

Appendix 3

The complete listings of the Tape Monitor and Printer Monitor system programs.

The following pages contain a complete listing of the TM and PM system programs. Each book page corresponds to a single assembler page consisting of 56 lines.

The listing itself consists of the following:

- The main Tape Monitor routine
- The TM subroutines
- The DUMP/DUMPT subroutine
- Subroutines used by DUMP/DUMPT
- The RDTAPE subroutines
- Subroutines used by RDTAPE
- The main Printer Monitor routine
- The PM subroutines
- The look-up table MESS
- The initialisation routine STEP
- The IBRES subroutine
- A survey of all the labels and their addresses

Aanhangsel 1

De listing van het systeemprogramma PME

De nu volgende pagina's vormen een combinatie van machinetaal en Engels, die bekend staat als de listing.

Op pagina 189/190 vindt men de instructies van de routine BINAR en PMBINA, die nodig zijn om (tijdelijk) op binair rekenen terug te schakelen. Voor details: zie Aanhangsel 4 in dit boek.

Enige belangrijke adressen:

EDITC	\$1500
BRK	\$1533
EDITW	\$153D
SEMIW	\$1667
SEACND	\$17C5
BINAR	\$17F6
PMBINA	\$17FA

0001: 14D8
 0002:
 0003:
 0004:
 0005:
 0006:
 0007:
 0008:
 0009:
 0010:
 0011:
 0012:
 0013:
 0014:
 0015:
 0016:
 0017:
 0018: 14D8
 0019: 14D8
 0020: 14D8
 0021: 14D8
 0022: 14D8
 0023: 14D8
 0024: 14D8
 0025: 14D8
 0026: 14D8
 0027: 14D8
 0028: 14D8
 0029: 14D8
 0030: 14D8
 0031: 14D8
 0032: 14D8
 0033: 14D8
 0034: 14D8
 0035: 14D8
 0036:
 0037:
 0038:
 0039:
 0040:
 0041: 14D8
 0042: 14D8
 0043: 14D8
 0044: 14D8
 0045: 14D8
 0046: 14D8
 0047: 14D8
 0048: 14D8
 0049: 14D8
 0050:
 0051:
 0052:
 0053:
 0054:
 0055:
 0056:
 0057: 14D8
 0058: 14D8
 0059: 14D8
 0060: 14D8
 0061: 14D8
 0062: 14D8
 0063: 14D8
 0064: 14D8
 0065: 14D8
 0066: 14D8
 0067:
 0068:

ORG ~~\$14D8~~ 14FD !

SOURCE LISTING OF THE PM EDITOR

WRITTEN BY G. NACHBAR

DATE : 25 JUNE 1981

THE PM EDITOR USES HEXADECIMAL LABELS

 POINTERS AND TEMPS IN PAGE ZERO

BEGADL *	\$00E2
BEGADH *	\$00E3
ENDADL *	\$00E4
ENDADH *	\$00E5
CURADL *	\$00E6
CURADH *	\$00E7
CENDL *	\$00E8
CENDH *	\$00E9
TABLEL *	\$00EC
TABLEH *	\$00ED
LABELS *	\$00EE
BYTES *	\$00F6
COUNT *	\$00F7
INH *	\$00F9
POINTL *	\$00FA
POINTH *	\$00FB
TEMPX *	\$00FD
NIBBLE *	\$00FE

 PME'S POINTERS AND TEMPS IN PAGE 1A

PARAL *	\$1A63
PARAH *	\$1A64
PARBL *	\$1A65
PARBH *	\$1A66
NMIL *	\$1A7A
NMIH *	\$1A7B
BRKTL *	\$1A7C
BRKTH *	\$1A7D
PBDD *	\$1A83

 ADDRESSES IN THE STANDARD EPROM

SAVE *	\$1C00
BEGIN *	\$1ED3
OPLN *	\$1E5C
LENACC *	\$1E60
ADCEND *	\$1EDC
RECEND *	\$1EEA
UP *	\$1E83
FILLWS *	\$1E47
NEXT *	\$1EF8
ASSEMB *	\$1F51

```

0069: *****
0070: SUBROUTINES IN PM
0071: *****
0072:
0073:
0074: 14D8      INPAR *      $1387
0075: 14D8      RECCHA *     $12AE
0076: 14D8      PRCHA *     $1334
0077: 14D8      PRBYT *     $128F
0078: 14D8      PRSP *      $11F3
0079: 14D8      ASHETT *     $141E
0080: 14D8      CRLF *      $11E8
0081: 14D8      RESIN *     $1268
0082: 14D8      RESTTY *    $14BC
0083:
0084: 14D8      STEP *      $14CF
0085:
0086:
0087:
0088:          IOCCORR: ADAPTS THE PBDD TO THE VERSION D SITUATION,
0089:          PRIOR TO ASSEMBLING
0090:
0091: 14D8 A9 1E      IOCCORR LDAIM $1E
0092: 14DA 8D 83 1A      STA  PBDD  PBO IS INPUT
0093: 14DD 4C 51 1F      JMP   ASSEMB ASSEMBLE THE PROGRAM
0094:
0095:          EDITC: COLD START ENTRY OF PME
0096:
0097: 14E0 20 68 12      EDITC JSR  RESIN  RESET INPUT BUFFERS
0098: 14E3 A0 00          LDYIM $00
0099: 14E5 20 1E 17      JSR  MESSA   "BEGAD,ENDAD:"
0100: 14E8 20 87 13      JSR  INPAR   GET PARAMETERS
0101: 14EB 30 F3         BMI  EDITC  TRY IT AGAIN, IF NOT DONE PROPERLY
0102: 14ED AD 63 1A      LDA  PARAL   LOAD BEGADL
0103: 14F0 85 E2          STAZ BEGADL
0104: 14F2 18            CLC
0105: 14F3 69 01          ADCIM $01
0106: 14F5 85 E8          STAZ CENDL  CENDL = BEGADL+1
0107: 14F7 AD 64 1A      LDA  PARAH
0108: 14FA 85 E3          STAZ BEGADH  LOAD BEGADH
0109: 14FC 69 00          ADCIM $00
0110: 14FE 85 E9          STAZ CENDH  CENDH = BEGADH+AARRY
0111: 1500 AD 65 1A      LDA  PARBL
0112: 1503 85 E4          STAZ ENDADL  LOAD ENDAD
0113: 1505 AD 66 1A      LDA  PARBH
0114: 1508 85 E5          STAZ ENDADH  LOAD ENDADH
0115: 150A 20 D3 1E      JSR  BEGIN   PRINT POINTER CURAD
0116: 150D A9 77          LDAIM $77   EQUALS BEGAD
0117: 150F A0 00          LDYIM $00
0118: 1511 91 E6          STAIY CURADL EOF 77 ON FIRST ADDRESS BEGAD
0119:
0120:          BRK: WARM START ENTRY OF PME, INCLUDING THE SPECIFI-
0121:          CATION OF THE BRK JUMP VECTOR
0122:
0123: 1513 A9 3D          BRK   LDAIM $3D
0124: 1515 8D 7C 1A      STA  BRKTL
0125: 1518 A9 15          LDAIM $15
0126: 151A 8D 7D 1A      STA  BRKTH
0127:
0128:          EDITW: MAY BE USED AS A WARM START ENTRY OF PME
0129:          PROVIDED THE BRK JUMP VECTOR HAS BEEN SPECIFIED
0130:          BEFOREHAND.
0131: 151D A0 20          EDITW LDYIM $20
0132: 151F 20 1E 17      JSR  MESSA   "PM EDITOR"
0133: 1522 A0 FF          LDYIM $FF
0134: 1524 9A            TXS          RESET STACK POINTER
0135:
0136:          THE CENTRAL LABEL "WARM" OF PME STARTS STARTS
0137:          WITH THE K-KEY ROUTINE

```



```

138:          ("DELETE")
139:
140: 1525 20 EB 16 WARM JSR PRINS PRINT CURRENT INSTRUCTION
141: 1528 20 AE 12 JSR RECCHA WAIT FOR A DEPRESSED KEY
142: 152B C9 4B CMPIM 'K IS IT THE K-KEY?
143: 152D D0 09 BNE LIST CONTINUE IF NOT
144: 152F 20 83 1E JSR UP ADAPT WORKSPACE
145: 1532 20 EA 1E JSR RECEND ADAPT CURRENT END ADDRESS POINTER
146: 1535 4C 25 15 JMP WARM READY
147: LABEL LIST: L-KEY ROUTINE
148:
149: 1538 C9 4C LIST CMPIM 'L L-KEY DEPRESSED?
150: 153A D0 12 BNE SKIP IF NOT CONTINUE
151: 153C 20 D3 1E JSR BEGIN START AT BEGAD
152: 153F 20 EB 16 LST JSR PRINS PRINT CURRENT INSTRUCTION
153: 1542 20 F8 1E JSR NEXT ADAPT INSTRUCTION POINTER
154: 1545 30 F8 BMI LST
155: 1547 A0 1F DONE LDYIM $1F
156: 1549 20 1E 17 MESS JSR MESSA PRINT "DONE"
157: 154C F0 D7 BEQ WARM READY
158:
159: LABEL SKIP: SPACE BAR KEY ROUTINE
160: LABEL INSERT: I-KEY ROUTINE
161:
162: 154E C9 20 SKIP CMPIM SPACE BAR DEPRESSED?
163: 1550 D0 07 BNE INSERT CONTINUE IF NOT
164: 1552 20 F8 1E JSR NEXT ADAPT INSTRUCTION POINTER
165: 1555 30 CE BMI WARM READY
166: 1557 10 EE BPL DONE PRINT "DONE" IF NO FURTHER INSTR. AVAILABLE
167: 1559 C9 49 INSERT CMPIM 'I I-KEY DEPRESSED?
168: 155B D0 17 BNE SEARCH IF NOT CONTINUE
169: 155D 20 A9 16 JSR READIN READ NEW INSTRUCTION
170: 1560 30 0A BMI ILLKEY PRINT "ILLEGAL KEY" IF NEW ISTR. ISN'T VALID
171: 1562 20 D6 16 MEM JSR CHECK
172: 1565 90 09 BCC FULL PRINT "FULL" IF MEMORY IS FULL
173: 1567 20 47 1E JSR FILLWS LOAD NEW INSTR. IN MEMORY
174: 156A F0 B9 BEQ WARM READY
175: 156C A0 0E ILLKEY LDYIM $0E
176: 156E D0 D9 BNE MESS PRINT "ILLEGAL KEY"
177: 1570 A0 1A FULL LDYIM $1A
178: 1572 D0 D5 BNE MESS PRINT "FULL"
179:
180: LABEL SEARCH: S-KEY ROUTINE
181: ANY BYTE PATTERN (INSTRUCTION) MAY BE SEARCHED FOR.
182: IF THE INSTRUCTION SEARCHED FOR IS FOUND,
183: ONE MAY SEARCH FOR THE SAME INSTRUCTION ON A HIGHER
184: ADDRESS, BY PRESSING THE KEY "Y". THE SEARCHING PRO-
185: CESS ALWAYS HAS TO BE CONCLUDED BY THE MESSAGE "DONE".
186: THIS OCCURS EITHER BECAUSE ALL OR NO INSTRUCTIONS ARE
187: FOUND, OR BY PRESSING AN ARBITRARY SOFTWARE-KEY (I.E.
188: A KEY GENERATING AN ASCII CODE WHEN DEPRESSED),
189: WITH THE EXCEPTION OF THE 'Y'-KEY.
190: 1574 C9 53 SEARCH CMPIM 'S S-KEY DEPRESSED?
191: 1576 D0 40 BNE BACK IF NOT CONTINUE
192: 1578 20 A9 16 JSR READIN READ INSTR. TO BE SEARCHED FOR
193: 157B 30 EF BMI ILLKEY "ILLEGAL KEY" IF NOT PROPERLY DONE
194: 157D 20 D3 1E JSR BEGIN START AT BEGAD
195: 1580 A5 FB SCAN LDAZ POINTH GET OPCODE OF SEARCH INSTR.
196: 1582 20 60 1E JSR LENACC GET LENGTH OF SEARCH INSTR.
197: 1585 A0 00 LDYIM $00
198: 1587 B1 E6 LDAIY CURADL GET OPCODE OF THE CURRENT INSTR.
199: 1589 C5 FB CMPZ POINTH COMPARE IT AGAINST OPCODE OF THE SEARCH INST.
200: 158B D0 20 BNE AGAIN IF NO MATCH THEN NEXT INSTR.
201: 158D C6 F6 DECZ BYTES
202: 158F F0 12 BEQ FOUND CONTINUE, IF INSTR. LENGTH IS 2 OR 3
203: 1591 C8 INY
204: 1592 B1 E6 LDAIY CURADL GET NEXT BYTE IN MEMORY
205: 1594 C5 FA CMPZ POINTL COMP. IT AGAINST 1ST OPERAND BYTE
206: 1596 D0 15 BNE AGAIN IF NO MATCH NEXT INSTR.

```

```

0207: 1598 C6 F6      DECZ  BYTES
0208: 159A F0 07      BEQ   FOUND  CONTINUE IF LENGTH OF CURR. INSTR. IS 3
0209: 159C C8          INY
0210: 159D B1 E6      LDAlY  CURADL  GET NEXT BYTE IN MEMORY
0211: 159F C5 F9      CMPZ   INH     COMP. IT AGAINST 2ND OPERAND BYTE
0212: 15A1 D0 0A      BNE   AGAIN  IF NO MATCH NEXT INSTR.
0213: 15A3 20 EB 16    FOUND  JSR   PRINS  PRINT OUT THE SPECIFIED INSTRUCTION
0214: 15A6 20 AE 12    JSR   RECCHA WAIT FOR A DEPRESSED KEY
0215: 15A9 C9 59      CMPIM  'Y     Y-KEY DEPRESSED?
0216: 15AB D0 08      BNE   DNE     IF NOT PRINT "DONE"
0217: 15AD 20 5C 1E    AGAIN  JSR   OPLEN  GET LENGTH OF THE CURR. INSTR.
0218: 15B0 20 F8 1E    JSR   NEXT   ADAPT INSTR. POINTER
0219: 15B3 30 CB      BMI   SCAN   CONTINUE SEARCH IF STILL INSTR. AVAILAB
0220: 15B5 4C 47 15    DNE   JMP   DONE   ELSE PRINT "DONE"
0221:
0222:                LABEL BACK: Z-KEY ROUTINE
0223:
0224: 15B8 C9 5A      BACK   CMPIM  'Z     Z-KEY DEPRESSED?
0225: 15BA D0 30      BNE   TOF     IF NOT CONTINUE
0226: 15BC A5 E2      LDAH  BEGADL
0227: 15BE 85 EC      STAZ  TABLEL
0228: 15C0 A5 E3      LDAH  BEGADH
0229: 15C2 85 ED      STAZ  TABLEH  TABLE:=BEGAD
0230: 15C4 C5 E7      CMPZ   CURADH  TABLE:=CURADH?
0231: 15C6 D0 08      BNE   BCKA   IF NOT INCREASE TABLE
0232: 15C8 A5 E2      LDAH  BEGADL
0233: 15CA C5 E6      CMPZ   CURADL  TABLE:=CURADL?
0234: 15CC D0 02      BNE   BCKA   IF NOT INCREASE TABLE
0235: 15CE F0 19      BEQ   BCKB   ELSE PRINT FIRST INSTRUCTION
0236: 15D0 A0 00      BCKA   LDYIM  $00
0237: 15D2 B1 EC      LDAlY  TABLEL  GET LENGTH OF INSTR. AT
0238: 15D4 20 60 1E    JSR   LENACC  WHICH POINT IS POINTED
0239: 15D7 20 80 16    JSR   INCTAB  INCREASE POINT BY BYTES
0240: 15DA A5 E6      LDAH  CURADL
0241: 15DC C5 EC      CMPZ   TABLEL  TABLE:=CURADL?
0242: 15DE D0 F0      BNE   BCKA   IF NOT INCREASE TABLE
0243: 15E0 A5 E7      LDAH  CURADH
0244: 15E2 C5 ED      CMPZ   TABLEH  TABLE:=CURADH
0245: 15E4 D0 EA      BNE   BCKA   IF NOT INCREASE TABLE
0246: 15E6 20 72 16    JSR   DECURA  DECREASE INSTRUCTION POINTER BY BYTES
0247: 15E9 4C 25 15    BCKB   JMP   WARM
0248:
0249:                LABEL TOF: T-KEY ROUTINE
0250:                LABEL SXTEEN: P-KEY ROUTINE
0251:
0252: 15EC C9 54      TOF     CMPIM  'T     T-KEY DEPRESSED?
0253: 15EE D0 06      BNE   SXTEEN  IF NOT CONTINUE
0254: 15F0 20 D3 1E    JSR   BEGIN  INSTR. POINTER = BEGAD
0255: 15F3 4C 25 15    TOFEND  JMP   WARM   READY
0256: 15F6 C9 50      SXTEEN  CMPIM  'P     P-KEY DEPRESSED?
0257: 15F8 D0 1C      BNE   ASMBLR  IF NOT CONTINUE
0258: 15FA A9 0F      LDAH  $0F     15 INSTR. TO BE PRINTED
0259: 15FC 85 F7      STAZ  COUNT
0260: 15FE 20 5C 1E    LINES  JSR   OPLEN  GET LENGTH OF CURRENT INSTR.
0261: 1601 20 F8 1E    JSR   NEXT   ADAPT INSTR. POINTER
0262: 1604 C6 F7      DECZ  COUNT
0263: 1606 F0 EB      BEQ   TOFEND  READY IF 15 INSTR. HAVE BEEN PRINTED
0264: 1608 A0 00      LDYIM  $00
0265: 160A B1 E6      LDAlY  CURADL
0266: 160C C9 77      CMPIM  $77
0267: 160E F0 E3      BEQ   TOFEND  READY IF OPCODE IS 77
0268: 1610 20 EB 16    JSR   PRINS  PRINT INSTRUCTION
0269: 1613 4C FE 15    JMP   LINES  NEXT INSTRUCTION
0270:                LABEL ASMBLR: X-KEY ROUTINE
0271:                LABEL ASSEND: RESTORE I/O AFTER ASSEMBLING
0272:
0273: 1616 C9 58      ASMBLR  CMPIM  'X     X-KEY DEPRESSED?
0274: 1618 D0 13      BNE   INPUT  IF NOT CONTINUE
0275: 161A A9 27      LDAH  ASSEND  SPECIFY NMI JUMP VECTOR

