6502 ASSEMBLER/DISASSEMBLER DATE: 6/19/80

PROGRAM LISTING

****

LINE NO.	ADDR.	OPCODE	OPERAND	COMMENTS
0001	1300	LDA...	0213	BEGIN INPUT SUBROUTINE.
0002	1303	PHA...	....	SAVE DATA IN ROM WORKSPACE.
0003	1304	LDA...	0214	
0004	1307	PHA...	....	
0005	1308	LDA...	0215	
0006	130B	PHA...	....	
0007	130C	LDA...	0216	
0008	130F	PHA...	....	
0009	1310	LDA...	13FF	LOAD INPUT DATA INTO WORKSPACE.
0010	1313	STA...	0214	
0011	1316	LDA...	13FE	
0012	1319	STA...	0215	
0013	131C	LDA...	13FD	
0014	131F	STA...	0216	
0015	1322	JSR...	1358	JUMP TO VECTOR DISPLAY AND KEY-
0016	1325	TAY...	....	BOARD INPUT SUBROUTINE. SAVE
0017	1326	STA...	13FC	TYPED CHARACTER AT 13FC AND Y
0018	1329	LDA...	0216	REGISTER.
0019	132C	STA...	13FD	SAVE INPUT DATA IN OTHER MEMORY.
0020	132F	LDA...	0215	
0021	1332	STA...	13FE	
0022	1335	LDA...	0214	
0023	1338	STA...	13FF	
0024	133B	PLA...	....	RESTORE DATA IN ROM WORKSPACE.
0025	133C	STA...	0216	
0026	133F	PLA...	....	
0027	1340	STA...	0215	
0028	1343	PLA...	....	
0029	1344	STA...	0214	
0030	1347	PLA...	....	
0031	1348	STA...	0213	
0032	134B	TYA...	....	LOAD ACCUMULATOR WITH KEY AND
0033	134C	RTS...	....	RETURN.
0034	134D	LDA...	..00	BEGIN ZERO ADDRESS OUTPUT ROUTINE.
0035	134F	JSR...	1450	DISPLAY MSD ('00').
0036	1352	LDA...	..00	
0037	1354	JSR...	1450	DISPLAY LSD ('00').
0038	1357	RTS...	....	RETURN.
0039	1358	LDA...	1412	BEGIN VECTOR DISPLAY AND KEY-
0040	135B	STA...	167C	BOARD INPUT SUBROUTINE.
0041	135E	LDA...	1413	SAVE CONTENTS OF CURSOR VECTOR.
0042	1361	STA...	13FB	
0043	1364	LDA...	..67	LOAD NEW CURSOR VECTOR (ADDRESS).
0044	1366	STA...	1412	
0045	1369	LDA...	..D3	
0046	136B	STA...	1413	
0047	136E	LDA...	14C4	LOAD ACCUMULATOR WITH FLAG2.
0048	1371	LSR...	....	TEST 'NEW ADDRESS' BIT.
0049	1372	BCS...	137A	IF SET, OUTPUT SECONDARY LOCATION
0050	1374	JSR...	134D	COUNTER AT BOTTOM OF SCREEN,
0051	1377	JMP...	1386	ELSE OUTPUT ZERO ADDRESS.
0052	137A	LDA...	14C6	
0053	137D	JSR...	1450	DISPLAY MSD OF LOCATION COUNTER.

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
0054      1380   LDA...   14C5
0055      1383   JSR...   1450     DISPLAY LSD OF LOCATION CTR.
0056      1386   JSR...   1426     OUTPUT SPACE.
0057      1389   LDY...   ..07     SET CHARACTER COUNTER.
0058      138B   LDA,Y.   13F3     LOAD CHARACTER FOR WORD 'VECTOR'
0059      138E   JSR...   1400       FROM TABLE AND DISPLAY VIA
0060      1391   DEY...   ....       OUTPUT SUBRTN. (1400)
0061      1392   BNE...   138B
0062      1394   LDA...   14C3     LOAD A REG. WITH FLAG1 BYTE.
0063      1397   LSR...   ....       TEST 'SAVE' MODE BIT.
0064      1398   LSR...   ....
0065      1399   BCC...   13AA     IF SET, OUTPUT 'SAVE' VECTOR,
0066      139B   LDA...   14C8       ELSE CHECK OTHER VECTORS.
0067      139E   JSR...   1450     DISPLAY MSD OF 'SAVE' VECTOR.
0068      13A1   LDA...   14C7
0069      13A4   JSR...   1450     DISPLAY LSD OF 'SAVE' VECTOR.
0070      13A7   JMP...   13C3     JUMP TO CURSOR RESTORE AND KEY
0071      13AA   LDA...   14C4       INPUT SUBRTN. (FD00)
0072      13AD   LSR...   ....     TEST 'MOVE' AND 'EDIT' STATUS
0073      13AE   LSR...   ....       BITS.
0074      13AF   BCS...   13B7     IF SET, THEN OUTPUT VECTOR,
0075      13B1   JSR...   134D       ELSE OUTPUT ZERO ADDRESS.
0076      13B4   JMP...   13C3
0077      13B7   LDA...   1DAF     DISPLAY MSD OF MODE VECTOR.
0078      13BA   JSR...   1450
0079      13BD   LDA...   1DAE     DISPLAY LSD OF MODE VECTOR.
0080      13C0   JSR...   1450
0081      13C3   LDA...   ..20     RESTORE CURSOR AND CURSOR
0082      13C5   JSR...   1411       VECTOR.
0083      13C8   LDA...   167C
0084      13CB   STA...   1412
0085      13CE   LDA...   13FB
0086      13D1   STA...   1413
0087      13D4   JSR...   FD00     JUMP TO KEYBOARD INPUT SUBRTN.
0088      13D7   RTS...   ....       RETURN.
0089      13D8   STA...   14C3     BEGIN ROUTINE TO CLEAR FLAGS.
0090      13DB   STA...   14C4
0091      13DE   RTS...   ....       RETURN.
0092      13DF   LDX...   14C0     BEGIN SECTION TO RESTORE REG.
0093      13E2   TXS...   ....       VALUES PRIOR TO EXECUTION.
0094      13E3   LDX...   14BE       RESTORE STACK PTR. AND X,
0095      13E6   LDY...   14BF         THE Y REGISTER,
0096      13E9   LDA...   14C1         AND PLACE P ON THE STACK.
0097      13EC   PHA...   ....
0098      13ED   LDA...   14BD       RESTORE ACCUMULATOR.
0099      13F0   PLP...   ....       RESTORE PROCESSOR FLAGS.
0100      13F1   JPI...   14C7       RETURN.
0101      13F3   ......   ....
0102      13F4   .....:   ....     BEGIN TABLE OF DATA FOR OUTPUT.
0103      13F5   .....R   ....
0104      13F6   .....O   ....
0105      13F7   .....T   ....
0106      13F8   .....C   ....
0107      13F9   .....E   ....
0108      13FA   .....V   ....     END TABLE.
0109-0112        (13FB-13FF)      DATA STORAGE.
```

