA DC        2180          DW      OPTIONAUX  ;OPTION
DF47 27 D9        2200          DW      RETURN     ;GO
DF49 27 D9        2220          DW      RETURN     ;LO
DF4B 27 D9        2240          DW      RETURN     ;HI
DF4D 27 D9        2260          DW      RETURN     ;STEP
DF4F 93 DC        2280          DW      NEXTAUX    ;NEXT
DF51 83 DC        2300          DW      STOREAUX   ;STORE
DF53 59 DC        2320          DW      PREVAUX    ;PREV
```

```
DF55                    2340            ;
DF55                    2360            ;
DF55                    2380            ;AUX STATE DEFINITION AND TEXT TABLES:
DF55 7C                 2400    MBR1H:  DB      TXTB        ;'BR1 HI.'
DF56 50                 2420            DB      TXTR
DF57 06                 2440            DB      TXT1
DF58 00                 2460            DB      TXTBLK
DF59 74                 2480            DB      TXTH
DF5A 84                 2500            DB      TXTID
DF5B CF DC              2520            DW      PRTOFF      ;STATE 0
DF5D 6C DA              2540            DW      PRTDAT      ;STATE 2
DF5F 7C                 2560    MBR1L:  DB      TXTB        ;'BR1 LO
DF60 50                 2580            DB      TXTR
DF61 06                 2600            DB      TXT1
DF62 00                 2620            DB      TXTBLK
DF63 38                 2640            DB      TXTL
DF64 DC                 2660            DB      TXTOD
DF65 CF DC              2680            DW      PRTOFF
DF67 6C DA              2700            DW      PRTDAT
DF69 6D                 2710    MSTEP:  DB      TXTS        ;'STEP
DF6A 78                 2720            DB      TXTT
DF6B 79                 2730            DB      TXTE
DF6C F3                 2740            DB      TXTPD
DF6D 00                 2750            DB      TXTBLK
DF6E 00                 2760            DB      TXTBLK
DF6F CF DC              2770            DW      PRTOFF
DF71 6C DA              2780            DW      PRTDAT
DF73 6D                 3040    MSRC:   DB      TXTS        ;'SRC HI.'
DF74 50                 3060            DB      TXTR
DF75 58                 3080            DB      TXTC
DF76 00                 3100            DB      TXTBLK
DF77 74                 3120            DB      TXTH
DF78 84                 3140            DB      TXTID
DF79 CF DC              3160            DW      PRTOFF
DF7B 6C DA              3180            DW      PRTDAT
DF7D 5E                 3200    MDES:   DB      TXTD        ;'DES HI.'
DF7E 79                 3220            DB      TXTE
DF7F 6D                 3240            DB      TXTS
DF80 00                 3260            DB      TXTBLK
DF81 74                 3280            DB      TXTH
```

```
DF82 84          3300          DB    TXTID
DF83 CF DC       3320          DW    PRTOFF
DF85 6C DA       3340          DW    PRTDAT
DF87 38          3360  MLEN:   DB    TXTL      ;'LEN HI.'
DF88 79          3380          DB    TXTE
DF89 37          3400          DB    TXTN
DF8A 00          3420          DB    TXTBLK
DF8B 74          3440          DB    TXTH
DF8C 84          3460          DB    TXTID
DF8D CF DC       3480          DW    PRTOFF
DF8F 6C DA       3500          DW    PRTDAT
DF91 58          3520  MCLR:   DB    TXTC      ;'CLRTST.'
DF92 38          3540          DB    TXTL
DF93 50          3560          DB    TXTR
DF94 78          3580          DB    TXTT
DF95 6D          3600          DB    TXTS
DF96 F8          3620          DB    TXTTD
DF97 CF DC       3640          DW    PRTOFF
DF99 D5 DC       3660          DW    PRTON
DF9B 73          3680  MPOP:   DB    TXTP      ;'POPPRM.'
DF9C 5C          3700          DB    TXTO
DF9D 73          3720          DB    TXTP
DF9E 73          3740          DB    TXTP
DF9F 50          3760          DB    TXTR
DFA0 D4          3780          DB    TXTMD
DFA1 CF DC       3800          DW    PRTOFF
DFA3 D5 DC       3820          DW    PRTON
DFA5 5E          3840  MDUP:   DB    TXTD      ;'DUPTST.'
DFA6 1C          3860          DB    TXTU
DFA7 73          3880          DB    TXTP
DFA8 78          3900          DB    TXTT
DFA9 6D          3920          DB    TXTS
DFAA F8          3940          DB    TXTTD
DFAB CF DC       3960          DW    PRTOFF
DFAD D5 DC       3980          DW    PRTON
DFAF 73          4000  MPRM:   DB    TXTP      ;'PROM 2'
DFB0 50          4020          DB    TXTR
DFB1 5C          4040          DB    TXTO
DFB2 54          4060          DB    TXTM
DFB3 00          4080          DB    TXTBLK
```

```
DFB4 5B          4100          DB      TXT2
DFB5 E7 DC       4120          DW      PRT708
DFB7 ED DC       4140          DW      PRT716
DFB9 54          4160  MMAP:   DB      TXTM        ;'MEMMAP.'
DFBA 79          4180          DB      TXTE
DFBB 54          4200          DB      TXTM
DFBC 54          4220          DB      TXTM
DFBD 77          4240          DB      TXTA
DFBE F3          4260          DB      TXTPD
DFBF DB DC       4280          DW      PRTRAM
DFC1 E1 DC       4300          DW      PRTROM
DFC3 7C          4320  MBAUD:  DB      TXTB        ;'BAUD
DFC4 77          4340          DB      TXTA
DFC5 1C          4360          DB      TXTU
DFC6 5E          4380          DB      TXTD
DFC7 00          4400          DB      TXTBLK
DFC8 00          4420          DB      TXTBLK
DFC9 C6 DD       4440          DW      DOBAUD
DFCB C6 DD       4460          DW      DOBAUD
DFCD             4480          ;
DFCD             4500          ;
DFCD             4520          ;
DFCD             4540          ;THE FOLLOWING CONSTANTS ARE COPIED INTO
DFCD             4560          ;SCRATCHPAD DURING PUP.  A SUBSET OF THESE
DFCD             4580          ;VARIABLES ARE COPIED IN DURING A RESET...
DFCD 00 03       4600  STRVAR:DW      DEFLTADD    ;HLBUF    DEFAULT ADR VARS (000 00
3)
DFCF 85 D9       4620          DW      HLDOUT      ;DSPUP    UPDATE ADR FOR MEM MODE
DFD1 F5 DE       4640          DW      TABLEM      ;TABLE    MEM MODE ROUTINES TABLE
DFD3 E7 DE       4660          DW      TASKA       ;TASK  ADR OF 1ST TASK IN TASK B)
CK
DFD5 C9          4680          DB      311Q        ;USEREVENTDEFAULT USER DEFINED EVE
NT (A RET)
DFD6 00 00       4700          DW      0000H       ;          2ND 2 BYTES OF USER EVE
NT
DFD8 06 E0       4720          DW      MSGA        ;DSPREG    DEFAULT REG MODE REG
DFDA 55 DF       4740          DW      MBR1H       ;AUXMESPTR ;DEFAULT AUX DSPLY REG
DFDC 43 FC       4760  SAUX:   DW      BR1H        ;AUXSTATE DEFAULT AUX DSPLY REG
DFDE 52 E0       4780          DW      BAUD300     ;BAUDPTR  BAUD RATE POINTER
DFE0 C3 FA DD    4800  BR1DAT:JMP     WEHITBR1    ;BR1USERD JMP SWAPPED WITH USER DA
TA FOR BREAK POINTS
```

```
DFE3 64          4840  SBR1H: DB     BR1SO     ;BREAK POINT 1 STATE OFFSET POIN
R
DFE4 00          4860         DB     00H       ;DATA VALUE
DFE5 64          4880  SBR1L: DB     BR1SO     ;BREAK POINT 1
DFE6 00          4900         DB     00H
DFE7 65          4910  SSTEP: DB     STEPSO    ;STEP
DFE8 00          4920         DB     00H
DFE9 66          5000  SSRC:  DB     SRCSO     ;STATE
DFEA 00          5020         DB     00H
DFEB 67          5040  SDES:  DB     DESSO     ;DESTINATION
DFEC 00          5060         DB     00H
DFED 68          5080  SLEN:  DB     LENSO     ;LENGTH
DFEE 00          5100         DB     00H
DFEF 69          5120  SCLR:  DB     CLRSO     ;CLEAR
DFF0 00          5140         DB     00H
DFF1 6A          5160  SPOP:  DB     POPSO     ;POP
DFF2 00          5180         DB     00H
DFF3 6B          5200  SDUP:  DB     DUPSO     ;DUPLICATION
DFF4 00          5220         DB     00H
DFF5 6C          5240  SPRM:  DB     PRMSO     ;PROM TYPE (2708/2716)
DFF6 00          5260         DB     00H
DFF7 6D          5280  SMAP:  DB     MAPSO     ;MEM MAP
DFF8 00          5300         DB     00H
DFF9 6E          5320  SBAUD: DB     BAUDSO    ;BAUD
DFFA 00          5340         DB     00H
DFFB FE 03       5360         DW     DEFLTUSP  ;DEFAULT USER STACK POINTER
DFFD 2F 01       5380         DW     001057Q   ;HALF BAUD TIMER (300 BAUD)
DFFF 63 01       5400         DW     001143Q   ;ONE BAUD INPUT TIMER
E001 63 01       5420  ENDVAR:DW     001143Q   ;ONE BAUD OUTPUT TIMER
E003             5440         ;
E003             5460         ;
E003             5480         ;REGISTER TEXT & LOCATION TABLE.
E003             5500         ;FORMAT FOR THIS TABLE IS:
E003             5520         ;   1ST BYTE: LO ADR OF REG CONTENTS
E003             5540         ;   2ND BYTE: 1ST CHAR OF REG NAME
E003             5560         ;   3RD BYTE: 2ND CHAR OF REG NAME
E003 0D          5580  MSGFL: DB     REGFL     ;REGISTER FLAGS
E004 71          5600         DB     TXTF      ;'FL.'
E005 B8          5620         DB     TXTLD
E006 0E          5640  MSGA:  DB     REGA      ;REGISTER A
```