```

```

0276: 161C 8D 7A 1A      STA  NMIL
0277: 161F A9 16          LDAIM ASSEND /256
0278: 1621 8D 7B 1A      STA  NMIH
0279: 1624 4C D8 14      JMP  IOCORR PREPARE I/O PRIOR TO ASSEMBLING
0280: 1627 20 BC 14      ASSEND JSR  RESTTY RESTORE I/O: PM SITUATION
0281: 162A 4C 68 17      JMP  LABLST LIST THE HEXADECIMAL LABELS
0282:
0283: LABEL INPUT: I-KEY ROUTINE
0284: NOTE: NO NON-NUMERICAL KEY HAS TO BE DEPRESSED
0285: PME AUTOMATICALLY ASSUMES THE INPUT KEY FUNCTION
0286: HAS BEEN OPTED FOR AS SOON AS THE DATA BELONGING
0287: TO THE NEW INSTR. HAS BEEN SPECIFIED
0288:
0289: 162D 20 91 16      INPUT JSR  BYT  GET THE 2ND NIBBLE OF THE 1ST BYTE
0290: 1630 30 12          BMI  WRONG
0291: 1632 20 AE 16      JSR  READ  GET THE OTHER BYTE(S)
0292: 1635 30 0D          BMI  WRONG
0293: 1637 20 5C 1E      JSR  OPLEN  GET LENGTH OF THE CURRENT INSTR.
0294: 163A 20 F8 1E      JSR  NEXT  ADAPT INSTR. POINTER
0295: 163D A5 FD          LDAZ  TEMPX  GET LENGTH OF NEW INSTR.
0296: 163F 85 F6          STAZ  BYTES  AND STORE IT IN BYTES
0297: 1641 4C 62 15      JMP  MEM  LOAD MEMORY WITH NEW INSTRUCTION
0298: 1644 4C 6C 15      WRONG JMP  ILLKEY PRINT "ILLEGAL KEY"
0299:
0300: END OF THE PME MAIN ROUTINE
0301:
0302:
0303:
0304: SEMI WARM START ("LUKE WARM") ENTRY OF PME
0305: FOLLOWING THE SPECIFICATION BY THE USER
0306: OF BEGAD AND ENDAD THE INSTRUCTION POINTER
0307: CURAD IS POINTED AT BEGAD AND THE CURRENT
0308: END ADDRESS POINTER CEND IS POINTED AT ENDAD.
0309: AFTER THIS PME IS ENTERED BY THE WARM START ENTRY
0310:
0311:
0312: 1647 20 68 12      SEMIW JSR  RESIN  RESET INPUT BUFFERS
0313: 164A A0 00          LDYIM $00
0314: 164C 20 1E 17      JSR  MESSA  "BEGAD,ENDAD:"
0315: 164F 20 87 13      JSR  INPAR  GET BEGAD AND ENDAD
0316: 1652 30 F3          BMI  SEMIW  TRY IT AGAIN IF NOT DONE PROPERLY
0317: 1654 AD 63 1A      LDA  PARAL
0318: 1657 85 E2          STAZ  BEGADL
0319: 1659 AD 64 1A      LDA  PARAH
0320: 165C 85 E3          STAZ  BEGADH
0321: 165E AD 65 1A      LDA  PARBL
0322: 1661 85 E4          STAZ  ENDADL
0323: 1663 85 E8          STAZ  CENDL
0324: 1665 AD 66 1A      LDA  PARBH
0325: 1668 85 E5          STAZ  ENDADH
0326: 166A 85 E9          STAZ  CENDH  CEND = ENDAD
0327: 166C 20 D3 1E      JSR  BEGIN  CURAD = BEGAD
0328: 166F 4C 13 15      JMP  BRK  JUMP TO WARM START
0329:
0330:
0331:
0332: *****
0333: SUBROUTINES OF PME
0334: *****
0335:
0336:
0337: DECURA
0338: THE INSTRUCTION POINTER IS DECREASED BY THE
0339: NUMBER OF BYTES - BEING THE LENGTH OF THE
0340: INSTRUCTION - WHICH PRECEEDS THE CURRENT
0341: INSTRUCTION IN MEMORY.
0342:
0343: 1672 38          DECURA SEC
0344: 1673 A5 E6          LDAZ  CURADL

```

```

0345: 1675 E5 F6      SBCZ  BYTES
0346: 1677 85 E6      STAZ  CURADL CURADL:=CURADL-BYTES
0347: 1679 A5 E7      LDAZ  CURADH
0348: 167B E9 00      SBCIM $00
0349: 167D 85 E7      STAZ  CURADH CURADH:=CURADH-BORROW
0350: 167F 60          RTS
0351:
0352:
0353:      SUBROUTINR INCTAB
0354:      INCREASES THE POINTER TABLE BY THE AMOUNT DETERMINED
0355:      BY THE CONTENTS OF BYTES (INSTRUCTION LENGTH).
0356:
0357: 1680 18          INCTAB CLC
0358: 1681 A5 EC      LDAZ  TABLE
0359: 1683 65 F6      ADCZ  BYTES
0360: 1685 85 EC      STAZ  TABLE TABLE:=TABLE + BYTES
0361: 1687 A5 ED      LDAZ  TABLEH
0362: 1689 69 00      ADCIM $00
0363: 168B 85 ED      STAZ  TABLEH
0364: 168D 60          RTS
0365:
0366:      THE SUBROUTINE BYTIN LOADS THE ACCU WITH
0367:      WITH DATA WHICH BELONGS TO TWO NUMERICAL KEYS
0368:      DEPRESSED. RETURNING FROM SUBROUTINE N=1 AND
0369:      Z=0 IF A NON-NUMERICAL KEY WAS DEPRESSED. TWO
0370:      SUCCESSIVELY DEPRESSED NUMERICAL KEYS WILL
0371:      RESULT INTO N=0 AND Z=1
0372:
0373: 168E 20 AE 12    BYTIN JSR  RECCHA WAIT FOR A DEPRESSED KEY
0374: 1691 20 1E 14    BYT  JSR  ASHETT CONVERT IT TO A DATA NIBBLE
0375: 1694 30 12      BMI  RETURN ERROR EXIT IF KEY <> 0...F
0376: 1696 0A          ASLA  ENTERED DATA IS NEW HIGHER DATA NIBB
0377: 1697 0A          ASLA
0378: 1698 0A          ASLA
0379: 1699 0A          ASLA
0380: 169A 85 FE      STAZ  NIBBLE SAVE HIGHER DATA NIBBLE
0381: 169C 20 AE 12    JSR  RECCHA WAIT FOR A DEPRESSED KEY
0382: 169F 20 1E 14    JSR  ASHETT CONVERT IT TO A DATA NIBBLE
0383: 16A2 30 04      BMI  RETURN ERROR EXIT IF KEY <> 0...F
0384: 16A4 05 FE      ORAZ  NIBBLE INSERT NEW LOWER DATA NIBBLE
0385: 16A6 A2 00      LDXIM $00 RESET N-FLAG, SET Z-FLAG
0386: 16A8 60          RETURN RTS
0387:
0388:
0389:      THE SUBROUTINE READIN LOADS EITHER ONE,
0390:      TWO OR THREE DATA BUFFERS DEPENDING
0391:      OF THE INSTRUCTION LENGTH SPECIFIED BY
0392:      THE OPCODE.
0393:      NORMAL EXIT: Z=1, N=0
0394:      ERROR EXIT: Z=0, N=1
0395:
0396: 16A9 20 8E 16    READIN JSR  BYTIN WAIT FOR OPCODE
0397: 16AC 30 27      BMI  RDB  ERROR IF KEY <> 0...F
0398: 16AE 85 FB      READ  STAZ  POINTH STORE OPCODE IN POINTH
0399: 16B0 20 60 1E    JSR  LENACC GET INSTRUCTION LENGTH
0400: 16B3 84 F7      STYZ  COUNT AND COPY IT INTO COUNT
0401: 16B5 84 FD      STYZ  TEMPX AS WELL AS IN TEMPX
0402: 16B7 C6 F7      DECZ  COUNT
0403: 16B9 F0 18      BEQ  RDA  RETURN IF ONE BYTE INSTR.
0404: 16BB 20 F3 11    JSR  PRSP  PRINT A SPACE
0405: 16BE 20 8E 16    JSR  BYTIN WAIT FOR (1ST) OPERAND
0406: 16C1 30 12      BMI  RDB  ERROR IF KEY <> 0...F
0407: 16C3 85 FA      STAZ  POINTL STORE (1ST) OPERND IN POINTL
0408: 16C5 C6 F7      DECZ  COUNT
0409: 16C7 F0 0A      BEQ  RDA  RETURN IF TWO BYTE INSTR.
0410: 16C9 20 F3 11    JSR  PRSP  PRINT A SPACE
0411: 16CC 20 8E 16    JSR  BYTIN WAIT FOR 2ND OPERAND
0412: 16CF 30 04      BMI  RDB  ERROR IF KEY <> 0...F
0413: 16D1 85 F9      STAZ  INH  STORE 2ND OPERAND IN INH

```

```

0414: 16D3 A2 00      RDA   LDXIM $00      Z=1, N=0
0415: 16D5 60          RDB   RTS
0416:
0417:
0418:
0419:
0420:
0421:
0422:
0423:
0424:
0425: 16D6 20 DC 1E   CHECK JSR   ADCEND ADJUST CURRENT ENDADDRESS POINTER
0426: 16D9 38          SEC
0427: 16DA A5 E4          LDZ   ENDADL
0428: 16DC E9 02          SBCIM $02
0429: 16DE E5 E8          SBCZ  CENDL
0430: 16E0 A5 E5          LDZ   ENDADH CARRY DETERMINED BY ENDAD-2-CEND
0431: 16E2 E5 E9          SBCZ  CENDH
0432: 16E4 90 04          BCC   CHKEND EXIT IF NO ROOM ANYMORE
0433: 16E6 20 EA 1E      JSR   RECDND READJUST CURRENT END ADDRESS POINTER
0434: 16E9 38          SEC
0435: 16EA 60          CHKEND RTS
0436:
0437:
0438:
0439:
0440:
0441:
0442:
0443:
0444:
0445:
0446: 16EB 20 E8 11   PRINS JSR   CRLF   PRINT A NEW LINE
0447: 16EE 20 5C 1E      JSR   OPLEN  GET LENGTH OF INSTRUCTION
0448: 16F1 A6 F6          LDZX  BYTES   TO BE PRINTED
0449: 16F3 A5 E7          LDZ   CURADH
0450: 16F5 20 8F 12      JSR   PRBYT  PRINT HIGHER ADDRESS BYTE
0451: 16F8 A5 E6          LDZ   CURADL
0452: 16FA 20 8F 12      JSR   PRBYT  PRINT LOWER ADDRESS BYTE
0453: 16FD A9 0F          LDAIM $0F
0454: 16FF 85 EE          STAZ  LABELS MAXIMUM NUMBER OF SPACES
0455: 1701 A0 00          LDYIM $00
0456: 1703 20 F3 11   PRT  JSR   PRSP   PRINT A SPACE
0457: 1706 B1 E6          LDYIM CURADL GET BYTE TO BE PRINTED
0458: 1708 20 8F 12      JSR   PRBYT  AND PRINT IT
0459: 170B 38          SEC
0460: 170C A5 EE          LDZ   LABELS DECREASE NUMBER OF SPACES
0461: 170E E9 03          SBCIM $03   TO BE PRINTED
0462: 1710 85 EE          STAZ  LABELS BY 3
0463: 1712 C8          INY
0464: 1713 CA          DEX
0465: 1714 D0 ED          BNE   PRT   IF THERE IS ANYONE
0466: 1716 20 F3 11   SP  JSR   PRSP   PRINT A NUMBER OF SPACES
0467: 1719 C6 EE          DECZ  LABELS
0468: 171B D0 F9          BNE   SP
0469: 171D 60          RTS      RETURN
0470:
0471:
0472:
0473:
0474:
0475:
0476:
0477: 171E 20 E8 11   MESSA JSR   CRLF   PRINT A NEW LINE
0478: 1721 B9 30 17   MA  LDAY  TXT   GET CHARACTER TO BE PRINTED
0479: 1724 C9 03          CMPIM $03   EOT CHARACTER?
0480: 1726 F0 07          BEQ   TXTEND IF YES RETURN
0481: 1728 20 34 13      JSR   PRCHA  PRINT A CHARACTER
0482: 172B C8          INY

```

```

0483: 172C 4C 21 17      JMP    MA
0484: 172F 60      TXTEND RTS      READY
0485:
0486:
0487:      LOOKUP TABLE TXT
0488:
0489: 1730 42      TXT    =      'B      Y=00
0490: 1731 45      =      'E
0491: 1732 47      =      'G
0492: 1733 41      =      'A
0493: 1734 44      =      'D
0494: 1735 2C      =      '
0495: 1736 45      =      'E
0496: 1737 4E      =      'N
0497: 1738 44      =      'D
0498: 1739 41      =      'A
0499: 173A 44      =      'D
0500: 173B 3A      =      ':'
0501: 173C 20      =      '
0502: 173D 03      =      '$03
0503: 173E 49      =      'I      Y=0E
0504: 173F 4C      =      'L
0505: 1740 4C      =      'L
0506: 1741 45      =      'E
0507: 1742 47      =      'G
0508: 1743 41      =      'A
0509: 1744 4C      =      'L
0510: 1745 20      =      '
0511: 1746 4B      =      'K
0512: 1747 45      =      'E
0513: 1748 59      =      'Y
0514: 1749 03      =      '$03
0515: 174A 46      =      'F      Y=1A
0516: 174B 55      =      'U
0517: 174C 4C      =      'L
0518: 174D 4C      =      'L
0519: 174E 03      =      '$03
0520: 174F 44      =      'D      Y=1F
0521: 1750 4F      =      'O
0522: 1751 4E      =      'N
0523: 1752 45      =      'E
0524: 1753 03      =      '$03
0525: 1754 50      =      'P      Y=24
0526: 1755 4D      =      'M
0527: 1756 20      =      '
0528: 1757 45      =      'E
0529: 1758 44      =      'D
0530: 1759 49      =      'I
0531: 175A 54      =      'T
0532: 175B 4F      =      'O
0533: 175C 52      =      'R
0534: 175D 03      =      '$03
0535: 175E 4C      =      'L      Y=2E
0536: 175F 41      =      'A
0537: 1760 42      =      'B
0538: 1761 20      =      '
0539: 1762 24      =      '$
0540: 1763 03      =      '$03
0541: 1764 3A      =      ':'      Y=34
0542: 1765 20      =      '
0543: 1766 24      =      '$
0544: 1767 03      =      '$03
0545:
0546:
0547:
0548:
0549:
0550:
0551: 1768 A0 FF      LABLST LDYIM $FF

