SIGNETICS PIPBUG M20
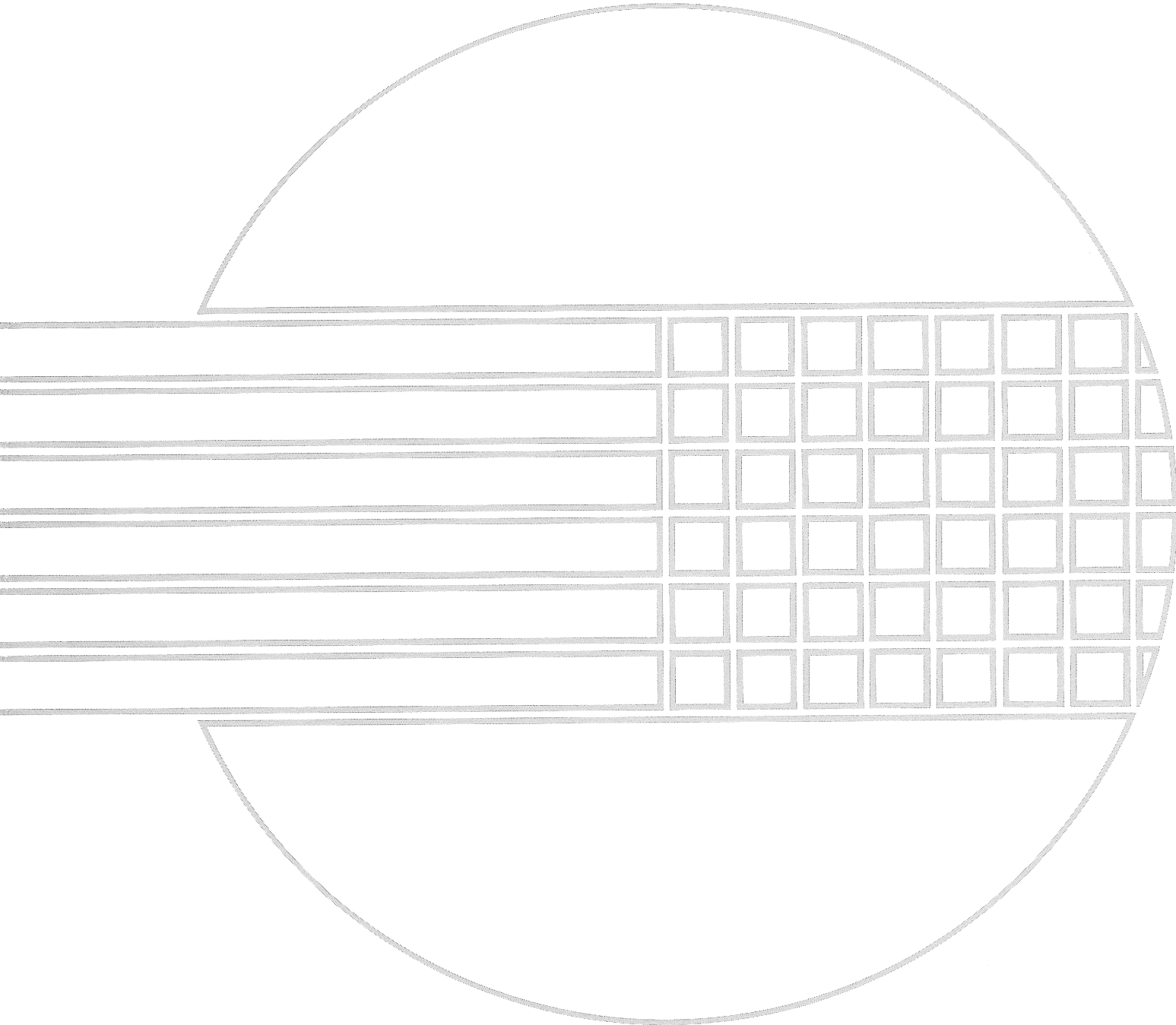
INTRODUCTION

The PIPBUG program is provided as part of the 2650 PC1001 and ABC1500 microcomputer boards, so that the user has immediately available to him the tools necessary to run programs on the 2650 microprocessor. Features include support of a user terminal, papertape load and dump, memory examine and alter, and breakpoints. The 2650 PC1001 card itself is described in detail in applications memo M14, while the ABC1500 is described in applications memo M19.

DESCRIPTION

The PIPBUG program is started by pressing the reset button on the card. It outputs the user prompt character of '*'. A command is then entered, starting with an alpha character indicating the operation wanted, followed by any required parameters separated by spaces. A carriage return terminates the command input. The parameters must be given as hexadecimal numbers. Leading zeros are unnecessary. For example, '008F' and '8F' are the same address. The error message for an illegal command or parameter is '?', after which the user can enter a new command line. The delete key can be used to delete the previous character.

The program fits in the first 1K bytes of memory. Also, the 63 bytes of RAM from location 1024_{10} to 1087_{10} are required for buffers and temporary storage. The program uses subroutines with a maximum nested depth of three.