```
E007 F7          5660          DB      TXTAD       ;'A. '
E008 00          5680          DB      TXTBLK
E009 10          5700  MSGB:   DB      REGB        ;REGISTER B
E00A FC          5720          DB      TXTBD       ;'B. '
E00B 00          5740          DB      TXTBLK
E00C 0F          5760  MSGC:   DB      REGC        ;REGISTER C
E00D D8          5780          DB      TXTCD       ;'C. '
E00E 00          5800          DB      TXTBLK
E00F 12          5820  MSGD:   DB      REGD        ;REGISTER D
E010 DE          5840          DB      TXTDD       ;'D. '
E011 00          5860          DB      TXTBLK
E012 11          5880  MSGE:   DB      REGE        ;REGISTER E
E013 F9          5900          DB      TXTED       ;'E. '
E014 00          5920          DB      TXTBLK
E015 15          5940  MSGH:   DB      USERHL+1    ;REGISTER H
E016 F4          5960          DB      TXTHD       ;'H. '
E017 00          5980          DB      TXTBLK
E018 14          6000  MSGL:   DB      USERHL      ;REGISTER L
E019 B8          6020          DB      TXTLD       ;'L. '
E01A 00          6040          DB      TXTBLK
E01B 5C          6060  MSGSH:  DB      USERSP+1    ;REGISER SP HI
E01C 6D          6080          DB      TXTS        ;'SH.'
E01D F4          6100          DB      TXTHD
E01E 5B          6120  MSGSL:  DB      USERSP      ;REGISER SP LO
E01F 6D          6140          DB      TXTS        ;'SL.'
E020 B8          6160          DB      TXTLD
E021             6180          ;
E021             6200          ;
E021             62 0          ;
E021             6240          ;TABLE OF UNDEFINED 8080 CODES
E021             6260          ;USED BY SINGLE-STEPPER
E021 CB          6280  UNDTBL:DB      313Q
E022 D9          6300          DB      331Q
E023 DD          6320          DB      335Q
E024 ED          6340          DB      355Q
E025 FD          6360          DB      375Q
E026 08          6363          DB      010Q
E027 10          6364          DB      020Q
E028 18          6365          DB      030Q
E029 20          6366          DB      040Q
```

```
E02A 28          6367          DB      050Q
E02B 30          6368          DB      060Q
E02C 38          6369          DB      070Q
E02D 00          6380          DB      000Q       ;INDICATES END OF TABLE
E02E             6400          ;
E02E             6420          ;
E02E             6440          ;SEVEN SEGMENT CODE TABLE
E02E 3F          6460  SSEGTB: DB      TXT0       ;0
E02F 06          6480          DB      TXT1       ;1
E030 5B          6500          DB      TXT2       ;2
E031 4F          6520          DB      TXT3       ;3
E032 66          6540          DB      TXT4       ;4
E033 6D          6560          DB      TXT5       ;5
E034 7D          6580          DB      TXT6       ;6
E035 07          6600          DB      TXT7       ;7
E036 7F          6620          DB      TXT8       ;8
E037 6F          6640          DB      TXT9       ;9
E038 77          6660          DB      TXTA       ;A
E039 7C          6680          DB      TXTB       ;B
E03A 58          6700          DB      TXTC       ;C
E03B 5E          6720          DB      TXTD       ;D
E03C 79          6740          DB      TXTE       ;E
E03D 71          6760          DB      TXTF       ;F
E03E             6780          ;
E03E             6800          ;
E03E             6820          ;BAUD FUNCTION TEXT AND CONSTANTS TABLE
E03E 00          6840  BAUD110: DB      TXTBLK     ;' 110'
E03F 06          6860          DB      TXT1
E040 06          6880          DB      TXT1
E041 3F          6900          DB      TXT0
E042 89 01       6920          DW      001211Q    ;110 BAUD TIMER CONSTANTS
E044 17 02       6940          DW      002027Q
E046 17 02       6960          DW      002027Q
E048 00          6980  BAUD150: DB      TXTBLK     ;' 150'
E049 06          7000          DB      TXT1
E04A 6D          7020          DB      TXT5
E04B 3F          7040          DB      TXT0
E04C 63 01       7060          DW      001143Q    ;150 BAUD TIMER CONSTANTS
E04E C7 01       7080          DW      001307Q
E050 C7 01       7100          DW      001307Q
```

```
E052 00          7120  BAUD300:      DB        TXTBLK ;' 300'
E053 4F          7140          DB    TXT3
E054 3F          7160          DB    TXT0
E055 3F          7180          DB    TXT0
E056 2F 01       7200          DW    001057Q   ;300 BAUD TIMER CONSTANTS
E058 63 01       7220          DW    001143Q
E05A 63 01       7240          DW    001143Q
E05C 00          7260  BAUD600:      DB        TXTBLK ;' 600'
E05D 7D          7280          DB    TXT6
E05E 3F          7300          DB    TXT0
E05F 3F          7320          DB    TXT0
E060 15 01       7340          DW    001025Q   ;600 BAUD TIMER CONSTANTS
E062 2F 01       7360          DW    001057Q
E064 2F 01       7380          DW    001057Q
E066 06          7400  BAUD1200:     DB        TXT1 ;'1200'
E067 5B          7420          DB    TXT2
E068 3F          7440          DB    TXT0
E069 3F          7460          DB    TXT0
E06A 08 01       7480          DW    001010Q   ;1200 BAUD TIMER CONSTANTS
E06C 15 01       7500          DW    001025Q
E06E 15 01       7520          DW    001025Q
E070             7540          ;
E070             7560          ;
E070             7580          ;
E070             7600          ;VARIOUS MESSAGES FOLLOW:
E070             7620          ;FIRST ALL 3-BYTE MESSAGES
E070             7640          ;
E070             7660          ;
E070 00          7680  MSGGO:  DB        TXTBLK    ;' GO'
E071 3D          7700          DB        TXTG
E072 5C          7720          DB        TXTO
E073 79          7740  MSGERR: DB        TXTE      ;'ERR'
E074 50          7760          DB        TXTR
E075 50          7780          DB        TXTR
E076 50          7800  MSGREG: DB        TXTR      ;'REG '
E077 79          7820          DB        TXTE
E078 3D          7840          DB        TXTG
E079 00          7860          DB        TXTBLK
E07A 40          7880  MSGDSH: DB        TXTDSH    ;'---'
E07B 40          7900          DB        TXTDSH
E07C 40          7920          DB        TXTDSH
```

```
E07D 5C          7940  MSGOFF:DB     TXTO       ;'OFF'
E07E 71          7960         DB     TXTF
E07F 71          7980         DB     TXTF
E080 00          8000  MSGON: DB     TXTBLK     ;' ON'
E081 5C          8020         DB     TXTO
E082 37          8040         DB     TXTN
E083 50          8060  MSGROM:DB     TXTR       ;'ROM'
E084 5C          8080         DB     TXTO
E085 54          8100         DB     TXTM
E086 50          8120  MSGRAM:DB     TXTR       ;'RAM'
E087 77          8140         DB     TXTA
E088 54          8160         DB     TXTM
E089 07          8180  MSG708:DB     TXT7       ;'708'
E08A 3F          8200         DB     TXT0
E08B 7F          8220         DB     TXT8
E08C 07          8240  MSG716:DB     TXT7       ;'716'
E08D 06          8260         DB     TXT1
E08E 7D          8280         DB     TXT6
E08F 00          8360  MSGTO: DB     TXTBLK     ;' TO.'
E090 78          8380         DB     TXTT
E091 DC          8400         DB     TXTOD
E092             8420         ;
E092             8440         ;
E092             8460         ;NOW 6-BYTE MESSAGES
E092             8480         ;
E092             8500         ;
E092 7C          8520  MSGBLO:DB     TXTB       ;'BLOCKS.'
E093 38          8540         DB     TXTL
E094 5C          8560         DB     TXTO
E095 58          8580         DB     TXTC
E096 70          8600         DB     TXTK
E097 ED          8620         DB     TXTSD
E098 5C          8640  MSGONES:       DB         TXTO ;'ONES...'
E099 37          8660         DB     TXTN
E09A 79          8680         DB     TXTE
E09B ED          8700         DB     TXTSD
E09C 80          8720         DB     TXTDOT
E09D 80          8740         DB     TXTDOT
E09E 5E          8760  MSGDATA:       DB         TXTD ;'DATA...'
E09F 77          8780         DB     TXTA
```