```
**********************************************************************
```

LINE NO.	ADDR.	OPCODE	OPERAND	COMMENTS
0113	1400	JSR...	1411	BEGIN OUTPUT SUBROUTINE
0114	1403	INC...	1412	OUTPUT CHARACTER AND
0115	1406	BNE...	140B	INCREMENT CURSOR VECTOR.
0116	1408	INC...	1413	
0117	140B	RTS...	....	RETURN.
0118	140B	LDA...	..A4	ALTERNATE CODE:
0119	140D	JSR...	1411	OUTPUT CURSOR CHARACTER
0120	1410	RTS...	....	(MODIFIED BY ASSEMBLER)
0121	1411	STA...	----	BEGIN CHARACTER OUTPUT.
0122	1414	RTS...	....	RETURN.
0123	1415	LDA...	..20	BEGIN SPACE OUTPUT SUBRTN.
0124	1417	JSR...	1411	OUTPUT SPACE.
0125	141A	RTS...	....	RETURN.
0126	141B	DEC...	14C2	BEGIN DELAY SUBROUTINE.
0127	141E	BNE...	141B	DECREMENT LOCATION 256
0128	1420	CLC...	....	TIMES x THE CONTENTS OF THE
0129	1421	ADC...	..FF	ACCUMULATOR.
0130	1423	BNE...	141B	
0131	1425	RTS...	....	RETURN.
0132	1426	LDA...	..20	BEGIN SPACE OUTPUT SUBRTN.
0133	1428	JSR...	1400	WITH CURSOR ACTION.
0134	142B	RTS...	....	RETURN.
0135	142C	LDA...	..2E	BEGIN PERIOD OUTPUT SUBRTN.
0136	142E	JSR...	1400	NO. OF PERIODS PRINTED =
0137	1431	DEY...	....	CONTENTS OF Y REGISTER.
0138	1432	BNE...	142C	
0139	1434	RTS...	....	RETURN.
0140	1435	CLC...	....	BEGIN CURSOR ADJUST SUBRTN.
0141	1436	LDA...	..10	INCREMENTS CURSOR TO BEG.
0142	1438	ADC...	1412	OF NEXT LINE.
0143	143B	STA...	1412	
0144	143E	BCC...	1443	
0145	1440	INC...	1413	
0146	1443	RTS...	....	RETURN.
0147	1444	LDA...	----	BEGIN VLC SUBROUTINE.
0148	1447	INC...	1445	LOAD ACCUMULATOR WITH DATA
0149	144A	BNE...	144F	POINTED TO BY A VIDEO LOC.
0150	144C	INC...	1446	COUNTER AND INCREMENT IT.
0151	144F	RTS...	....	RETURN.
0152	1450	TAY...	....	BEGIN HEX OUTPUT SUBRTN.
0153	1451	JSR...	145B	OUTPUT TWO HEX CHARACTERS
0154	1454	TYA...	....	REPRESENTING DATA IN
0155	1455	AND...	..0F	THE ACCUMULATOR.
0156	1457	JSR...	145F	
0157	145A	RTS...	....	RETURN.
0158	145B	LSR...	....	BEGIN HEX CHARACTER PROCESSOR
0159	145C	LSR...	....	SUBROUTINE (CALLED BY 1450)
0160	145D	LSR...	....	
0161	145E	LSR...	....	
0162	145F	TAX...	....	
0163	1460	LDA,X.	14F0	GET CHARACTER FROM TABLE.
0164	1463	JSR...	1400	OUTPUT THE CHARACTER.
0165	1466	RTS...	....	RETURN.
0166	1467	LDA...	..C7	BEGIN ASSEMBLER MESSAGES SUBRTN.
0167	1469	STA...	1412	SET UP CURSOR.
0168	146C	LDA...	..D0	
0169	146E	STA...	1413	

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0170    1471   LDX...   ..09      SET UP CHARACTER COUNTER.
  0171    1473   LDA,X.   14CE      GET CHARACTER FROM TABLE.
  0172    1476   JSR...   1400      OUTPUT THE CHARACTER.
  0173    1479   DEX...   ....      FORMS 9 CHARACTER WORD:
  0174    147A   BNE...   1473        'ASSEMBLER'
  0175    147C   JSR...   1426      OUTPUT SPACE.
  0176    147F   LDY...   ..18      SET UP FOR 4 CHARACTER MODE
  0177    1481   LDA...   14C3      LOAD ACCUMULATOR WITH
  0178    1484   LSR...   ....        FLAG1 BYTE FOR TEST OF
  0179    1485   BCC...   1495        MODE BIT.
  0180    1487   LDX...   ..04      SET UP CHARACTER COUNTER.
  0181    1489   LDA,Y.   14D7      GET CHARACTER FROM TABLE.
  0182    148C   JSR...   1400      OUTPUT THE CHARACTER.
  0183    148F   DEY...   ....      WORD EXAMPLE:
  0184    1490   DEX...   ....        'CODE'
  0185    1491   BNE...   1489
  0186    1493   BEQ...   149B
  0187    1495   DEY...   ....      BIT TEST FAILED - ADJUST Y
  0188    1496   DEY...   ....        REGISTER AND PROCEED TO
  0189    1497   DEY...   ....        NEXT BIT IN FLAG1 BYTE.
  0190    1498   DEY...   ....
  0191    1499   BNE...   1494
  0192    149B   JSR...   1426      OUTPUT SPACE.
  0193    149E   LDA...   14C3      CHECK BASE BIT - 'H' OR 'D'
  0194    14A1   ASL...   ....
  0195    14A2   LDA...   ..48
  0196    14A4   BCC...   14A8
  0197    14A6   LDA...   ..44
  0198    14A8   JSR...   1400      OUTPUT THE CHARACTER.
  0199    14AB   JSR...   1415      BLANK OUT CURSOR.
  0200    14AE   RTS...   ....      RETURN.
  0201    14AF   LDY...   ..8D    BEGIN CODE MODIFICATION SUBRTN.
  0202    14B1   STY...   1444      ALLOWS DATA TO BE STORED IN
  0203    14B4   JSR...   1444      MEMORY POINTED TO BY VLC.
  0204    14B7   LDY...   ..AD      RESTORE CODE.
  0205    14B9   STY...   1444
  0206    14BC   RTS...   ....      RETURN.
  0207    14BD   .....    ....    ACCUMULATOR STORAGE.
  0208    14BE   .....    ....    X REGISTER STORAGE.
  0209    14BF   .....    ....    Y REGISTER STORAGE.
  0210    14C0   .....    ....    S REGISTER STORAGE.
  0211    14C1   .....    ....    P REGISTER STORAGE.
  0212    14C2   .....    ....    UTILITY.
  0213    14C3   .....    ....    FLAG1 BYTE.
  0214    14C4   .....    ....    FLAG2 BYTE.
  0215    14C5   .....    ....    NEXT-LINE ADDRESS (LO).
  0216    14C6   .....    ....    NEXT-LINE ADDRESS (HI).
  0217    14C7   .....    ....    PRIMARY LOCATION COUNTER (LO).
  0218    14C8   .....    ....    PRIMARY LOCATION COUNTER (HI).
  0219    14C9   .....    ....    UTILITY.
  0220    14CA   .....    ....    UTILITY.
  0221    14CB   .....    ....    UTILITY.
  0222    14CC   .....    ....    UTILITY.
  0223    14CD   .....    ....    UTILITY.
  0224    14CE   .....    ....    SCREEN LINE COUNTER.
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
  0225    1630   STA...  14BD     BEGIN COLD START.
  0226    1633   PHP...  ....       SAVE REGISTER CONTENTS FOR
  0227    1634   PLA...  ....       LATER RECALL BY OPERATOR.
  0228    1635   STA...  14C1
  0229    1638   STX...  14BE
  0230    163B   STY...  14BF
  0231    163E   TSX...  ....
  0232    163F   STX...  14C0
  0233    1642   CLD...  ....     CLEAR BCD INDICATOR BIT.
  0234    1643   LDX...  ..00     SET UP REGISTERS FOR CLEARING
  0235    1645   LDA...  ..20       SCREEN.
  0236    1647   STA,X.  D000     LOOP THROUGH VIDEO MEMORY,
  0237    164A   STA,X.  D100       STORING '20' IN EACH LOCATION.
  0238    164D   STA,X.  D200
  0239    1650   STA,X.  D300
  0240    1653   INX...  ....
  0241    1654   BNE...  1647
  0242    1656   LDA...  ..52     DISPLAY 'R/S' PROMPTS ON SCREEN.
  0243    1658   STA...  D0C7
  0244    165B   LDA...  ..2F
  0245    165D   STA...  D0C8
  0246    1660   LDA...  ..53
  0247    1662   STA...  D0C9
  0248    1665   JSR...  FD00     JUMP TO INPUT SUBROUTINE FOR
  0249    1668   CMP...  ..53       REPLY TO PROMPTS.
  0250    166A   BEQ...  1672     COMPARE REPLY TO 'S' AND 'R'.
  0251    166C   CMP...  ..52
  0252    166E   BEQ...  1686
  0253    1670   BNE...  1665     INVALID REPLY.  REPEAT INPUT.
  0254    1672   LDA...  ..00     REPLY=S.  START INITIALIZATION.
  0255    1674   STA...  167C       SET UP COUNTERS TO CLEAR
  0256    1677   TAX...  ....       MEMORY IN WORKSPACE.
  0257    1678   LDY...  ..13       SET UP PAGE COUNTER.
  0258    167A   STA,X.  --00         LOOP THROUGH PAGE, STORING
  0259    167D   INX...  ....         '0' IN EACH LOCATION.
  0260    167E   BNE...  167A
  0261    1680   INC...  167C       GO TO NEXT PAGE.
  0262    1683   DEY...  ....       DECREMENT PAGE COUNTER.
  0263    1684   BNE...  167A
  0264    1686   LDX...  ..FF       INITIALIZE STACK POINTER.
  0265    1688   TXS...  ....
  0266    1689   LDA...  ..4C       INITIALIZE 'BREAK' ROUTINE.
  0267    168B   STA...  01C0         ALLOWS AN EXECUTED 'BRK'
  0268    168E   LDA...  ..30         INSTRUCTION TO TRANSFER
  0269    1690   STA...  01C1         TO THE ASSEMBLER
  0270    1693   LDA...  ..16         IMMEDIATE CONTROL.
  0271    1695   STA...  01C2
  0272    1698   LDA...  ..20     REPLY=R.  RESTART ASSEMBLER.
  0273    169A   STA...  14C3     INITIALIZE FLAG BYTES TO BEGIN
  0274    169D   LDA...  ..00       OPERATION WITH 'CODE-DISPLAY'.
  0275    169F   STA...  14C4
  0276    16A2   STA...  14C7     INIT. PRIMARY LOCATION COUNTER.
  0277    16A5   STA...  14C8
  0278    16A8   LDA...  ..60     MODIFY CODE TO INHIBIT CURSOR
  0279    16AA   STA...  140B       OUTPUT.
  0280    16AD   JSR...  1467     PRINT MESSAGES AT TOP OF SCREEN.
  0281    16B0   LDA...  14C7     LOAD VLC WITH PLC (INITIALIZE
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
  0282    16B3   STA...  1445       VIDEO LOCATION COUNTER).
  0283    16B6   LDA...  14C8
  0284    16B9   STA...  1446
  0285    16BC   LDA...  ..07     INITIALIZE CURSOR.
  0286    16BE   STA...  1412
  0287    16C1   LDA...  ..D1
  0288    16C3   STA...  1413
  0289    16C6   LDA...  ..12     INITIALIZE SCREEN LINE COUNTER.
  0290    16C8   STA...  14CE
  0291    16CB   LDA...  1446     OUTPUT CONTENTS OF VLC TO SCREEN.
  0292    16CE   JSR...  1450       DISPLAY HI PART.
  0293    16D1   LDA...  1445
  0294    16D4   JSR...  1450       DISPLAY LO PART.
  0295    16D7   LDA...  14C3     LOAD ACCUMULATOR WITH FLAG1 BYTE.
  0296    16DA   ASL...  ....       IS ASSEMBLER IN WRITE MODE?
  0297    16DB   ASL...  ....
  0298    16DC   BCC...  16E3       NO. GO ON TO NEXT CHECK.
  0299    16DE   LDY...  ..A9       YES. MODIFY CODE FOR CURSOR
  0300    16E0   STY...  140B         OUTPUT.
  0301    16E3   PHA...  ....         SAVE ACCUMULATOR FOR OUTPUT
  0302    16E4   JSR...  1426         OF SPACE.
  0303    16E7   PLA...  ....         RESTORE ACCUMULATOR.
  0304    16E8   BCC...  16ED
  0305    16EA   JMP...  187C         JUMP TO WRITE SECTION.
  0306    16ED   ASL...  ....     NEXT CHECK - IS ASSEMBLER SET FOR
  0307    16EE   ASL...  ....       DATA DISPLAY?
  0308    16EF   BCC...  16F4       NO. GO ON TO NEXT CHECK.
  0309    16F1   JMP...  1871       YES. JUMP TO DATA DISPLAY SEC.
  0310    16F4   ASL...  ....     NEXT CHECK - IS ASSEMBLER SET FOR
  0311    16F5   BCC...  170D       CHARACTER DISPLAY?
  0312    16F7   LDY...  ..05       YES. OUTPUT 5 PERIODS TO SCREEN
  0313    16F9   JSR...  142C         IN INSTRUCTION FIELD,
  0314    16FC   JSR...  1444         GET CODE POINTED TO BY VLC,
  0315    16FF   JSR...  1400         OUTPUT CHARACTER CODE,
  0316    1702   JSR...  1426         AND OUTPUT SPACE
  0317    1705   LDY...  ..04         AND 4 PERIODS TO DATA FIELD.
  0318    1707   JSR...  142C
  0319    170A   JMP...  1770         JUMP TO END OF DISPLAY SEC.
  0320    170D   JSR...  174A     START MNEMONIC OUTPUT - JUMP TO
  0321    1710   CPX...  ..00       INSTRUCTION SEARCH FOR SINGLE
  0322    1712   BEQ...  175B       ADDRESSING MODE INSTRUCTIONS.
  0323    1714   LDY...  ..03     SEARCH SUCCESSFUL - USING RE-
  0324    1716   INX...  ....       TURNED POINTER TO TABLE IN X,
  0325    1717   LDA,X.  1514       PRINT 3-CHARACTER MNEMONIC.
  0326    171A   JSR...  1400
  0327    171D   DEY...  ....
  0328    171E   BNE...  1716
  0329    1720   LDY...  ..03       OUTPUT INSTRUCTION FIELD
  0330    1722   JSR...  142C       FILLER PERIODS.
  0331    1725   JSR...  1426       OUTPUT SPACE BETWEEN FIELDS.
  0332    1728   CPX...  ..24     DETERMINE BY POSITION IN TABLE
  0333    172A   BPL...  175B       IF INSTRUCTION IS A BRANCH.
  0334    172C   JSR...  1444       YES. GET DISPLACEMENT AND
  0335    172F   TAX...  ....         CALCULATE EFFECTIVE ADDRESS.
  0336    1730   CLC...  ....
  0337    1731   ADC...  1445
  0338    1734   PHA...  ....
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0339    1735   LDA...   ..00
  0340    1737   ADC...   1446
  0341    173A   TAY...   ....
  0342    173B   TXA...   ....
  0343    173C   BPL...   173F
  0344    173E   DEY...   ....
  0345    173F   TYA...   ....
  0346    1740   JSR...   1450     OUTPUT HI PART OF ADDRESS.
  0347    1743   PLA...   ....
  0348    1744   JSR...   1450     OUTPUT LO PART OF ADDRESS.
  0349    1747   JMP...   1770     JUMP TO END OF DISPLAY SECTION.
  0350    174A   JSR...   1444     BEGIN SINGLE ADDRESS MODE INS.
  0351    174D   LDX...   ..A0       SEARCH SUBRTN.  RETURNS POS.
  0352    174F   CMP,X.   1514       IN TABLE IN X REGISTER.
  0353    1752   BEQ...   175A       IF X=0, SEARCH FAILED.
  0354    1754   DEX...   ....
  0355    1755   DEX...   ....
  0356    1756   DEX...   ....
  0357    1757   DEX...   ....
  0358    1758   BNE...   174F
  0359    175A   RTS...   ....       RETURN.
  0360    175B   JMP...   17E8       (EXTENDS BRANCH AT LINE 322).
  0361    175E   CPX...   ..30     INS. IS NOT A BRANCH.  IS IT A
  0362    1760   BPL...   1794       JUMP INSTRUCTION?
  0363    1762   JSR...   1444       YES. GET BOTH PARTS OF THE
  0364    1765   PHA...   ....         EFFECTIVE ADDRESS,
  0365    1766   JSR...   1444
  0366    1769   JSR...   1450         OUTPUT THE HI PART,
  0367    176C   PLA...   ....
  0368    176D   JSR...   1450         AND OUTPUT THE LO PART.
  0369    1770   JSR...   1415     LINE COMPLETED - BLANK CURSOR.
  0370    1773   JSR...   1435       ADJUST CURSOR FOR NEXT LINE.
  0371    1776   DEC...   14CE       DECREMENT LINE COUNTER.
  0372    1779   BEQ...   1791         IF ZERO, WAIT FOR INPUT.
  0373    177B   LDA...   ..11     HAS ASSEMBLER FINISHED SECOND LINE
  0374    177D   CMP...   14CE       FROM TOP?