In the explanations of the commands CR means the carriage return key and LF means the line feed key. The symbol ƀ means there must be at least one space.

COMMANDS
Command Summary

A Alter memory
B Set Breakpoint
C Clear Breakpoint
D Dump memory to papertape
G Goto address
L Load memory from papertape
S See and alter registers

1. Alter Memory   Aaaaa   CR
   Action: Outputs aaaabcc where 'aaaa' is a memory location and 'cc' is its content. User can respond with:
   1) CR which ends the command
   2) LF which will display the next memory location
   3) nn CR which will replace 'cc' by 'nn' at location 'aaaa' and end the command
   4) nn LF which will replace 'cc' by 'nn' and then display the next location.

2. Load from Papertape   L   CR
   Action: Will start reading papertape expecting blocks of data in the hex object format (see applications memo M21). In case of illegal characters, a BCC error, or a length error, the papertape will be stopped and the command ended with the standard error message.

   At the end of a successful load, control is passed to the address in the EOF block. This would usually be back to the PIPBUG program.

3. Dump to Papertape   Dssssƀeeee CR
   Action: Will punch a leader of 50 blanks and then output the contents of locations 'ssss' to 'eeee', inclusive, in hex object format. When done, the EOF block and a trailer of 50 blanks are punched.

4. See and Set Microprocessor Registers   Sn   CR
   Action: The parameter 'n' is in the range 0 to 8 and selects a particular register;
   0 = register 0
   1 = register 1 bank #0
   2 = register 2 bank #0
   3 = register 3 bank #0
   4 = register 1 bank #1
   5 = register 2 bank #1
   6 = register 3 bank #1
   7 = PSW upper
   8 = PSW lower

   The contents of the selected register will be displayed. The user can respond with:
   1) CR which ends the command
   2) LF which displays the next register's content
   3) nn CR which resets the register to 'nn' and ends the command
   4) nn LF which resets the register to 'nn' and displays the next register's content

5. Go To   Gaaaa   CR
   Action: Control will be transferred to location 'aaaa' after restoring the register contents.

6. Set Breakpoint   Biƀaaaa CR
   Action: Will set the ith breakpoint (i = 1 or 2) at the address 'aaaa.'

7. Clear Breakpoint   Ci   CR
   Action: Will clear the ith breakpoint. If the ith breakpoint is not set, gives error message.

BREAKPOINTS

Breakpoints provide a means to get a "snapshot" of the program and microprocessor's status immediately prior to executing the instruction at the breakpoint address. PIPBUG allows two breakpoints to be set.

Setting a breakpoint at location '1053' with the command 'B1 1053' causes the two bytes of program at '1053' and '1054' to be stored in a table in PIPBUG's RAM area. They are replaced by the two byte instruction 'ZBRR *VEC'. At location VEC in the program is the address of the 1st breakpoint handling routine. There is a separate routine for the 2nd breakpoint.

When the user program executes the instruction at location '1053', the ZBRR instruction jumps to the breakpoint routine. This routine first saves the microprocessor registers, then restores the two bytes of user program to locations '1053' and '1054', prints the breakpoint address '1053', and finally jumps to PIPBUG. Now the user can use the See command to examine the microprocessor registers.

Since the breakpoints are software implemented and are cleared when reached, there will not be another breakpoint when the user program is re-executed. It must be explicitly re-set with the Set Breakpoint command. Breakpoints will remain in memory until executed or explicitly cleared with the Clear Breakpoint command.

SUGGESTIONS FOR USING PIPBUG

After having written and assembled a program, the user has a papertape containing the object code for the program. The Load command is used to read the code into the RAM of the prototyping card. In the operand field of the END directive of the program, the user should put blanks or a zero, so that after reading the tape PIPBUG restarts itself.

Most commonly the loaded program is still under development. The user wants to run and test only parts of the program. He can use the Goto and Breakpoint commands to isolate the particular code sequence. The two breakpoints can be set at the normal and error exits of the code. Using the Goto command the user then transfers control to the starting address of the code. Remember that the microprocessor's registers can be pre-set using the See command.

If there is a bug, the user can make machine language patches to the program with the Alter command. Great care should be taken when doing this, since assemblers are more methodical than people. The Dump command can be used to save on papertape the program and all patches so that the debugging can be continued at some later time.