```

THE INSTRUCTIONS FOLLOWING THE LABEL LABLST
DEAL WITH THE PRINTING OF ALL HEXADECIMAL LABELS
AFTER THE PROGRAM HAS BEEN ASSEMBLED


```

0552: 176A 20 E8 11 LBLSTA JSR CRLF PRINT ON A NEW LINE
0553: 176D A2 04 LDXIM $04 4 LABELS ON EACH LINE
0554: 176F 84 FD LBLSTB STYZ TEMPX SAVE Y
0555: 1771 A0 2E LDYIM $2E "LAB $"
0556: 1773 20 21 17 JSR MA
0557: 1776 A4 FD LDYZ TEMPX RESTORE Y
0558: 1778 B1 EC LDAIY TABLEL GET LABEL NUMBER
0559: 177A 20 8F 12 JSR PRBYT PRINT LABEL NUMBER
0560: 177D 88 DEY
0561: 177E 84 FD STYZ TEMPX SAVE Y
0562: 1780 A0 34 LDYIM $34 ": $"
0563: 1782 20 21 17 JSR MA
0564: 1785 A4 FD LDYZ TEMPX RESTORE Y
0565: 1787 B1 EC LDAIY TABLEL GET HIGHER ADDRESS BYTE
0566: 1789 20 8F 12 JSR PRBYT AND PRINT IT
0567: 178C 88 DEY
0568: 178D B1 EC LDAIY TABLEL GET LOWER ADDRESS BYTE
0569: 178F 20 8F 12 JSR PRBYT AND PRINT IT
0570: 1792 20 F3 11 JSR PRSP PRINT A SPACE
0571: 1795 88 DEY SETUP FOR NEXT LABEL
0572: 1796 C4 EE CPYZ LABELS ALL LABELS PRINTED
0573: 1798 F0 05 BEQ LBLSTC IF NOT SET UP FOR NEXT LABEL
0574: 179A CA DEX
0575: 179B D0 D2 BNE LBLSTB ON THE SAME LINE
0576: 179D F0 CB BEQ LBLSTA OR ON A NEW LINE
0577: 179F 20 D3 1E LBLSTC JSR BEGIN INSTRUCTION POINTER = BEGAD
0578: 17A2 4C 1D 15 JMP EDITW WARM START ENTRY OF PME WITHOUT
0579: BRK JUMP VECTOR SPECIFICATION
0580:
0581:
0582:
0583: THE WARM CEND ENTRY (LABEL SEACND) OF PME
0584: RESTORES THE POSITION OF THE CURRENT END
0585: ADDRESS POINTER CEND, AFTER SPECIFICATION,
0586: BY THE USER, OF BEGAD AND ENDAD, AND AFTER FINDING A
0587: PSEUDO OPCODE 77, I.E. AN EOF CHARACTER. THIS IS
0588: FOLLOWED BY A JUMP TO THE WARM START ENTRY OF PME.
0589:
0590:
0591: 17A5 20 68 12 SEACND JSR RESIN RESET INPUT BUFFERS
0592: 17A8 A0 00 LDYIM $00
0593: 17AA 20 1E 17 JSR MESSA "BEGAD,ENDAD"
0594: 17AD 20 87 13 JSR INPAR GET BEGAD AND ENDAD
0595: 17B0 30 F3 BMI SEACND TRY IT AGAIN IF NOT PROPERLY DONE
0596: 17B2 20 D3 1E JSR BEGIN INSTR. POINTER=BEGAD
0597: 17B5 A0 00 SCNDA LDYIM $00
0598: 17B7 B1 E6 LDAIY CURADL GET CURRENT OPCODE
0599: 17B9 C9 77 CMPIM $77 IS IT OPCODE 77?
0600: 17BB F0 09 BEQ SCNDB IF NOT CONTINUE
0601: 17BD 20 60 1E JSR LENACC GET CURRENT INSTR. LENGTH
0602: 17C0 20 F8 1E JSR NEXT ADJUST INSTR. POINTER
0603: 17C3 4C B5 17 JMP SCNDA AND CHECK AGAIN FOR OPCODE 77
0604: 17C6 18 SCNDB CLC
0605: 17C7 A5 E6 LDAZ CURADL
0606: 17C9 69 01 ADCIM $01
0607: 17CB 85 E8 STAZ CENDL CENDL:=CURADL+1
0608: 17CD A5 E7 LDAZ CURADH
0609: 17CF 69 00 ADCIM $00
0610: 17D1 85 E9 STAZ CENDH CENDH:=CURADH+CARRY
0611: 17D3 4C 13 15 JMP BRK WARM START ENTRY OF PME
0612:
0613:
0614: THE INSTRUCTIONS FOLLOWING THE LABEL BINAR
0615: AND THE INSTRUCTIONS FOLLOWING THE LABEL PMBINA ARE
0616: AN EXTENSION OF THE STANDARD MONITOR AND OF PM
0617: RESPECTIVELY. IN CASE OF SINGLE STEPPING THROUGH
0618: AN USER PROGRAM WITH DECIMAL ARITHMETIC, THERE
0619: WILL BE A TEMPORARY SWITCH BACK ON BINARY ARITHMETIC
0620: PROVIDED THE NMI JUMP VECTOR HAS BEEN SPECIFIED

```

```

0621:                                ON 17F6 OR 17FA RESPECTIVELY.
0622:
0623:
0624: 17D6 D8          BINAR CLD          BINARY ARITHMETIC
0625: 17D7 4C 00 1C     JMP      SAVE      SAVE INPUT PROPER
0626: 17DA D8          PMBINA CLD         BINARY ARITHMETIC
0627: 17DB 4C CF 14     JMP      STEP      SAVE INPUT PROPER
0628:
0629:
0630:                                ***** END OF THE PME PROGRAM *****
0631:
0632:
0633:                                VICINUS ELABORABAT PROGRAMMAM XXV VI MCMLXXXI
0634:                                VIR NOCTIS IMPOSIT PROGRAMMAM IN MACHINAM
0635:
ID=

```

-T

```

SYMBOL TABLE 3000 3276
ADCEND 1EDC      AGAIN 15AD      ASHETT 141E      ASMBLR 1616
ASSEMB 1F51      ASSEND 1627     BACK   15B8      BCKA   15D0
BCKB  15E9      BEGADH 00E3     BEGADL 00E2     BEGIN  1ED3
BINAR  17D6     BRKTH  1A7D     BRKTL  1A7C     BRK    1513
BYTES  00F6     BYTIN  168E     BYT     1691     CENDH  00E9
CENDL  00E8     CHECK  16D6     CHKEND 16EA     COUNT  00F7
CRLF   11E8     CURADH 00E7     CURADL 00E6     DECURA 1672
DNE    15B5     DONE   1547     EDITC  14E0     EDITW  151D
ENDADH 00E5     ENDADL 00E4     FILLWS 1E47     FOUND  15A3
FULL   1570     ILLKEY 156C     INCTAB 1680     INH     00F9
INPAR  1387     INPUT  162D     INSERT 1559     IOCORR 14D8
LABELS 00EE     LABLST 1768     LBLSTA 176A     LBLSTB 176F
LBLSTC 179F     LENACC 1E60     LINES  15FE     LIST   1538
LST     153F     MA      1721     MEM     1562     MESS   1549
MESSA  171E     NEXT   1EF8     NIBBLE 00FE     NMIH   1A7B
NMIL   1A7A     OPLEN  1E5C     PARAH  1A64     PARAL  1A63
PARBH  1A66     PARBL  1A65     PBDD    1A83     PMBINA 17DA
POINTH 00FB     POINTL 00FA     PRBYT   128F     PRCHA  1334
PRINS  16EB     PRSP    11F3     PRT     1703     RDA    16D3
RDB     16D5     READ   16AE     READIN 16A9     RECCHA 12AE
RECEAD 1EEA     RESIN  1268     RESTTY 14BC     RETURN 16A8
SAVE    1C00     SCAN   1580     SCNDA  17B5     SCNDB  17C6
SEACND 17A5     SEARCH 1574     SEMIW  1647     SKIP   154E
SP       1716     STEP   14CF     SXTEEN 15F6     TABLEH 00ED
TABLEL 00EC     TEMPX  00FD     TOFEND 15F3     TOF    15EC
TXTEND 172F     TXT     1730     UP      1E83     WARM   1525
WRONG   1644

```


**De listing van het systeemprogramma TM
en
de listing van het systeemprogramma PM**

Elke boekpagina bevat 69 assembler-regels. Na het algemene gedeelte (deklaraties) komen achtereenvolgens aan de orde:

- Het hoofdprogramma TM
- De subroutines van TM
- De "super-subroutine" DUMP/DUMPT
- De subroutines van DUMP/DUMPT
- De "super-subroutine" RDTAPE
- De subroutines van RDTAPE
- Het hoofdprogramma PM
- De subroutines van PM
- De opzoektabel MESS van PM
- De STEP-voorroutine van PM
- De subroutine IPBRES van PM
- En tot slot een "symbol table", met een overzicht van alle labels en niet-anonieme geheugenplaatsen.

```

0001: 0800          ORG    $0800  SOFTWARE OF JUNIOR COMPUTER 3,4
0002:
0003:
0004:          SOURCE LISTING OF THE TAPE/PRINTER MONITOR
0005:
0006:          WRITTEN BY A. NACHTMANN
0007:
0008:
0009:          DATE: 17 DEC. 1980
0010:
0011:
0012:
0013:          *****
0014:          *** DEFINITIONS OF DUMPT & RDTAPE ***
0015:          *****
0016:
0017:
0018:          DUMPT STORES A PROGRAM ON TAPE
0019:          1) STORE A PROGRAM ON TAPE
0020:          2) ID IS THE IDENTITY NUMBER
0021:          3) SA IS THE START ADDRESS
0022:          4) EA IS THE END ADDRESS + 1
0023:          5) ID = 00 & ID = FF SHOULD NOT BE USED
0024:
0025:          RDTAPE READS A PROGRAM FROM TAPE
0026:          1) READ A PROGRAM FROM TAPE WITH A CERTAIN ID
0027:          2) DISCARD DATA IN MEMORY BEGINNING AT SA
0028:          3) ID = 00 & ID = FF ARE SPECIAL IDS
0029:
0030:
0031:
0032:          THE DATA FORMAT ON TAPE IS ASCII:
0033:
0034:          255SYN<>*<>ID<>SAL<>SAH<>DATA<>/<>CHKL<>CHKH<>EOT<>EOT
0035:
0036:          I/O DEFINITION
0037:
0038:          PAD      *      $1A80  PORT A DATA REGISTER
0039:          PADD     *      $1A81  PORT A DATA DIRECTION
0040:          PBD      *      $1A82  PORT B DATA REGISTER
0041:          PBDD     *      $1A83  PORT B DATA DIRECTION
0042:
0043:
0044:          TIMER REGISTERS
0045:
0046:          CNTA     *      $1AF4  CLK1T, DISABLE TIMER IRQ
0047:          CNTC     *      $1AF6  CLK64T, DISABLE TIMER IRQ
0048:          RDFLAG   *      $1AD5  READ FLAG REGISTER, B7 IS TIMER FLAG
0049:          RDTDIS   *      $1AD4  READ CONTENTS OF TIMER CELL, IRQ DISABL
0050:
0051:
0052:          TEMPORARY DATA BUFFERS
0053:
0054:          SY       *      $1A69  SYN COUNTER
0055:          BYTE     *      $1A6A  BYTE FROM TAPE
0056:          CHAR     *      $1A6B  CHARACTER FROM TAPE
0057:          HIGHER   *      $1A6C  3600 HZ HALF PERIODE DELAY
0058:          LOWER    *      $1A6D  2400 HZ HALF PERIODE DELAY
0059:          CHKL     *      $1A6E  CHECK SUM LOW
0060:          CHKH     *      $1A6F  CHECK SUM HIGH
0061:          SAL      *      $1A70  START ADDRESS LOW
0062:          SAH      *      $1A71  START ADDRESS HIGH
0063:          EAL      *      $1A72  END ADDRESS LOW + 1
0064:          EAH      *      $1A73  END ADDRESS HIGH
0065:          SYNCNT   *      $1A74
0066:          BITS     *      $1A75  AMOUNT OF BITS
0067:          FIRST    *      $1A76  HALF PERIODE AMOUNT OF 3600 HZ
0068:          SECOND   *      $1A77  HALF PERIODE AMOUNT OF 2400 HZ
0069:          GANG     *      $1A78  TEMP OF PBD-BITS

```

```

0070: 0800      ID      *      $1A79  ID OF THE DUMPT PROGRAM
0071: 0800      NMI     *      $1A7A  NMI VECTOR
0072:
0073:      *****
0074:      *** DEFINITIONS OF THE TTY MONITOR ***
0075:      *****
0076:
0077:
0078:      I/O DEFINITION: SEE DUMPT & RDTAPE
0079:
0080:
0081:      CPU REGISTERS
0082:
0083: 0800      PCL      *      $00EF  PROGRAM COUNTER
0084: 0800      PCH      *      $00F0
0085: 0800      PREG      *      $00F1  STATUS REGISTER
0086: 0800      SPUSER    *      $00F2  USER STACK POINTER
0087: 0800      ACC       *      $00F3  ACCUMULATOR
0088: 0800      YREG      *      $00F4  INDEX Y
0089: 0800      XREG      *      $00F5  INDEX X
0090:
0091:
0092:      INPUT BUFFERS
0093:
0094: 0800      INL       *      $00F8  LOWER BYTE
0095: 0800      INH       *      $00F9  UPPER BYTE
0096:
0097:
0098:      ADDRESS POINTER
0099:
0100: 0800      POINTL    *      $00FA
0101: 0800      POINTH    *      $00FB
0102:
0103:
0104:      TEMPORARAY DATA BUFFERS
0105:
0106: 0800      TEMP      *      $00FC
0107: 0800      TEMPX     *      $00FD
0108: 0800      STPBIT    *      $1A59  NUMBER OF STOP BITS + 1
0109: 0800      CNTRL     *      $1A5A  BIT TIME BUFFER
0110: 0800      CNTH      *      $1A5B
0111: 0800      CNTHL     *      $1A5C  HALF BIT TIME BUFFER
0112: 0800      CNTHH     *      $1A5D
0113: 0800      TIML      *      $1A5E  COUNT DOWN BUFFER
0114: 0800      TIMH      *      $1A5F
0115: 0800      TEMPA     *      $1A60  TEMPS
0116: 0800      TEMPB     *      $1A61
0117: 0800      CHA       *      $1A62  CHARACTER BUFFER
0118: 0800      PARAL     *      $1A63  PARAMETER BUFFERS
0119: 0800      PARAH     *      $1A64
0120: 0800      PARBL     *      $1A65
0121: 0800      PARBH     *      $1A66
0122: 0800      PRIEMP    *      $1A67  TTY BUFFER
0123: 0800      BRKT      *      $1A7C  BREAK TEST VECTOR
0124:
0125:
0126:
0127:
0128:      *****
0129:      *** BUFFERS & EXTERNAL ADDRESSES ***
0130:      *****
0131:
0132: 0800      DISCNT     *      $1A68  DISPLAY COUNTER
0133: 0800      COLDST     *      $1CB5  EDITOR COLD START
0134: 0800      WARMST     *      $1CCA  EDITOR WARM START
0135: 0800      BEGIN      *      $1ED3  EDITOR SUBROUTINE
0136: 0800      RESET      *      $1C1D  RESET OF VERSION D
0137: 0800      GETKEY     *      $1DF9  COMPUTE THE KEY VALUE
0138: 0800      LDAINH     *      $1DA7  PART OF SCAND/SCANDS

```

```

0139: 0800      BEGADL *      $00E2  BEGIN ADDRESS POINTER
0140: 0800      BEGADH *      $00E3
0141: 0800      ENDADL *      $00E4  END ADDRESS POINTER
0142: 0800      ENDADH *      $00E5
0143: 0800      CENDL  *      $00E8  CURRENT END ADDRESS POINTER
0144: 0800      CENDH  *      $00E9

0145:
0146:      *****
0147:      *** TAPE MANAGEMENT ***
0148:      *****
0149:
0150:
0151:      >PC KEY: READ A DATA BLOCK WITH ID=01...FE FROM TAPE
0152:              1) IF ID = 00, NO SA MAY BE ENTERED
0153:              2) IF ID = FF, SAH & SAL SHOULD BE ENTERED
0154:      >AD KEY: SAVE A DATA BLOCK FROM SA TO EA-1 ON TAPE
0155:              1) ENTER ID = 01...FE
0156:              2) ENTER SAH & SAL
0157:              3) ENTER EAH & EAL + 1
0158:      >DA KEY: 1) EDITOR COLD START ENTRY
0159:              2) FILE IS DEFINED BY BEGH,BEGL & ENDH,ENDL
0160:      >+ KEY: DISPLAY ID-SAH-SAL-EAH-EAL-BEGH-BEGL-ENDH-
0161:              ENDL-ID-SAH...
0162:      >GO KEY: SAVE AN EDITOR FILE BETWEEN BEGAD AND CEND
0163:              ON TAPE; THEN JUMP TO WARM START ENTRY OF
0164:              THE EDITOR AND DISPLAY THE FIRST INSTRUCTION
0165:
0166:
0167:
0168: 0800 20 72 0C  GETA  JSR  ADJPNT DECREMENT FILE POINTER
0169: 0803 A5 FA      LDAZ  POINTL POINT := SA
0170: 0805 8D 70 1A      STA  SAL
0171: 0808 A5 FB      LDAZ  POINTH
0172: 080A 8D 71 1A      STA  SAH
0173: 080D 4C 4E 08      JMP  GETB
0174:
0175: 0810 A2 00      TPINIT LDXIM $00
0176: 0812 8E 79 1A      STX  ID      RESET ALL TAPE PARAMETERS
0177: 0815 8E 70 1A      STX  SAL
0178: 0818 8E 71 1A      STX  SAH
0179: 081B 8E 72 1A      STX  EAL
0180: 081E 8E 73 1A      STX  EAH
0181:
0182: 0821 8E 68 1A  TPI   STX  DISCNT RESET DISPLAY COUNTER
0183: 0824 A9 06      LDAIM $06
0184: 0826 8D 82 1A      STA  PBD      DISPLAY OFF
0185: 0829 A9 1E      LDAIM $1E
0186: 082B 8D 83 1A      STA  PBDD
0187:
0188: 082E 20 36 09  TPTXT JSR  TAPDIS DISPLAY TAPE PARAMETERS
0189: 0831 D0 FB      BNE  TPTXT KEY RELEASED?
0190:
0191: 0833 20 36 09  TTX   JSR  TAPDIS
0192: 0836 F0 FB      BEQ  TTX   ANY KEY DEPRESSED?
0193: 0838 20 36 09      JSR  TAPDIS DEBOUNCE KEY
0194: 083B F0 F6      BEQ  TTX   KEY STILL DEPRESSED?
0195: 083D 20 F9 1D      JSR  GETKEY RETURN WITH KEY IN ACCU
0196:
0197: 0840 C9 14      GET   CMPIM $14  PC KEY?
0198: 0842 D0 0E      BNE  SAVE
0199: 0844 20 02 0B      JSR  RDTAPE READ DATA FROM TAPE
0200: 0847 A2 FF      LDXIM $FF
0201: 0849 EC 79 1A      CPX  ID      CHECK, IF ID = FF
0202: 084C F0 B2      BEQ  GETA  IF YES, ADJUST FILE POINTER
0203:
0204: 084E E8      GETB  INX
0205: 084F 4C 21 08      JMP  TPI  RESET DISCNT
0206:                                SHOW 'ID XX'
0207: 0852 C9 10      SAVE  CMPIM $10  AD KEY?