  0375    1780   BNE...   178E       NO. JUMP TO BEG. OF DISPLAY SEC.
  0376    1782   LDA...   1445       YES. SAVE SECOND ADDRESS ON
  0377    1785   STA...   14C5         SCREEN FOR BECOMING FIRST
  0378    1788   LDA...   1446         ADDRESS WHEN SCREEN IS SCROLLED
  0379    178B   STA...   14C6         1 LINE (REPRINTED).
  0380    178E   JMP...   16CB     BACK TO BEG. FOR NEXT LINE.
  0381    1791   JMP...   188D     SCREEN FULL - GO TO INPUT SEC.
  0382    1794   LDY...   ..04     BEGIN ROUTINE TO FILL DATA FIELD
  0383    1796   JSR...   142C       WITH 4 FILLER PERIODS.
  0384    1799   BEQ...   1770       RETURN TO CONTROLLING ROUTINE.
  0385    179B   STA...   14CD     BEGIN MULTI-ADDRESS MODE INS.
  0386    179E   TAY...   ....       SEARCH SUBRTN.
  0387    179F   AND...   ..E3       GENERALIZE CODE IN ACCUMULATOR.
  0388    17A1   LDX...   ..69       IF X=0, SEARCH FAILED.
  0389    17A3   CMP,X.   15B3       RETURNS POS. IN TABLE IN X.
  0390    17A6   BEQ...   17B0       RETURNS POS. OF ADDRESSING MODE
  0391    17A8   DEX...   ....         MODIFIER IN MODIFIER TABLE
  0392    17A9   DEX...   ....         AT LOC. 14CC.
  0393    17AA   DEX...   ....       RETURNS CODE ITSELF AT LOC. 14CD.
  0394    17AB   DEX...   ....
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0395    17AC   DEX...   ....
  0396    17AD   BNE...   17A3
  0397    17AF   RTS...   ....       RETURN IF SEARCH FAILED.
  0398    17B0   STX...   14CB       SAVE POSITION IN TABLE.
  0399    17B3   TYA...   ....       RESTORE CODE TO ACCUMULATOR.
  0400    17B4   CLC...   ....
  0401    17B5   AND...   ..1C       NOW MASK ALL BUT BITS 2,3,4.
  0402    17B7   CPX...   ..18       IS INS. ONE OF THE FIRST FOUR
  0403    17B9   BPL...   17C1         IN TABLE? (CPX,CPY,LDX,LDY)
  0404    17BB   CMP...   ..00         YES. ADJUST ADDRESS MODE
  0405    17BD   BNE...   17C1           INDIC. (BITS 2,3,4) TO
  0406    17BF   LDA...   ..08           COMPENSATE FOR VARIATION.
  0407    17C1   CPY...   ..BE       IS CODE 'LDX,Y'? (STILL IN Y
  0408    17C3   BNE...   17C7         REGISTER).
  0409    17C5   LDA...   ..18         YES. ADJUST ADDRESS MODE.
  0410    17C7   LDY...   ..00       SET UP REGISTERS TO FIND
  0411    17C9   LDX...   ..18         ADDR. MODE MODIFIER IN TABLE.
  0412    17CB   INY...   ....       LOOP THROUGH THE 8 SELECTIONS.
  0413    17CC   CMP,X.   14FF
  0414    17CF   BEQ...   17D6       FOUND IT.
  0415    17D1   DEX...   ....
  0416    17D2   DEX...   ....
  0417    17D3   DEX...   ....
  0418    17D4   BNE...   17CB
  0419    17D6   STX...   14CC       STORE POS. AT LOC. 14CC.
  0420    17D9   LDX...   14CB       LOAD X WITH POS. IN CODE TABLE.
  0421    17DC   LDA,X.   15B7       GET ADDRESS MODE INFO. FROM TABLE.
  0422    17DF   LSR...   ....       SHIFT TO APPROPRIATE BIT
  0423    17E0   DEY...   ....         INDICATING (IF SET) VALID
  0424    17E1   BNE...   17DF         ADDRESS MODE.
  0425    17E3   BCS...   17E7       VALID MODE - OK TO PRINT.
  0426    17E5   LDX...   ..00       INVALID - NO MNEMONIC TO PRINT.
  0427    17E7   RTS...   ....       RETURN.
  0428    17E8   JSR...   179B     BACK TO MAIN CODE. JUMP TO ABOVE
  0429    17EB   CPX...   ..00       MULTI-ADDRESS MODE SEARCH.
  0430    17ED   BNE...   1802     UNSUCCESSFUL.  OUTPUT DATA ONLY.
  0431    17EF   LDY...   ..06       FILL INS. FIELD WITH PERIODS.
  0432    17F1   JSR...   142C
  0433    17F4   JSR...   1426       OUTPUT A SPACE.
  0434    17F7   LDY...   ..02       SEND 2 PERIODS TO DATA FIELD.
  0435    17F9   JSR...   142C
  0436    17FC   LDA...   14CD       GET UNDISASSEMBLED CODE...
  0437    17FF   JMP...   176D         AND DISPLAY IN DATA FIELD.
  0438    1802   LDY...   ..03     SUCCESSFUL.  DISPLAY 3 CHAR.
  0439    1804   INX...   ....       MNEMONIC.
  0440    1805   LDA,X.   15B3
  0441    1808   JSR...   1400       OUTPUT CHARACTER.
  0442    180B   DEY...   ....
  0443    180C   BNE...   1804
  0444    180E   INY...   ....       INCREMENT Y TO '01'...
  0445    180F   JSR...   142C         TO OUTPUT 1 PERIOD.
  0446    1812   LDX...   14CC       LOAD X WITH POS. OF MODIFIER.
  0447    1815   LDY...   ..02       LOAD CHARACTER COUNTER.
  0448    1817   DEX...   ....
  0449    1818   LDA,X.   14FF
  0450    181B   JSR...   1400       DISPLAY 2 CHAR. MODIFIER.
  0451    181E   DEY...   ....
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0452    181F   BNE...   1817
  0453    1821   JSR...   1426       OUTPUT SPACE BETWEEN FIELDS.
  0454    1824   CPX...   ..08       DETERMINE BY POS. IN TABLE
  0455    1826   BPL...   182B         IF INS. IS ABSOLUTE ADDR.
  0456    1828   JMP...   1762         YES. JUMP TO ADDR. OUTPUT.
  0457    182B   CPX...   ..0D         IS INS. IMMEDIATE? (DATA?)
  0458    182D   BEQ...   1838         YES. DISPLAY DATA IN FIELD.
  0459    182F   JSR...   1444         NO. INS. MUST BE O-PAGE
  0460    1832   PHA...   ....           INSTRUCTION.  LOAD UP
  0461    1833   LDA...   ..00           O-PAGE ADDRESS AND
  0462    1835   JMP...   1769           DISPLAY.
  0463    1838   JSR...   1444       GET DATA FOLLOWING IMM. INS.
  0464    183B   TAX...   ....       SAVE IN X REGISTER.
  0465    183C   LDA...   14C3       GET FLAG1 BYTE FOR BASE STATUS.
  0466    183F   ASL...   ....       IS BIT SET? (BASE=DECIMAL?)
  0467    1840   BCS...   1865         YES. DISPLAY DATA BASE 10.
  0468    1842   LDY...   ..02         NO. DISPALY DATA BASE 2.
  0469    1844   JSR...   142C           FIRST PRINT 2 FILL PERIODS..
  0470    1847   TXA...   ....           TRANSFER DATA TO ACCUMULATOR.
  0471    1848   JMP...   176D           JUMP TO DATA DISP. CODE.
  0472    184B   LDX...   ..03         SET UP REGS. FOR BASE 10 CONV.
  0473    184D   LDY...   ..00         SUBROUTINE.
  0474    184F   SEC...   ....         ROUTINE COUNTS NO. OF SUB-
  0475    1850   SBC,X.   1620           TRACTIONS WHEN SUBTRACTING
  0476    1853   INY...   ....           '100', '10', '1' IN THAT
  0477    1854   BCS...   1850           ORDER, GENERATING THE DEC.
  0478    1856   ADC,X.   1620           DIGITS OF THE NO.
  0479    1859   PHA...   ....
  0480    185A   LDA,Y.   14EF       BASED ON OFFSET IN Y, GET CHAR.
  0481    185D   JSR...   1400         FROM TABLE AND DISPLAY.
  0482    1860   PLA...   ....
  0483    1861   DEX...   ....
  0484    1862   BNE...   184D
  0485    1864   RTS...   ....
  0486    1865   LDY...   ..01       SEND ONE FILL PERIOD TO DATA
  0487    1867   JSR...   142C         FIELD.
  0488    186A   TXA...   ....       RESTORE DATA TO ACCUMULATOR.
  0489    186B   JSR...   184B       PERFORM CONVERSION.
  0490    186E   JMP...   1770       JUMP TO END OF DISPLAY SEC.
  0491    1871   LDY...   ..06       BEGIN ROUTINE TO FILL INS. FIELD.
  0492    1873   JSR...   142C         PRINT 6 FILL PERIODS.
  0493    1876   JSR...   1426         PRINT SPACE BETWEEN FIELDS.
  0494    1879   JMP...   1838       RETURN TO CONTROLLING RTN.
  0495    187C   ASL...   ....     BEGIN ROUTINE TO DET. IF IN DATA
  0496    187D   ASL...   ....       MODE FOR WRITING.
  0497    187E   BCC...   188D       NO. GO ON TO MAIN INPUT ROUTINE.
  0498    1880   LDY...   ..06       YES. FILL INS. FIELD.
  0499    1882   JSR...   142C
  0500    1885   JSR...   1426       OUTPUT SPACE BETWEEN FIELDS.
  0501    1888   LDX...   ..04       SET UP INPUT CHAR. COUNTER.
  0502    188A   JMP...   188F       JUMP INTO INPUT ROUTINE.
  0503    188D   LDX...   ..0B     BEGIN KEY INPUT AND TEST RTN.
  0504    188F   STX...   14CB       SAVE KEY COUNTER.
  0505    1892   JMP...   1FD1       GENERAL KEY INPUT.
  0506    1895   PHA...   ....       SAVE KEY IN STACK.
  0507    1896   LDY...   ..06       LOAD KEY COMPARISON COUNTER.
  0508    1898   CMP,Y.   1623       COMPARE TO TABLE OF CTRL KEYS.
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
0509      189B   BEQ...  18A2     MATCHED CTRL A,C,D,L,S,E.
0510      189D   DEY...  ....
0511      189E   BNE...  1898
0512      18A0   BEQ...  18B3       TESTS FAILED - GO ON TO OTHERS.
0513      18A2   LDA...  14C3       SUCCESS - GET FLAG1 BYTE.
0514      18A5   AND...  ..C0       MASK ANY ACTIVE MODE BIT.
0515      18A7   CPY...  ..04       EXAMINE CMP. COUNTER TO DET.
0516      18A9   BCC...  18AD         IF KEY IS CTRL L,S,E.
0517      18AB   AND...  ..80       YES. TURN OFF WRITE BIT.
0518      18AD   ORA,Y.  1629       SET APPROPRIATE BIT FOR KEY.
0519      18B0   JMP...  1FE0       CLEAR FLAG2 BYTE.
0520      18B3   CMP...  ..02     IS KEY CTRL B?
0521      18B5   BNE...  18ED       NO. CONTINUE TESTING.
0522      18B7   LDA...  14C3       YES. GET FLAG1 BYTE.
0523      18BA   EOR...  ..80       COMPLEMENT BASE INDIC. BYTE.
0524      18BC   STA...  14C3       PLACE BACK IN MEMORY.
0525      18BF   ASL...  ....
0526      18C0   ASL...  ....       SHIFT WRITE BIT INTO CARRY.
0527      18C1   PLA...  ....       GET CHARACTER AGAIN.
0528      18C2   BCC...  18EA       BIT CLEAR.  RETURN TO DISP. SEC.
0529      18C4   LDA...  1412       SAVE CURSOR ADDRESS IN STACK.
0530      18C7   PHA...  ....
0531      18C8   LDA...  1413
0532      18CB   PHA...  ....
0533      18CC   JSR...  1467       UPDATE INFO. AT SCREEN TOP.
0534      18CF   PLA...  ....       RESTORE CURSOR ADDRESS.
0535      18D0   STA...  1413
0536      18D3   PLA...  ....
0537      18D4   STA...  1412
0538      18D7   JSR...  1415       REMOVE CURSOR.
0539      18DA   LDA...  1412       LDA WITH LO PART OF CURSOR. ADDR.
0540      18DD   CLC...  ....
0541      18DE   ADC...  14CB       ADJUST TO BEG. POS. OF LINE.
0542      18E1   SEC...  ....
0543      18E2   SBC...  ..10
0544      18E4   STA...  1412       RESTORE CURSOR ADDRESS.
0545      18E7   JMP...  16CB       RETURN TO DISP. SEC.
0546      18EA   JMP...  16A8       RETURN TO DISP. SEC.
0547      18ED   CMP...  ..17     IS KEY CTRL W?
0548      18EF   BNE...  190B       NO. CONTINUE TESTING.
0549      18F1   LDA...  14C3       YES. GET FLAG1 BYTE.
0550      18F4   EOR...  ..40       COMPLEMENT WRITE BIT.
0551      18F6   STA...  14C3       RESTORE FLAG1 BYTE.
0552      18F9   AND...  ..07       MASK ALL BUT CTRL L,S,E BITS.
0553      18FB   BEQ...  1907       ARE ANY OF THEM SET?
0554      18FD   LDA...  14C3       YES. CLEAR THEM.
0555      1900   AND...  ..F8
0556      1902   ORA...  ..20       SET THE CODE BIT.
0557      1904   JSR...  13D8       UPDATE THE FLAGS.
0558      1907   PLA...  ....       GET THE KEY OF THE STACK.
0559      1908   JMP...  16A8       RETURN TO DISP. SECTION.
0560      190B   CMP...  ..1B     IS KEY 'ESC'?
0561      190D   BNE...  1913       NO. CONTINUE TESTING.
0562      190F   PLA...  ....       YES. GET KEY OFF STACK AND
0563      1910   JMP...  1642         WARM START ASSEMBLER.
0564      1913   CMP...  ..11     IS KEY CTRL Q?
0565      1915   BNE...  192A       NO. PREPARE TO EXIT INPUT RTN.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0566    1917   LDA...   ..20       YES. CLEAR LOWER 3/4 OF SCRN.
  0567    1919   LDX...   ..00
  0568    191B   STA,X.   D100
  0569    191E   STA,X.   D200
  0570    1921   STA,X.   D300
  0571    1924   INX...   ....
  0572    1925   BNE...   191B
  0573    1927   JMP...   1D0F     DETERMINE WRITE STATUS.
  0574    192A   LDA...   14C3   LOAD ACCUMULATOR WITH FLAG1 BYTE.
  0575    192D   ASL...   ....
  0576    192E   ASL...   ....   DROP WRITE BIT INTO CARRY.
  0577    192F   PLA...   ....   GET KEY OFF OF STACK.
  0578    1930   LDX...   14CB   GET CHAR. CTR. (USED DURING WRITE).
  0579    1933   BCC...   1938   BIT IS SET - GO ON TO ASSEMBLY.
  0580    1935   JMP...   19C0
  0581    1938   LDY...   ..10   BIT CLEAR - SET UP TO DETERMINE
  0582    193A   CMP,Y.   14EF     IF KEY IS HEX  CHARACTER.
  0583    193D   BEQ...   1944     YES. PREPARE TO SAVE KEY.
  0584    193F   DEY...   ....
  0585    1940   BNE...   193A
  0586    1942   BEQ...   196A     NO. GO ON TO NEXT TEST.
  0587    1944   LSR...   14C4       SHIFT RIGHTMOST BIT OF FLAG2.
  0588    1947   BCS...   1952       IF SET, KEY IS NOT FIRST KEY
  0589    1949   LDX...   ..00         TO FOLLOW LAST CR.
  0590    194B   STX...   14C5       OTHERWISE, ZERO SECONDARY LOC.
  0591    194E   STX...   14C6         COUNTER THAT HEX KEYS ARE
  0592    1951   SEC...   ....         ROTATED INTO.
  0593    1952   ROL...   14C4       SET HEX KEY INDIC. BIT.
  0594    1955   DEY...   ....
  0595    1956   TYA...   ....
  0596    1957   ASL...   .....
  0597    1958   ASL...   ....
  0598    1959   ASL...   ....
  0599    195A   ASL...   ....     GENERATE LEAST 4 BITS OF HEX CHAR.
  0600    195B   LDY...   ..04     LOOP THROUGH A SHIFTING RTN.
  0601    195D   ASL...   ....       WHICH ROTATES KEY INTO LOC.
  0602    195E   ROL...   14C5       COUNTER.
  0603    1961   ROL...   14C6
  0604    1964   DEY...   ....