APPENDIX

```
PIP ASSEMBLER VERSION 3 LEVEL 1                                    PAGE    1

LINE ADDR B1 B2 B3 B4 ERR SOURCE

    1 0001                 P       EQU     1
    2 0002                 N       EQU     2
    3 0000                 Z       EQU     0
    4 0002                 LCOM    EQU     H'02'           LOGICAL COMPARE
    5 0001                 CAR     EQU     H'01'           CARRY
    6 0080                 SENS    EQU     H'80'           SENSE
    7 0040                 FLAG    EQU     H'40'           FLAG
    8 0020                 II      EQU     H'20'           INTERRUPT INHIB
    9 0020                 IDC     EQU     H'20'           INTER DIGIT CAR
   10 0004                 OVF     EQU     H'04'           OVEFFLOW
   11 0000                 R0      EQU     0
   12 0001                 R1      EQU     1
   13 0002                 R2      EQU     2
   14 0003                 R3      EQU     3
   15 0003                 UN      EQU     3
   16 0000                 EQ      EQU     0
   17 0002                 LT      EQU     2
   18 0001                 GT      EQU     1
   19 0008                 WC      EQU     H'08'
   20 0010                 RS      EQU     H'10'
   21 0020                 SPAC    EQU     H'20'
   22 0001                 BMAX    EQU     1               NO. BKPTS - 1
   23 007F                 DELE    EQU     H'7F'
   24 000D                 CR      EQU     13
   25 000A                 LF      EQU     10
   26 0014                 BLEN    EQU     20
   27 003A                 STAR    EQU     A':'
   28                      *
   29                              ORG     0
   30 0000 07 3F           INIT    LODI.R3 63              ZERO MARK VECTOR AND O
   31 0002 20                      EORZ    R0
   32 0003 CF 44 00        AINI    STRA.R0 COM.R3.-
   33 0006 5B 7B                   BRNR.R3 AINI
   34 0008 04 77                   LODI.R0 H'77'
   35 000A CC 04 09                STRA.R0 XGOT            LOAD THE RAM CODE TO S
   36 000D 04 1B                   LODI.R0 H'1B'
   37 000F CC 04 0B                STRA.R0 XGOT+2
   38 0012 04 80                   LODI.R0 H'80'
   39 0014 CC 04 0C                STRA.R0 XGOT+3
   40 0017 1B 09                   BCTR.UN MBUG
   41 0019 01 60           VEC     ACON    BK01            BREAKPOINT VECTOR
   42 001B 01 6E                   ACON    BK02
   43                      *
   44                      * COMMAND HANDLER
   45 001D 04 3F           EBUG    LODI.R0 A'?'            ERROR RETURN FOR ALL R
   46 001F 3F 02 B4                BSTA.UN COUT
   47 0022 75 FF           MBUG    CPSL    H'FF'           START OF CMD LOOP. RES
   48 0024 3F 00 8A                BSTA.UN CRLF
   49 0027 04 2A                   LODI.R0 A'*'
   50 0029 3F 02 B4                BSTA.UN COUT
   51 002C 3B 2D                   BSTR.UN LINE            DONT CARE IF THERE IS
   52 002E 20                      EORZ    R0
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   2

```
LINE ADDR B1 B2 B3 B4 ERR SOURCE
  53 002F CC 04 27          STRA.R0     BPTR
  54 0032 0C 04 13          LODA.R0     BUFF
  55 0035 E4 41             COMI.R0     A'A'
  56 0037 1C 00 AB           BCTA.EQ     ALTE
  57 003A E4 42             COMI.R0     A'B'
  58 003C 1C 01 E5           BCTA.EQ    BKPT
  59 003F E4 43             COMI.R0      A'C'
  60 0041 1C 01 CA          BCTA.EQ    CLR
  61 0044 E4 44             COMI.R0     A'D'
  62 0046 1C 03 10           BCTA.EQ    DUMP
  63 0049 E4 47             COMI.R0      A'G'
  64 004B 1C 01 3A           BCTA.EQ    GOTO
  65 004E E4 4C             COMI.R0     A'L'
  66 0050 1C 03 B5           BCTA.EQ     LOAD
  67 0053 E4 53             COMI.R0     A'S'
  68 0055 1C 00 F4           BCTA.EQ    SREG
  69 0058 1F 00 1D          BCTA.UN    EBUG
  70                        * INPUT A CMD LINE INTO BUFFER
  71                        * CODE IS 1=CR  2=LF  3=MSG+CR  4=MSG+LF
  72 005B 07 FF      LINE   LODI.R3     -1
  73 005D CF 04 27          STRA.R3     BPTR
  74 0060 E7 14      LLIN   COMI.R3     BLEN
  75 0062 18 19             BCTR.EQ     ELIN       ON BUFFER OVERFLOW FOR
  76 0064 3F 02 86          BSTA.UN     CHIN       GET CHAR
  77 0067 E4 7F             COMI.R0     DELE
  78 0069 98 0E             BCFR.EQ     ALIN
  79 006B E7 FF             COMI.R3     -1         ECHO AND BACK PTR
  80 006D 18 71             BCTR.EQ     LLIN
  81 006F 0F 64 13          LODA.R0     BUFF,R3
  82 0072 3F 02 B4          BSTA.UN     COUT
  83 0075 A7 01             SUBI.R3     1
  84 0077 1B 67             BCTR.UN     LLIN
  85 0079 E4 0D      ALIN   COMI.R0     CR
  86 007B 98 18             BCFR.EQ     BLIN
  87 007D 05 01      ELIN   LODI.R1     1
  88 007F 03        CLIN   LODZ        R3
  89 0080 1A 02             BCTR.N      DLIN
  90 0082 85 02             ADDI.R1     2
  91 0084 CD 04 2A   DLIN   STRA.R1     CODE
  92 0087 CF 04 29          STRA.R3     CNT
  93 008A 04 0D     CRLF   LODI.R0     CR
  94 008C 3F 02 B4          BSTA.UN     COUT
  95 008F 04 0A             LODI.R0     LF
  96 0091 3F 02 B4          BSTA.UN     COUT
  97 0094 17               RETC.UN
  98 0095 05 02     BLIN   LODI.R1     2
  99 0097 E4 0A             COMI.R0     LF
 100 0099 18 64             BCTR.EQ     CLIN
 101 009B CF 24 13          STRA.R0     BUFF,R3,+   STROE CHAR AND ECHO
 102 009E 3F 02 B4          BSTA.UN     COUT
 103 00A1 1F 00 60          BCTA.UN     LLIN
 104                        *
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   3