```
E0A0 78          8800          DB      TXTT
E0A1 F7          8820          DB      TXTAD
E0A2 80          8840          DB      TXTDOT
E0A3 80          8860          DB      TXTDOT
E0A4             8880          ;
E0A4             8900          ;
E0A4             8920          ;NOW 8 CHAR MESSAGES
E0A4             8940          ;
E0A4             8960          ;
E0A4 37          8980  MSGNOT: DB      TXTN        ;'NOT.STORED'
E0A5 5C          9000          DB      TXTO
E0A6 F8          9020          DB      TXTTD
E0A7 6D          9040          DB      TXTS
E0A8 78          9060          DB      TXTT
E0A9 5C          9080          DB      TXTO
E0AA 50          9100          DB      TXTR
E0AB 79          9120          DB      TXTE
E0AC 5E          9140          DB      TXTD
E0AD 38          9160  MSGLOADIN:      DB          TXTL ;'LOADING...'
E0AE 5C          9180          DB      TXTO
E0AF 77          9200          DB      TXTA
E0B0 5E          9220          DB      TXTD
E0B1 04          9240          DB      TXTI
E0B2 37          9260          DB      TXTN
E0B3 BD          9280          DB      TXTGD
E0B4 80          9300          DB      TXTDOT
E0B5 80          9320          DB      TXTDOT
E0B6 5E          9520  MSGDUMPIN:      DB          TXTD ;'DUMPING...'
E0B7 1C          9540          DB      TXTU
E0B8 54          9560          DB      TXTM
E0B9 73          9580          DB      TXTP
E0BA 04          9600          DB      TXTI
E0BB 37          9620          DB      TXTN
E0BC BD          9640          DB      TXTGD
E0BD 80          9660          DB      TXTDOT
E0BE 80          9680          DB      TXTDOT
E0BF 5E          9700  MSGDUDONE:      DB          TXTD ;'DUMP DONE'
E0C0 1C          9720          DB      TXTU
E0C1 54          9740          DB      TXTM
E0C2 73          9760          DB      TXTP
```

```
_0C3 00          9780          DB      TXTBLK
E0C4 5E          9800          DB      TXTD
E0C5 5C          9820          DB      TXTO
E0C6 37          9840          DB      TXTN
E0C7 79          9860          DB      TXTE
E0C8 5C          9880  MSGONEGOD:     DB          TXTO ;'ONES GOOD'
E0C9 37          9900          DB      TXTN
E0CA 79          9920          DB      TXTE
E0CB 6D          9940          DB      TXTS
E0CC 00          9960          DB      TXTBLK
E0CD 3D          9980          DB      TXTG
E0CE 5C          10000         DB      TXTO
E0CF 5C          10020         DB      TXTO
E0D0 5E          10040         DB      TXTD
E0D1 5C          10060 MSGONEBAD:     DB          TXTO ;'ONES  BAD'
E0D2 37          10080         DB      TXTN
E0D3 79          10100         DB      TXTE
E0D4 6D          10120         DB      TXTS
E0D5 00          10140         DB      TXTBLK
E0D6 00          10160         DB      TXTBLK
E0D7 7C          10180         DB      TXTB
E0D8 77          10200         DB      TXTA
E0D9 5E          10220         DB      TXTD
E0DA 73          10240 MSGPOPING:     DB          TXTP ;'POPPING...'
E0DB 5C          10260         DB      TXTO
E0DC 73          10280         DB      TXTP
E0DD 73          10300         DB      TXTP
E0DE 04          10320         DB      TXTI
E0DF 37          10340         DB      TXTN
E0E0 BD          10360         DB      TXTGD
E0E1 80          10380         DB      TXTDOT
E0E2 80          10400         DB      TXTDOT
E0E3 80          10420         DB      TXTDOT
E0E4 73          10440 MSGPOPDON:     DB          TXTP ;'POP  DONE'
E0E5 5C          10460         DB      TXTO
E0E6 73          10480         DB      TXTP
E0E7 00          10500         DB      TXTBLK
E0E8 00          10520         DB      TXTBLK
E0E9 5E          10540         DB      TXTD
E0EA 5C          10560         DB      TXTO
E0EB 37          10580         DB      TXTN
```

```
E0EC 79          10600          DB      TXTE
E0ED 5E          10620 MSGDATGOD:      DB          TXTD ;'DATA GOOD'
E0EE 77          10640          DB      TXTA
E0EF 78          10660          DB      TXTT
E0F0 77          10680          DB      TXTA
E0F1 00          10700          DB      TXTBLK
E0F2 3D          10720          DB      TXTG
E0F3 5C          10740          DB      TXTO
E0F4 5C          10760          DB      TXTO
E0F5 5E          10780          DB      TXTD
E0F6 5E          10800 MSGDATBAD:      DB          TXTD ;'DATA  BAD'
E0F7 77          10820          DB      TXTA
E0F8 78          10840          DB      TXTT
E0F9 77          10860          DB      TXTA
E0FA 00          10880          DB      TXTBLK
E0FB 00          10900          DB      TXTBLK
E0FC 7C          10920          DB      TXTB
E0FD 77          10940          DB      TXTA
E0FE 5E          10960          DB      TXTD
E0FF 7C          10980 MSGBREAK1:      DB          TXTB ;'BREAK PT.1'
E100 50          11000          DB      TXTR
E101 79          11020          DB      TXTE
E102 77          11040          DB      TXTA
E103 70          11060          DB      TXTK
E104 00          11080          DB      TXTBLK
E105 73          11100          DB      TXTP
E106 F8          11120          DB      TXTTD
E107 06          11140          DB      TXT1
E108             11340          ;EXEC VERSION NUMBER IS:
E108 79          11360 MSGXECVER:      DB          TXTE ;'EXEC.READY'
E109 64          11380          DB      TXTX
E10A 79          11400          DB      TXTE
E10B B9          11420          DB      271Q        ;NON-STANDARD 'C.'
E10C 50          11440          DB      TXTR
E10D 79          11460          DB      TXTE
E10E 77          11480          DB      TXTA
E10F 5E          11500          DB      TXTD
E110 6E          11520          DB      TXTY
E111 6D          11540 MSGSTAOFF:      DB          TXTS ;'STATE. OFF'
E112 78          11560          DB      TXTT
```

```
E113 77          11580          DB      TXTA
E114 78          11600          DB      TXTT
E115 F9          11620          DB      TXTED
E116 00          11640          DB      TXTBLK
E117 5C          11660          DB      TXTO
E118 71          11680          DB      TXTF
E119 71          11700          DB      TXTF
CASS
E11A F5             20  CDUMP:  PUSH    PSW         ;SAVE EXEC REGISTERS
E11B C5             30          PUSH                B
E11C E5             40          PUSH    H
E11D 21 00 80       60          LXI     H           200000Q;GENERATE 10 SECOND LEADER
E120 3E 02          70  LEADER: MVI     A           002Q;CASS OUT BIT ONLY AT LOGIC 1
E
E122 06 09          80          MVI     B           011Q;ONE TIMER
E124 D3 05          90          OUT     005Q
E126 05            100          DCR     B
E127 C2 26 E1      110          JNZ     $-4;LOOP ON DCR B
E12A 3E 00         120          MVI     A           000Q;CASS OUT BIT ONLY AT ZERO
E12C 06 07         130          MVI     B           007Q ;ZERO TIMER
E12E D3 05         140          OUT     005Q
E130 05            150          DCR     B
E131 C2 30 E1      160          JNZ     $-4;LOOP ON DCR B
E134 2B            170          DCX     H           ;DECREMENT 10 SEC TIMER
E135 7C            180          MOV     A           H
E136 A7            190          ANA     A
E137 C2 20 E1      200          JNZ     LEADER
E13A 3A 4A FC      210          LDA     SA;SEND SA NOTE ZERO IS 39 CP SHORT
E13D 67            220          MOV     H           A ;H=SA HI
E13E 00            230          NOP     ;TIMER TRIMMER
E13F CD 95 E1      240          CALL    CASSOUT;EXACT
E142 3A 22 FC      250          LDA     EA;SEND EA
E145 47            260          MOV     B           A
E146 CD 95 E1      270          CALL    CASSOUT
E149 2E 00         280  DUMP1:  MVI     L           000Q
E14B 7C            290  ENDCHECK:      MOV         A H ;DOES SA HI = EA?
E14C B8            300          CMP     B
E14D CA 58 E1      310          JZ      EOD         ;JUMP TO END OF DUMP
E150 7E            320          MOV     A           M ;NO SEND BYTE
E151 23            330          INX     H
E152 CD 95 E1      340          CALL    CASSO'IT
```