  0605    1965   BNE...   195D
  0606    1967   JMP...   19A8     COMPLETED. GO TO END OF SEC.
  0607    196A   CMP...   ..0D   IS KEY 'RETURN'?
  0608    196C   BNE...   1984     NO. GO ON TO NEXT TEST.
  0609    196E   LDA...   14C5     YES. TRANSFER CONTENTS OF
  0610    1971   STA...   14C7       SEC. LOC. COUNTER TO
  0611    1974   LDA...   14C6       PRIMARY LOC. COUNTER.
  0612    1977   STA...   14C8
  0613    197A   LSR...   14C4       CLEAR HEX KEY INDIC. BIT.
  0614    197D   CLC...   ....
  0615    197E   ROL...   14C4
  0616    1981   JMP...   16A8     BACK TO DISPLAY SEC.
  0617    1984   CMP...   ..1D   IS KEY (SHIFT) 'RETURN'?
  0618    1986   BNE...   19A1     NO. GO ON TO NEXT TEST.
  0619    1988   LDA...   14C4     YES. FIRST CHECK FOR HEX CHARS..
  0620    198B   LSR...   ....
  0621    198C   BCC...   1991
  0622    198E   JMP...   196E     NO. LOAD PLC.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0623    1991   DEC...   14C7       YES. PRIMARY LOC. COUNTER.
  0624    1994   LDY...   ..FF
  0625    1996   CPY...   14C7
  0626    1999   BNE...   199E
  0627    199B   DEC...   14C8
  0628    199E   JMP...   16A8       RETURN TO DISPLAY SECTION.
  0629    19A1   CMP...   ..18     IS KEY CTRL X? (FORMERLY R).
  0630    19A3   BNE...   19A8       NO. GO ON TO END OF INPUT SEC.
  0631    19A5   JMP...   13DF       EXECUTE CODE POINTED BY PLC.
  0632    19A8   JMP...   1DEB     FINAL KEY CHECK - LINEFEED?
  0633    19AB   LSR...   ....     SHIFT 'MOVE', 'SAVE', and 'EDIT'
  0634    19AC   BCC...   19B1       INDIC. BITS OF FLAG1 BYTE INTO
  0635    19AE   JMP...   1E04       CARRY TO DETERMINE IF ASSEMBLER
  0636    19B1   LSR...   ....       SHOULD JUMP TO 'MOVE', 'SAVE',
  0637    19B2   BCC...   19B7       or 'EDIT' CODE SECTIONS.
  0638    19B4   JMP...   1D1C
  0639    19B7   LSR...   ....
  0640    19B8   BCC...   19BD
  0641    19BA   JMP...   1D79
  0642    19BD   JMP...   1892       NO BITS SET - BACK TO INPUT SEC.
  0643    19C0   CMP...   ..0A     ASSEMBLING/TRANSLATING SECTION.
  0644    19C2   BNE...   19CA       KEY NOT LINEFEED - NEXT CHECK.
  0645    19C4   JSR...   1447       YES. INCREMENT VIDEO PC AND
  0646    19C7   JMP...   18D7         JUMP TO CURSOR ADJUST CODE.
  0647    19CA   CMP...   ..1A
  0648    19CC   BNE...   19DE       KEY NOT (SHIFT) LINEFEED....
  0649    19CE   DEC...   1445       YES. DECREMENT VIDEO PC (VPC).
  0650    19D1   LDY...   ..FF
  0651    19D3   CPY...   1445
  0652    19D6   BNE...   19DB
  0653    19D8   DEC...   1446
  0654    19DB   JMP...   18D7       JUMP TO CURSOR ADJUSTING CODE.
  0655    19DE   CMP...   ..7F
  0656    19E0   BNE...   19F3       KEY NOT RUB-OUT - NEXT CHECK.
  0657    19E2   CPX...   ..0B       ARE THERE ANY CHARS. TO RUB?
  0658    19E4   BEQ...   19F0         NO. SKIP CURSOR ADJUSTMENT.
  0659    19E6   INX...   ....         YES. INCREMENT SPACE CTR.
  0660    19E7   JSR...   1415         BLANK OUT CURSOR.
  0661    19EA   DEC...   1412         DECREMENT CURSOR ADDRESS.
  0662    19ED   JSR...   140B         REPRINT CURSOR.
  0663    19F0   JMP...   188F         BACK TO INPUT SECTION.
  0664    19F3   CMP...   ..0D
  0665    19F5   BNE...   19FA       KEY NOT 'RETURN' - NEXT CHECK.
  0666    19F7   JMP...   1A31         YES. JUMP TO LINE ASSEMBLING.
  0667    19FA   CMP...   ..20     IS KEY CODE LESS THAN ACCEPTABLE?
  0668    19FC   BCC...   19F0         YES. BACK TO INPUT SECTION.
  0669    19FE   CMP...   ..5B     IS KEY CODE MORE THAN ACCEPTABLE?
  0670    1A00   BCS...   19F0         YES. BACK TO INPUT SECTION.
  0671    1A02   DEX...   ....     DEC. SPACE AVAILABLE COUNTER.
  0672    1A03   BPL...   1A08     LINE NOT FULL - CONTINUE.
  0673    1A05   INX...   ....
  0674    1A06   BEQ...   19F0     LINE FULL - BACK TO INPUT SEC.
  0675    1A08   CMP...   ..20     IS KEY SPACEBAR?
  0676    1A0A   BEQ...   1A11       YES. GO TO SPACE FILLER RTN.
  0677    1A0C   JSR...   1400       NO. OUTPUT CHAR. WHATEVER IT IS.
  0678    1A0F   BNE...   1A14
  0679    1A11   JSR...   1A17         JUMP TO SUBROUTINE.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0680    1A14   JMP...   188F       BACK TO INPUT SECTION.
  0681    1A17   LDY...   ..01     BEGIN SPACEBAR FILLER ROUTINE.
  0682    1A19   CPX...   ..05       BASED ON SPACE-REMAINING CTR.
  0683    1A1B   BMI...   1A24       IN X, AUTOMATICALLY POSITIONS
  0684    1A1D   JSR...   142C       CURSOR AT BEG. OF DATA FIELD
  0685    1A20   INY...   ....       UNLESS CURSOR IS IN DATA FIELD.
  0686    1A21   DEX...   ....
  0687    1A22   BNE...   1A19
  0688    1A24   JSR...   1426
  0689    1A27   RTS...   ....       RETURN.
  0690    1A28   LDA,Y.   ----     BEGIN ROUTINE TO LOAD TYPED CHAR.
  0691    1A2B   RTS...   ....       ON SCREEN LINE IN ACCUMULATOR.
  0692    1A2C   .....A   ....     REGISTER DATA TABLE.
  0693    1A2D   .....X   ....
  0694    1A2E   .....Y   ....
  0695    1A2F   .....S   ....
  0696    1A30   .....P   ....
  0697    1A31   STX...   14C2     LINE ASSEMBLING SECTION.
  0698    1A34   SEC...   ....
  0699    1A35   LDA...   ..0C       BASED ON COUNT STORED AT 14C2,
  0700    1A37   SBC...   14CB       CALCULATE ADDR. FOR SBRTN. AT
  0701    1A3A   EOR...   ..FF       LINE 690.
  0702    1A3C   CLC...   ....
  0703    1A3D   ADC...   ..01
  0704    1A3F   ADC...   1412
  0705    1A42   STA...   1A29       STORE GENERATED LO PART OF ADDR.
  0706    1A45   LDA...   1413       GET HI PART FROM CURSOR ADDRESS.
  0707    1A48   STA...   1A2A
  0708    1A4B   CPX...   ..05       EXAMINE COUNTER TO DET. IF DATA
  0709    1A4D   BCC...   1AB7         FIELD MUST BE FILLED BY ASSEM.
  0710    1A4F   DEX...   ....
  0711    1A50   JSR...   1A17       BRING CURSOR TO BEG. OF DATA FLD.
  0712    1A53   LDY...   ..02       IS SECOND CHAR. '?'?
  0713    1A55   JSR...   1A28         GET CHARACTER.
  0714    1A58   CMP...   ..3F         MAKE COMPARISON.
  0715    1A5A   BNE...   1A9C         NO. GO ON TO NEXT COMPARISON.
  0716    1A5C   DEY...   ....         YES. GET FIRST CHAR. ON LINE.
  0717    1A5D   JSR...   1A28           LOAD COMPARISON COUNTER.
  0718    1A60   LDY...   ..05
  0719    1A62   CMP,Y.   1A2B           RUN THROUGH TABLE AT 692.
  0720    1A65   BEQ...   1A6C           MATCH - PREPARE TO DISP. DATA.
  0721    1A67   DEY...   ....
  0722    1A68   BNE...   1A62
  0723    1A6A   BEQ...   1AB7           NO MATCH - GO TRY TO ASSEMBLE.
  0724    1A6C   LDA,Y..  14BC         GET REGISTER DATA..
  0725    1A6F   PHA...   ....         SAVE DATA.
  0726    1A70   LDA...   14C3         LOAD FLAG1 BYTE INTO ACCUMULATOR.
  0727    1A73   ASL...   ....         DROP BASE DISPLAY BIT INTO CARRY.
  0728    1A74   BCS...   1A90         SET - DISPLAY BASE 10.
  0729    1A76   LDY...   ..02         CLEAR - DISPLAY BASE 16 WITH 2
  0730    1A78   JSR...   142C           FILLER PERIODS.
  0731    1A7B   PLA...   ....         GET DATA AND...
  0732    1A7C   JSR...   1450           OUTPUT HEX CHARACTERS.
  0733    1A7F   JSR...   1415         BLANK OUT CURSOR.
  0734    1A82   JSR...   1435         ADVANCE CURSOR TO NEXT LINE.
  0735    1A85   DEC...   14CE         DECREMENT SCREEN LINE COUNTER.
  0736    1A88   BEQ...   1A8D
```

```
**********************************************************************
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
  0737    1A8A   JMP...  16CB       RETURN TO DISPLAY SECTION.
  0738    1A8D   JMP...  16BC       RETURN TO DISPLAY SECTION.
  0739    1A90   LDY...  ..01     ROUTINE TO DISPLAY 1 PERIOD
  0740    1A92   JSR...  142C       AND DATA (BASE 10) IN DATA
  0741    1A95   PLA...  ....       FIELD.
  0742    1A96   JSR...  184B
  0743    1A99   JMP...  1A7F
  0744    1A9C   LDA...  14C3     LDA WITH FLAG1 BYTE.
  0745    1A9F   AND...  ..08     MASK ALL BUT 'ASCI' BIT.
  0746    1AA1   BEQ...  1AB7   'ASCI' WRITE SECTION.
  0747    1AA3   DEY...  ....     LET Y EQUAL 0.
  0748    1AA4   JSR...  1A28     GET FIRST CHARACTER ON LINE.
  0749    1AA7   LDY...  ..0B
  0750    1AA9   CPY...  14C2     WERE ANY CHARACTERS TYPED?
  0751    1AAC   BNE...  1AB0
  0752    1AAE   LDA...  ..20     NO. DEFAULT TO BLANK CHAR.
  0753    1AB0   JSR...  14AF     PLACE DATA IN MEMORY.
  0754    1AB3   PHA...  ....
  0755    1AB4   JMP...  1A70     JUMP TO DATA FLD. DATA DISP.
  0756    1AB7   DEX...  ....   SECTION FILLS OUT DATA FIELD.
  0757    1AB8   BMI...  1AC1
  0758    1ABA   LDY...  ..01
  0759    1ABC   JSR...  142C     PRINT A PERIOD.
  0760    1ABF   BEQ...  1AB7     BACK TO TOP OF LOOP.
  0761    1AC1   LDA...  14C3   WRITING CODE OR DATA? (CHK FLAG)
  0762    1AC4   AND...  ..10
  0763    1AC6   BEQ...  1ACB     ASSEMBLE LINE OF MNEMONIC CODE.
  0764    1AC8   JMP...  1C45     PLACE DATA IN MEMORY.
  0765    1ACB   LDY...  ..01   BEFORE ASSEMBLING, GET FIRST
  0766    1ACD   JSR...  1A28     CHAR. IN INSTRUCTION FIELD.
  0767    1AD0   CMP...  ..2E     IS IT A PERIOD?
  0768    1AD2   BEQ...  1AC8     YES. ASSUME LINE HAS DATA ONLY.
  0769    1AD4   LDX...  14C2   GET CHAR. COUNTER FOR THIS LINE.
  0770    1AD7   CPX...  ..05     WAS ANY DATA TYPED?
  0771    1AD9   BMI...  1ADE     YES.  ASSEMBLE 2-BYTE INST.
  0772    1ADB   JMP...  1C71     NO. ASSEMBLE 1-BYTE INST.
  0773    1ADE   LDX...  ..69   BEGIN 2/3-BYTE ASSEMBLING SEC.
  0774    1AE0   LDY...  ..03     (MULTI-ADDRESSING MODE INST.)
  0775    1AE2   JSR...  1A28     GET Yth CHAR. OF MNEMONIC FOR
  0776    1AE5   CMP,X.  15B6       COMPARISON INTO TABLE.
  0777    1AE8   BNE...  1AF0     TEST FAILED ON Yth CHAR.-MOVE
  0778    1AEA   DEX...  ....       TABLE PTR. TO NEXT INST.
  0779    1AEB   DEY...  ....
  0780    1AEC   BNE...  1AE2     TEST SUCCEEDED.  NEXT CHAR.
  0781    1AEE   BEQ...  1AFA     FULL 3-CHAR. MATCH - EXIT SEARCH.
  0782    1AF0   DEX...  ....
  0783    1AF1   DEY...  ....
  0784    1AF2   BNE...  1AF0
  0785    1AF4   DEX...  ....
  0786    1AF5   DEX...  ....
  0787    1AF6   BNE...  1AE0     START SEARCH ON NEW INST.
  0788    1AF8   BEQ...  1ADB     NO MATCHES - GO TO NEXT SRCH.
  0789    1AFA   STX...  14CC   SAVE TABLE POS. OF INSTRUC.
  0790    1AFD   LDX...  ..18   BEGIN ADDRESSING MODE SEARCH.
  0791    1AFF   LDY...  ..05
  0792    1B01   JSR...  1A28     GET 5/6 CHAR. ON LINE.
  0793    1B04   CMP,X.  14FE     COMPARE INTO MODIFIER TABLE.
  0794    1B07   BNE...  1B12     TEST FAILED - NEXT MODIFIER.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0795    1B09   INY...   ....     SEARCH ON ADDR. MODIFIER CONT.
  0796    1B0A   JSR...   1A28     GET LAST CHARACTER OF MODIFIER.
  0797    1B0D   CMP,X.   14FD       COMPARE INTO MODIFIER TABLE.
  0798    1B10   BEQ...   1B1C       MATCHED - EXIT SEARCH.
  0799    1B12   DEX...   ....       NO MATCH - NEXT MODIFIER.
  0800    1B13   DEX...   ....
  0801    1B14   DEX...   ....
  0802    1B15   BNE...   1AFF       BACK TO TOP OF SEARCH.
  0803    1B17   LDX...   ..00       SEARCH FAILED - ERROR CONDITION;
  0804    1B19   JMP...   188F         RETURN TO INPUT SECTION.
  0805    1B1C   LDY...   ..04     ERROR TRAP -- CHECK FOR INVALID
  0806    1B1E   JSR...   1A28       CHAR. BETWEEN INST. AND MODIFIER.
  0807    1B21   AND...   ..FC
  0808    1B23   CMP...   ..2C
  0809    1B25   BNE...   1B17       INVALID - EXIT LINE ASSEM. SEC.
  0810    1B27   LDY...   ..07     ERROR TRAP -- CHECK FOR NON-
  0811    1B29   JSR...   1A28       BLANK CHAR. BETWEEN FIELDS.
  0812    1B2C   CMP...   ..20
  0813    1B2E   BNE...   1B17       INVALID - EXIT LINE ASSEM. SEC.
  0814    1B30   CPX...   ..0F     IS INST. IMMEDIATE ADDRESSING?
  0815    1B32   BNE...   1B45       NO. CONTINUE PROCESSING.
  0816    1B34   LDY...   ..08       YES. LOOP THROUGH DATA FIELD;
  0817    1B36   JSR...   1A28         IF ALL 4 POSITIONS CONTAIN
  0818    1B39   CMP...   ..2E         HEX DIGIT, ASSUME ABSOLUTE
  0819    1B3B   JMP...   1C5D         ADDRESSING INSTEAD.
  0820    1B3E   INY...   ....
  0821    1B3F   CPY...   ..0C       CHK OUT TO POSITION.11.
  0822    1B41   BNE...   1B36       BACK TO TOP OF LOOP.
  0823    1B43   LDX...   ..03       SET TO ABSOLUTE ADDRESSING.
  0824    1B45   LDY...   14CC     GET TABLE POS. OF INSTRUCTION.
  0825    1B48   LDA,Y.   15BA     GET ADDR. MODE INFO ON INSTRUCTION.
  0826    1B4B   STX...   14CD
  0827    1B4E   ASL...   ....     SHIFT AN INDIC. BIT INTO THE CARRY.
  0828    1B4F   DEX...   ....
  0829    1B50   DEX...   ....
  0830    1B51   DEX...   ....
  0831    1B52   BNE...   1B4E     BACK TO TOP OF LOOP.
  0832    1B54   BCC...   1B17     BIT NOT SET ON THAT MODE - INVALID.
  0833    1B56   LDX...   14CD
  0834    1B59   LDA,X.   14FF     GET MODE DATA SO THE INST. CODE
  0835    1B5C   CMP...   ..08       CAN BE CREATED.  THE FOLLOWING
  0836    1B5E   BNE...   1B66       SECTION ADJUSTS PECULIARITIES IN