LINE ADDR B1 B2 B3 B4 ERR SOURCE

```
105                              * SUBR THAT STORES DOUBLE PRECISION INTO TEMP
106 00A4 CD 04 0D         STRT    STRA.R1      TEMP
107 00A7 CE 04 0E                 STRA.R2      TEMP+1
108 00AA 17                       RETC.UN
109                              * DISPLAY AND ALTER MEMORY
110 00AB 3F 02 DB         ALTE    BSTA.UN      GNUM
111 00AE 3B 74            LALT    BSTR.UN      STRT
112 00B0 3F 02 69                 BSTA.UN      BOUT
113 00B3 0D 04 0E                 LODA.R1      TEMP+1
114 00B6 3F 02 69                 BSTA.UN      BOUT
115 00B9 3F 03 5B                 BSTA.UN      FORM
116 00BC 0D 84 0D                 LODA.R1      *TEMP          DISPLAY CONTENT
117 00BF 3F 02 69                 BSTA.UN      BOUT
118 00C2 3F 03 5B                 BSTA.UN      FORM
119 00C5 3F 00 5B                 BSTA.UN      LINE
120 00C8 0C 04 2A                 LODA.R0      CODE
121 00CB E4 02                    COMI.R0      2
122 00CD 1E 00 22                 BCTA.LT      MBUG
123 00D0 18 11                    BCTR.EQ      DALT
124 00D2 CC 04 11         CALT    STRA.R0      TEMR
125 00D5 3F 02 DB                 BSTA.UN      GNUM
126 00D8 CE 84 0D                 STRA.R2      *TEMP          UPDATE CONTENTS
127 00DB 0C 04 11                 LODA.R0      TEMR
128 00DE E4 04                    COMI.R0      4
129 00E0 9C 00 22                 BCFA.EQ      MBUG
130 00E3 06 01            DALT    LODI.R2      1              INCR CURRENT ADDRESS
131 00E5 8E 04 0E                 ADDA.R2      TEMP+1
132 00E8 05 00                    LODI.R1      0
133 00EA 77 08                    PPSL         WC
134 00EC 8D 04 0D                 ADDA.R1      TEMP
135 00EF 75 08                    CPSL         WC
136 00F1 1F 00 AE                 BCTA.UN      LALT
137                              * SELECTIVELY DISPLAY AND ALTER REGISTERS
138 00F4 3F 02 DB         SREG    BSTA.UN      GNUM           GET INDEX OF REG
139 00F7 E6 08            LSRE    COMI.R2      8              CHECK RANGE
140 00F9 1D 00 1D                 BCTA.GT      EBUG
141 00FC CE 04 11                 STRA.R2      TEMR
142 00FF 0E 64 00                 LODA.R0      COM,R2         DISPLAY CONTENTS
143 0102 C1                       STRZ         R1
144 0103 3F 02 69                 BSTA.UN      BOUT
145 0106 3F 03 5B                 BSTA.UN      FORM
146 0109 3F 00 5B                 BSTA.UN      LINE
147 010C 0C 04 2A                 LODA.R0      CODE
148 010F E4 02                    COMI.R0      2
149 0111 1E 00 22                 BCTA.LT      MBUG           CR
150 0114 18 1C                    BCTR.EQ      CSRE           LF
151 0116 CC 04 0F         ASRE    STRA.R0      TEMQ           UPDATE CONTENTS, THEN
152 0119 3F 02 D3                 BSTA.UN      GNUM
153 011C 02                       LODZ         R2
154 011D 9E 04 11                 LODA.R2      TEMR
155 0120 CE 64 00                 STRA.R0      COM,R2
156 0123 E6 08                    COMI.R2      8              MUST UPDATE PSW LOWER
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   4

LINE ADDR B1 B2 B3 B4 ERR SOURCE