```
E155 C3 4B E1     350           JMP     ENDCK_CK
E158 E1           360    EOD:   POP     H          ;RESTORE EXEC REIGSTERS
E159 C1           370           POP     B
E15A F1           380           POP     PSW
E15B C9           390           RET
E15C C5           400    CLOAD: PUSH    B          ;SAVE EXEC REGISTERS
E15D D5           410           PUSH    D
E15E E5           420           PUSH    H
E15F F5           430           PUSH    PSW
E160              431;THIS ROUTINE AUTO-SENSES LEADER
E160 21 00 40     432           LXI     H          100000Q ;SETS TRANSITION COUNTER
0 4K(IE 2 SEC OF LEADER)
E163 DB 01        433    A:     IN      001Q       ;INPUT CASS
E165 E6 02        434           ANI     002Q       ;MASK CASS BIT
E167 47           435           MOV     B          A ;SAVE BIT
E168 DB 01        436    B:     IN      001Q       ;INPUT CASS AGAIN
E16A E6 02        437           ANI     002Q       ;MASK CASS BIT
E16C A8           438           XRA     B          ;RESULT IS 0 IF BIT HAS NOT CHANGE
D
E16D CA 68 E1     439           JZ      B          ;SPIN UNTIL TRANSISTION
E170 2B           440           DCX     H          ;DECREMENT TRANSISITION COUNTER
E171 7C           441           MOV     A          H ;TEST TRANS COUNTER FOR 0
E172 B5           442           ORA     L
E173 C2 63 E1     443           JNZ     A          ;CONTINUE COUNTDOWN
E176              444;4K PULSES COUNTED
E176 2E 00        520           MVI     L          000Q ;BEGIN CALL LOAD/CLEAR L
E178 CD FC E1     530           CALL    CASSIN;GET SA
E17B 32 63 FC     535           STA     SABACK     ;SAVE HIGH ADDRESS FROM TAPE
E17E 67           540           MOV     H          A ;SETUP SA HI
E17F CD FC E1     550           CALL    CASSIN;GET EA
E182 32 22 FC     555           STA     EA         ;SAVE LOW ADDRESS FROM TAPE
E185 47           560           MOV     B          A ;SETUP EA  HI
E186 CD FC E1     570    GETBYTE:       CALL       CASSIN
E189 77           580           MOV     M          A ;STORE BYTE
E18A 23           590           INX     H          ;SA+1
E18B 78           620    ENDCK: MOV     A          B
E18C BC           630           CMP     H
E18D C2 86 E1     640           JNZ     GETBYTE
E190 F1           650    POPALL:POP     PSW
E191 E1           660           POP     H
```

```
E192 D1          670          POP    D
E193 C1          680          POP    B
E194 C9          690          RET
E195 F5          700  CASSOUT:       PUSH       PSW ;SENDS ONE BYTE TO THE CASSE
E
E196 C5          710          PUSH   B
E197 D5          720          PUSH   D
E198 E5          730          PUSH   H
E199 4F          740          MOV    C          A
E19A 21 09 04    750          LXI    H          LONG
E19D 3E F8       760          MVI    A          370Q
E19F CD DB E1    770          CALL   PULSE
E1A2 00          780          NOP
E1A3 00          790          NOP
E1A4 16 08       800          MVI    D          010Q
E1A6 79          810  CTOUTB: MOV    A          C
E1A7 0F          820          RRC
E1A8 4F          830          MOV    C          A
E1A9 21 04 08    840          LXI    H          SHORT
E1AC 3E FC       850          MVI    A          374Q
E1AE DA B8 E1    860          JC     CTOUTD
E1B1 21 09 04    870          LXI    H          LONG
E1B4 3D          880          DCR    A
E1B5 7F          890          MOV    A          A
E1B6 00          900          NOP
E1B7 00          910          NOP
E1B8 CD DB E1    920  CTOUTD: CALL   PULSE
E1BB 15          930          DCR    D
E1BC C2 A6 E1    940          JNZ    CTOUTB
E1BF 16 02       950          MVI    D          002Q
E1C1 C3 C8 E1    960          JMP    CTOUTG
E1C4 00          970  CTOUTE: NOP
E1C5 00          980          NOP
E1C6 00          990          NOP
E1C7 7F          1000         MOV    A          A
E1C8 7F          1010 CTOUTG: MOV    A          A
E1C9 00          1020         NOP
E1CA 21 04 08    1030         LXI    H          SHORT
E1CD 3E FC       1040         MVI    A          374Q
E1CF CD DB E1    1050         CALL   PULSE
E1D2 15          1060         DCR    D
```

```
E1D3 C2 C4 E1     1070          JNZ     CTOUTE
E1D6 E1           1080          POP     H
E1D7 D1           1090          POP     D
E1D8 C1           1100          POP     B
E1D9 F1           1110          POP     PSW
E1DA C9           1120          RET
E1DB 85           1130  PULSE:  ADD     L
E1DC 47           1140          MOV     B         A
E1DD 3E 05        1150  PULSA:  MVI     A         005Q
E1DF D3 05        1160          OUT     005Q
E1E1 05           1170          DCR     B
E1E2 F2 DD E1     1180          JP      PULSA
E1E5 45           1190          MOV     B         L
E1E6 3E 07        1200  PULSB:  MVI     A         007Q
E1E8 D3 05        1210          OUT     005Q
E1EA 05           1220          DCR     B
E1EB F2 E6 E1     1230          JP      PULSB
E1EE 3E 05        1240          MVI     A         005Q
E1F0 D3 05        1250          OUT     005Q
E1F2 25           1260          DCR     H
E1F3 C8           1270          RZ
E1F4 45           1280          MOV     B         L
E1F5 05           1290          DCR     B
E1F6 05           1300          DCR     B
E1F7 3E 05        1310          MVI     A         005Q
E1F9 C3 DD E1     1320          JMP     PULSA
E1FC C5           1330  CASSIN:PUSH     B         ;SAVE EXEC REGISTERS
E1FD D5           1340          PUSH    D
E1FE E5           1350          PUSH    H
E1FF 21 08 08     1360          LXI     H         MEDIUM
E202 06 00        1370  CTINA:  MVI     B         000Q;IGNORE SHORT PULSES
E204 DB 01        1380  CTINB:  IN      001Q
E206 E6 02        1390          ANI     002Q
E208 CA 04 E2     1400          JZ      CTINB
E20B 04           1410  CTINC:  INR     B         ;TIME PULSE DURATION
E20C DB 01        1420          IN      001Q
E20E E6 02        1430          ANI     002Q
E210 C2 0B E2     1440          JNZ     CTINC
E213 78           1450          MOV     A         B ;IS B > L ?
E214 BD           1460          CMP     L         ;IF A < L CARRY = 1
E215 DA 02 E2     1470          JC      CTINA;SHORT SO CONTINUE TO WAIT
```

```
E218 25          1480          DCR    H          ;LONG,MUST BE 1ST PULSE OF START B
IT
E219 25          1490          DCR    H          ;START BIT
E21A C2 02 E2    1500          JNZ    CTINA;WAIT TO END OF START BIT
E21D 0E 08       1510          MVI    C          010Q
E21F 21 08 08    1520  CTIND:  LXI    H          MEDIUM
E222 06 00       1530  CTINE:  MVI    B          000Q
E224 DB 01       1540  CTINF:  IN     001Q;WAIT FOR START OF A PULSE
E226 E6 02       1550          ANI    002Q
E228 CA 24 E2    1560          JZ     CTINF
E22B 04          1570  CTING:  INR    B          ;TIME PULSE DURATION
E22C DB 01       1580          IN     001Q
E22E E6 02       1590          ANI    002Q
E230 C2 2B E2    1600          JNZ    CTING
E233 78          1610          MOV    A          B
E234 BD          1620          CMP    L          ;CARRY=1(LONG)=0(SHORT)
E235 DA 39 E2    1630          JC     CTINH
E238 25          1640          DCR    H          ;DCR TWICE FOR LONG,ONCE FOR SHORT

E239 25          1650  CTINH:  DCR    H
E23A C2 22 E2    1660          JNZ    CTINE
E23D 7A          1670          MOV    A          D ;END OF BIT,SAVE BIT VALUE IN D
E23E 1F          1680          RAR
E23F 57          1690          MOV    D          A
E240 0D          1700          DCR    C          ;CHECK FOR LAST BIT
E241 C2 1F E2    1710          JNZ    CTIND
E244 E1          1720          POP    H          ;RESTORE REGISTERS
E245 D1          1730          POP    D
E246 C1          1740          POP    B
E247 C9          1750          RET
PROM
E248                    5;MD2 EPROM PROGRAMMER SOFTWARE.
E248                    6;WRITTEN BY M. VESLOCKI, ENTERED BY A. SINGER, 2/24/80.
E248                    7;
E248                    8;SYSTEM SCRATCHPAD REGISTERS.
E248                    9          ;BEGINNING OF RESERVED SCRATCH PAD FOR PROM PGMMER
E248                    10    ;CONTROL:DB 00H;B7:0=2707/1=2716 B0:VERIFY 0=DATA/1=ONES
E248                    20    ;SSTART:DB 00H;SOURCE HI ADRESS STARTING POINT
E248                    30    ;SADR:DW 0000H;DYNAMIC SOURCE ADRESS
E24C                    40    ;SDATA:DB 00H;SOURCE DATA
```