  0837    1B60   CPY...   ..14       THE INSTRUCTION SET.
  0838    1B62   BPL...   1B66
  0839    1B64   LDA...   ..00
  0840    1B66   ORA,Y.   15B6
  0841    1B69   CMP...   ..BA
  0842    1B6B   BNE...   1B6F
  0843    1B6D   LDA...   ..BE
  0844    1B6F   STA...   14CB     SAVE CREATED INST. CODE.
  0845    1B72   JSR...   1BBA     GET THE DATA FOR THE INST. (HEX)
  0846    1B75   LDX...   14CD
  0847    1B78   CPX...   ..0F     IS INST. IMMEDIATE ADDRESSING?
  0848    1B7A   BNE...   1B85       NO. SKIP DECIMAL DATA OPTION.
  0849    1B7C   LDA...   14C3       YES. GET FLAG1 BYTE.
  0850    1B7F   ASL...   ....       DROP BASE BIT INTO CARRY.
  0851    1B80   BCC...   1B85       DEC. MODE NOT SET.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
0852      1B82   JSR...   1BF9     GET DECIMAL DATA FROM FIELD.
0853      1B85   LDX...   14CD
0854      1B88   CPX...   ..0A     IS THE ADDRESSING AN ABSOLUTE FORM?
0855      1B8A   BMI...   1B93       YES. CONTINUE PROCESSING.
0856      1B8C   LDA...   14C9       NO. MAKE SURE THAT HI PART OF
0857      1B8F   CMP...   ..00         DATA (ADDRESS) IS 00.
0858      1B91   BNE...   1B17         ERROR - EXIT LINE ASSEM. SEC.
0859      1B93   LDA...   14CB     STORE INST. AND DATA IN MEMORY
0860      1B96   JSR...   14AF       SEQUENTIALLY.
0861      1B99   LDA...   14CA
0862      1B9C   JSR...   14AF
0863      1B9F   CPX...   ..0A       STORE THIRD BYTE ONLY IF
0864      1BA1   BPL...   1BA9         ABSOLUTE FORM OF ADDRESSING IS
0865      1BA3   LDA...   14C9         USED.
0866      1BA6   JSR...   14AF
0867      1BA9   JMP...   1A7F       EXIT ASSEMBLING SECTION.
0868      1BAC   LDY...   ..00     INITIALIZATION FOR DATA SUBROUTINES
0869      1BAE   STY...   14C9       (HEX AND DECIMAL).
0870      1BB1   STY...   14CA       CLEAR DATA STORAGE BYTES.
0871      1BB4   LDY...   ..08       GET FIRST CHARACTER IN DATA FIELD.
0872      1BB6   JSR...   1A28
0873      1BB9   RTS...   ....       RETURN.
0874      1BBA   JMP...   1CF8     BEGIN DATA RETRIEVING SUBROUTINE
0875      1BBD   CMP...   ..2F       (HEX FORMAT).
0876      1BBF   BNE...   1BCB
0877      1BC1   INY...   ....       FIRST CHAR. WAS '/' - TAKE CHAR.
0878      1BC2   JSR...   1A28         FOLLOWING AS THE DATA.
0879      1BC5   STA...   14CA
0880      1BC8   JMP...   1BF8
0881      1BCB   LDX...   ..10       COMPARE DATA CHAR. TO HEX CHARS.
0882      1BCD   JSR...   1A28         IN THE TABLE.
0883      1BD0   CMP,X.   14EF
0884      1BD3   BEQ...   1BE1       MATCHED - CHAR. ALL RIGHT.
0885      1BD5   DEX...   ....
0886      1BD6   BNE...   1BD0       NO MATCH YET - NEXT CHARACTER.
0887      1BD8   CMP...   ..2E       NO MATCHES - IS IT A PERIOD?
0888      1BDA   BEQ...   1BF3         YES. CONTINUE PROCESSING.
0889      1BDC   PLA...   ....         NO. EXIT ROUTINE BY PULLING
0890      1BDD   PLA...   ....           ADDRESS OFF STACK AND
0891      1BDE   JMP...   1B17           EXITING ASSEMBLING SECTION.
0892      1BE1   DEX...   ....
0893      1BE2   TXA...   ....       TRANSFER COUNTER CONTENTS TO
0894      1BE3   LDX...   ..04         ACCUMULATOR AND ROTATE DATA
0895      1BE5   ASL...   ....         INTO DESTINATION.
0896      1BE6   ASL...   ....
0897      1BE7   ASL...   ....
0898      1BE8   ASL...   ....
0899      1BE9   ASL...   ....
0900      1BEA   ROL...   14CA
0901      1BED   ROL...   14C9
0902      1BF0   DEX...   ....
0903      1BF1   BNE...   1BE9       CONTINUE IF 4 BITS NOT SHIFTED.
0904      1BF3   INY...   ....
0905      1BF4   CPY...   ..0C       REACHED END OF DATA FIELD?
0906      1BF6   BNE...   1BCB         NO. GET NEXT CHARACTER.
0907      1BF8   RTS...   ....         YES. RETURN.
0908      1BF9   JSR...   1BAC     BEGIN DATA RETRIEVING SUBROUTINE.
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
  0909    1BFC   CMP...   ..2F       (DEC. FORMAT).
  0910    1BFE   BEQ...   1BC1       CHECK FOR '/' AS IN LAST SUBRTN.
  0911    1C00   LDY...   ..0B       INITIALIZE SCREEN CHAR. PTR.
  0912    1C02   STY...   14CC
  0913    1C05   LDA...   ..00       INITIALIZE DIGIT COUNTER.
  0914    1C07   STA...   167C
  0915    1C0A   PHA...   ....       INITIALIZE DATA VALUE IN STACK.
  0916    1C0B   LDY...   14CC
  0917    1C0E   JSR...   1A28       GET Yth CHAR. IN DATA FIELD.
  0918    1C11   LDX...   ..0A       COMPARE TO THE 10 DEC. DIGITS.
  0919    1C13   CMP,X.   14EF
  0920    1C16   BEQ...   1C23       MATCHED - CHAR. ALL RIGHT.
  0921    1C18   DEX...   ....
  0922    1C19   BNE...   1C13       NO MATCH YET - NEXT CHARACTER.
  0923    1C1B   TAX...   ....       NO MATCHES.  RESTORE DATA TO
  0924    1C1C   PLA...   ....         ACCUMULATOR AND CHECK FOR
  0925    1C1D   CPX...   ..2E         PERIOD CHARACTER.
  0926    1C1F   BEQ...   1C37         O.K.  GO TO BOTTOM OF LOOP.
  0927    1C21   BNE...   1BDC         INVALID - EXIT ROUTINE.
  0928    1C23   DEX...   ....
  0929    1C24   PLA...   ....       RESTORE DATA TO ACCUMULATOR.
  0930    1C25   LDY...   167C       ROUTINE NOW SUBTRACTS AS MANY
  0931    1C28   DEX...   ....         OF THE CURRENT POWER OF 10
  0932    1C29   BMI...   1C34         AS POSSIBLE TO GENERATE A BIT
  0933    1C2B   CLC...   ....         VALUE (STORED AT 14CA).
  0934    1C2C   ADC,Y.   1621         POWERS OF 10 ARE IN TABLE OF
  0935    1C2F   BCS...   1BDC         DATA.
  0936    1C31   DEX...   ....
  0937    1C32   BPL...   1C2C
  0938    1C34   INC...   167C       INCREMENT DIGIT COUNTER.
  0939    1C37   DEC...   14CC       DECREMENT SCREEN CHAR. PTR.
  0940    1C3A   LDY...   ..07
  0941    1C3C   CPY...   14CC       END OF DATA FIELD?
  0942    1C3F   BNE...   1C0A         NO. BACK TO TOP OF LOOP.
  0943    1C41   STA...   14CA         YES. SAVE DATA AND EXIT.
  0944    1C44   RTS...   ....       RETURN.
  0945    1C45   JSR...   1BBA     SECTION FOR READING DATA FIELD ONLY.
  0946    1C48   LDA...   14C3       GET FLAG1 BYTE FOR BASE STATUS.
  0947    1C4B   ASL...   ....       DROP BASE BIT INTO CARRY.
  0948    1C4C   BCC...   1C51
  0949    1C4E   JSR...   1BF9       BIT SET - GET DECIMAL DATA.
  0950    1C51   LDA...   ..00       EXAMINE HI PART OF DATA
  0951    1C53   CMP...   14C9         DESTINATION.  IF IT IS NOT
  0952    1C56   BNE...   1BDE         ZERO, DATA IS TOO LARGE.
  0953    1C58   LDX...   ..0B       DATA O.K. - EXIT TO DATA
  0954    1C5A   JMP...   1B99         STORING SECTION.
  0955    1C5D   BEQ...   1C62     SECTION FOR USE AT 1B3B ONLY.
  0956    1C5F   JMP...   1B3E       (SEE LINE 819).
  0957    1C62   LDY...   14CC
  0958    1C65   LDA,Y.   15BA
  0959    1C68   AND...   ..08
  0960    1C6A   BNE...   1C6E
  0961    1C6C   LDX...   ..03
  0962    1C6E   JMP...   1B48
  0963    1C71   LDX...   ..A0     BEGIN SINGLE-ADDRESSING MODE INST.
  0964    1C73   LDY...   ..03       SEARCH.  (COMPLEMENT TO SEARCH
  0965    1C75   JSR...   1A28       AT LINE 773).
```

```
LINE NO.  ADDR.  OPCODE    OPERAND  COMMENTS
0966      1C78   CMP,X.    1517       COMPARE Yth CHAR. OF MNEMONIC
0967      1C7B   BNE...    1C83         TO Xth CHAR. IN TABLE.
0968      1C7D   DEX...    ....       TEST SUCCEEDED ON Yth CHAR. -
0969      1C7E   DEY...    ....         MOVE PTRS. TO NEXT CHARS.
0970      1C7F   BNE...    1C75
0971      1C81   BEQ...    1C8D       FULL 3-CHAR. MATCH - EXIT LOOP.
0972      1C83   DEX...    ....       TEST FAILED - ADJUST POINTER
0973      1C84   DEY...    ....         TO NEXT INSTRUCTION.
0974      1C85   BNE...    1C83
0975      1C87   DEX...    ....
0976      1C88   BNE...    1C73       START SEARCH ON NEXT INST.
0977      1C8A   JMP...    1B17       INVALID MNEMONIC - EXIT SECTION.
0978      1C8D   LDA,X.    1517     SEARCH SUCCESSFUL - GET THE HEX
0979      1C90   STA...    14CB       CODE FROM TABLE ALSO AND SAVE IT.
0980      1C93   CPX...    ..20       CHECK RELATIVE POS. IN TABLE.
0981      1C95   BMI...    1CA9       INST. IS A BRANCH - PROCESS ADDR.
0982      1C97   CPX...    ..2C
0983      1C99   BPL...    1CA3       INST. IS 1 BYTE - STORE DIRECTLY.
0984      1C9B   JSR...    1BBA       INST. IS 3 BYTE JUMP - GET ADDR.
0985      1C9E   LDX...    ..00         FOR INSTRUCTION AND JUMP TO
0986      1CA0   JMP...    1B93         DATA STORING SECTION.
0987      1CA3   JSR...    14AF     STORE SINGLE-BYTE INSTRUCTION
0988      1CA6   JMP...    1A7F       AND EXIT ASSEMBLING SECTION.
0989      1CA9   JSR...    1BBA   BRANCH ADDRESS CONVERTING SECTION.
0990      1CAC   SEC...    ....     ADDRESS IS AT 14C9-14CA.
0991      1CAD   LDA...    14CA
0992      1CB0   SBC...    ..02       SUBTRACT 2 FROM ADDRESS.
0993      1CB2   STA...    14CA
0994      1CB5   BCS...    1CBA
0995      1CB7   DEC...    14C9
0996      1CBA   SEC...    ....       SUBTRACT ADDRESS OF FOLLOWING
0997      1CBB   LDA...    14CA       INST. FROM BRANCH ADDRESS TO GIVE
0998      1CBE   SBC...    1445       ROUGH DISPLACEMENT (IN X REG.).
0999      1CC1   TAX...    ....
1000      1CC2   LDA...    14C9
1001      1CC5   SBC...    1446
1002      1CC8   PHP...    ....       SAVE FLAGS FOLLOWING SUBTRACTION.
1003      1CC9   CMP...    ..00       CHECK FOR DISPLACEMENT OUT OF
1004      1CCB   BEQ...    1CD5         BOUNDS (HI PART TOO HIGH).
1005      1CCD   CMP...    ..FF
1006      1CCF   BEQ...    1CD5
1007      1CD1   PLP...    ....       OUT OF BOUNDS - PULL FROM STACK
1008      1CD2   JMP...    1B17         AND EXIT ASSEMBLING SECTION.
1009      1CD5   TXA...    ....       CHECK DIRECTION AND SIZE OF
1010      1CD6   LDY...    ..FF         DISPLACEMENT TO AVOID EXCEEDING
1011      1CD8   CPY...    14C9         LIMITS:
1012      1CDB   BNE...    1CE1           FWD DISP: 00 - 7F
1013      1CDD   PLP...    ....           BWD DISP: 80 - FF.
1014      1CDE   JMP...    1CE9
1015      1CE1   PLP...    ....
1016      1CE2   BCC...    1CE9
1017      1CE4   ASL...    ....
1018      1CE5   BCS...    1CD2
1019      1CE7   BCC...    1CEC
1020      1CE9   ASL...    ....
1021      1CEA   BCC...    1CD2
1022      1CEC   ROR...    ....
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
  1023    1CED   TAX...  ....       SAVE DISP. IN X REGISTER.
  1024    1CEE   LDA...  14CB       STORE HEX CODE FOR BRNCH IN MEMORY.
  1025    1CF1   JSR...  14AF
  1026    1CF4   TXA...  ....       RESTORE DISP. TO ACCUMULATOR.
  1027    1CF5   JMP...  1BA6       JUMP TO DATA STORING SECTION.
  1028    1CF8   LDY...  ..08     SECTION FOR USE AT 1BBA ONLY.
  1029    1CFA   JSR...  1A28       (SEE LINE 874).
  1030    1CFD   CMP...  ..2E       LOOPS THROUGH DATA FIELD TO
  1031    1CFF   BNE...  1D09         SEE IF ANY DIGITS WERE TYPED.
  1032    1D01   INY...  ....
  1033    1D02   CPY...  ..0C
  1034    1D04   BNE...  1CFA
  1035    1D06   JMP...  1BDC       NO CHARS. - EXIT ASSEMBLING SEC.
  1036    1D09   JSR...  1BAC       YES CHARS. - INITIALIZE.
  1037    1D0C   JMP...  1BBD       RETURN TO THE SUBROUTINE.
  1038    1D0F   LDA...  14C3     SECTION FOR USE AT 1927 ONLY.
  1039    1D12   ASL...  ....       (SEE LINE 573).
  1040    1D13   ASL...  ....       DETERMINES WRITE BIT STATUS.
  1041    1D14   BCS...  1D19
  1042    1D16   JMP...  18F1       JUMP INTO INPUT SECTION.
  1043    1D19   JMP...  16BC       JUMP INTO DISPLAY SECTION.
  1044    1D1C   LDA...  13FC     CASSETTE-SAVE SECTION.  (CTRL S)
  1045    1D1F   CMP...  ..20       WAS SPACEBAR THE INPUT?
  1046    1D21   BNE...  1D63       NO. SEND CHAR. TO CASSETTE PORT.
  1047    1D23   LDA...  14C4       YES. GET FLAG2 BYTE TO MAKE
  1048    1D26   LSR...  ....         SURE THAT NO ADDRESS CHAR.
  1049    1D27   BCS...  1D63         HAS BEEN TYPED.
  1050    1D29   LDA...  14C7       NO ADDR. CHARS., SO MOVE NEXT
  1051    1D2C   STA...  1D36         PRIMARY ADDRESS TO THE 'SAVE'
  1052    1D2F   LDA...  14C8         VECTOR.
  1053    1D32   STA...  1D37
  1054    1D35   LDA...  ----         THE 'SAVE' VECTOR.
  1055    1D38   TAY...  ....         SAVE BYTE IN Y REGISTER.
  1056    1D39   LSR...  ....         SHIFT BYTE TO TURN IT INTO A
  1057    1D3A   LSR...  ....           DISPLACEMENT INTO THE HEX
  1058    1D3B   LSR...  ....           CHARACTER TABLE.
  1059    1D3C   LSR...  ....
  1060    1D3D   JSR...  1D6C         SEND THE CORRESPONDING CHAR.
  1061    1D40   TYA...  ....         RETRIEVE THE BYTE TO SEND THE
  1062    1D41   AND...  ..0F           SECOND HEX CHARACTER.
  1063    1D43   JSR...  1D6C
  1064    1D46   LDA...  ..0D         SEND A CARRIAGE-RETURN.
  1065    1D48   JSR...  1D70
  1066    1D4B   INC...  1D36         INCREMENT THE VECTOR.
  1067    1D4E   BNE...  1D5B
  1068    1D50   INC...  1D37
  1069    1D53   LDA...  14C5         COMPARE VECTOR TO ADDR. OF
  1070    1D56   CMP...  1D36           NEXT INSTRUCTION.
  1071    1D59   BNE...  1D35         NOT EQUAL - SEND NEXT BYTE.
  1072    1D5B   LDA...  ..01         REDUCE INPUT DELAY.
  1073    1D5D   STA...  13FF