```
157 0125 98 03                    BCFR,EQ     BSRE
158 0127 CC 04 0A                 STRA,R0     XGOT+1
159 012A 0C 04 0F      BSRE       LODA,R0     TEMQ
160 012D E4 03                    COMI,R0     3
161 012F 1C 00 22                 BCTA,EQ     MBUG
162 0132 0E 04 11      CSRE       LODA,R2     TEMR
163 0135 86 01                    ADDI,R2     1
164 0137 1F 00 F7                 BCTA,UN     LSRE
165                    *  GOTO ADDRESS
166 013A 3F 02 DB      GOTO       BSTA,UN     GNUM
167 013D 3F 00 A4                 BSTA,UN     STRT        PUT ADDR IN RAM
168 0140 0C 04 07                 LODA,R0     COM+7
169 0143 92                       LPSU
170 0144 0D 04 01                 LODA,R1     COM+1        BANK ZERO
171 0147 0E 04 02                 LODA,R2     COM+2
172 014A 0F 04 03                 LODA,R3     COM+3
173 014D 77 10                    PPSL        RS           BANK ONE
174 014F 0D 04 04                 LODA,R1     COM+4
175 0152 0E 04 05                 LODA,R2     COM+5
176 0155 0F 04 06                 LODA,R3     COM+6
177 0158 0C 04 00                 LODA,R0     COM
178 015B 75 FF                    CPSL        H'FF'
179 015D 1F 04 09                 BCTA,UN     XGOT         AND BCTA,UN $TEMP
180                    *
181                    *BREAKPOINT RUNTIME CODE
182 0160 CC 04 00      BK01       STRA,R0     COM          ENTRY FOR BKPT-1 VIA V
183 0163 13                       SPSL
184 0164 CC 04 08                 STRA,R0     COM+8
185 0167 CC 04 0A                 STRA,R0     XGOT+1       IN RAM FOR REG RESTORE
186 016A 04 00                    LODI,R0     0            BKPT INDEX
187 016C 1B 0C                    BCTR,UN     BKEN
188 016E CC 04 00      BK02       STRA,R0     COM          ENTRY FOR BKPT-2
189 0171 13                       SPSL
190 0172 CC 04 08                 STRA,R0     COM+8
191 0175 CC 04 0A                 STRA,R0     XGOT+1       IN RAM FOR REG RESTORE
192 0178 04 01                    LODI,R0     1
193 017A CC 04 11      BKEN       STRA,R0     TEMR
194 017D 12                       SPSU
195 017E CC 04 07                 STRA,R0     COM+7
196 0181 77 10                    PPSL        RS
197 0183 CD 04 04                 STRA,R1     COM+4
198 0186 CE 04 05                 STRA,R2     COM+5
199 0189 CF 04 06                 STRA,R3     COM+6
200 018C 75 18                    CPSL        RS+WC        FORCE TO BANK ZERO
201 018E CD 04 01                 STRA,R1     COM+1
202 0191 CE 04 02                 STRA,R2     COM+2
203 0194 CF 04 03                 STRA,R3     COM+3
204 0197 0E 04 11                 LODA,R2     TEMR
205 019A 3B 0F                    BSTR,UN     CLBK
206 019C 0D 04 0D                 LODA,R1     TEMP         PRINT BKPT ADDR
207 019F 3F 02 69                 BSTA,UN     BOUT
208 01A2 0D 04 0E                 LODA,R1     TEMP+1
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   5

LINE ADDR B1 B2 B3 B4 ERR SOURCE

```
209 01A5 3F 02 69              BSTA.UN    BOUT
210 01A8 1F 00 22              BCTA.UN    MBUG
211                         * SUBR TO CLEAR A BKPT     LIKE MANY SUBR HAS REL ADDR
212 01AB 20            CLBK    EORZ       R0
213 01AC CE 64 2D              STRA.R0    MARK.R2
214 01AF 0E 64 33              LODA.R0    HADR.R2
215 01B2 CC 04 0D              STRA.R0    TEMP
216 01B5 0E 64 35              LODA.R0    LADR.R2
217 01B8 CC 04 0E              STRA.R0    TEMP+1
218 01BB 0E 64 2F              LODA.R0    HDAT.R2
219 01BE CC 84 0D              STRA.R0    *TEMP
220 01C1 0E 64 31              LODA.R0    LDAT.R2
221 01C4 07 01                 LODI.R3    1
222 01C6 CF E4 0D              STRA.R0    *TEMP.R3
223 01C9 17                    RETC.UN
224                         * BREAK POINT    MARK INDICATES IF SET
225                         * HADR +LADR IS BKPT ADDR.   HDAT + LDAT  IS TWO BYTE
226 01CA 3B 0B        CLR      BSTR.UN    NOK
227 01CC 0E 64 2D              LODA.R0    MARK.R2       CLEAR IT IF SET
228 01CF 1C 00 1D              BCTA.Z     EBUG
229 01D2 3B 57                 BSTR.UN    CLBK
230 01D4 1F 00 22              BCTA.UN    MBUG
231 01D7 3F 02 DB     NOK      BSTA.UN    GNUM          CHECK RANGE ON BKPT NUMB
232 01DA A6 01                 SUBI.R2    1
233 01DC 1E 02 50              BCTA.N     ABRT
234 01DF E6 01                 COMI.R2    BMAX
235 01E1 1D 02 50              BCTA.GT    ABRT
236 01E4 17                    RETC.UN
237 01E5 3B 70       BKPT      BSTR.UN    NOK          SET BKPT AND CLR ANY E
238 01E7 0E 64 2D              LODA.R0    MARK.R2
239 01EA BC 01 AB              BSFA.Z     CLBK          CLEAR EXISTING
240 01ED CE 04 11              STRA.R2    TEMR
241 01F0 3F 02 DB              BSTA.UN    GNUM          GET BKPT ADDR
242 01F3 3F 00 A4              BSTA.UN    STRT          SUBR TO STORE R1-R2 IN
243 01F6 0F 04 11              LODA.R3    TEMR
244 01F9 02                    LODZ       R2
245 01FA CF 64 35              STRA.R0    LADR.R3
246 01FD 01                    LODZ       R1
247 01FE CF 64 33              STRA.R0    HADR.R3
248 0201 0C 84 0D              LODA.R0    *TEMP         SAVE CONTENTS
249 0204 CF 64 2F              STRA.R0    HDAT.R3
250 0207 05 9B                 LODI.R1    H'9B'         = ZBRR
251 0209 CD 84 0D              STRA.R1    *TEMP
252 020C 06 01                 LODI.R2    1
253 020E 0E E4 0D              LODA.R0    *TEMP.R2
254 0211 CF 64 31              STRA.R0    LDAT.R3
255 0214 0F 62 22              LODA.R0    DISP.R3
256 0217 CE E4 0D              STRA.R0    *TEMP.R2
257 021A 04 FF                 LODI.R0    -1
258 021C CF 64 2D              STRA.R0    MARK.R3
259 021F 1F 00 22              BCTA.UN    MBUG
260 0222 99         DISP       DATA       VEC+H'80'
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   6