```

```

0208: 0854 D0 08      BNE  PLUS
0209: 0856 20 DF 09    JSR  DUMP  WRITE DATA ON TAPE
0210: 0859 A2 00      LDXIM $00
0211: 085B 4C 21 08    JMP  TPI   SHOW 'ID XX'
0212:
0213: 085E C9 12      PLUS CMPIM $12  + KEY?
0214: 0860 D0 11      BNE  FILES
0215: 0862 EE 68 1A    INC  DISCNT SET UP PARAMETER COUNTER
0216: 0865 A0 09      LDYIM $09  LIMIT COUNT
0217: 0867 CC 68 1A    CPY  DISCNT
0218: 086A D0 C2      BNE  TPTXT
0219: 086C A0 00      LDYIM $00  RESET PARAMETER COUNTER
0220: 086E 8C 68 1A    STY  DISCNT
0221: 0871 F0 BB      BEQ  TPTXT
0222:
0223: 0873 C9 13      FILES CMPIM $13  GO KEY?
0224: 0875 D0 25      BNE  DAT
0225: 0877 A5 E2      LDAZ  BEGADL
0226: 0879 8D 70 1A    STA  SAL   BEGAD := SA
0227: 087C A5 E3      LDAZ  BEGADH
0228: 087E 8D 71 1A    STA  SAH
0229: 0881 A5 E8      LDAZ  CENDL  CEND := EA
0230: 0883 8D 72 1A    STA  EAL
0231: 0886 A5 E9      LDAZ  CENDH
0232: 0888 8D 73 1A    STA  EAH
0233:
0234: 088B 20 DF 09    JSR  DUMP  WRITE DATA = FILE ON TAPE
0235: 088E 20 D3 1E    JSR  BEGIN  SHOW THE FIRST INSTRUKTION
0236: 0891 A9 1E      LDAIM $1E  DIFINE I/O
0237: 0893 8D 83 1A    STA  PBDD
0238: 0896 A2 FF      LDXIM $FF  RESET STACK
0239: 0898 9A      TXS
0240: 0899 4C CA 1C    JMP  WARMST RETURN TO EDITOR
0241:
0242: 089C C9 11      DAT  CMPIM $11  DA KEY?
0243: 089E D0 03      BNE  SHIFT IT WAS A DATA KEY
0244: 08A0 4C B5 1C    JMP  COLDST EDITOR COLD START ENTRY
0245:
0246: 08A3 06 F9      SHIFT ASLZ  INH   SHIFT KEY INTO DISPLAY BUFFER
0247: 08A5 06 F9      ASLZ  INH
0248: 08A7 06 F9      ASLZ  INH
0249: 08A9 06 F9      ASLZ  INH
0250: 08AB 05 F9      ORAZ  INH
0251: 08AD 85 F9      STAZ  INH
0252: 08AF A0 00      LDYIM $00  RESET PARAMETER COUNTER
0253: 08B1 CC 68 1A    CPY  DISCNT ID TO DISPLAY?
0254: 08B4 D0 06      BNE  STSAH
0255: 08B6 8D 79 1A    STA  ID   SAVE ID
0256: 08B9 4C 2E 08    JMP  TPTXT
0257:
0258: 08BC C8      STSAH INY
0259: 08BD CC 68 1A    CPY  DISCNT SAH TO DISPLAY?
0260: 08C0 D0 06      BNE  STSAL
0261: 08C2 8D 71 1A    STA  SAH   SAVE SAH
0262: 08C5 4C 2E 08    JMP  TPTXT
0263:
0264: 08C8 C8      STSAL INY
0265: 08C9 CC 68 1A    CPY  DISCNT SAL TO DISPLAY?
0266: 08CC D0 06      BNE  STEAH
0267: 08CE 8D 70 1A    STA  SAL   SAVE SAL
0268: 08D1 4C 2E 08    JMP  TPTXT
0269:
0270: 08D4 C8      STEAH INY
0271: 08D5 CC 68 1A    CPY  DISCNT EAH TO DISPLAY?
0272: 08D8 D0 06      BNE  STEAL
0273: 08DA 8D 73 1A    STA  EAH   SAVE EAH
0274: 08DD 4C 2E 08    JMP  TPTXT
0275:
0276: 08E0 C8      STEAL INY

```

```

0277: 08E1 CC 68 1A      CPY   DISCNT EAL TO DISPLAY?
0278: 08E4 D0 06         BNE   STBEGH
0279: 08E6 8D 72 1A      STA   EAL      SAVE EAL
0280: 08E9 4C 2E 08      JMP   TPTXT
0281:
0282: 08EC C8             STBEGH INY
0283: 08ED CC 68 1A      CPY   DISCNT BEGH TO DISPLAY?
0284: 08F0 D0 05         BNE   STBEGH
0285: 08F2 85 E3         STAZ  BEGADH  SAVE BEGADH
0286: 08F4 4C 2E 08      JMP   TPTXT
0287:
0288: 08F7 C8             STBEGH INY
0289: 08F8 CC 68 1A      CPY   DISCNT BEGADL TO DISPLAY?
0290: 08FB D0 05         BNE   STENDH
0291: 08FD 85 E2         STAZ  BEGADL  SAVE BEGADL
0292: 08FF 4C 2E 08      JMP   TPTXT
0293:
0294: 0902 C8             STENDH INY
0295: 0903 CC 68 1A      CPY   DISCNT ENDADH TO DISPLAY?
0296: 0906 D0 05         BNE   STENDL
0297: 0908 85 E5         STAZ  ENDADH  SAVE ENDADH
0298: 090A 4C 2E 08      JMP   TPTXT
0299:
0300: 090D C8             STENDL INY
0301: 090E CC 68 1A      CPY   DISCNT ENDADL TO DISPLAY?
0302: 0911 D0 05         BNE   TPVEC  ERROR EXIT
0303: 0913 85 E4         STAZ  ENDADL  SAVE ENDADL
0304: 0915 4C 2E 08      JMP   TPTXT
0305:
0306:
0307: 0918 4C 10 08      TPVEC  JMP   TPINIT
0308:
0309:          SUBROUTINES OF 'TAPE MANAGEMENT'
0310:
0311: 091B B9 BB 09      TDISP  LDAY  TLOOK  TEXT LOOKUP
0312: 091E 8D 80 1A      STA   PAD   OUTPUT TEXT PATTERN
0313: 0921 8E 82 1A      STX   PBD   DISPLAY ENABLE
0314: 0924 98             TYA             SAVE INDEX Y
0315: 0925 A0 7F         LDYIM $7F
0316:
0317: 0927 88             DELY  DEY             DELAY
0318: 0928 D0 FD         BNE   DELY
0319: 092A A8             TAY             RESTORE INDEX Y
0320: 092B E8             INX             SET UP FOR NEXT DISPLAY
0321: 092C E8             INX
0322: 092D C8             INY             ADJUST TEXT POINTER
0323: 092E E0 10         CPXIM $10        4 DISPLAYS SCANNED?
0324: 0930 D0 E9         BNE   TDISP
0325: 0932 20 A7 1D      JSR   LDAINH  DISPLAY DATA BUFFER INH
0326: 0935 60             RTS
0327:
0328: 0936 A9 7F         TAPDIS LDAIM $7F
0329: 0938 8D 81 1A      STA   PADD  I/O DEFINITION
0330:
0331: 093B A2 08         SID   LDXIM $08  INIT DISPLAY SWITCH
0332: 093D A0 00         LDYIM $00
0333: 093F CC 68 1A      CPY   DISCNT ID TO DISPLAY?
0334: 0942 D0 09         BNE   SSAH
0335: 0944 AD 79 1A      LDA   ID      ID TO DISPLAY
0336: 0947 85 F9         STAZ  INH
0337:
0338: 0949 20 1B 09      COMPNT JSR   TDISP  SHOW ANY PARAMETER
0339: 094C 60             RTS
0340:
0341: 094D C8             SSAH  INY
0342: 094E CC 68 1A      CPY   DISCNT SAH TO DISPLAY?
0343: 0951 D0 09         BNE   SSAL
0344: 0953 AD 71 1A      LDA   SAH   SAH TO DISPLAY
0345: 0956 85 F9         STAZ  INH

```


0346:	0958	A0 04		LDYIM \$04	LOOKUP POINTER
0347:	095A	D0 ED		BNE COMPNT	SHOW SAH
0348:					
0349:	095C	C8	SSAL	INY	
0350:	095D	CC 68 1A		CPY DISCNT	SAL TO DISPLAY?
0351:	0960	D0 09		BNE SEAH	
0352:	0962	AD 70 1A		LDA SAL	SAL TO DISPLAY
0353:	0965	85 F9		STAZ INH	
0354:	0967	A0 08		LDYIM \$08	LOOKUP POINTER
0355:	0969	D0 DE		BNE COMPNT	SHOW SAL
0356:					
0357:	096B	C8	SEAH	INY	
0358:	096C	CC 68 1A		CPY DISCNT	EAH TO DISPLAY?
0359:	096F	D0 09		BNE SEAL	
0360:	0971	AD 73 1A		LDA EAH	EAH TO DISPLAY
0361:	0974	85 F9		STAZ INH	
0362:	0976	A0 0C		LDYIM \$0C	LOOKUP POINTER
0363:	0978	D0 CF		BNE COMPNT	SHOW EAH
0364:					
0365:	097A	C8	SEAL	INY	
0366:	097B	CC 68 1A		CPY DISCNT	
0367:	097E	D0 09		BNE SBEGH	
0368:	0980	AD 72 1A		LDA EAL	
0369:	0983	85 F9		STAZ INH	
0370:	0985	A0 10		LDYIM \$10	LOOKUP POINTER
0371:	0987	D0 C0		BNE COMPNT	SHOW EAL
0372:					
0373:	0989	C8	SBEGH	INY	
0374:	098A	CC 68 1A		CPY DISCNT	
0375:	098D	D0 08		BNE SBEGH	
0376:	098F	A5 E3		LDAZ BEGADH	
0377:	0991	85 F9		STAZ INH	
0378:	0993	A0 14		LDYIM \$14	
0379:	0995	D0 B2		BNE COMPNT	
0380:					
0381:	0997	C8	SBEGH	INY	
0382:	0998	CC 68 1A		CPY DISCNT	
0383:	099B	D0 08		BNE SENDH	
0384:	099D	A5 E2		LDAZ BEGADL	
0385:	099F	85 F9		STAZ INH	
0386:	09A1	A0 18		LDYIM \$18	
0387:	09A3	D0 A4		BNE COMPNT	
0388:					
0389:	09A5	C8	SENDH	INY	
0390:	09A6	CC 68 1A		CPY DISCNT	
0391:	09A9	D0 08		BNE SENDL	
0392:	09AB	A5 E5		LDAZ ENDADH	
0393:	09AD	85 F9		STAZ INH	
0394:	09AF	A0 1C		LDYIM \$1C	
0395:	09B1	D0 96		BNE COMPNT	
0396:					
0397:	09B3	A5 E4	SENDL	LDAZ ENDADL	
0398:	09B5	85 F9		STAZ INH	
0399:	09B7	A0 20		LDYIM \$20	
0400:	09B9	D0 8E		BNE COMPNT	
0401:					
0402:					
0403:					
0404:					
0405:					
0406:	09BB	66	TLOOK	= \$66	ID PATTERN
0407:	09BC	21		= \$21	
0408:	09BD	7F		= \$7F	
0409:	09BE	7F		= \$7F	
0410:					
0411:	09BF	52		= \$52	SAH PATTERN
0412:	09C0	08		= \$08	
0413:	09C1	09		= \$09	
0414:	09C2	7F		= \$7F	

```

0415:
0416: 09C3 52      =      $52    SAL PATTERN
0417: 09C4 08      =      $08
0418: 09C5 47      =      $47
0419: 09C6 7F      =      $7F
0420:
0421: 09C7 06      =      $06    EAH PATTERN
0422: 09C8 08      =      $08
0423: 09C9 09      =      $09
0424: 09CA 7F      =      $7F
0425:
0426: 09CB 06      =      $06    EAL PATTERN
0427: 09CC 08      =      $08
0428: 09CD 47      =      $47
0429: 09CE 7F      =      $7F
0430:
0431: 09CF 03      =      $03    BEGH PATTERN
0432: 09D0 06      =      $06
0433: 09D1 42      =      $42
0434: 09D2 09      =      $09
0435:
0436: 09D3 03      =      $03    BEGL PATTERN
0437: 09D4 06      =      $06
0438: 09D5 42      =      $42
0439: 09D6 47      =      $47
0440:
0441: 09D7 06      =      $06    ENDH PATTERN
0442: 09D8 2B      =      $2B
0443: 09D9 21      =      $21
0444: 09DA 09      =      $09
0445:
0446: 09DB 06      =      $06    ENDL PATTERN
0447: 09DC 2B      =      $2B
0448: 09DD 21      =      $21
0449: 09DE 47      =      $47
0450:
0451:
0452:
0453:
0454: 09DF A9 7D    DUMP   LDAIM $7D    3600HZ HALF PERIODE DELAY
0455: 09E1 8D 6C 1A   STA    HIGHER
0456: 09E4 A9 C3     LDAIM $C3    2400 HZ HALF PERIODE DELAY
0457: 09E6 8D 6D 1A   STA    LOWER
0458: 09E9 A9 03     LDAIM $03    TOGGLE 3 TIMES AT 3600 HZ
0459: 09EB 8D 76 1A   STA    FIRST
0460: 09EE A9 02     LDAIM $02    TOGGLE 2 TIMES AT 2400 HZ
0461: 09FO 8D 77 1A   STA    SECOND
0462:
0463: 09F3 A9 47     DUMPT   LDAIM $47    PBD PATTERN
0464: 09F5 A2 FF     LDXIM $FF    PBDD PATTERN
0465: 09F7 8D 82 1A   STA    PBD    INPUT OFF,OUTPUT ON,STROBE DISABLED
0466: 09FA 8D 78 1A   STA    GANG   SAVE BITS OF PBD
0467: 09FD 8E 83 1A   STX    PBDD    PB7...PBO IS OUTPUT
0468: 0A00 A9 00     LDAIM $00    PAD PATTERN
0469: 0A02 A2 7F     LDXIM $7F    PADD PATTERN
0470: 0A04 8D 80 1A   STA    PAD    SEGMENTS ON BUT DISABLED
0471: 0A07 8E 81 1A   STX    PADD    ALL LINES OUTPUT EXEPT PA7
0472: 0A0A 8D 6E 1A   STA    CHKL   RESET CHECK SUM
0473: 0A0D 8D 6F 1A   STA    CHKH
0474: 0A10 AD 70 1A   LDA    SAL    INITIALIZE DUMPT POINTER
0475: 0A13 85 FA     STAZ   POINTL
0476: 0A15 AD 71 1A   LDA    SAH
0477: 0A18 85 FB     STAZ   POINTH
0478: 0A1A A2 FF     LDXIM $FF    SET SYNC COUNTER
0479: 0A1C 8E 74 1A   STX    SYNCNT
0480:
0481: 0A1F A9 16     SYNCs  LDAIM '    SYN CHARACTER
0482: 0A21 20 A3 0A   JSR    OUTCH  OUTPUT 255 SYN CHARACTERS
0483: 0A24 CE 74 1A   DEC    SYNCNT