```
E246                50    ;DDATA:DB 00H;DESTINATION DATA
E248                60    ;DADR:DW 0000H;DYNAMIC DESTINATION ADRESS
E248                90    ;STATUS:DB 00H;RESULT OF TESTING B0=0(OK),=1(ERROR)
E248                100;
E248                110;INITIALIZATION ROUTINE.
E248 3E 04          120   INI:   MVI    A          NOPROGPLS ;TURN PROGRAMMING PULSE
OFF BY SENDING
E24A D3 0A          130          OUT    PRTSEROUT ;PROGRAMMING PULSE BIT TO SERIAL O
UTPUT PORT
E24C 11 00 00       140          LXI    D          000000Q ;SET DESTINATION ADDRESS
O ZERO
E24F 3A 4A FC       150          LDA    SSTART     ;SETUP SOURCE ADDRESS
E252 67             160          MOV    H          A
E253 2E 00          170          MVI    L          000Q
E255 C9             180          RET    ;EXIT FROM INI
E256                190;
E256                200;VERIFY DATA OR VERFY ONES ROUTINE.
E256 C5             210   VERIFY:PUSH   B          ;PUSH ALL REGISTERS
E257 D5             220          PUSH   D
E258 E5             230          PUSH   H
E259 F5             240          PUSH   PSW
E25A CD 48 E2       250          CALL   INI        ;INITIALIZE
E25D 7A             260   LOOP1: MOV    A          D ;SETUP DESTINATION ADDRESS HIGH
E25E E6 07          270          ANI    PRMRBIT    ;MASK PROM READ/WRITE BIT=0="READ"

E260 D3 01          280          OUT    PRTLEDS1   ;SEND R/W BIT AND DEST HIGH ADDRES
S TO PROM
E262 7B             290          MOV    A          E ;SEND DESTINATION LOW ADDRESS
 PROM
E263 D3 00          300          OUT    PRTLEDS0
E265 DB 08          310   RDPROM:IN     PRTPRMIN   ;READ PROM DATA
E267 32 27 FC       320          STA    DDATA      ;SAVE IN DEST DATA REGISTER
E26A 22 24 FC       330          SHLD   SADR       ;STORE SOURCE ADRESS
E26D EB             340          XCHG   ;STORE DESTINATION ADDRESS
E26E 22 28 FC       350          SHLD   DADR
E271 EB             360          XCHG   ;RESTORE REGISTERS
E272 47             370   BLANK: MOV    B          A ;TEST: VERIFY ONES OR VERIFY D
A
E273 3A 23 FC       380          LDA    CONTROL    ;FETCH CONTROL WORD
E276 0F             390          RRC    ;POSITION TO BIT
E277 D2 8D E2       400          JNC    RDSOURCE;  IF NOT CARRY, VERIFY DATA
```

```
E27A 3E FF       410   ONES:  MVI    A        377Q ;OTHERWISE PUT ALL ONES IN A
TO VERIFY ONES
E27C 32 26 FC    420          STA    SDATA    ;STORE ONES IN SOURCE DATA
E27F B8          430          CMP    B
E280 CA 9C E2    440          JZ     INCDADR  ;JUMP IF ONES FOUND
E283 3E 01       450   ERROR: MVI    A        001Q ;VERIFY FAILS SET ERROR FLAG
IN STATUS WORD
E285 32 2A FC    460          STA    STATUS
E288 F1          470   POPNRET:     POP      PSW ;POP ALL REGISTERS
E289 E1          480          POP    H
E28A D1          490          POP    D
E28B C1          500          POP    B
E28C C9          510          RET
E28D 7E          520   RDSOURCE:    MOV      A M ;READ SOURCE DATA FROM MEMOR.
E28E 32 26 FC    530          STA    SDATA    ;SAVE IN SCRATCH REGISTER
E291 47          540          MOV    B        A ;SET UP VERIFICATION
E292 3A 27 FC    550          LDA    DDATA    ;GET DESTINATION DATA FROM PROM
E295 B8          560          CMP    B
E296 CA 9C E2    570          JZ     INCDADR  ;JUMP IF DATA IN PROM MATCHES MEMO
RY
E299 C3 83 E2    580          JMP    ERROR    ;OTHERWISE DATA VERIFY FAILS, GO S
ET ERROR FLAG
E29C 13          590   INCDADR:     INX      D ;INCREMENT DESTINATION ADDRESS
E29D 23          600          INX    H        ;INCREMENT SOURCE ADDRESS
E29E 3A 23 FC    610   LIMIT: LDA    CONTROL  ;FETCH CONTROL TO CHECK FOR 1K OR
2K LIMIT
E2A1 07          620          RLC    ;POSITION TO 1K/2K(2708/2716) BIT
E2A2 07          630          RLC
E2A3 07          640          RLC
E2A4 E6 04       650          ANI    004Q     ;MASK OUT 1K/2K BIT
E2A6 C6 04       660          ADI    004Q     ;COMPUTE 1K/2K LIMIT FOR TEST
E2A8 BA          670          CMP    D        ;TEST: DESTINATION ADDRESS HIGH
CONTROL LIMIT?
E2A9 C2 5D E2    680          JNZ    LOOP1    ;LIMIT NOT REACHED, CONTINUE VERIF
YING
E2AC 3E 00       690          MVI    A        000Q ;LIMIT REACHED, VERIFY OK, SE
T STATUS ACCORDINGLY
E2AE 32 2A FC    700          STA    STATUS
E2B1 C3 88 E2    710          JMP    POPNRET  ;GO TO RESTORE REGS AND EXIT FROM
VERIFY ROUTINE
```

```
E2B4                    720;
E2B4                    730;SYMBOLIC NAMES FOR DEVICE ADDRESSES.
E2B4                    740   PRTLEDS0:     EQU       000Q ;OUT 0, LEDS AND PROM POPPER
LOW ADDR
E2B4                    750   PRTLEDS1:     EQU       001Q ;OUT 1, LEDS AND PROM POPPER
HI ADDR,R/W BIT
E2B4                    755   PRTLEDS2:     EQU       002Q ;OUT 2, LEDS AND PROM POPPER
DATA WRITE
E2B4                    760   PRTSEROUT:    EQU       012Q;SPLIT REGISTER;BD2=PGM PULSE
BD3=RAM 374 ON/OFF
E2B4                    765   PRTPRMIN:     EQU       010Q ;IN 10 PROM POPPER READ DATA
PORT
E2B4                    770;
E2B4                    780;SYMBOLIC NAMES FOR BIT MASKS.
E2B4                    790   NOPROGPLS:    EQU       004Q ;MASK TO TURN PROM PROGRAMMIN
G PULSE OFF
E2B4                    795   PROGPLS:      EQU       000Q ;MASK TO TURN PROGRAMMING PUL
SE ON
E2B4                    800   PRMRBIT:      EQU       007Q ;MASK TO FORCE PROM R/W BIT T
O R(READ)
E2B4                    805   PRMWBIT:      EQU       100Q ;MASK TO FORCE PROM R/W BIT T
O WRITE
E2B4                    810;
E2B4                    820;PROM POPPING ROUTINE
E2B4 C5                 830   POPPER:PUSH   B         ;PUSH ALL REGISTERS
E2B5 D5                 840          PUSH   D
E2B6 E5                 850          PUSH   H
E2B7 F5                 860          PUSH   PSW
E2B8 CD 48 E2           870          CALL   INI       ;INITIALIZE POPPER
E2BB 3A 23 FC           880   COUNTER:               LDA CONTROL ;SET A LOOP COUNTER TO

E2BE 07                 890          RLC    ;  250 DECIMAL IF 2708(NO CARRY)
E2BF DA C7 E2           900          JC     COUNT1;1IF2716(CARRY)
E2C2 06 FA              910          MVI    B         372Q ;SET UP 250 DECIMAL
E2C4 C3 C9 E2           920          JMP    SENDDADR
E2C7 06 01              930   COUNT1:MVI    B         001Q;SETUP A 1
E2C9 3E 04              940   SENDDADR:     MVI       A NOPROGPLS ;SET PROM PROGRAM PULS
E BIT OFF
E2CB D3 0A              950          OUT    PRTSEROUT ;TRANSMIT OFF VALUE TO PULSE BIT
E2CD 7A                 960          MOV    A         D ;SETUP DESTINATION ADDRESS HIGH
E2CE F6 40              970          ORI    PRMWBIT   ;SETUP R/WAIT TO W(WRITE) MODE
```