```
LINE ADDR B1 B2 B3 B4 ERR SOURCE

261 0223 9B                        DATA       VEC+H'80'+2
262                             *
263                             * INPUT TWO HEX CHARS AND FORM  AS BYTE IN R1
264 0224 3F 02 86               BIN    BSTA.UN    CHIN
265 0227 3B 1D                         BSTR.UN    LKUP
266 0229 D3                            RRL.R3
267 022A D3                            RRL.R3
268 022B D3                            RRL.R3
269 022C D3                            RRL.R3
270 022D CF 04 12                      STRA.R3    TEMS
271 0230 3F 02 86                      BSTA.UN    CHIN
272 0233 3B 11                         BSTR.UN    LKUP
273 0235 6F 04 12                      IORA.R3    TEMS
274 0238 03                            LODZ       R3
275 0239 C1                            STRZ       R1
276 023A 3B 01                         BSTR.UN    CBCC
277 023C 17                            RETC.UN
278                             * CALCULATE THE BCC CHAR,  EOR AND THEN ROTATE LEFT
279 023D 01                     CBCC   LODZ       R1
280 023E 2C 04 2C                      EORA.R0    BCC
281 0241 D0                            RRL.R0
282 0242 CC 04 2C                      STRA.R0    BCC
283 0245 17                            RETC.UN
284                             * LOOKUP ASCII CHAR IN HEX VALUE TABLE
285 0246 07 10                  LKUP   LODI.R3    16
286 0248 EF 42 59               ALKU   COMA.R0    ANSI,R3,-
287 024B 14                            RETC.EQ
288 024C E7 01                         COMI.R3    1
289 024E 9A 78                         BCFR.LT    ALKU
290                             * ABORT EXIT FROM ANY LEVEL OF SUBR
291                             *  USE RAS PTR SINCE POSSIBLE BKPT PROG USING IT
292 0250 0C 04 07               ABRT   LODA.R0    COM+7
293 0253 64 40                         IORI.R0    H'40'
294 0255 92                            LPSU
295 0256 1F 00 1D                      BCTA.UN    EBUG
296 0259 30 31 32 33            ANSI   DATA       A'0123456789ABCDEF'
         34 35 36 37
         38 39 41 42
         43 44 45 46
297                             * BYTE IN R1 OUTPUT IN HEX
298 0269 CD 04 12               BOUT   STRA.R1    TEMS
299 026C 3B 4F                         BSTR.UN    CBCC
300 026E 51                            RRR.R1
301 026F 51                            RRR.R1
302 0270 51                            RRR.R1
303 0271 51                            RRR.R1
304 0272 45 0F                         ANDI.R1    H'0F'
305 0274 0D 62 59                      LODA.R0    ANSI,R1
306 0277 3F 02 B4                      BSTA.UN    COUT
307 027A 0D 04 12                      LODA.R1    TEMS
308 027D 45 0F                         ANDI.R1    H'0F'
309 027F 0D 62 59                      LODA.R0    ANSI,R1
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   7