  1074    1D60   JMP...  196E         UPDATE PRIMARY LOC. COUNTER.
  1075    1D63   LDA...  13FC       LOAD ACCUMULATOR WITH TYPED CHAR.
  1076    1D66   JSR...  1D70         AND SEND IT.
  1077    1D69   JMP...  1892         RETURN TO INPUT SECTION.
  1078    1D6C   TAX...  ....       HEX CHAR. SENDING SUBROUTINE.
  1079    1D6D   LDA,X.  14F0
```

```
**********************************************************************

LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS

  1080    1D70   STA...  F001      OUTPUT CHAR. TO CASSETTE PORT.
  1081    1D73   LDA...  ..12      SET DELAY FACTOR.
  1082    1D75   JSR...  141B      DELAY SUBROUTINE EXECUTION.
  1083    1D78   RTS...  ....      RETURN TO 'SAVE' SECTION.
  1084    1D79   LDA...  13FC    DATA MOVING SECTION.  (CTRL M)
  1085    1D7C   CMP...  ..4D      WAS THE CHAR. 'M' TYPED?
  1086    1D7E   BNE...  1DE8      NO. EXIT SECTION.
  1087    1D80   LDA...  14C4      GET FLAG2 BYTE.
  1088    1D83   LSR...  ....      DROP ADDR. FLAG INTO CARRY.
  1089    1D84   BCS...  1D8B      NEW ADDR. ENTERED - SAVE IT.
  1090    1D86   LSR...  ....      DROP EXECUTION BIT INTO CARRY.
  1091    1D87   BCS...  1D97      IF SET, THEN MOVING SECTION HAS
  1092    1D89   BCC...  1DDF        BEEN EXECUTED AT LEAST ONCE.
  1093    1D8B   LDA...  14C5      MOVE TYPED ADDRESS TO STORAGE.
  1094    1D8E   STA...  1DAE        (MOVE VECTOR INITIALIZATION).
  1095    1D91   LDA...  14C6
  1096    1D94   STA...  1DAF
  1097    1D97   LDA...  1DAF      EXAMINE HI PART OF 'MOVE' VECTOR.
  1098    1D9A   CMP...  ..13
  1099    1D9C   BCS...  1DDF      OUT OF LIMITS - SKIP MOVING SEC.
  1100    1D9E   LDA...  14C7      INITIALIZE 'MOVE' SOURCE ADDR.
  1101    1DA1   STA...  1DAB
  1102    1DA4   LDA...  14C8
  1103    1DA7   STA...  1DAC
  1104    1DAA   LDA...  ----      LOAD DATA FROM THE SOURCE ADDRESS.
  1105    1DAD   STA...  ----      STORE VIA 'MOVE' VECTOR.
  1106    1DB0   INC...  1DAB      INCREMENT SOURCE ADDRESS.
  1107    1DB3   BNE...  1DB8
  1108    1DB5   INC...  1DAC
  1109    1DB8   INC...  1DAE      INCREMENT DESTINATION ADDR.
  1110    1DBB   BNE...  1DC0        ('MOVE' VECTOR).
  1111    1DBD   INC...  1DAF
  1112    1DC0   LDA...  1DAF      EXAMINE HI PART OF VECTOR.
  1113    1DC3   CMP...  ..13
  1114    1DC5   BEQ...  1DDF      NOW OUT OF LIMITS - SKIP TO END.
  1115    1DC7   LDA...  1DAB      COMPARE VECTOR TO BOTTOM SCREEN
  1116    1DCA   CMP...  1445        ADDRESS (DONE WITH MOVING?).
  1117    1DCD   BNE...  1DAA
  1118    1DCF   LDA...  1DAC
  1119    1DD2   CMP...  1446
  1120    1DD5   BNE...  1DAA
  1121    1DD7   LDA...  ..02      MOVE COMPLETED - SET EXEC. FLAG.
  1122    1DD9   STA...  14C4
  1123    1DDC   JMP...  16A8      RETURN TO DISPLAY SECTION.
  1124    1DDF   LDA...  ..00      OUT-OF-LIMITS SECTION.
  1125    1DE1   STA...  14C4        CLEAR FLAGS AND JUMP INTO
  1126    1DE4   PHA...  ....        INPUT SECTION TO EXIT 'MOVE'.
  1127    1DE5   JMP...  18FD
  1128    1DE8   JMP...  1892      A JUMP TO BEGINNING OF INPUT SEC.
  1129    1DEB   CMP...  ..0A    SCREENFULL (LF) KEY CHECK.
  1130    1DED   BNE...  1DFE      SECTION FOR USE AT 19A8 ONLY.
  1131    1DEF   LDA...  1445        (SEE LINE 632).
  1132    1DF2   STA...  14C7      IF LINEFEED IS PRESSED WHILE
  1133    1DF5   LDA...  1446      NOT 'WRITING', NEXT SCREENFULL
  1134    1DF8   STA...  14C8      OF CODE IS DISPLAYED BY UPDATING
  1135    1DFB   JMP...  16A8      PRIMARY LOCATION COUNTER.
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
1136      1DFE   LDA...  14C3
1137      1E01   JMP...  19AB     RETURN TO PATCHED CODE.
1138      1E04   JMP...  1F33     EDITING SECTION.  (CTRL E)
1139      1E07   CMP...  ..59       IS TYPED CHAR. AN (SHIFT) 'I'?
1140      1E09   BEQ...  1E19       YES. ADJUST ADDRESSES FOR INSERTION.
1141      1E0B   CMP...  ..5F       IS TYPED CHAR. A (SHIFT) 'O'?
1142      1E0D   BNE...  1DE8       NEITHER. RETURN TO INPUT SECTION.
1143      1E0F   LDX...  ..FF       YES. ADJUST ADDRESSES FOR DELETION.
1144      1E11   STX...  14C9         INITIALIZE DISPLACEMENT BYTES.
1145      1E14   STX...  14CA
1146      1E17   BNE...  1E22
1147      1E19   LDX...  ..01
1148      1E1B   STX...  14C9
1149      1E1E   DEX...  ....
1150      1E1F   STX...  14CA
1151      1E22   LDA...  14C4       GET FLAG2 BYTE.
1152      1E25   LSR...  ....       DROP ADDR. CHAR. FLAG INTO CARRY.
1153      1E26   BCS...  1E2E       IF SET, UPDTAE 'EDIT' VECTOR.
1154      1E28   CLC...  ....
1155      1E29   LSR...  ....       DROP EXECUTION FLAG INTO CARRY.
1156      1E2A   BCS...  1E3F       IF SET, SKIP INITIALIZATION OF VECTOR.
1157      1E2C   BCC...  1DDF       IF CLEAR, NO ADDR. MEANS NO EDITING.
1158      1E2E   LDA...  14C5       INITIALIZE 'EDIT' VECTOR.
1159      1E31   STA...  1DAE         (THE POINT AT WHICH EDITING
1160      1E34   LDA...  14C6          BEGINS.)
1161      1E37   STA...  1DAF
1162      1E3A   LDA...  ..02       SET EXECUTION FLAG IN FLAG2 BYTE.
1163      1E3C   STA...  14C4
1164      1E3F   LDA...  1DAE       TRANSFER VECTOR CONTENTS FOR VLC
1165      1E42   STA...  1445         SUBROUTINE.
1166      1E45   LDA...  1DAF         (SEE LINES 147 TO 151).
1167      1E48   STA...  1446
1168      1E4B   CMP...  ..01       INITIAL CHECK FOR OUT-OF-LIMITS.
1169      1E4D   BEQ...  1DDF         CONTROL TRANSFERS TO 1DDF
1170      1E4F   CMP...  ..01         IF VECTOR OUT OF WORKSPACE.
1171      1E51   BEQ...  1DDF
1172      1E53   CMP...  ..13
1173      1E55   BCS...  1DDF
1174      1E57   LDA...  1446       TOP-OF-LOOP CHECK FOR VLC OUT
1175      1E5A   CMP...  ..14 /13     OF LIMITS.
1176      1E5C   BNE...  1E61
1177      1E5E   JMP...  16A8       OUT-OF-LIMITS EXIT POINT.
1178      1E61   CMP...  ..01
1179      1E63   BEQ...  1E5E
1180      1E65   JSR...  174A       FIRST ATTEMPT AT DISASSEMBLING INST.
1181      1E68   CPX...  ..00       IF X=0 THEN ATTEMPT FAILED.
1182      1E6A   BEQ...  1ECB       BRANCH TO NEXT DISASSEM. ATTEMPT.
1183      1E6C   CPX...  ..58       IS INST. 'BRK'? (BASED ON X POS.)
1184      1E6E   BEQ...  1E5E       YES.  STOP EDITING.
1185      1E70   CPX...  ..24       IS INST. A BRANCH?
1186      1E72   BCC...  1EDF       YES. PROCESS ELSEWHERE.
1187      1E74   CPX...  ..30       IS INST. ONE BYTE ONLY?
1188      1E76   BCS...  1E57       YES. PROCESS ELSEWHERE.
1189      1E78   JSR...  1444       INST. MUST BE A JUMP; GET LO PART
1190      1E7B   STA...  14CC         OF ADDRESS AND SAVE IT.
1191      1E7E   LDA...  1446       BEFORE CONTINUING, CHECK LIMITS.
1192      1E81   CMP...  ..13
```

```
LINE NO.  ADDR.  OPCODE   OPERAND  COMMENTS
********************************************************************************
    1193  1E83   BEQ...   1E51
    1194  1E85   CMP...   ..01
    1195  1E87   BEQ...   1E51
    1196  1E89   JSR...   1444     GET HI PART OF ADDRESS AND SAVE IT.
    1197  1E8C   STA...   14CD
    1198  1E8F   CMP...   ..FF
    1199  1E91   CMP...   ..01     BEFORE CONTINUING, CHECK LIMITS.
    1200  1E93   BEQ...   1E57     MAKE SURE THAT ADDRESS OF INST.
    1201  1E95   CMP...   ..13       REFERENCES WORK AREA.
    1202  1E97   BCS...   1E57
    1203  1E99   SEC...   ....     DETERMINE IF JMP ADDRESS PRE-
    1204  1E9A   LDA...   14CC       CEDES POINT OF INS./DEL.
    1205  1E9D   SBC...   14C7     IF IT DOESN'T (CARRY CLEAR),
    1206  1EA0   LDA...   14CD       NO ADJUSTMENT IS NEEDED.
    1207  1EA3   SBC...   14C8
    1208  1EA6   BCC...   1E57
    1209  1EA8   LDA...   1445     BEFORE ADJUSTING ADDRESS, RESET
    1210  1EAB   SBC...   ..02       VLC TO SECOND BYTE OF INST.
    1211  1EAD   STA...   1445
    1212  1EB0   BCS...   1EB5
    1213  1EB2   DEC...   1446
    1214  1EB5   CLC...   ....     NOW ADJUST AND STORE ADDRESS,
    1215  1EB6   LDA...   14CC       USING DISPLACEMENT PREVIOUSLY
    1216  1EB9   ADC...   14C9       DETERMINED  (SEE LINE 1144).
    1217  1EBC   JSR...   14AF
    1218  1EBF   LDA...   14CD
    1219  1EC2   ADC...   14CA
    1220  1EC5   JSR...   14AF
    1221  1EC8   JMP...   1E57     BACK TO TOP OF LOOP.
    1222  1ECB   JSR...   179B     SECOND DISASSEMBLING ATTEMPT.
    1223  1ECE   CPX...   ..00     IF X=0 THEN CODE NOT AN INST.
    1224  1ED0   BEQ...   1E5E
    1225  1ED2   LDA...   14CC     EXAMINE ADDRESSING MODE OF INST.
    1226  1ED5   CMP...   ..0A     IF CARRY SET, INST. IS THREE BYTES.
    1227  1ED7   BCS...   1EDC     IF CLEAR, INST. IS ONLY TWO BYTES.
    1228  1ED9   JSR...   1447       AFTER ADJUSTING VLC FOR NO.
    1229  1EDC   JMP...   1FE8       OF BYTES, RETURN TO LOOP TOP.
    1230  1EDF   JSR...   1444     ELSEWHERE - BRANCH PROCESSING.
    1231  1EE2   TAY...   ....     SAVE BRANCH OFFSET.
    1232  1EE3   LDA...   14C7     SUBTRACT VLC ADDRESS FROM
    1233  1EE6   SEC...   ....       ADDRESS OF INS./DEL. TO
    1234  1EE7   SBC...   1445       DETERMINE IF BRANCH IS OVER
    1235  1EEA   STA...   14CC       THE POINT OF INS./DEL. AND
    1236  1EED   LDA...   14C8       MUST BE ADJUSTED.
    1237  1EF0   SBC...   1446
    1238  1EF3   BCS...   1F1E
    1239  1EF5   CMP...   ..FF     BRANCH IS BEYOND POINT OF INS./DEL.
    1240  1EF7   BNE...   1F1B     MORE THAN A PAGE OUT OF LIMITS.
    1241  1EF9   TYA...   ....     RESTORE BRANCH OFFSET TO A REG.
    1242  1EFA   BPL...   1F1B     NOT A BACKWARD BRANCH (WHICH IT
    1243  1EFC   SEC...   ....       SHOULD BE PAST PT. OF INS./DEL.)
    1244  1EFD   SBC...   14CC     COMPARE OFFSET TO DISTANCE BE-
    1245  1F00   BCS...   1F1B       TWEEN BRANCH AND THE POINT
    1246  1F02   INY...   ....       OF INSERTION/DELETION.
    1247  1F03   LDA...   14C9     BASED ON 'EDIT' DISPLACEMENT,
    1248  1F06   BMI...   1F0A       ADJUST OFFSET IN Y REGISTER.
    1249  1F08   DEY...   ....
```

```
LINE NO.  ADDR.  OPCODE  OPERAND  COMMENTS
1250      1F09   DEY...  ....
1251      1F0A   DEC...  1445       RESET VLC TO THE BRANCH OFFSET.
1252      1F0D   LDX...  ..FF
1253      1F0F   CPX...  1445
1254      1F12   BNE...  1F17
1255      1F14   DEC...  1446
1256      1F17   TYA...  ....       PLACE OFFSET IN ACCUMULATOR.
1257      1F18   JSR...  14AF       STORE IN MEMORY OVER OLD OFFSET.
1258      1F1B   JMP...  1E57       BACK TO TOP OF LOOP.
1259      1F1E   CMP...  ..00       BRANCH IS PRIOR TO POINT OF INS./DEL.
1260      1F20   BNE...  1F1B       MORE THAN A PAGE OUT OF LIMITS.
1261      1F22   TYA...  ....       RESTORE BRANCH OFFSET TO A REG.
1262      1F23   BMI...  1F1B       NOT A FORWARD BRANCH (WHICH IT
1263      1F25   SEC...  ....         SHOULD BE).
1264      1F26   INY...  ....
1265      1F27   SBC...  14CC       COMPARE OFFSET TO DISTANCE BE-
1266      1F2A   BCC...  1F1B         TWEEN BRANCH AND THE POINT OF
1267      1F2C   JMP...  1FEE         INSERTION/DELETION.
1268      1F2F   BMI...  1F08       JUMPS TO UPDATE BRANCH OFFSET.
1269      1F31   BPL...  1F0A
1270      1F33   LDA...  14C8     'INSERTING/DELETING PORTION OF 'EDIT'.
1271      1F36   CMP...  ..01       MAKE SURE PRIMARY LOCATION CTR.
1272      1F38   BEQ...  1F42       IS WITHIN WORK AREA LIMITS.
1273      1F3A   CMP...  ..01
1274      1F3C   BEQ...  1F42
1275      1F3E   CMP...  ..13
1276      1F40   BCC...  1F45
1277      1F42   JMP...  1DDF       OUT OF LIMITS - EXIT.
1278      1F45   LDX...  14C7
1279      1F48   LDA...  13FC       GET CHARACTER TYPED.
1280      1F4B   CMP...  ..49       IS IT AN 'I'?
1281      1F4D   BEQ...  1F56       YES. PERFORM AN INSERTION.
1282      1F4F   CMP...  ..4F       IS IT AN 'O'?
1283      1F51   BEQ...  1F8A       YES. PERFORM A DELETION.
1284      1F53   JMP...  1E07       RETURN TO BEGINNING OF 'EDIT' SEC.
1285      1F56   STX...  1F69     BEGIN INSERTION ROUTINE.
1286      1F59   STX...  1F6C       INITIALIZE SOURCE AND DESTINATION
1287      1F5C   LDX...  14C8       ADDRESSES.
1288      1F5F   STX...  1F6A
1289      1F62   STX...  1F6D
1290      1F65   LDA...  ..EA       INSERT A 'NOP' INST. (HEX=EA).
1291      1F67   TAX...  ....       BEGIN INSERTION LOOP.
1292      1F68   LDA...  ----
1293      1F6B   STX...  ----
1294      1F6E   INC...  1F69       INCREMENT ADDRESSES.
1295      1F71   INC...  1F6C
1296      1F74   BNE...  1F7C
1297      1F76   INC...  1F6A
1298      1F79   INC...  1F6D
1299      1F7C   LDX...  1F6D       EXAMINE HI PART OF ADDRESSES
1300      1F7F   CPX...  ..01         FOR OUT-OF-LIMITS CONDITION.
1301      1F81   BEQ...  1F87       OUT OF LIMITS - EXIT.
1302      1F83   CPX...  ..13
1303      1F85   BNE...  1F67       CONTINUE SHIFTING DATA.
1304      1F87   JMP...  197A       EXIT 'EDIT' AND GO TO INPUT SEC.
1305      1F8A   STX...  1FB2     BEGIN DELETION ROUTINE.
1306      1F8D   INX...  ....       INITIALIZE SOURCE AND DESTINATION
```

```
**********************************************************************
```