```



```

0484: 0A27 D0 F6      BNE   SYNC5
0485:
0486: 0A29 A9 2A      LDAIM  '*'      OUTPUT START CHARACTER
0487: 0A2B 20 A3 0A    JSR    OUTCH
0488: 0A2E AD 79 1A    LDA    ID        OUTPUT ID
0489: 0A31 20 8B 0A    JSR    OUTBT
0490: 0A34 AD 70 1A    LDA    SAL      OUTPUT START ADDRESS
0491: 0A37 20 7A 0A    JSR    OUTBTC   AND START CHECK SUM COMPUTATION
0492: 0A3A AD 71 1A    LDA    SAH
0493: 0A3D 20 7A 0A    JSR    OUTBTC
0494:
0495: 0A40 A5 FB      DATATR LDAZ  POINTH
0496: 0A42 CD 73 1A    CMP    EAH      ENTIRE FILE TRANSMITTED?
0497: 0A45 D0 23      BNE    HEXDAT
0498: 0A47 A5 FA      LDAZ  POINTL
0499: 0A49 CD 72 1A    CMP    EAL
0500: 0A4C D0 1C      BNE    HEXDAT
0501:
0502: 0A4E A9 2F      LDAIM  '/'      OUTPUT END OF DATA CHARACTER
0503: 0A50 20 A3 0A    JSR    OUTCH    STOP WITH CHECK SUM COMPUTATION
0504: 0A53 AD 6E 1A    LDA    CHKL     OUTPUT CHECK SUM
0505: 0A56 20 8B 0A    JSR    OUTBT
0506: 0A59 AD 6F 1A    LDA    CHKH
0507: 0A5C 20 8B 0A    JSR    OUTBT
0508: 0A5F A9 04      LDAIM  '        EOT CHARACTER
0509: 0A61 20 A3 0A    JSR    OUTCH    OUTPUT EOT CHARACTER
0510: 0A64 A9 04      LDAIM  '        BOT CHARACTER
0511: 0A66 20 A3 0A    JSR    OUTCH
0512:
0513: 0A69 60          RTS          RETURN TO CALLER
0514:
0515: 0A6A A0 00      HEXDAT LDYIM $00
0516: 0A6C B1 FA      LDAIY  POINTL   FETCH CURRENT DATA BYTE
0517: 0A6E 20 7A 0A    JSR    OUTBTC   TRANSMIT CURRENT DATA BYTE
0518: 0A71 E6 FA      INCZ   POINTL   AND COMPUT CHECK SUM
0519: 0A73 D0 CB      BNE    DATATR   SET UP FOR NEXT DATA BYTE
0520: 0A75 E6 FB      INCZ   POINTH
0521: 0A77 4C 40 0A    JMP    DATATR
0522:
0523:                *** END OF DUMP/DUMPT ***
0524:                DUMP'S SUBROUTINES
0525:
0526:
0527:                OUTBTC OUTPUTS A HEX BYTE AS TWO ASCII CHARACTERS
0528:                TO THE TAPE RECORDER. ALSO THE CHECK SUM IS COM-
0529:                PUTED.
0530:
0531:                OUTBT OUTPUTS A HEX BYTE AS TWO ASCII CHARACTERS
0532:                TO THE TAPE RECORDER WITHOUT CHECK SUM COMPUTATION.
0533:
0534:                NIBOUT CONVERTS A HEX NIBBLE TO AN ASCII CHARACTER
0535:                AND TRANSMITS THE 8 BIT ASCII WORD TO THE TAPE.
0536:
0537:                HIGH AND LOW PRODUCE THE DELAYS OF THE 3600 HZ AND
0538:                THE 2400 HZ FREQUENCY.
0539:
0540:
0541:                *** OUTBTC/OUTBT ***
0542:
0543: 0A7A A8          OUTBTC TAY          SAVE ACCU
0544: 0A7B 18          CLC
0545: 0A7C 6D 6E 1A    ADC    CHKL     CHECK SUM COMPUTATION
0546: 0A7F 8D 6E 1A    STA    CHKL
0547: 0A82 AD 6F 1A    LDA    CHKH
0548: 0A85 69 00      ADCIM  $00     CHK:= CHK + BYTE
0549: 0A87 8D 6F 1A    STA    CHKH
0550: 0A8A 98          TYA          GET ACCU AGAIN
0551:
0552: 0A8B A8          OUTBT  TAY          SAVE ACCU TEMP

```

```

0553: 0A8C 4A          LSRA          GET UPPER NIBBLE
0554: 0A8D 4A          LSRA
0555: 0A8E 4A          LSRA
0556: 0A8F 4A          LSRA
0557: 0A90 20 9A 0A    JSR      NIBOUT OUTPUT UPPER NIBBLE AS ASCII CHAR.
0558: 0A93 98          TYA          GET BYTE AGAIN
0559: 0A94 29 0F        ANDIM $0F GET LOWER NIBBLE
0560: 0A96 20 9A 0A    JSR      NIBOUT OUTPUT LOWER NIBBLE AS ASCII CHAR.
0561: 0A99 60          RTS
0562:
0563:          *** END OF OUTBTC/OUTBT ***
0564:
0565:
0566:          *** NIBOUT/OUTCH ***
0567:
0568: 0A9A C9 0A          NIBOUT CMPIM $0A   CONVERT A NIBBLE TO AN ASCII CHAR.
0569: 0A9C 18             CLC
0570: 0A9D 30 02          BMI  NIB
0571: 0A9F 69 07          ADCIM $07
0572:
0573: 0AA1 69 30          NIB  ADCIM $30
0574:
0575: 0AA3 A2 08          OUTCH LDXIM $08   SET UP FOR 8 BITS
0576: 0AA5 8E 75 1A      STX  BITS
0577:
0578: 0AA8 4A             ONE   LSRA          SHIFT OUT BIT BY BIT
0579: 0AA9 48             PHA          SAVE CHARACTER
0580: 0AAA 90 0C          BCC  ZERO
0581: 0AAC 20 C8 0A        JSR  HIGH   START AT 3600 HZ
0582: 0AAF 20 E5 0A        JSR  LOW
0583: 0AB2 20 E5 0A        JSR  LOW   END AT 2400 HZ
0584: 0AB5 4C C1 0A        JMP  ZRO
0585:
0586: 0AB8 20 C8 0A        ZERO JSR  HIGH   START AT 3600 HZ
0587: 0ABB 20 C8 0A        JSR  HIGH
0588: 0ABE 20 E5 0A        JSR  LOW   END AT 2400 HZ
0589:
0590: 0AC1 68             ZRO   PLA          GET CHARACTER AGAIN
0591: 0AC2 CE 75 1A      DEC  BITS   ALL BITS SHIFTED OUT
0592: 0AC5 D0 E1          BNE  ONE
0593: 0AC7 60             RTS
0594:
0595:          *** END OF NIBOUT/OUTCH ***
0596:
0597:
0598:          *** HIGH ***
0599:
0600: 0AC8 AE 76 1A      HIGH  LDX  FIRST  AMOUNT OF HALF PERIODES.
0601:
0602: 0ACB 2C D5 1A      HIG   BIT   RDFLAG TIME OUT?
0603: 0ACE 10 FB          BPL  HIG
0604: 0AD0 AD 6C 1A      LDA  HIGHER SET UP TIMER FOR SHORT PERIODE
0605: 0AD3 8D F4 1A      STA  CNTA  DISABLE TIMER IRQ, CLK1T
0606: 0AD6 AD 78 1A      LDA  GANG
0607: 0AD9 49 80          EORIM $80  TOGGLE OUTPUT
0608: 0ADB 8D 82 1A      STA  PBD
0609: 0ADE 8D 78 1A      STA  GANG  SAVE STATUS QUO
0610: 0AE1 CA             DEX
0611: 0AE2 D0 E7          BNE  HIG
0612: 0AE4 60             RTS
0613:
0614:          *** END OF HIGH ***
0615:
0616:
0617:          *** LOW ***
0618:
0619: 0AE5 AE 77 1A      LOW   LDX  SECOND AMOUNT OF HALF PERIODES
0620:
0621: 0AE8 2C D5 1A      LO    BIT   RDFLAG TIME OUT?

```

```

0622: 0AEB 10 FB      BPL    LO
0623: 0AED AD 6D 1A   LDA    LOWER  SET UP TIMER FOR LONG PERIODE
0624: 0AF0 8D F4 1A   STA    CNTA   DISABLE TIMER IRQ, CLK1T
0625: 0AF3 AD 78 1A   LDA    GANG
0626: 0AF6 49 80       EORIM  $80   TOGGLE OUTPUT
0627: 0AF8 8D 82 1A   STA    PBD
0628: 0AFB 8D 78 1A   STA    GANG   SAVE STATUS QUO
0629: 0AFE CA        DEX
0630: 0AFF D0 E7      BNE    LO
0631: 0B01 60         RTS
0632:
0633:                *** END OF LOW ***
0634:
0635:                *****
0636:                * RDTAPE IS A SUBROUTINE *
0637:                *****
0638:
0639:                ID = 00: IGNORE ID ON TAPE; DISCARD DATA BLOCK
0640:                AT SAL, SAH COMING FROM TAPE
0641:
0642:                ID = FF: IGNORE ID ON TAPE; DISCARD DATA BLOCK
0643:                AT SAL, SAH STORED IN JUNIORS MEMORY
0644:
0645: 0B02 A9 32      RDTAPE LDAIM $32   INPUT RECORDER IS ON
0646: 0B04 8D 82 1A   STA    PBD   OUTPUT RECORDER IS OFF, PB7 IS ENABLED
0647: 0B07 8D 78 1A   STA    GANG   STROBE '9' IS ENABLED
0648: 0B0A A9 7E      LDAIM $7E   PB0 IS INPUT
0649: 0B0C 8D 83 1A   STA    PBDD   PB7 IS INPUT
0650: 0B0F A9 7F      LDAIM $7F   PA6...PA0 IS OUTPUT
0651: 0B11 8D 81 1A   STA    PADD
0652: 0B14 A9 00      LDAIM $00   SEGMENTS ON
0653: 0B16 8D 6E 1A   STA    CHKL   RESET CHECK SUM
0654: 0B19 8D 6F 1A   STA    CHKH
0655:
0656: 0B1C A9 FF      SYNC    LDAIM $FF  RESET FOR SYN CHARACTER
0657: 0B1E 8D 6B 1A   STA    CHAR
0658:
0659: 0B21 20 C2 0B   SYNCA   JSR    RDBIT  READ A BIT FROM TAPE
0660: 0B24 6E 6B 1A   ROR    CHAR   RIGHT SHIFT
0661: 0B27 AD 6B 1A   LDA    CHAR   GET CURRENT CHARACTER
0662: 0B2A 20 E8 0B   JSR    BTWEEN  DISPLAY NOT SYN CHARACTER
0663: 0B2D C9 16      CMPIM  '      SYN CHARACTER?
0664: 0B2F D0 F0      BNE    SYNCA  IF NOT, RESYNC
0665: 0B31 A0 0A      LDYIM  $0A   TRY IT FOR 10 SYNS AT LEAST
0666: 0B33 8C 69 1A   STY    SY     SYN COUNTER
0667:
0668: 0B36 20 36 0C   TENSYN JSR    RDCH
0669: 0B39 20 5D 0C   JSR    CHARVU  DISPLAY SYN CHARACTER
0670: 0B3C C9 16      CMPIM  '      STILL SYN CHARACTER?
0671: 0B3E D0 DC      BNE    SYNC  IF NOT, RETURN
0672: 0B40 CE 69 1A   DEC    SY     10 SYNS RECEIVED?
0729: 0BBA AD 71 1A   LDA    SAH
0730: 0BBD 85 FB      STAZ   POINTH
0731: 0BBF 4C 70 0B   JMP    FILMEM
0732:
0733:                *** END OF RDTAPE ***
0734:
0735:                SUBROUTINES OF RDTAPE
0736:
0737:                *** RDBIT ***
0738:
0739:                RDBIT READS 1 BIT FROM TAPE.
0740:                LOG 1: IT RETURNS WITH C = 1
0741:                LOG 0: IT RETURNS WITH C = 0
0742:
0743: 0BC2 2C 82 1A   RDBIT  BIT    PBD    3600 HZ?
0744: 0BC5 10 FB      BPL    RDBIT
0745: 0BC7 AD D4 1A   LDA    RDTDIS  GET COUNT DOWN
0746: 0BCA A0 FF      LDYIM  $FF

```

```

0747: OBCC 8C F6 1A      STY  CNTC  START TIMER FOR 2400 HZ COUNT DOWN
0748: OBCF AO 14      LDYIM $14
0749:
0750: OBD1 88          RDBA  DEY          DELAY JITTER TIME
0751: OBD2 DO FD          BNE  RDBA
0752:
0753: OBD4 2C 82 1A      RDBB  BIT  PBD    2400 HZ?
0754: OBD7 30 FB          BMI  RDBB
0755: OBD9 38          SEC
0756: OBDA ED D4 1A      SBC  RDTDIS SET OR RESET C-FLAG
0757: OBDD AO FF          LDYIM $FF
0758: OBDF 8C F6 1A      STY  CNTC  START TIMER FOR 3600 HZ COUNT DOWN
0759: OBE2 AO 07          LDYIM $07  DELAY FOR JITTER
0760:
0761: OBE4 88          RDBC  DEY
0762: OBE5 DO FD          BNE  RDBC
0763: OBE7 60          RTS
0764:
0765:      *** END OF RDBIT ***
0766:
0767:
0768:
0769:      *** BTWEEN ***
0770:
0771:      DISPLAY THE BETWEEN CHARACTER
0772:
0773: OBE8 48          BTWEEN PHA          SAVE ACCU
0774: OBE9 A9 36          LDAIM $36      OUTPUT BETWEEN CHARACTER
0775: OBEB 8D 80 1A      STA  PAD
0776: OBEE 68          PLA          GET ACCU AGAIN
0777: OBEF 20 64 OC      JSR  VU      DON'T CHANGE BETWEEN
0778: OBF2 60          RTS
0779:
0780:      *** END OF BTWEEN ***
0781:      *** RDBYT ***
0782:
0783:
0784:      READ 1 HEX BYTE = 2 ASCII CHARACTERS FROM TAPE.
0785:      COMPOSE THE HEX VALUES OF THE CHARACTERS IN ACCU
0786:      AND RETURN.
0787:
0788:      1) N = 1: A NOT VALID HEX CHARACTER WAS TRANSMITTED
0789:      2) Z = 1: END OF DATA TRANSMISSION
0790:      3) Z = 0: VALID HEX BYTE IN ACCU
0791:      4) THE N-FLAG HAS ALWAYS PRIORITY
0792:
0793: OBF3 20 36 OC      RDBYT JSR  RDCH  READ ANY ASCII CHARACTER FROM TAPE
0794: OBF6 C9 2F          CMPIM '/'      END OF DATA CHARACTER?
0795: OBF8 DO 01          BNE  RBB
0796:
0797: OBFA 60          RBA  RTS          ERROR EXIT
0798:
0799: OBF8 20 19 OC      RBB  JSR  ASCHEX ASCII HEX CONVERSION
0800: OBFE 30 FA          BMI  RBA      NOT VALID CHARACTER
0801: OC00 0A          ASLA          SHIFT NIBBLE TO LEFT
0802: OC01 0A          ASLA
0803: OC02 0A          ASLA
0804: OC03 0A          ASLA
0805: OC04 8D 6A 1A      STA  BYTE      SAVE HIGH ORDER NIBBLE
0806: OC07 20 36 OC      JSR  RDCH  READ NEXT CHARACTER
0807: OC0A C9 2F          CMPIM '/'      END OF DATA CHARACTER
0808: OC0C FO EC          BEQ  RBA
0809: OC0E 20 19 OC      JSR  ASCHEX ASCII HEX CONVERSION
0810: OC11 30 E7          BMI  RBA      NOT VALID CHARACTER
0811: OC13 OD 6A 1A      ORA  BYTE      BYTE = HIGH ORDER AND LOW ORDER NIBBL
0812: OC16 AO 01          LDYIM $01      BE SHURE THAT CHARACTER
0813: OC18 60          RTS          NORMAL EXIT
0814:
0815:      *** END OF RDBYT ***

```

```

0816:      *** ASCHEX ***
0817:
0818:
0819:      CONVERT AN ASCII CHARACTER TO A HEX DATA NIBBLE.
0820:
0821:      1) RETURN WITH CONVERTED HEX NUMBER IN ACCU
0822:      2) N = 1, IF NOT VALID HEX NUMBER
0823:      3) Z = 1, IF VALID HEX NUMBER
0824:      4) *** ASCHEX IS ALSO USED IN THE PRINTER SOFTWARE ***
0825:
0826: 0C19 C9 30  ASCHEX CMPIM $30  IGNORE 00...2F
0827: 0C1B 30 0C      BMI  NOTVAL
0828: 0C1D C9 3A      CMPIM $3A
0829: 0C1F 30 0B      BMI  VALID
0830: 0C21 C9 41      CMPIM $41  IGNORE 3A...40
0831: 0C23 30 04      BMI  NOTVAL
0832: 0C25 C9 47      CMPIM $47  IGNORE 47...7F
0833: 0C27 30 03      BMI  VALID
0834:
0835: 0C29 A0 FF      NOTVAL LDYIM $FF  SET N-FLAG
0836: 0C2B 60          RTS          ERROR EXIT
0837:
0838: 0C2C C9 40      VALID  CMPIM $40  ASCII HEX CONVERSION
0839: 0C2E 30 03      BMI  VAL
0840: 0C30 18          CLC
0841: 0C31 69 09      ADCIM $09
0842:
0843: 0C33 29 0F      VAL   ANDIM $0F  HEX DATA IS LOW ORDER NIBBLE IN ACCU
0844: 0C35 60          RTS
0845:
0846:      *** END OF ASCHEX ***
0847:      *** RDCH ***
0848:
0849:
0850:      READ AN ASCII CHARACTER FROM TAPE AND
0851:      STORE IT IN ACCU
0852:
0853: 0C36 A2 08      RDCH   LDXIM $08  SET UP FOR 8 BITS
0854:
0855: 0C38 20 C2 0B  READ  JSR   RDBIT  READ A BIT FROM TAPE
0856: 0C3B 6E 6B 1A      ROR   CHAR  SHIFT BIT INTO CHARACTER
0857: 0C3E CA          DEX          ALL BITS READ?
0858: 0C3F D0 F7      BNE  READ
0859: 0C41 2E 6B 1A      ROL   CHAR  B7 MUST BE ZERO
0860: 0C44 4E 6B 1A      LSR   CHAR
0861: 0C47 AD 6B 1A      LDA   CHAR  RECEIVED CHARACTER TO ACCU
0862: 0C4A 60          RTS
0863:
0864:      *** END OF RDCH ***
0865:
0866:      *** CHKSUM ***
0867:
0868:
0869:      COMPUTE CHECK SUM OF RECEIVED DATA
0870:
0871: 0C4B 48      CHKSUM PHA          SAVE ACCU
0872: 0C4C 18      CLC
0873: 0C4D 6D 6E 1A  ADC  CHKL  CHK := CHK + BYTE
0874: 0C50 8D 6E 1A  STA  CHKL
0875: 0C53 AD 6F 1A  LDA  CHKH
0876: 0C56 69 00      ADCIM $00
0877: 0C58 8D 6F 1A  STA  CHKH
0878: 0C5B 68      PLA          GET ACCU AGAIN
0879: 0C5C 60      RTS
0880:
0881:      *** END OF CHKSUM ***
0882:
0883:
0884:      *** CHARVU ***