LINE ADDR B1 B2 B3 B4 ERR SOURCE

```
310 0282 3F 02 B4              BSTA.UN    COUT
311 0285 17                    RETC.UN
312                     * 110 BAUD INPUT FOR PAPERTAPE AND CHAR   1MHZ CLOCK
313 0286 77 10          CHIN   PPSL       RS
314 0288 04 80                 LODI.R0    H'80'       ENABLE TAPE READER
315 028A B0                    WRTC.R0
316 028B 05 00                 LODI.R1    0
317 028D 06 08                 LODI.R2    8
318 028F 12             ACHI   SPSU
319 0290 1A 74                 BCTR.LT    CHIN        LOOK FOR START BIT
320 0292 20                    EORZ       R0
321 0293 B0                    WRTC.R0                DISABLE TAPE READER
322 0294 3B 17                 BSTR.UN    DLY
323 0296 3B 10          BCHI   BSTR.UN    DLAY        WAIT TO MIDDLE OF DATA
324 0298 12                    SPSU
325 0299 44 80                 ANDI.R0    H'80'       MOVE BIT 7 OF R0 INTO
326 029B 51                    RRR.R1
327 029C 61                    IORZ       R1
328 029D C1                    STRZ       R1
329 029E FA 76                 BDRR.R2    BCHI
330 02A0 3B 06                 BSTR.UN    DLAY
331 02A2 45 7F                 ANDI.R1    H'7F'       DELETE PARITY BIT
332 02A4 01                    LODZ       R1
333 02A5 75 18                 CPSL       RS+WC
334 02A7 17                    RETC.UN
335                     * DELAY FOR ONE BIT TIME
336 02A8 20             DLAY   EORZ       R0
337 02A9 F8 7E                 BDRR.R0    $
338 02AB F8 7E                 BDRR.R0    $
339 02AD F8 7E          DLY    BDRR.R0    $
340 02AF 04 E5                 LODI.R0    H'E5'
341 02B1 F8 7E                 BDRR.R0    $
342 02B3 17                    RETC.UN
343                     *
344 02B4 77 10          COUT   PPSL       RS
345 02B6 76 40                 PPSU       FLAG
346 02B8 C2                    STRZ       R2
347 02B9 05 08                 LODI.R1    8
348 02BB 3B 6B                 BSTR.UN    DLAY
349 02BD 3B 69                 BSTR.UN    DLAY
350 02BF 74 40                 CPSU       FLAG
351 02C1 3B 65          ACOU    BSTR.UN    DLAY
352 02C3 52                    RRR.R2
353 02C4 1A 04                 BCTR.LT    ONE
354 02C6 74 40                 CPSU       FLAG
355 02C8 1B 02                 BCTR.UN    ZERO
356 02CA 76 40          ONE    PPSU       FLAG
357 02CC F9 73          ZERO   BDRR.R1    ACOU
358 02CE 3B 58                 BSTR.UN    DLAY
359 02D0 76 40                 PPSU       FLAG
360 02D2 75 10                 CPSL       RS
361 02D4 17                    RETC.UN
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   8

LINE ADDR B1 B2 B3 B4 ERR SOURCE

```
362                              *
363                              * GET A NUMBER FROM THE BUFFER INTO R1 - R2
364  02D5 0C 04 2A     DNUM    LODA.R0     CODE
365  02D8 18 07                BCTR.Z      LNUM        SKIP SPACES UNTIL REAC
366  02DA 17                   RETC.UN                 OR SPACE ENDING NUMBER
367  02DB 20           GNUM    EORZ        R0
368  02DC C1                   STRZ        R1
369  02DD C2                   STRZ        R2
370  02DE CC 04 2A             STRA.R0     CODE
371  02E1 0F 04 27     LNUM    LODA.R3     BPTR
372  02E4 EF 04 29             COMA.R3     CNT         CHECK FOR E O B
373  02E7 14                   RETC.EQ
374  02E8 0F 24 13             LODA.R0     BUFF.R3.+   GET CHAR
375  02EB CF 04 27             STRA.R3     BPTR
376  02EE E4 20                COMI.R0     SPAC
377  02F0 18 63                BCTR.EQ     DNUM
378  02F2 3F 02 46     BNUM    BSTA.UN     LKUP
379  02F5 04 0F        CNUM    LODI.R0     H'0F'       R1=AB R2=DD
380  02F7 D2                   RRL.R2
381  02F8 D2                   RRL.R2
382  02F9 D2                   RRL.R2
383  02FA D2                   RRL.R2
384  02FB 42                   ANDZ        R2
385  02FC D1                   RRL.R1
386  02FD D1                   RRL.R1
387  02FE D1                   RRL.R1
388  02FF D1                   RRL.R1
389  0300 45 F0                ANDI.R1     H'F0'
390  0302 46 F0                ANDI.R2     H'F0'       R0=C R1=B0 R2=D0 R3=V
391  0304 61                   IORZ        R1
392  0305 C1                   STRZ        R1
393  0306 03                   LODZ        R3
394  0307 62                   IORZ        R2
395  0308 C2                   STRZ        R2          R1=BC R2=DV
396  0309 04 01                LODI.R0     1
397  030B CC 04 2A             STRA.R0     CODE
398  030E 1B 51                BCTR.UN     LNUM
399                            * DUMP TO PAPER TAPE IN OBJECT FORMAT
400  0310 3B 49        DUMP    BSTR.UN     GNUM        START ADDRESS
401  0312 3F 00 A4             BSTA.UN     STRT        SUBR TO STORE R1-R2 IN
402  0315 3B 44                BSTR.UN     GNUM
403  0317 86 01                ADDI.R2     1
404  0319 77 08                PPSL        WC
405  031B 85 00                ADDI.R1     0
406  031D 75 08                CPSL        WC          MAKE END ADDR NOT INCL
407  031F CD 04 0F             STRA.R1     TEMQ
408  0322 CE 04 10             STRA.R2     TEMQ+1
409  0325 3B 38        FDUM    BSTR.UN     GAP
410  0327 04 FF                LODI.R0     -1
411  0329 CC 04 29             STRA.R0     CNT
412  032C 3F 00 8A             BSTA.UN     CRLF        PUNCH FOR CR/LF AND ST
413  032F 04 3A                LODI.R0     STAR
```

PIP ASSEMBLER VERSION 3 LEVEL 1   PAGE   9