LINE NO.	ADDR.	OPCODE	OPERAND	COMMENTS
1307	1F8E	STX...	1FA2	ADDRESSES.
1308	1F91	PHP...	....	(SOURCE ADDR. MUST BE ONE LARGER
1309	1F92	LDX...	14C8	THAN DESTINATION ADDR.)
1310	1F95	STX...	1FB3	
1311	1F98	STX...	1FA3	
1312	1F9B	PLP...	....	
1313	1F9C	BNE...	1FA1	
1314	1F9E	INC...	1FA3	
1315	1FA1	LDA...	----	LOAD FROM SOURCE ADDR. - LOOP TOP.
1316	1FA4	LDX...	1FA3	EXAMINE SOURCE ADDRESS FOR
1317	1FA7	CPX...	..01	OUT-OF-LIMITS CONDITION.
1318	1FA9	BEQ...	1FAF	
1319	1FAB	CPX...	..13	
1320	1FAD	BNE...	1FB1	
1321	1FAF	LDA...	..00	IF OUT OF LIMITS, STORE 0 IN MEMORY.
1322	1FB1	STA...	----	STORE AT DESTINATION ADDR.
1323	1FB4	INC...	1FA2	INCREMENT ADDRESSES.
1324	1FB7	BNE...	1FBC	
1325	1FB9	INC...	1FA3	
1326	1FBC	INC...	1FB2	
1327	1FBF	BNE...	1FC4	
1328	1FC1	INC...	1FB3	
1329	1FC4	LDX...	1FB3	EXAMINE DESTINATION ADDRESS FOR
1330	1FC7	CPX...	..01	OUT-OF-LIMITS CONDITION.
1331	1FC9	BEQ...	1F87	EXIT.
1332	1FCB	CPX...	..13	
1333	1FCD	BEQ...	1F87	EXIT.
1334	1FCF	BNE...	1FA1	BACK TO TOP OF LOOP.
1335	1FD1	JSR...	1300	SECTION FOR USE AT 1892 ONLY.
1336	1FD4	PHA...	....	(SEE LINE 505).
1337	1FD5	CMP...	13FE	CODE PATCHED IN TO DISCRIMINATE
1338	1FD8	BNE...	1FDD	BETWEEN 'RETURN' AND 'CTRL M'.
1339	1FDA	JMP...	190B	
1340	1FDD	JMP...	1896	
1341	1FE0	LDY...	..00	SECTION FOR USE AT 18B0 ONLY.
1342	1FE2	STY...	14C4	(SEE LINE 519).
1343	1FE5	JMP...	18BC	CLEARS FLAG2 BYTE.
1344	1FE8	JSR...	1447	SECTION FOR USE AT 1EDC ONLY.
1345	1FEB	JMP...	1E57	(SEE LINE 1229).
1346	1FEE	PHP...	....	SECTION FOR USE AT 1F2C ONLY.
1347	1FEF	LDA...	14C9	(SEE LINE 1267).
1348	1FF2	TAX...	....	CODE PATCHED TO ELIMINATE
1349	1FF3	PLP...	....	BRANCH (FWD OFFSET) PROBLEM.
1350	1FF4	BNE...	1FFC	
1351	1FF6	TXA...	....	
1352	1FF7	BPL...	1FFC	
1353	1FF9	JMP...	1F1B	RETURN - DO NOT ADJUST OFFSET.
1354	1FFC	TXA...	....	
1355	1FFD	JMP...	1F2F	RETURN - ADJUST BRANCH OFFSET.