```

```

0885:
0886:
0887:          OUTPUT ANY CHARACTER TO 7 SEGMENT DISPLAY
0888:          CHARACTER CHANGE IS ENABLED/DISABLED
0889:
0890: 0C5D 48      CHARVU PHA          SAVE ACCU
0891: 0C5E 49 7F    EORIM $7F        OUTPUT INVERTED CHAR. TO SEGMENTS
0892: 0C60 8D 80 1A STA PAD
0893: 0C63 68      PLA          RESTORE ACCU
0894:
0895: 0C64 48      VU      PHA
0896: 0C65 AD 78 1A LDA GANG
0897: 0C68 49 02    EORIM $02        CHANGE DISPLAYS
0898: 0C6A 8D 82 1A STA PBD
0899: 0C6D 8D 78 1A STA GANG        SAVE STATUS QUO
0900: 0C70 68      PLA
0901: 0C71 60      RTS
0902:
0903:          *** END OF CHARVU ***
0904:
0905:
0906:          ADJPNT ADDRESS POINTER ADJUSTMENT
0907: 0C72 38      ADJPNT SEC        16 BIT SUBTRACTION
0908: 0C73 A5 FA    LDAZ POINTL
0909: 0C75 E9 01    SBCIM $01
0910: 0C77 85 FA    STAZ POINTL
0911: 0C79 A5 FB    LDAZ POINTH
0912: 0C7B E9 00    SBCIM $00
0913: 0C7D 85 FB    STAZ POINTH
0914: 0C7F 60      RTS
0915:
0916: 0C8D          *** END OF ADJPNT ***
ID=B4 0E
R      0F
X
0001: 1000          ORG $1000
0002:
0003:
0004:          *****
0005:          *** MAIN PROGRAM OF THE PRINTER MONITOR ***
0006:          *****
0007:
0008:
0009:          COMMANDS OF THE PRINTER MONITOR:
0010:
0011:          > '-' DECREMENT THE CURRENT ADDRESS BY ONE
0012:          > '+' INCREMENT THE CURRENT ADDRESS BY ONE
0013:          > 'SPACE' 1) PRINT THE ADDRESS IN THE INPUT BUFFER
0014:                  2) SHOW THE DATA OF THIS ADDRESS
0015:          > '.' STORE INPUT DATA AT THE CURRENT ADDRESS
0016:          > 'R' START PROGRAM EXECUTION AT THE CURRENT ADDRESS
0017:          > 'L' LIST THE CONTENTS OF ALL CPU REGISTERS
0018:          > 'P' PRINT OUT THE LAST PROGRAM CENTER
0019:          > 'M' PRINT A HEXDUMP SPECIFIED BY THE INPUT PARAMETER
0020:          > 'G' READ A DATA BLOCK WITH A CERTAIN ID FROM TAPE
0021:          > 'S' STORE A DATA BLOCK BETWEEN SA AND EA-1 ON TAPE
0022:
0023:
0024:
0025: 1000 D8      INITPR CLD          RESET SEQUENCE OF THE PRINTER PROGRAM
0026: 1001 78      SEI              IRQ LINE IS DISABLED
0027: 1002 A9 67    LDAIM $67
0028: 1004 8D 82 1A STA PBD          PBD: 01100111
0029: 1007 A9 00    LDAIM $00
0030: 1009 8D 80 1A STA PAD          PAD: 00000000
0031: 100C A2 FE    LDXIM $FE        RESET BIT TIME COUNTER
0032: 100E 8E 5A 1A STX CNTL
0033: 1011 E8      INX

```


0034:	1012	8E	5B	1A	STX	CNTH	
0035:	1015	9A			TXS		RESET COMPUTER STACK POINTER
0036:	1016	86	F2		STXZ	SPUSER	RESET USER STACK POINTER
0037:	1018	A9	7F		LDAIM	\$7F	
0038:	101A	8D	81	1A	STA	PADD	PADD: 01111111
0039:	101D	8D	83	1A	STA	PBDD	PBDD: 01111111
0040:	1020	A2	02		LDXIM	\$02	
0041:	1022	8E	59	1A	STX	STPBIT	TRANSMIT NONE PARITY & ONE STOP BIT
0042:	1025	A9	5F		LDAIM	LABJUN	
0043:	1027	8D	7C	1A	STA	BRKT	SETUP BREAK VECTOR
0044:	102A	A9	10		LDAIM	LABJUN	/256
0045:	102C	8D	7D	1A	STA	BRKT	+01
0046:	102F	A9	CF		LDAIM	STEP	SETUP STEP BY STEP VECTOR
0047:	1031	8D	7A	1A	STA	NMI	
0048:	1034	A9	14		LDAIM	STEP	/256
0049:	1036	8D	7B	1A	STA	NMI	+01
0050:							
0051:	1039	2C	80	1A	STRTBT	BIT	PAD WAIT FOR A START BIT
0052:	103C	30	FB		BMI	STRTBT	
0053:	103E	20	E0	12	JSR	COMTIM	COMPUTE THE START BIT TIME
0054:	1041	4E	5F	1A	LSR	TIMH	DIVIDE BY 2
0055:	1044	6E	5E	1A	ROR	TIML	
0056:	1047	AD	5E	1A	LDA	TIML	
0057:	104A	8D	5C	1A	STA	CNTHL	SAVE HALF START BIT TIME
0058:	104D	AD	5F	1A	LDA	TIMH	
0059:	1050	8D	5D	1A	STA	CNTHH	
0060:	1053	A2	08		LDXIM	\$08	
0061:	1055	20	03	13	JSR	DELHBT	
0062:	1058	20	BE	12	JSR	RECD	GET THE REST OF THE CHARACTER
0063:	105B	C9	7F		CMPIM	\$7F	WAS IT THE RUBOUT CHARACTER?
0064:	105D	D0	A1		BNE	INITPR	
0065:							
0066:	105F	20	46	12	LABJUN	JSR	JUNIOR PRINT 'JUNIOR'
0067:	1062	20	E8	11	JSR	CRLF	
0068:	1065	A2	FF		LDXIM	\$FF	
0069:	1067	9A			TXS		RESET STACK POINTER WHEN A BREAK OCCURS
0070:	1068	86	F2		STXZ	SPUSER	
0071:							
0072:	106A	20	59	12	RESALL	JSR	RESPAR RESET PARAMETER BUFFER
0073:	106D	20	68	12	JSR	RESIN	RESET INPUT BUFFER
0074:							
0075:	1070	20	AE	12	READCH	JSR	RECCHA WAIT FOR A CHARACTER
0076:							
0077:	1073	C9	2B		PLU	CMPIM	'+' MINUS
0078:	1075	D0	09		BNE		
0079:	1077	20	13	12	JSR	INCPNT	INCREMENT CURRENT ADDRESS
0080:	107A	20	F8	11	JSR	PRBUFS	OPEN NEXT CELL
0081:	107D	4C	6A	10	JMP	RESALL	
0082:							
0083:	1080	C9	2D		MINUS	CMPIM	'-' MINUS
0084:	1082	D0	09		BNE	SPACE	
0085:	1084	20	1A	12	JSR	DECPNT	DECREMENT CURRENT ADDRESS
0086:	1087	20	F8	11	JSR	PRBUFS	OPEN PREVIOUS CELL
0087:	108A	4C	6A	10	JMP	RESALL	
0088:							
0089:	108D	C9	20		SPACE	CMPIM	' ' PNT
0090:	108F	D0	0E		BNE		
0091:	1091	A5	F8		LDAZ	INL	OUTPUT THE ADDRESS
0092:	1093	85	FA		STAZ	POINTL	IN THE INPUT BUFFER
0093:	1095	A5	F9		LDAZ	INH	
0094:	1097	85	FB		STAZ	POINTH	
0095:	1099	20	F8	11	JSR	PRBUFS	OPEN CURRENT CELL
0096:	109C	4C	6A	10	JMP	RESALL	
0097:							
0098:	109F	C9	2E		PNT	CMPIM	' ' PNT
0099:	10A1	D0	0F		BNE	RUN	
0100:	10A3	A5	F8		LDAZ	INL	
0101:	10A5	AC	00		LDYIM	\$00	
0102:	10A7	91	FA		STAY	POINTL	STORE CURRENT DATA BYTE IN MEMORY

```

0103: 10A9 20 13 12      JSR  INCPNT
0104: 10AC 20 F8 11      JSR  PRBUFS OPEN NEXT CELL
0105: 10AF 4C 6A 10      JMP  RESALL
0106:
0107: 10B2 C9 52      RUN  CMPIM 'R
0108: 10B4 D0 13      BNE  LIST
0109: 10B6 A6 F2      LDZX SPUSER START PROGRAM EXECUTION
0110: 10B8 9A      TXS      AT THE CURRENT DISPLAYED ADDRESS
0111: 10B9 A5 FB      LDAZ POINTH
0112: 10BB 48      PHA
0113: 10BC A5 FA      LDAZ POINTL
0114: 10BE 48      PHA
0115: 10BF A5 F1      LDAZ PREG
0116: 10C1 48      PHA
0117: 10C2 A6 F5      LDZX XREG
0118: 10C4 A4 F4      LDYZ YREG
0119: 10C6 A5 F3      LDAZ ACC
0120: 10C8 40      RTI
0121:
0122: 10C9 C9 4C      LIST  CMPIM 'L
0123: 10CB D0 52      BNE  PC
0124: 10CD A0 14      LDYIM $14
0125: 10CF 20 D6 11  JSR  MESSY ACC:
0126: 10D2 A5 F3      LDAZ ACC
0127: 10D4 20 8F 12  JSR  PRBYT
0128: 10D7 A0 1A      LDYIM $1A
0129: 10D9 20 D6 11  JSR  MESSY Y :
0130: 10DC A5 F4      LDAZ YREG
0131: 10DE 20 8F 12  JSR  PRBYT
0132: 10E1 A0 20      LDYIM $20
0133: 10E3 20 D6 11  JSR  MESSY X :
0134: 10E6 A5 F5      LDAZ XREG
0135: 10E8 20 8F 12  JSR  PRBYT
0136: 10EB A0 26      LDYIM $26
0137: 10ED 20 D6 11  JSR  MESSY PC :
0138: 10F0 A5 F0      LDAZ PCH
0139: 10F2 20 8F 12  JSR  PRBYT
0140: 10F5 A5 EF      LDAZ PCL
0141: 10F7 20 8F 12  JSR  PRBYT
0142: 10FA A0 2C      LDYIM $2C
0143: 10FC 20 D6 11  JSR  MESSY SP :
0144: 10FF A9 01      LDAIM $01
0145: 1101 20 8F 12  JSR  PRBYT
0146: 1104 A5 F2      LDAZ SPUSER
0147: 1106 20 8F 12  JSR  PRBYT
0148: 1109 A0 32      LDYIM $32
0149: 110B 20 D6 11  JSR  MESSY PR :
0150: 110E 20 28 12  JSR  SHOWPR PRINT OUT FLAGS
0151: 1111 A0 38      LDYIM $38
0152: 1113 20 D6 11  JSR  MESSY NV BDIZC
0153: 1116 20 F3 11  JSR  PRSP
0154: 1119 4C 6A 10  JMP  RESALL
0155:
0156: 111C 4C B0 11  CONTIN JMP  GETTAP
0157:
0158:
0159: 111F C9 50      PC    CMPIM 'P
0160: 1121 D0 0E      BNE  MATRIX
0161: 1123 A5 EF      LDAZ PCL RESTORE LAST PROGRAM COUNTER
0162: 1125 85 FA      STAZ POINTL
0163: 1127 A5 F0      LDAZ PCH
0164: 1129 85 FB      STAZ POINTH
0165: 112B 20 F8 11  JSR  PRBUFS OPEN CURRENT CELL
0166: 112E 4C 6A 10  JMP  RESALL
0167:
0168: 1131 C9 4D      MATRIX CMPIM 'M
0169: 1133 D0 E7      BNE  CONTIN
0170: 1135 A0 52      LDYIM $52
0171: 1137 20 D6 11  JSR  MESSY HEXDUMP:

```



```

172: 113A 20 87 13      JSR  INPAR  READ PARAMETERS
173: 113D 10 03          BPL   MATF
174:
175: 113F 4C 5F 10      MATD  JMP    LABJUN RETURN, IF INVALID CHARACTER
176:
177: 1142 38              MATF  SEC     VALID INPUT PARAMETERS?
178: 1143 AD 65 1A        LDA  PARBL
179: 1146 ED 63 1A        SBC  PARAL  PARA < PARB?
180: 1149 AD 66 1A        LDA  PARBH
181: 114C ED 64 1A        SBC  PARAH
182: 114F 90 EE          BCC  MATD
183: 1151 20 E8 11        JSR  CRLF
184: 1154 A2 06          LDXIM $06
185:
186: 1156 20 F3 11      MATG  JSR  PRSP
187: 1159 CA              DEX
188: 115A D0 FA          BNE  MATG
189: 115C A0 00          LDYIM $00  RESET COLUMN COUNTER
190:
191: 115E 98              MATH  TYA
192: 115F 20 9B 12        JSR  PRNIBL OUTPUT COLUMNS
193: 1162 20 F3 11        JSR  PRSP
194: 1165 20 F3 11        JSR  PRSP
195: 1168 C8              INY
196: 1169 C0 10          CPYIM $10  PRINT COLUMNS 0...F
197: 116B D0 F1          BNE  MATH
198: 116D AD 63 1A        LDA  PARAL
199: 1170 85 FA          STAZ  POINTL SETUP MATRIX POINTER
200: 1172 AD 64 1A        LDA  PARAH
201: 1175 85 FB          STAZ  POINTH
202:
203: 1177 20 E8 11      MATJ  JSR  CRLF
204: 117A A2 10          LDXIM $10
205: 117C A5 FB          LDAZ  POINTH
206: 117E 20 8F 12        JSR  PRBYT  OUTPUT CURRENT MATRIX ADDRESS
207: 1181 A5 FA          LDAZ  POINTL
208: 1183 20 8F 12        JSR  PRBYT
209: 1186 A0 17          LDYIM $17
210: 1188 20 D9 11        JSR  ME
211:
212: 118B 38              MATK  SEC
213: 118C AD 65 1A        LDA  PARBL  HEXDUMP FINISHED?
214: 118F E5 FA          SBCZ  POINTL
215: 1191 AD 66 1A        LDA  PARBH
216: 1194 E5 FB          SBCZ  POINTH
217: 1196 B0 06          BCS  MATL
218: 1198 20 E8 11        JSR  CRLF
219: 119B 4C 5F 10        JMP  LABJUN  HEXDUMP IS FINISHED
220:
221: 119E A0 00          MATL  LDYIM $00
222: 11A0 B1 FA          LDAIY POINTL  FETCH CURRENT DATA BYTE
223: 11A2 20 8F 12        JSR  PRBYT  OUTPUT CURRENT DATA BYTE
224: 11A5 20 F3 11        JSR  PRSP
225: 11A8 20 13 12        JSR  INCPNT
226: 11AB CA              DEX
227: 11AC D0 DD          BNE  MATK
228: 11AE F0 C7          BEQ  MATJ
229:
230: 11B0 C9 47          GETTAP CMPIM 'G
231: 11B2 D0 0B          BNE  SAVID
232: 11B4 20 8A 14        JSR  GETID  READ DATA FROM TAPE SPEC. BY ID
233: 11B7 30 03          BMI  GETERR  ILLEGAL CHARACTER?
234: 11B9 4C 6A 10        JMP  RESALL  NORMAL EXIT
235:
236: 11BC 4C 5F 10      GETERR JMP  LABJUN
237:
238: 11BF C9 53          SAVID  CMPIM 'S
239: 11C1 D0 08          BNE  VALNUM  NO COMMAND KEY WAS DEPRESSED
240: 11C3 20 3B 14        JSR  SAID   STORE DATA ON TAPE SPECIFIED BY THE PARAMETERS