```
E2D0 D3 01        980          OUT    PRTLEDS1  ;SEND DEST HIGH ADDRESS AND R/W PI
T
E2D2 7B           990          MOV    A         E ;SETUP DEST LO ADDRESS
E2D3 D3 00        1000         OUT    PRTLEDS0  ;SEND DEST LOW ADDRESS
E2D5 7E           1010 SENDDATA:    MOV       A M ;SET UP THE DATA FOR PROGRAMMI
NG
E2D6 D3 02        1020         OUT    PRTLEDS2  ;SEND IT TO THE PROM SOCKET
E2D8 3A 23 FC     1030 DOPULSE:     LDA       CONTROL ;SETUP THE CONTROL WORD
E2DB 17           1040         RAL    ;DO 0.5 MS TIMER FOR 2708, 50 MS FOR 2716
E2DC DA E4 E2     1050         JC     SETC100   ;IF CARRY, 2716 AND 50 MS PULSE
E2DF 0E 01        1060         MVI    C         001Q ;OTHERWISE 2708 AND 0.5 MS PU
LSE
E2E1 C3 E6 E2     1070         JMP    PULSEON
E2E4 0E 6A        1080 SETC100:     MVI       C 152Q ;SET COUNT FOR 100 ITERATIO
NS AND 50 MS
E2E6 7A           1090 PULSEON:     MOV       A D ;
E2E7 F6 40        1100         ORI    100Q      ;RETAIN MODE
E2E9 3E 00        1110         MVI    A         PROGPLS ;SET UP PROGRAM PULSE BIT
E2EB D3 0A        1120         OUT    PRTSEROUT ;TURN PROGRAM PULSE LINE ON
E2ED 3E 16        1130 TIMER:       MVI       A 026Q ;PULSE TIMING GENERATOR COD
E
E2EF 3D           1140 PTIMELOOP:   DCR       A
E2F0 C2 EF E2     1150         JNZ    PTIMELOOP
E2F3 0D           1160         DCR    C
E2F4 C2 ED E2     1170         JNZ    TIMER
E2F7 3E 04        1180 PULSEOFF:    MVI       A NOPROGPLS ;TURN PROM PROGRAMMING
 PULSE OFF
E2F9 D3 0A        1190         OUT    PRTSEROUT
E2FB 7A           1200         MOV    A         D ;SET UP ADDRESS AND R/W BIT
E2FC F6 40        1210         ORI    PRMWBIT
E2FE D3 01        1220         OUT    PRTLEDS1  ;SEND ADDRESS AND R/W BIT
E300 13           1230 DADRLIMIT:   INX       D ;TEST DEST ADDRESS = LIMIT
E301 23           1240         INX    H
E302 3A 23 FC     1250         LDA    CONTROL   ;GET CONTROL WORD TO DETERMINE 1 ?
K LIMIT
E305 07           1260         RLC
E306 07           1270         RLC
E307 07           1280         RLC
E308 E6 04        1290         ANI    004Q      ;MASK FOR TEST
E30A C6 04        1300         ADI    004Q
```

```
E30C BA          1310          CMP     D           ;DO TEST
E30D C2 C9 E2    1320          JNZ     SENDDADR    ;IF NOT AT LIMIT CONTINUE PROGRAMM
ING
E310 CD 48 E2    1330          CALL    INI         ;OTHERWISE CHECK FOR MORE PROGRAMM
ING ITERATIONS
E313 05          1340  DCRLOOP:      DCR         B ;DECREMENT ITERATION COUNTER
E314 C2 C9 E2    1350  LOOPEQ0:      JNZ         SENDDADR ;IF NOT DONE ITERATING, S
TART OVER
E317 3E 00       1360  FINALCOND:    MVI         A 000Q ;SETUP R/W BIT
E319 D3 01       1370          OUT     PRTLEDS1    ;SEND TO POPPER
E31B 3E 04       1380          MVI     A           NOPROGPLS ;TURN OFF PROGRAMMING PU
LSE LINE
E31D D3 0A       1390          OUT     PRTSEROUT
E31F C3 88 E2    1400          JMP     POPNRET     ;GO TO RESTORE REGISTERS AND EXIT
TYIN
E322 C5          30    TTYIN:  PUSH    B
E323 D5          40            PUSH    D
E324 E5          50            PUSH    H
E325 06 08       60            MVI     B           010Q
E327 DB 01       70    TEST:   IN      001Q
E329 0F          80            RRC
E32A DA 27 E3    90            JC      TEST
E32D CD 47 E3    100           CALL    WAITH
E330 DB 01       110           IN      001Q
E332 0F          120           RRC
E333 DA 27 E3    130           JC      TEST
E336 CD 53 E3    140   TTYIB:  CALL    WAIT
E339 DB 01       150           IN      001Q
E33B 0F          160           RRC
E33C 79          170           MOV     A           C
E33D 1F          180           RAR
E33E 4F          190           MOV     C           A
E33F 05          200           DCR     B
E340 C2 36 E3    210           JNZ     TTYIB
E343 E1          220           POP     H
E344 D1          230           POP     D
E345 C1          240           POP     B
E346 C9          260           RET
E347 2A 5D FC    270   WAITH:  LHLD    TYHLFBD     ;LOAD HALF BAUD TIME CONST.
E34A 2B          280   DEC:    DCX     H
```

```
E34B 7C           290          MOV    A          H
E34C A7           300          ANA    A
E34D C2 4A E3     310          JNZ    DEC
E350 00           320          NOP
E351 00           330          NOP
E352 C9           340          RET
E353 2A 5F FC     370  WAIT:   LHLD   TYONEBDI   ;LOAD ONE BAUD INPUT TIME CONST.
E356 2B           380  DC:     DCX    H
E357 7C           390          MOV    A          H
E358 A7           400          ANA    A
E359 C2 56 E3     410          JNZ    DC
E35C 00           420          NOP
E35D 00           430          NOP
E35E 00           440          NOP
E35F 00           450          NOP
E360 C9           460          RET
TYOUT
E361 F5           20   TTYOUT: PUSH   PSW
E362 C5           30           PUSH   B
E363 D5           40           PUSH   D
E364 E5           50           PUSH   H
E365 06 09        60           MVI    B          011Q
E367 07           70           RLC
E368 4F           80           MOV    C          A
E369 97           90           SUB    A
E36A D3 05        100  SNDTTY: OUT    005Q
E36C CD 90 E3     110          CALL   REST
E36F 79           120          MOV    A          C
E370 0F           130          RRC
E371 4F           140          MOV    C          A
E372 E6 01        150          ANI    001Q
E374 05           160          DCR    B
E375 C2 6A E3     170          JNZ    SNDTTY
E378 3E 01        180          MVI    A          001Q
E37A D3 05        190          OUT    005Q
E37C CD 90 E3     200          CALL   REST
E37F CD 90 E3     210          CALL   REST
E382 21 02 01     220          LXI    H          001002Q
E385 2B           230  DCXH:   DCX    H
E386 7C           240          MOV    A          H
E387 A7           250          ANA    A
```

```
E388 C2 85 E3     260           JNZ     DCXH
E38B E1           270           POP     H
E38C D1           280           POP     D
E38D C1           290           POP     B
E38E F1           300           POP     PSW
E38F C9           310           RET
E390 2A 61 FC     320   REST:   LHLD    TYONEBDO  ;LOAD ONE BAUD OUTPUT TIME CONST.
E393 2B           330   DCREST: DCX     H
E394 7C           340           MOV     A         H
E395 A7           350           ANA     A
E396 C2 93 E3     360           JNZ     DCREST
E399 00           370           NOP
E39A 00           380           NOP
E39B C9           390           RET
E39C 00           395           NOP
E39D 32 03 FC     400           STA     SIM4;THIS IS THE TTY PATCH
E3A0 3E 05        410           MVI     A         005Q;THE BIT SET CODE FOR TTY
E3A2 C3 06 00     420           JMP     KE2;JUMP TO EXEC
SCRATCHB
E3A5              100                             ;SCRATCHPAD DEFINITIONS FOLLOW:
E3A5              200                             ;THESE BYTES DO NOT NEED TO BE
E3A5              300                             ;INITIALEZED DURING PUP OR RESET
E3A5              400           ORG     FSCR-001000Q ;START OF SCRATCHPAD
FC00 00           500   SIM1:   DB      00H       ;SIMULATION BYTES USED BY SINGLE
TEPPER
FC01 00           600   SIM2:   DB      00H
FC02 00           700   SIM3:   DB      00H
FC03 00           800   SIM4:   DB      00H
FC04 00           900   SIM5:   DB      00H
FC05 00           1000  SIM6:   DB      00H
FC06 00           1010  STEPAREA1:      DB        00H ;USED BY SINGLE STEPPER
FC07 00           1020  STEPAREA2:      DB        00H
FC08 00           1030  STEPAREA3:      DB        00H
FC09 00           1040  HLBUF1:DB       0000H     ;ANOTHER BYTE USED BY SINGLE STEPP
ER!
FC0A 00           1050  VECT1:  DB      00H
FC0B 00           1060  VECT2:  DB      00H
FC0C 00           1070  VECT3:  DB      00H
FC0D              1100                             ;USER REGISTERS FOLLOW:
FC0D 00           1200  REGFL:  DB      00H       ;FLAGS
```