****

TABLES OF DATA

SCREEN HEADING DATA (14CF - 14EF):

```
52 45 4C 42 4D 45 53 53 41 45 44 4F 43 41
54 41 44 49 43 53 41 45 56 4F 4D 45 56 41
53 54 49 44 45
```

HEXADECIMAL CHARACTER TABLE (14F0 - 14FF):

```
30 31 32 33 34 35 36 37 38 39 41 42 43 44
45 46
```

ADDRESSING-MODE MODIFIER TABLE (1500 - 1517):

```
2E 2E 0C 2E 58 1C 2E 59 18 2E 5A 04 2E 2E
08 58 49 00 59 49 10 49 5A 14
```

SINGLE ADDRESSING-MODE INSTRUCTION TABLE (1518 - 15B7):

```
90 42 43 43 B0 42 43 53 F0 42 45 51 30 42
4D 49 D0 42 4E 45 10 42 50 4C 50 42 56 43
70 42 56 53 4C 4A 4D 50 6C 4A 50 49 20 4A
53 52 4A 4C 53 52 0A 41 53 4C 2A 52 4F 4C
6A 52 4F 52 58 43 4C 49 B8 43 4C 56 CA 44
45 58 88 44 45 59 E8 49 4E 58 C8 49 4E 59
00 42 52 4B 18 43 4C 4C D8 43 4C 44 EA 4E
4F 50 48 50 48 41 08 50 48 50 68 50 4C 41
28 50 4C 50 40 52 54 49 60 52 54 53 38 53
45 43 F8 53 45 44 78 53 45 49 AA 54 41 58
A8 54 41 59 BA 54 53 58 8A 54 58 41 9A 54
58 53 98 54 59 41
```

MULTI ADDRESSING-MODE INSTRUCTION TABLE (15B8 - 1620):

```
E0 43 50 58 98 C0 43 50 59 98 A2 4C 44 58
B9 A0 4C 44 59 D9 C1 43 40 50 FF 61 41 44
43 FF 21 41 4E 44 FF C2 44 45 43 D1 41 45
4F 52 FF E2 49 4E 43 D1 A1 4C 44 41 FF 02
41 53 4C D1 20 42 49 54 90 42 4C 53 52 D1
01 4F 52 41 FF 22 52 4F 4C D1 62 52 4F 52
D1 E1 53 42 43 FF 81 53 54 41 F7 82 53 54
58 91 80 53 54 59 91
```

POWERS OF TEN TABLE (1621 - 1623):

```
01 0A 64
```

CONTROL CHARACTER TABLE (1624 - 162F):

```
03 04 01 0D 13 05 20 10 08 04 02 01
```

****

NOTES ON ASSEMBLER DATA STORAGE

There are 18 bytes (14BD - 14CE) that are used by the Assembler to store calculations and intermediate data, and -- with the necessary exception of 18 bytes (01EE - 01FF) of stack memory and various memory pointers throughout the Assembler -- these are the only bytes used for storage by the Assembler. All remaining memory below the Assembler is considered its workspace and is available for general use (note that stack memory may not be EDITed).

Of the 18 bytes formerly mentioned, the first five (14BD - 14C1) are used to preserve the contents of the accumulator (the A register) and the X, Y, S, and P registers immediately prior to starting (executing) the Assembler, either with a jump to location 1630 or the execution of a 'BRK' instruction (provided that the proper initialization took place). These contents may be examined with the Assembler, and they are restored to their registers when program execution (by Control X/R) resumes. This is done to facilitate program operation and debugging.

Other bytes are described below:

14C2: used as a counter in the delay subroutine (see line 126) and for short-term data storage.

14CE: used to count and limit the screen lines as they printed.

14C7/14C8: contains the address seen on the first screen line and is the Primary Location Counter (PLC).

14C5/14C6: contains the address seen on the second screen line (the next-line address) and is the Secondary Location Counter (SLC). This is transferred to the PLC when 'RETURN' is pressed, giving a scrolling effect. When an address is entered, these bytes are cleared and the address moved to them, so that -- when 'RETURN' is pressed -- the PLC will be correspondingly updated.

1412/1413: referred to as the cursor vector, these bytes contain the video position (+ 1) of the last character sent to the screen by the Assembler. The cursor is visible only if location 140B does not contain the 'RTS' instruction. At this point, the Assembler modifies itself (see lines 117/118).

1445/1446: contains the address of the byte that the Assembler is currently accessing/displaying and is called the Video Location Counter (VLC or VPC). Note that, just as the subroutine at line 147 reads from this address, the subroutine at line 201 writes to it. That is also a point where the Assembler is self-modifying.

****

NOTES ON ASSEMBLER DATA STORAGE (cont.)

167C/167D: used to clear memory (load bytes with 00). As the high part of the address is variable, 167C is also used for short-term storage.

1A29/1A2A: contains the video address of a character of the line that the Assembler is currently attempting to assemble, and it is used to access that character.

1D36/1D37: contains the address of the byte that the 'SAVE' section is currently sending to the cassette port.

1DAB/1DAC: contains the source address of data that is being MOVEd or EDITed.

1DAE/1DAF: contains corresponding destination to the above.

1F69/1F6A: contains the source address of data shifting from an insertion.

1F6C/1F6D: contains corresponding destination to the above.

1FA2/1FA3: contains the source address of data shifting to a deletion.

1FB2/1FB3: contains corresponding destination to the above.

14C3: referred to as the FLAG1 byte, it contains 8 1-bit flags. When a flag is set, the following condition is true.

Bit 0: 'EDIT' is on.	Bit 1: 'SAVE' is on.
Bit 2: 'MOVE' is on.	Bit 3: 'ASCI' is on.
Bit 4: 'DATA' is on.	Bit 5: 'CODE' is on.
Bit 6: 'WRITE' is on.	Bit 7: Data is in decimal form.

Note that Bits 0, 1, and 2 are never set when Bit 6 is set, and not more than one of Bits 0 through 5 may be set at any time.

14C4: referred to as the FLAG2 byte, it contains 2 1-bit flags. When Bit 0 is set, an address character has been typed and not displayed with 'RETURN'. It is used primarily to tell the Assembler when it can and cannot clear the SLC.

When Bit 1 is set, data has been MOVEd or EDITed since the last Control M/E was typed. It is used to tell the Assembler whether or not 'MOVE' or 'EDIT' has yet been successfully initialized with an address from the operator and executed.

NOTES ON 8K/4K VERSION DIFFERENCES

The primary difference between versions is that all references to addresses from 1300 to 1FFF in the 8K Assembler are changed to the corresponding addresses from 0300 to 0FFF in the 4K Assembler. Furthermore, the following occurrences of the data 13(hex) in the 8K version are changed to 03(hex) in the 4K version:

1679	1D9B	1DC4	1E54	1E5B	1E82	1E96	1F3F	1F84
			1FAC		1FCC			

No other differences in code exist.