```

```

0241: 11C6 30 F4      BMI   GETERR ILLEGAL PARAMETER WAS ENTERED
0242: 11C8 4C 6A 10   JMP   RESALL
0243:
0244: 11CB 20 6F 12   VALNUM JSR   HEXNUM INPUT DATA TO BUFFER
0245: 11CE D0 03      BNE   VNA
0246: 11D0 4C 70 10   JMP   READCH
0247:
0248: 11D3 4C 6A 10   VNA    JMP   RESALL
0249:
0250: *****
0251: *** SUBROUTINES OF THE PRINTER MONITOR ***
0252: *****
0253:
0254:
0255: MESSY PRINTS A MESSAGE, POINTED BY Y REGISTER
0256:
0257: 11D6 20 E8 11   MESSY JSR   CRLF   PRINT A CR&LF
0258:
0259: 11D9 B9 BD 13   ME     LDAY   MESS   LOAD CHARACTERS
0260: 11DC C9 03      CMPIM  $03   ETX CHARACTER ?
0261: 11DE F0 07      BEQ    MESEND
0262: 11E0 20 34 13   JSR    PRCHA  CHARACTER TO TTY
0263: 11E3 C8        INY
0264: 11E4 4C D9 11   JMP    ME
0265:
0266: 11E7 60        MESEND RTS
0267:
0268:
0269: CRLF PRINT CARRIAGE RETURN & LINE FEED
0270: PRSP PRINT A SPACE
0271:
0272: 11E8 A9 0D      CRLF   LDAIM  $0D
0273: 11EA 20 34 13   JSR    PRCHA  OUTPUT CR
0274: 11ED A9 0A      LDAIM  $0A
0275:
0276: 11EF 20 34 13   CLEND  JSR    PRCHA  OUTPUT LF
0277: 11F2 60        RTS
0278:
0279: 11F3 A9 20      PRSP   LDAIM  $20
0280: 11F5 4C EF 11   JMP    CLEND  OUTPUT A SPACE
0281:
0282:
0283: PRBUFS OUTPUT ADDRESS AND DATA
0284:
0285: 11F8 20 E8 11   PRBUFS JSR   CRLF
0286: 11FB A5 FB      LDAZ   POINTH
0287: 11FD 20 8F 12   JSR    PRBYT  OUTPUT HIGH ORDER ADDR
0288: 1200 A5 FA      LDAZ   POINTL
0289: 1202 20 8F 12   JSR    PRBYT  OUTPUT LOW ORDER ADDR
0290: 1205 20 F3 11   JSR    PRSP
0291: 1208 A0 00      LDYIM  $00
0292: 120A B1 FA      LDAIY  POINTL  FETCH DATA FROM MEMORY
0293: 120C 20 8F 12   JSR    PRBYT
0294: 120F 20 F3 11   JSR    PRSP
0295: 1212 60        RTS
0296:
0297:
0298: INCPNT INCRMENT ADDR POINTER BY ONE
0299:
0300: 1213 E6 FA      INCPNT INCZ   POINTL
0301: 1215 D0 02      BNE    IP
0302: 1217 E6 FB      INCZ   POINTH
0303:
0304: 1219 60        IP     RTS
0305:
0306:
0307: DECPNT DECREMENT ADDR POINTER BY ONE
0308:
0309: 121A 38        DECPNT SEC

```

```

0310: 121B A5 FA      LDAZ POINTL 16 BIT SUBTRACTION
0311: 121D E9 01      SBCIM $01
0312: 121F 85 FA      STAZ POINTL
0313: 1221 A5 FB      LDAZ POINTH
0314: 1223 E9 00      SBCIM $00
0315: 1225 85 FB      STAZ POINTH
0316: 1227 60        RTS
0317:
0318:
0319:
0320:                SHOWPR SHOW THE CONTENTS OF THE P REGISTER
0321: 1228 A5 F1      SHOWPR LDAZ PREG
0322: 122A 8D 67 1A    STA PRTEMP
0323: 122D A2 08        LDXIM $08      BIT COUNTER
0324:
0325: 122F 0E 67 1A    SPRA ASL PRTEMP SHIFT OUT THE BITS
0326: 1232 90 09        BCC SPRB      IS IT A 0 OR 1 ?
0327: 1234 A9 01        LDAIM $01
0328: 1236 20 9B 12    JSR PRNIBL PRINT A '1'
0329: 1239 CA          DEX
0330: 123A D0 F3        BNE SPRA      ALL BITS PRINTED ?
0331: 123C 60        RTS
0332:
0333: 123D A9 00        SPRB LDAIM $00
0334: 123F 20 9B 12    JSR PRNIBL PRINT A '0'
0335: 1242 CA          DEX
0336: 1243 D0 EA        BNE SPRA      ALL BITS PRINTED ?
0337: 1245 60        RTS
0338:
0339:
0340:                JUNIOR PRINT 'JUNIOR'
0341:                EDITOR PRINT 'EDITOR'
0342:                ASSEM PRINT 'ASSEMBLER'
0343:
0344: 1246 A0 00        JUNIOR LDYIM $00
0345:
0346: 1248 20 D6 11    JUN JSR MESSY
0347: 124B 20 E8 11    JSR CRLF
0348: 124E 60        RTS
0349:
0350: 124F A0 07        EDITOR LDYIM $07
0351: 1251 4C 48 12    JMP JUN
0352:
0353: 1254 A0 0E        ASSEM LDYIM $0E
0354: 1256 4C 48 12    JMP JUN
0355:
0356:
0357:                RESET SUBROUTINES
0358:
0359: 1259 A0 00        RESPAR LDYIM $00
0360: 125B 8C 63 1A    STY PARAL
0361: 125E 8C 64 1A    STY PARAH
0362: 1261 8C 65 1A    STY PARBL
0363: 1264 8C 66 1A    STY PARBH
0364: 1267 60        RTS
0365:
0366: 1268 A0 00        RESIN LDYIM $00
0367: 126A 84 F8        STYZ INL
0368: 126C 84 F9        STYZ INH
0369: 126E 60        RTS
0370:
0371:                HEXNUM >CONVERT AN ASCII CHAR. TO A HEX NIBBLE
0372:                >SHIFT HEX DATA NIBBLE INTO BUFFER
0373:                >PRINT 'WHAT?' IF CHARACTER WAS NOT VALID
0374:                >RETURN WITH Z=1, IF VALID CHARACTER
0375:                >RETURN WITH N=1, IF NOT VALID CHARACTER
0376:
0377: 126F 20 1E 14    HEXNUM JSR ASHETT ASCII HEX CONVERSION
0378: 1272 30 10        BMI HNUB      NOT VALID ?

```

```

0379: 1274 A2 04          LDXIM $04      SET NIBBLE COUNTER
0380:
0381: 1276 06 F8      HNUA  ASLZ  INL
0382: 1278 26 F9          ROLZ  INH
0383: 127A CA          DEX
0384: 127B D0 F9          BNE  HNUA
0385: 127D 05 F8          ORAZ  INL      NIBBLE TO INPUT BUFFER
0386: 127F 85 F8          STAZ  INL
0387: 1281 A0 00          LDYIM $00      SET Z FLAG
0388: 1283 60          RTS
0389:
0390: 1284 A0 46      HNUB  LDYIM $46
0391: 1286 20 D6 11          JSR  MESSY  'WHAT?'
0392: 1289 20 E8 11          JSR  CRLF
0393: 128C A0 FF          LDYIM $FF      SET N FLAG
0394: 128E 60          RTS
0395:
0396:
0397:          PRBYT >CONVERT AN BYTE STORED IN ACCU TO
0398:          TWO ASCII CHARACTERS AND PRINT THEM
0399:
0400: 128F 48      PRBYT  PHA          SAVE BYTE
0401: 1290 4A          LSRA          GET UPPER NIBBLE
0402: 1291 4A          LSRA
0403: 1292 4A          LSRA
0404: 1293 4A          LSRA
0405: 1294 20 A4 12          JSR  NIBASC  NIBBLE TO ASCII CONVERSION
0406: 1297 20 34 13          JSR  PRCHA  PRINT UPPER NIBBLE
0407: 129A 68          PLA          GET BYTE AGAIN
0408:
0409: 129B 29 0F      PRNIBL ANDIM $0F      GET LOWER NIBBLE
0410: 129D 20 A4 12          JSR  NIBASC
0411: 12A0 20 34 13          JSR  PRCHA  PRINT LOWER NIBBLE
0412: 12A3 60          RTS
0413:
0414:
0415:          NIBASC >CONVERT A HEX DATA NIBBLE TO AN ASCII CHARACTER
0416:
0417: 12A4 C9 0A      NIBASC CMPIM $0A      NUMBER OR LETTER ?
0418: 12A6 18          CLC
0419: 12A7 30 02          BMI  NA
0420: 12A9 69 07          ADCIM $07
0421:
0422: 12AB 69 30      NA      ADCIM $30
0423: 12AD 60          RTS
0424:
0425:
0426:          RECCHA >RECEIVE 1 ASCII CHARACTER FROM PRINTER
0427:          >RETURN WITH ASCII CHARACTER IN ACCU
0428:          >SAVE X REGISTER
0429:
0430: 12AE 2C 80 1A      RECCHA BIT  PAD  WAIT FOR START BIT
0431: 12B1 30 FB          BMI  RECCHA
0432: 12B3 8E 61 1A          STX  TEMPB  SAVE INDEX X
0433: 12B6 A2 08          LDXIM $08      BIT COUNTER IS 8
0434: 12B8 20 03 13          JSR  DELHBT  DELAY HALF BIT TIME
0435:
0436: 12BB 20 12 13      RECA  JSR  DELBIT  DELAY ONE BIT TIME
0437:
0438: 12BE 2C 80 1A      RECD  BIT  PAD  ONE/ZERO CHECK
0439: 12C1 10 0A          BPL  RECB  BRANCH IF ZERO
0440: 12C3 38          SEC          BIT IS '1'
0441: 12C4 6E 62 1A          ROR  CHA  ROTATE CARRY INTO CHARACTER
0442: 12C7 CA          DEX          SET UP FOR NEXT BIT
0443: 12C8 D0 F1          BNE  RECA  ALL BITS READ?
0444: 12CA 4C D4 12          JMP  RECC
0445:
0446: 12CD 18      RECB  CLC          BIT IS '0'
0447: 12CE 6E 62 1A          ROR  CHA

```

```

0448: 12D1 CA          DEX
0449: 12D2 D0 E7      BNE RECA
0450:
0451: 12D4 20 12 13 RECC JSR DELBIT WAIT FOR STOP BIT TIME
0452: 12D7 AD 62 1A      LDA CHA
0453: 12DA 29 7F          ANDIM $7F BIT 7 MUST BE ZERO
0454: 12DC AE 61 1A      LDX TEMPB RESTORE INDEX X
0455: 12DF 60            RTS
0456:
0457:
0458: COMTIM >COMPUTE BIT TIME
0459:
0460: 12E0 18          COMTIM CLC
0461: 12E1 AD 5A 1A      LDA CNTL 16 BIT ADD
0462: 12E4 69 01          ADCIM $01
0463: 12E6 8D 5A 1A      STA CNTL
0464: 12E9 AD 5B 1A      LDA CNTH
0465: 12EC 69 00          ADCIM $00
0466: 12EE 8D 5B 1A      STA CNTH
0467: 12F1 2C 80 1A      BIT PAD START BIT FINISHED ?
0468: 12F4 10 EA          BPL COMTIM
0469: 12F6 AD 5A 1A      LDA CNTL SET UP FOR
0470: 12F9 8D 5E 1A      STA TIML HALF BIT TIME COMPUTATION
0471: 12FC AD 5B 1A      LDA CNTH
0472: 12FF 8D 5F 1A      STA TIMH
0473: 1302 60            RTS
0474:
0475:
0476: DELBIT >DELAY 1 BIT TIME
0477: DELHBT >DELAY 1/2 BIT TIME
0478:
0479: 1303 AD 5C 1A DELHBT LDA CNTHL FETCH 1/2 BIT TIME
0480: 1306 8D 5E 1A      STA TIML
0481: 1309 AD 5D 1A      LDA CNTHH
0482: 130C 8D 5F 1A      STA TIMH
0483: 130F 4C 1E 13      JMP CNTDN START WITH BIT COUNT DOWN
0484:
0485: 1312 AD 5A 1A DELBIT LDA CNTL FETCH 1 BIT TIME
0486: 1315 8D 5E 1A      STA TIML
0487: 1318 AD 5B 1A      LDA CNTH
0488: 131B 8D 5F 1A      STA TIMH
0489:
0490: 131E 38          CNTDN SEC
0491: 131F AD 5E 1A      LDA TIML 16 BIT SUBTRACTION
0492: 1322 E9 01          SBCIM $01 COUNT DOWN
0493: 1324 8D 5E 1A      STA TIML
0494: 1327 AD 5F 1A      LDA TIMH
0495: 132A E9 00          SBCIM $00
0496: 132C 8D 5F 1A      STA TIMH
0497: 132F EA          NOP EQUALIZE 4 MICRO SEC
0498: 1330 EA          NOP
0499: 1331 B0 EB          BCS CNTDN COUNT DOWN FINISHED ?
0500: 1333 60            RTS
0501:
0502: PRCHA >TRANSMIT AN ASCII CHARACTER STORED IN
0503: ACCU TO PRINTER
0504: >SAVE INDEX X
0505:
0506: 1334 8E 60 1A PRCHA STX TEMPA SAVE INDEX X
0507: 1337 8D 62 1A      STA CHA
0508: 133A AD 82 1A      LDA PBD
0509: 133D 29 FE          ANDIM $FE TRANSMIT START BIT
0510: 133F 8D 82 1A      STA PBD
0511: 1342 20 12 13      JSR DELBIT DELAY OF START BIT
0512: 1345 A2 07          LDXIM $07 SET UP FOR 7 DATA BITS
0513:
0514: 1347 4E 62 1A PRA LSR CHA SHIFT OUT CHARACTER
0515: 134A 90 30          BCC PRC BRANCH IF '0'
0516: 134C AD 82 1A      LDA PBD

```

```

0517: 134F 09 01      ORAIM $01  OUTPUT A LOG '1'
0518: 1351 8D 82 1A    STA  PBD
0519:
0520: 1354 20 12 13 PRB JSR  DELBIT DELAY 1 BIT TIME
0521: 1357 CA            DEX                SET UP FOR NEXT BIT
0522: 1358 D0 ED            BNE  PRA  ALL BITS TRANSMITTED ?
0523: 135A AE 59 1A      LDX  STPBIT X := AMOUNT OF STOP BITS + 1
0524:
0525: 135D AD 82 1A PRD  LDA  PBD
0526: 1360 09 01      ORAIM $01  FIRST NONE PARITY
0527: 1362 8D 82 1A    STA  PBD  AND THEN 1 STOP BIT
0528: 1365 20 12 13    JSR  DELBIT
0529: 1368 CA            DEX
0530: 1369 D0 F2            BNE  PRD
0531: 136B 2C 80 1A      BIT  PAD  TEST FOR BREAK
0532: 136E 10 04      BPL  BRKTST
0533: 1370 AE 60 1A      LDX  TEMPX RESTORE INDEX X
0534: 1373 60            RTS
0535:
0536: 1374 2C 80 1A BRKTST BIT  PAD  KEY RELEASED ?
0537: 1377 10 FB      BPL  BRKTST
0538: 1379 6C 7C 1A      JMI  BRKT  JUMP TO AN USER SELECTABLE VECTOR
0539:
0540: 137C AD 82 1A PRC  LDA  PBD
0541: 137F 29 FE      ANDIM $FE  OUTPUT A LOG '0'
0542: 1381 8D 82 1A    STA  PBD
0543: 1384 4C 54 13      JMP  PRB
0544:
0545:                INPAR >PARAMETER INPUT OF MATRIX
0546:
0547: 1387 20 AE 12 INPAR JSR  RECCHA WAIT FOR A CHARACTER
0548: 138A C9 2C      CMPIM ' '  IS IT A COLON ?
0549: 138C F0 07      BEQ  IPA
0550: 138E 20 6F 12    JSR  HEXNUM FILLUP INPUT BUFFER
0551: 1391 30 29      BMI  IPD  RETURN, IF NOT VALID
0552: 1393 F0 F2      BEQ  INPAR ELSE CONTINUE
0553:
0554: 1395 A5 F8      IPA  LDAZ INL  INPUT TO PARAMETER BUFFER
0555: 1397 8D 63 1A    STA  PARAL
0556: 139A A5 F9      LDAZ INH
0557: 139C 8D 64 1A    STA  PARAH
0558: 139F 20 68 12    JSR  RESIN
0559:
0560: 13A2 20 AE 12 IPB  JSR  RECCHA WAIT FOR A CHARACTER
0561: 13A5 C9 0D      CMPIM $0D  WAS IT A CARRIAGE RETURN ?
0562: 13A7 F0 07      BEQ  IPC
0563: 13A9 20 6F 12    JSR  HEXNUM
0564: 13AC 30 0E      BMI  IPD  VALID CHARACTER ?
0565: 13AE F0 F2      BEQ  IPB
0566:
0567: 13B0 A5 F9      IPC  LDAZ INH  INPUT TO PARAMETER BUFFER
0568: 13B2 8D 66 1A    STA  PARBH
0569: 13B5 A5 F8      LDAZ INL
0570: 13B7 8D 65 1A    STA  PARBL
0571: 13BA A0 00      LDYIM $00
0572:
0573: 13BC 60          IPD  RTS
0574:
0575:
0576:                STRING LOOKUP TABLE
0577:
0578: 13BD 4A          MESS = 'J' Y = 00
0579: 13BE 55          = 'U'
0580: 13BF 4E          = 'N'
0581: 13C0 49          = 'I'
0582: 13C1 4F          = 'O'
0583: 13C2 52          = 'R'
0584: 13C3 03          = $03
0585: 13C4 45          = 'E' Y = 07