```
FC0E 00          1300  REGA:  DB      00H       ;A
FC0F 00          1400  REGC:  DB      00H       ;C
FC10 00          1500  REGB:  DB      00H       ;B
FC11 00          1600  REGE:  DB      00H       ;E
FC12 00          1700  REGD:  DB      00H       ;D
FC13 00          1800  PUSHRG:DB      00H       ;(UNUSED BUT NEEDED AS POINTER)
FC14 00 00       1900  USERHL:DW      0000H     ;L&H
FC16 00 00       1910  USERHL1:       DW        0000H ;TEMP L&H USED DURING RESET
FOR STORING USER REG'S
FC18 00 00       2000  USERSC:DW      0000H     ;TEMP USED BY SINGLE STEPPER
FC1A 00 00       2100  EXECSP:DW      0000H     ;TEMP USED BY SINGLE STEPPER
FC1C 00 00       2110  STEPADR:       DW        0000H ;ADRESS FOR SINGLE STEP
FC1E 00          2200  INSTATE:       DB        00H ;INPUT STATE REGISTER
FC1F 00          2300  LASTKEY:       DB        00H ;ALWAYS CONTAINS LAST KEY STRO
CK
FC20 00          2310  LASTKEYCO:     DB        00H ;LAST ***COMPRESSED*** KEY CO
E
FC21 00          2400  ERRFLAG:       DB        00H
FC22 00          2500  EA:    DB      00H       ;END ADR FOR UTILITIES
FC23             2502                           ;PROM PROGRAMMER SYSTEM SCRATCHPAD
 REGISTERS
FC23 00          2503  CONTROL:       DB        00H ;B7:0=2708,1=2716; B0:0=VERIFY
 DATA,1=VERIFY ONES
FC24 00 00       2510  SADR:  DW      000H      ;DYNAMIC SOURCE ADRESS
FC26 00          2520  SDATA: DB      00H       ;SOURCE DATA
FC27 00          2530  DDATA: DB      00H       ;DESTINATION DATA
FC28 00 00       2540  DADR:  DW      0000H     ;DYNAMIC DESTINATION ADRESS
FC2A 00          2550  STATUS:DB      00H       ;RESULT OF TESTING B0=0(OK)=1(ERRO
R)
FC2B             2595;
FC2B             2596;
FC2B             2597;
FC2B             2598;
FC2B             2599;
FC2B             2600                           ;THE FOLLOWING BYTES DO NEED TO BE
 INITIALIZED
FC2B             2700                           ;AT ONE TIME OR ANOTHER
FC2B 00          2800  STATEBUF:       DB        00H ;AUX STATE OFFSET BUFFER
FC2C 00          2900  DATBUF:DB      00H       ;DATA BUFFER, GENERAL ACCUMULATOR
FC2D 00 00       3000  HLBUF:  DW      0000H     ;HI&LO ADD. VARIABLES
FC2F 00 00       3100  DSPUP:  DW      0000H     ;DISPLAY UPDATE ROUTINE POINTER
```

```
FC31 00 00        3200  TABLE: DW      0000H     ;CURRENT COMMAND TABLE POINTER
FC33 00 00        3300  TASK:  DW      0000H     ;CURRENT TASK POINTER
FC35 00 00        3310  USEREVENT:     DW        0000H ;USER DEFINED EVENT ( INITIAL
LY NOP )
FC37 00           3311        DB      00H       ;3RD BYTE OF USEREVENT (A JMP, U"J
ALLY)
FC38 00 00        3400  DSPREG:DW      0000H     ;CURRENT DISPLAY REG POINTER
FC3A 00 00        3600  AUXMESPTR:     DW        0000H ;AUX MESSAGE POINTER
FC3C 00 00        3700  AUXSTATE:      DW        0000H ;AUX STATE POINTER
FC3E 00 00        3710  BAUDPTR:       DW        0000H ;BAUD RATE POINTER
FC40 00 00        3800  BR1USERD:      DW        0000H ;BREAK POINT 1 USER DATA 0
JMP TO WEHITBR1
FC42 00           3900        DB      00H       ;AUXSTATE LOCATOR AND DATA TABLE
FC43 00 00        4200  BR1H:  DW      0000H     ;BREAK POINT 1 HI
FC45 00 00        4300  BR1L:  DW      0000H     ;BREAK POINT 1 LO
FC47 00 00        4400  STEP1: DW      0000H     ;STEP (ONE/DATA)
FC49 00 00        4600  SRC:   DW      0000H     ;SOURCE REGISTER
FC4B              4700  SA:    EQU     SRC+001Q  ;STARTING ADR FOR UTILITIES
FC4B              4710  SSTART:EQU     SRC+001Q  ;DYNAMIC SOURCE ADRESS
FC4B 00 00        4800  DES:   DW      0000H     ;DESTINATION REGISTER
FC4D 00 00        4900  LEN:   DW      0000H     ;LENGTH REGISTER
FC4F 00 00        5000  CLR:   DW      0000H     ;CLEAR TEST FOR PROM REGISTER
FC51 00 00        5100  POP:   DW      0000H     ;POP PROM? REGISTER
FC53 00 00        5200  DUP:   DW      0000H     ;DUPLICATION TEST??? REGISTER
FC55 00 00        5210  PRM:   DW      0000H     ;PROM TYPE (2708/2716)
FC57 00 00        5300  MAP:   DW      0000H     ;MEMORY MAP
FC59 00 00        5400  BAUD:  DW      0000H     ;BAUD REGISTER
FC5B 00 00        5410  USERSP:DW      0000H     ;USER STACK POINTER
FC5D 00 00        5420  TYHLFBD:       DW        0000H ;TTY HALF BAUD TIME CONSTA

FC5F 00 00        5425  TYONEBDI:      DW        0000H ;TTY ONE BAUD IN TIME CONST.

FC61 00 00        5430  TYONEBDO:      DW        0000H ;TTY ONE BAUD OUT TIME CONST

FC63 00           5450  SABACK:DB      00H       ;START ADDRESS FOR CLOAD TO RETUF
(USES EA ALSO)
FC64              5497;
FC64              5498;
FC64              5499;
FC64              5500                          ;AUX REGISTER STATE OFFSETS
FC64 00           5600  BR1SO: DB      00H       ;BREAK POINT 1
FC65 00           5700  STEPSO:DB      00H       ;STEP
```

```
FC66 00          5800  SRCSO: DB     00H       ;SOURCE
FC67 00          5900  DESSO: DB     00H       ;DESTINATION
FC68 00          6000  LENSO: DB     00H       ;LENGTH
FC69 00          6100  CLRSO: DB     00H       ;CLEAR
FC6A 00          6200  POPSO: DB     00H       ;POP
FC6B 00          6300  DUPSO: DB     00H       ;DUPLICATION
FC6C 00          6310  PRMSO: DB     00H       ;PROM TYPE (2708/2716)
FC6D 00          6400  MAPSO: DB     00H       ;MEM MAP
FC6E 00          6500  BAUDSO:DB     00H       ;BAUD
FC6F             6510          ;
FC6F             6520          ;
FC6F             6530          ;
6000-6C14  13
```

(Blank Page)

SYMBOL TABLE CROSS REFERENCE LISTING

```
>>SYM
FA0 D800       FSCR FD00      FSCRHI 00FC
F0 00D8        F1 00D9        F2 00DA
F3 00DB        F4 00DC        F5 00DD
F6 00DE        F7 00DF        F10 00E0
KE1 0B9D       KE2 0006       DIPBASE 0020
DIPLEDS 0008   DIPPUPRES 0010   BITPUP 0040
DEFLTADD 0300   DEFLTUSP 03FE   LONG 0409
MEDIUM 0808    SHORT 0804      TXT0 003F
TXT1 0006      TXT2 005B       TXT3 004F
TXT4 0066      TXT5 006D       TXT6 007D
TXT7 0007      TXT8 007F       TXT9 006F
TXTAD 00F7     TXTA 0077       TXTBD 00FC
TXTB 007C      TXTCD 00D8      TXTC 0058
TXTDD 00DE     TXTD 005E       TXTED 00F9
TXTE 0079      TXTFD 00F1      TXTF 0071
TXTGD 00BD     TXTG 003D       TXTH 0074
TXTHD 00F4     TXTI 0004       TXTID 0084
TXTJ 000E      TXTJD 008E      TXTK 0070
TXTKD 00F0     TXTLD 00B8      TXTL 0038
TXTM 0054      TXTMD 00D4      TXTN 0037
TXTND 00B7     TXTO 005C       TXTOD 00DC
TXTP 0073      TXTPD 00F3      TXTR 0050
TXTRD 00D0     TXTS 006D       TXTSD 00ED
TXTT 0078      TXTTD 00F8      TXTU 001C
TXTUD 009C     TXTV 003E       TXTVD 00BE
TXTW 007E      TXTWD 00FE      TXTX 0064
TXTXD 00E4     TXTY 006E       TXTYD 00EE
TXTZ 005B      TXTZD 00DB      TXTDSH 0040
TXTBLK 0000    TXTDOT 0080     PORT0 0000
PORT1 0001     PORT2 0002      PORT8279C 0003
PORT8279D 0004   PORTPUP 0001   PORTDIPS 0001
KEYNEXT 001F   KEYSTORE 001E   KEYPREV 001D
KEYSTEP 001C   KEYHIGH 0017   KEYLOW 0016
KEYGO 0015     KEYOPTION 0014   KEYLOAD 000F
KEYDUMP 000E   KEYPROM 000D   KEYCOPY 000C
KEYMEM 0007    KEYREGS 0006   KEYAUX 0005
KEYCANCEL 0004   EXECB D800     MAP2 D80A
SAVEGOREG D81B   PTA D837       RESET D840
SETSP D847     SET8279 D84A   RESVAR D856
```