```

0586:	13C5	44	=	'D	
0587:	13C6	49	=	'I	
0588:	13C7	54	=	'T	
0589:	13C8	4F	=	'O	
0590:	13C9	52	=	'R	
0591:	13CA	03	=	\$03	
0592:	13CB	41	=	'A	Y = 0E
0593:	13CC	53	=	'S	
0594:	13CD	53	=	'S	
0595:	13CE	45	=	'E	
0596:	13CF	4D	=	'M	
0597:	13D0	03	=	\$03	
0598:	13D1	41	=	'A	Y = 14
0599:	13D2	43	=	'C	
0600:	13D3	43	=	'C	
0601:	13D4	3A	=	:'	Y = 17
0602:	13D5	20	=	:'	
0603:	13D6	03	=	\$03	
0604:	13D7	59	=	'Y	Y = 1A
0605:	13D8	20	=	:'	
0606:	13D9	20	=	:'	
0607:	13DA	3A	=	:'	
0608:	13DB	20	=	:'	
0609:	13DC	03	=	\$03	
0610:	13DD	58	=	'X	Y = 20
0611:	13DE	20	=	:'	
0612:	13DF	20	=	:'	
0613:	13E0	3A	=	:'	
0614:	13E1	20	=	:'	
0615:	13E2	03	=	\$03	
0616:	13E3	50	=	'P	Y = 26
0617:	13E4	43	=	'C	
0618:	13E5	20	=	:'	
0619:	13E6	3A	=	:'	
0620:	13E7	20	=	:'	
0621:	13E8	03	=	\$03	
0622:	13E9	53	=	'S	Y = 2C
0623:	13EA	50	=	'P	
0624:	13EB	20	=	:'	
0625:	13EC	3A	=	:'	
0626:	13ED	20	=	:'	
0627:	13EE	03	=	\$03	
0628:	13EF	50	=	'P	Y = 32
0629:	13F0	52	=	'R	
0630:	13F1	20	=	:'	
0631:	13F2	3A	=	:'	
0632:	13F3	20	=	:'	
0633:	13F4	03	=	\$03	
0634:	13F5	20	=	:'	Y = 38
0635:	13F6	20	=	:'	
0636:	13F7	20	=	:'	
0637:	13F8	20	=	:'	
0638:	13F9	20	=	:'	
0639:	13FA	4E	=	'N	
0640:	13FB	56	=	'V	
0641:	13FC	20	=	:'	
0642:	13FD	42	=	'B	
0643:	13FE	44	=	'D	
0644:	13FF	49	=	'I	
0645:	1400	5A	=	'Z	
0646:	1401	43	=	'C	
0647:	1402	03	=	\$03	
0648:	1403	57	=	'W	Y = 46
0649:	1404	48	=	'H	
0650:	1405	41	=	'A	
0651:	1406	54	=	'T	
0652:	1407	3F	=	'?	
0653:	1408	03	=	\$03	
0654:	1409	52	=	'R	Y = 4C


```

0655: 140A 45      =      'E
0656: 140B 41      =      'A
0657: 140C 44      =      'D
0658: 140D 59      =      'Y
0659: 140E 03      =      $03
0660: 140F 48      =      'H      Y = 52
0661: 1410 45      =      'E
0662: 1411 58      =      'X
0663: 1412 44      =      'D
0664: 1413 55      =      'U
0665: 1414 4D      =      'M
0666: 1415 50      =      'P
0667: 1416 3A      =      ':'
0668: 1417 20      =
0669: 1418 03      =      $03
0670: 1419 53      =      'S      Y = 5C
0671: 141A 41      =      'A
0672: 141B 3A      =      ':'
0673: 141C 20      =
0674: 141D 03      =      $03
0675:
0676:      ASHETT = ASCHEX
0677:
0678:
0679:      CONVERT AN ASCII CHARACTER TO A HEX DATA NIBBLE.
0680:
0681:      1) RETURN WITH CONVERTED HEX NUMBER IN ACCU
0682:      2) N = 1, IF NOT VALID HEX NUMBER
0683:      3) Z = 1, IF VALID HEX NUMBER
0684:
0685: 141E C9 30      ASHETT CMPIM $30      IGNORE 00...2F
0686: 1420 30 0C      BMI      NOTVAT
0687: 1422 C9 3A      CMPIM $3A
0688: 1424 30 0B      BMI      VALIT
0689: 1426 C9 41      CMPIM $41      IGNORE 3A...40
0690: 1428 30 04      BMI      NOTVAT
0691: 142A C9 47      CMPIM $47      IGNORE 47...7F
0692: 142C 30 03      BMI      VALIT
0693:
0694: 142E A0 FF      NOTVAT LDYIM $FF      SET N-FLAG
0695: 1430 60      RTS      ERROR EXIT
0696:
0697: 1431 C9 40      VALIT CMPIM $40      ASCII HEX CONVERSION
0698: 1433 30 03      BMI      VALT
0699: 1435 18      CLC
0700: 1436 69 09      ADCIM $09
0701:
0702: 1438 29 0F      VALT ANDIM $0F
0703: 143A 60      RTS
0704:
0705: 143B 20 AE 12   SAID JSR      RECCHA WAIT FOR A CHARACTER
0706: 143E C9 2C      CMPIM      ,
0707: 1440 F0 07      BEQ      SIC      IT WAS A DELIMITER
0708: 1442 20 6F 12   JSR      HEXNUM READ PARAMETER = ID
0709: 1445 30 3F      BMI      SIB
0710: 1447 F0 F2      BEQ      SAID
0711:
0712: 1449 A5 F8      SIC LDAZ INL      SAVE ID
0713: 144B 8D 79 1A   STA      ID
0714: 144E C9 00      CMPIM $00      ID = 00 & FF ARE NOT VALID
0715: 1450 F0 35      BEQ      SIA
0716: 1452 C9 FF      CMPIM $FF
0717: 1454 F0 31      BEQ      SIA
0718: 1456 20 68 12   JSR      RESIN RESET INPUT BUFFER
0719: 1459 20 87 13   JSR      INPAR READ SA AND EA
0720: 145C 30 28      BMI      SIB NOT VALID CHARACTER
0721: 145E AD 63 1A   LDA      PARAL SAVE ALL PARAMETERS
0722: 1461 8D 70 1A   STA      SAL
0723: 1464 AD 64 1A   LDA      PARAH

```

```

0724: 1467 8D 71 1A      STA  SAH
0725: 146A AD 65 1A      LDA  PARBL
0726: 146D 8D 72 1A      STA  EAL
0727: 1470 AD 66 1A      LDA  PARBH
0728: 1473 8D 73 1A      STA  EAH
0729: 1476 20 DF 09      JSR  DUMP      STORE DATA ON TAPE
0730: 1479 20 BC 14      JSR  RESTTY   I/O RESET
0731: 147C A0 4C          LDYIM $4C      'READY'
0732: 147E 20 D6 11      JSR  MESSY
0733: 1481 20 E8 11      JSR  CRLF
0734: 1484 A0 00          LDYIM $00      NORMAL EXIT
0735:
0736: 1486 60              SIB  RTS
0737:
0738: 1487 A0 FF          SIA  LDYIM $FF      ERROR EXIT
0739: 1489 60              RTS
0740:
0741:
0742: 148A 20 A2 13      GETID JSR  IPB      READ ID
0743: 148D 30 17          BMI  GA      NOT VALID PARAMETER
0744: 148F 8D 79 1A      STA  ID      SAVE ID
0745: 1492 C9 FF          CMPIM $FF     SPECIAL ID ?
0746: 1494 F0 11          BEQ  IDPAR
0747:
0748: 1496 20 02 0B      GB  JSR  RDTAPE  READ DATA FROM TAPE
0749: 1499 20 BC 14      JSR  RESTTY   I/O RESET
0750: 149C A0 4C          LDYIM $4C      'READY'
0751: 149E 20 D6 11      JSR  MESSY
0752: 14A1 20 E8 11      JSR  CRLF
0753: 14A4 A0 00          LDYIM $00      NORMAL EXIT
0754:
0755: 14A6 60              GA  RTS
0756:
0757: 14A7 A0 5C          IDPAR LDYIM $5C      'SA: '
0758: 14A9 20 D6 11      JSR  MESSY
0759: 14AC 20 EB 14      JSR  IPBRES  READ A SPECIAL ID
0760: 14AF 30 F5          BMI  GA      NOT VALID PARAMETER
0761: 14B1 8D 70 1A      STA  SAL      SAVE START ADDRESS
0762: 14B4 A5 F9          LDAZ  INH
0763: 14B6 8D 71 1A      STA  SAH
0764: 14B9 4C 96 14      JMP  GB
0765:
0766:
0767:
0768: 14BC A9 67          RESTTY LDAIM $67
0769: 14BE 8D 82 1A      STA  PBD
0770: 14C1 A9 00          LDAIM $00
0771: 14C3 8D 80 1A      STA  PAD
0772: 14C6 A9 7F          LDAIM $7F
0773: 14C8 8D 81 1A      STA  PADD
0774: 14CB 8D 83 1A      STA  PBDD
0775: 14CE 60              RTS
0776: *****
0777: *** STEP BY STEP PROGRAM OF THE PRINTER MONITOR ***
0778: *****
0779:
0780:
0781: > THE NMI VECTOR FOR STEP BY STEP IS SET AUTOMATICALLY
0782: > STEP SWITCH: ON POSITION
0783: > K4 AND K5 DISABLE THE SYNC SIGNAL OF THE PROCESSOR
0784: > A HARDWARE MODIFICATION IS REQUIRED (SEE BOOK 3)
0785:
0786:
0787: 14CF 85 F3          STEP STAZ  ACC      SAVE ACCU
0788: 14D1 68              PLA      GET PREG
0789: 14D2 85 F1          STAZ  PREG
0790: 14D4 68              PLA      GET PCL
0791: 14D5 85 EF          STAZ  PCL
0792: 14D7 85 FA          STAZ  POINTL PC OF THE NEXT INSTRUCTION

```

```

0793: 14D9 68          PLA          GET PCH
0794: 14DA 85 F0       STAZ PCH
0795: 14DC 85 FB       STAZ POINTH
0796: 14DE 84 F4       STYZ YREG
0797: 14E0 86 F5       STXZ XREG
0798: 14E2 BA          TSX          GET OLD STACK POINTER
0799: 14E3 86 F2       STXZ SPUSER AND SAVE IT
0800: 14E5 20 F8 11    JSR PRBUFS OPEN NEXT CELL
0801: 14E8 4C 6A 10    JMP RESALL BACK TO MONITOR
0802:
0803:
0804:
0805:          *** SPECIAL ID ***
0806:
0807: 14EB A9 00       IPBRES LDAIM $00
0808: 14ED 85 F8       STAZ INL      RESET INPUT BUFFER
0809: 14EF 85 F9       STAZ INH
0810: 14F1 4C A2 13    JMP IPB      CONTINUE
0811:
0812:
0813:          *****
0814:          *** END OF THE PROGRAM ***
0815:          *****
ID=

```

-T

SYMBOL TABLE 3000 3594			
ACC	00F3	ADJPNT	0C72
ASSEM	1254	BEGADH	00E3
BITS	1A75	BRKT	1A7C
BYTE	1A6A	CENDH	00E9
CHARVU	0C5D	CHA	1A62
CHKID	0B9E	CHKL	1A6E
CNTA	1AF4	CNTC	1AF6
CNTHH	1A5D	CNTHL	1A5C
COMPNT	0949	COMTIM	12E0
DATATR	0A40	DAT	089C
DELHBT	1303	DELY	0927
DUMPT	09F3	EAH	1A73
ENDADH	00E5	ENDADL	00E4
FIRST	1A76	FMA	0B84
GB	1496	GETA	0800
GETID	148A	GETKEY	1DF9
HEXDAT	0A6A	HEXNUM	126F
HIG	0ACB	HNUA	1276
IDPAR	14A7	INCPNT	1213
INL	00F8	INPAR	1387
IPBRES	14EB	IPB	13A2
JUNIOR	1246	JUN	1248
LIST	10C9	LO	0AE8
MATD	113F	MATF	1142
MATJ	1177	MATK	118B
ME	11D9	MESEND	11E7
MINUS	1080	NA	12AB
NIB	0AA1	NMI	1A7A
ONE	0AA8	OUTBT	0A8B
PADD	1A81	PAD	1A80
PARBH	1A66	PARBL	1A65
PC	111F	PCH	00F0
PLU	1073	PNT	109F
PRA	1347	PRBUFS	11F8
PRCHA	1334	PRC	137C
PRNIBL	129B	PRSP	11F3
RBB	0BFB	RDBA	0BD1
RDBIT	0BC2	RDBYT	0BF3
RDSA	0B60	RDTAPE	0B02
READCH	1070	RECA	12BB
RECCHA	12AE	RECD	12BE
RESIN	1268	RESPAR	1259
SAH	1A71	SAID	143B
SAVID	11BF	SBEGH	0989
SEAL	097A	SECOND	1A77
SHIFT	08A3	SHOWPR	1228
SIC	1449	SID	093B
SPRB	123D	SPUSER	00F2
STAR	0B45	STARA	0B55
STEAH	08D4	STEAL	08E0
STEP	14CF	STPBIT	1A59
STSAL	08C8	SY	1A69
SYNCNT	1A74	SYNCS	0A1F
TDISP	091B	TEMP	00FC
TEMPX	00FD	TENSYN	0B36
TLOOK	09BB	TPINIT	0810
TPVEC	0918	TTXT	0833
VALNUM	11CB	VALT	1438
VU	0C64	WARMST	1CCA
ZERO	0AB8	ZRO	0AC1
		ASCHEX	0C19
		BEGADL	00E2
		BRKTST	1374
		CENDL	00E8
		CHECK	0B8A
		CHKSUM	0C4B
		CNTDN	131E
		CNTL	1A5A
		CONTIN	111C
		DECPNT	121A
		DISCNT	1A68
		EAL	1A72
		FILES	0873
		GA	14A6
		GETB	084E
		GETTAP	11B0
		HIGH	0AC8
		HNUB	1284
		INH	00F9
		IP	1219
		IPC	13B0
		LABJUN	105F
		LOWER	1A6D
		MATG	1156
		MATL	119E
		MESS	13BD
		NIBASC	12A4
		NOTVAL	0C29
		OUTBTC	0A7A
		PARAH	1A64
		PBDD	1A83
		PCL	00EF
		POINTH	00FB
		PRBYT	128F
		PRD	135D
		PRTEMP	1A67
		RDBB	0BD4
		RDCH	0C36
		RDTDIS	1AD4
		RECB	12CD
		RESALL	106A
		RESTTY	14BC
		SAL	1A70
		SBEGL	0997
		SENDH	09A5
		SIA	1487
		SPACE	108D
		SSAH	094D
		STBEGH	08EC
		STENDH	0902
		STRBT	1039
		SYNC	0B1C
		SYNVEC	0B9B
		TEMPA	1A60
		TIMH	1A5F
		TPI	0821
		VALID	0C2C
		VAL	0C33
		XREG	00F5
		ASHETT	141E
		BEGIN	1ED3
		BTWEEN	0BE8
		CHAR	1A6B
		CHKH	1A6F
		CLEND	11EF
		CNTH	1A5B
		COLDST	1CB5
		CRLF	11E8
		DELBIT	1312
		DUMP	09DF
		EDITOR	124F
		FILMEM	0B70
		GANG	1A78
		GETERR	11BC
		GET	0840
		HIGHER	1A6C
		ID	1A79
		INITPR	1000
		IPA	1395
		IPD	13BC
		LDAINH	1DA7
		LOW	0AE5
		MATH	115E
		MATRIX	1131
		MESSY	11D6
		NIBOUT	0A9A
		NOTVAT	142E
		OUTCH	0AA3
		PARAL	1A63
		PBD	1A82
		PLUS	085E
		POINTNL	00FA
		PRB	1354
		PREG	00F1
		RBA	0BFA
		RDBC	0BE4
		RDFLAG	1AD5
		READ	0C38
		RECC	12D4
		RESET	1C1D
		RUN	10B2
		SAVE	0852
		SEAH	096B
		SENDL	09B3
		SIB	1486
		SPRA	122F
		SSAL	095C
		STBEGL	08F7
		STENDL	090D
		STSAH	08BC
		SYNCA	0B21
		TAPDIS	0936
		TEMPB	1A61
		TIML	1A5E
		TPTXT	082E
		VALIT	1431
		VNA	11D3
		YREG	00F4