```
PUPTST D865   DOPUP D86C    CLRSCR D86F
PUPVAR D875   MAIN1 D889    COMDIS D899
MEM D8AC      CANCLEM D8B8  UPDATE D8C5
KEYRD D8C9    ADD1 D8DC     HEX1 D8E4
ROTATE D8EA   OCT1 D8F2     ACCUM D8FE
SETTASKA D915  CLR7SG D91C   DSPOUT D91D
RETURN D927   SSEG D928     DATOUT D940
DSPLY D943    DSPLY1 D948   HEXPR D951
LSTDIG D961   OCTPR D96E    HLDOUT D985
ENDHLD D9A6   STPERR D9AA   MSGAT6 D9B0
MSGGEN D9B4   MSGAT0 D9C0   PREVM D9C7
SAVEHL D9CB   STOREM D9D1   DODASH D9DC
TIMEOUT D9E2  NEXTM D9EC    HIM D9F3
LOM D9FD      GOM DA07      SETUPGO DA17
NOBRONCO DA32  PREVR DA43    SHOWRG DA52
REGOUT DA5C   PRTDAT DA6C   PRT2TST DA73
STORER DA81   NEXTR DA8A    REG DA9C
CANCLR DAA8   STEP DAAE     UNDEF DACE
EXPLIC DADE   GROUPS DB00   BYTES3 DB2A
BYTES2 DB2E   BYTE1 DB32    DOINST DB38
EXRET DB48    FINISH DB49   FINISH1 DB53
FINISH2 DB54  EXPCHL DB59   MET DB5F
EXCALL DB6D   EXJMP DB74    EXRST DB7C
EXCOND DB87   NOTMET DB9E   ENDSTP DBAD
ERRERR DBB9   SWAP1 DBBD    SWAP2 DBC6
COPYBTOH DBD5  STUCK DBE1    COPY DBE7
LOOP2 DC08    NOMEM DC20    MSG8 DC23
AUX DC2B      CANCLEAUX DC37  AUXOUTA DC44
PREVAUX DC59  PREVAUX1 DC6B  NOTSTRAUX DC71
STOREAUX DC83  NEXTAUX DC93  NOTENDAUX DCA8
OPTIONAUX DCBA  PRTOFF DCCF   PRTON DCD5
PRTRAM DCDB   PRTROM DCE1   PRT708 DCE7
PRT716 DCED   SETTASKC DCF3  LOAD DCFA
DUMP DD2D     PROM DD4D     DOCLR DD71
CLRCOOD DD89  CLRBAD DD8F   DOPOP DD96
DODUP DDA5    DOBAUD DDC6   INRBAUD DDD3
RESBAUD DDDC  INRBAUD1 DDDF  DSPLYBAUD DDE6
WEHITBR1 DDFA  PTB DE06      SETBP DE1F
REMOVEBP DE1F  SRCTST DE2D   STAOFF DE37
DESTST DE3F   LENTST DE46   SETUPBR1 DE4D
```

```
STEPD DE56     CANCEL DE63    MULTISTEP DE6E
MORESTEPS DE78   PT2 DE90       STEPBP DE9C
BPADRCHK DEA2   STEPC DEBA     SETMEMMAP DEC2
SAVEM DECC     TASKA DEE7     TASKG DEE9
TASKB DEEB     TASKC DEED     TASKD DEEF
TASKE DEF1     TASKF DEF3     TABLEM DEF5
TABLER DF15    TABLEAUX DF35   MBR1H DF55
MBR1L DF5F     MSTEP DF69     MSRC DF73
MDES DF7D      MLEN DF87      MCLR DF91
MPOP DF9B      MDUP DFA5      MPRM DFAF
MMAP DFB9      MBAUD DFC3     STRVAR DFCD
SAUX DFDC      BR1DAT DFE0    SBR1H DFE3
SBR1L DFE5     SSTEP DFE7     SSRC DFE9
SDES DFEB      SLEN DFED      SCLR DFEF
SPOP DFF1      SDUP DFF3      SPRM DFF5
SMAP DFF7      SBAUD DFF9     ENDVAR E001
MSGFL E003     MSGA E006      MSGB E009
MSGC E00C      MSGD E00F      MSGE E012
MSGH E015      MSGL E018      MSGSH E01B
MSGSL E01E     UNDTBL E021    SSEGTB E02E
BAUD110 E03E   BAUD150 E048   BAUD300 E052
BAUD600 E05C   BAUD1200 E066   MSGGO E070
MSGERR E073    MSGREG E076    MSGDSH E07A
MSGOFF E07D    MSGON E080     MSGROM E083
MSGRAM E086    MSG708 E089    MSG716 E08C
MSGTO E08F     MSGBLO E092    MSGONES E098
MSGDATA E09E   MSGNOT E0A4    MSGLOADIN E0AD
MSGDUMPIN E0B6   MSGDUDONE E0BF   MSGONEGOD E0C8
MSGONEBAD E0D1   MSGPOPING E0DA   MSGPOPDON E0E4
MSGDATGOD E0ED   MSGDATBAD E0F6   MSGBREAK1 E0FF
MSGXECVER E108   MSGSTAOFF E111   CDUMP E11A
LEADER E120    DUMP1 E149     ENDCHECK E14B
EOD E158       CLOAD E15C     A E163
B E168         GETBYTE E186   ENDCK E18B
POPALL E190    CASSOUT E195   CTOUTB E1A6
CTOUTD E1B8    CTOUTE E1C4    CTOUTG E1C8
PULSE E1DB     PULSA E1DD     PULSB E1E6
CASSIN E1FC    CTINA E202     CTINB E204
CTINC E20B     CTIND E21F     CTINE E222
CTINF E224     CTING E22B     CTINH E239
INI E248       VERIFY E256    LOOP1 E25D
RDPROM E265    BLANK E272     ONES E27A
```

```
ERROR E283     POPNRET E288   RDSOURCE E28D
INCDADR E29C   LIMIT E29E     PRTLEDS0 0000
PRTLEDS1 0001  PRTLEDS2 0002  PRTSEROUT 000A
PRTPRMIN 0008  NOPROGPLS 0004  PROGPLS 0000
PRMRBIT 0007   PRMWBIT 0040   POPPER E2B4
COUNTER E2BB   COUNT1 E2C7    SENDDADR E2C9
SENDDATA E2D5  DOPULSE E2D8   SETC100 E2E4
PULSEON E2E6   TIMER E2ED     PTIMELOOP E2EF
PULSEOFF E2F7  DADRLIMIT E300  DCRLOOP E313
LOOPEQ0 E314   FINALCOND E317  TTYIN E322
TEST E327      TTYIB E336     WAITH E347
DEC E34A       WAIT E353      DC E356
TTYOUT E361    SNDTTY E36A    DCXH E385
REST E390      DCREST E393    SIM1 FC00
SIM2 FC01      SIM3 FC02      SIM4 FC03
SIM5 FC04      SIM6 FC05      STEPAREA1 FC06
STEPAREA2 FC07  STEPAREA3 FC08  HLBUF1 FC09
VECT1 FC0A     VECT2 FC0B     VECT3 FC0C
REGFL FC0D     REGA FC0E      REGC FC0F
REGB FC10      REGE FC11      REGD FC12
PUSHRG FC13    USERHL FC14    USERHL1 FC16
USERSC FC18    EXECSP FC1A    STEPADR FC1C
INSTATE FC1E   LASTKEY FC1F   LASTKEYCO FC20
ERRFLAG FC21   EA FC22        CONTROL FC23
SADR FC24      SDATA FC26     DDATA FC27
DADR FC28      STATUS FC2A    STATEBUF FC2B
DATBUF FC2C    HLBUF FC2D     DSPUP FC2F
TABLE FC31     TASK FC33      USEREVENT FC35
DSPREG FC38    AUXMESPTR FC3A  AUXSTATE FC3C
BAUDPTR FC3E   BR1USERD FC40  BR1H FC43
BR1L FC45      STEP1 FC47     SRC FC49
SA FC4A        SSTART FC4A    DES FC4B
LEN FC4D       CLR FC4F       POP FC51
DUP FC53       PRM FC55       MAP FC57
BAUD FC59      USERSP FC5B    TYHLFBD FC5D
TYONEBDI FC5F  TYONEBDO FC61  SABACK FC63
BR1SO FC64     STEPSO FC65    SRCSO FC66
DESSO FC67     LENSO FC68     CLRSO FC69
POPSO FC6A     DUPSO FC6B     PRMSO FC6C
MAPSO FC6D     BAUDSO FC6E
```

(Blank Page)