```
LINE ADDR B1 B2 B3 B4 ERR SOURCE

414 0331 3F 02 B4              BSTA.UN    COUT
415 0334 20                    EORZ       R0
416 0335 CC 04 2C              STRA.R0    BCC
417 0338 0D 04 0F              LODA.R1    TEMQ
418 033B 0E 04 10              LODA.R2    TEMQ+1
419 033E AE 04 0E              SUBA.R2    TEMP+1         GET BYTE COUNT
420 0341 77 08                 PPSL       WC
421 0343 AD 04 0D              SUBA.R1    TEMP
422 0346 75 08                 CPSL       WC
423 0348 1E 00 1D              BCTA.N     EBUG           START >  END ADDR
424 034B 19 1C                 BCTR.P     ADUM           CNT >  NORMAL BLOCK SI
425 034D 5A 1C                 BRNR.R2    BDUM           THIS IS SHORT BLOCK
426 034F 07 04                 LODI.R3    4              EOF,   PUNCH ZERO BLK
427 0351 3F 02 69    CDUM      BSTA.UN    BOUT
428 0354 FB 7B                 BDRR.R3    CDUM
429 0356 3B 07                 BSTR.UN    GAP
430 0358 1F 00 22              BCTA.UN    MBUG
431                    * SUBRS FOR OUTPUTTING BLANKS
432 035B 07 03       FORM      LODI.R3    3
433 035D 1B 02                 BCTR.UN    AGAP
434 035F 07 32       GAP       LODI.R3    50
435 0361 04 20       AGAP      LODI.R0    SPAC
436 0363 3F 02 B4              BSTA.UN    COUT
437 0366 FB 79                 BDRR.R3    AGAP
438 0368 17                    RETC.UN
439 0369 06 FF       ADUM      LODI.R2    255
440 036B CE 04 28    BDUM      STRA.R2    MCNT
441 036E 0D 04 0D              LODA.R1    TEMP           STARTING ADDRESS
442 0371 3F 02 69              BSTA.UN    BOUT
443 0374 0D 04 0E              LODA.R1    TEMP+1
444 0377 3F 02 69              BSTA.UN    BOUT
445 037A 0D 04 28              LODA.R1    MCNT           COUNT OF DATA BYTES IN
446 037D 3F 02 69              BSTA.UN    BOUT
447 0380 0D 04 2C              LODA.R1    BCC
448 0383 3F 02 69              BSTA.UN    BOUT
449 0386 0F 04 29    DDUM      LODA.R3    CNT
450 0389 0F A4 0D              LODA.R0    *TEMP,R3,+
451 038C EF 04 28              COMA.R3    MCNT
452 038F 18 09                 BCTR.EQ    EDUM           OUTPUT BCC
453 0391 CF 04 29              STRA.R3    CNT
454 0394 C1                    STRZ       R1
455 0395 3F 02 69              BSTA.UN    BOUT
456 0398 1B 6C                 BCTR.UN    DDUM
457 039A 0D 04 2C    EDUM      LODA.R1    BCC
458 039D 3F 02 69              BSTA.UN    BOUT
459 03A0 0E 04 0E              LODA.R2    TEMP+1
460 03A3 8E 04 28              ADDA.R2    MCNT
461 03A6 05 00                 LODI.R1    0
462 03A8 77 08                 PPSL       WC
463 03AA 8D 04 0D              ADDA.R1    TEMP
464 03AD 75 08                 CPSL       WC
465 03AF 3F 00 A4              BSTA.UN    STRT
```

```
LINE ADDR B1 B2 B3 B4 ERR SOURCE

466 03B2 1F 03 25          BCTA.UN     FDUM
467                    * LOAD FROM PAPERTAPE IN OBJECT FORMAT
468 03B5 3F 02 86    LOAD  BSTA.UN     CHIN          LOOK FOR START CHAR
469 03B8 E4 3A             COMI.R0     STAR
470 03BA 98 79             BCFR.EQ     LOAD
471 03BC 20               EORZ         R0
472 03BD CC 04 2C          STRA.R0     BCC
473 03C0 3F 02 24          BSTA.UN     BIN           READ ADDR AND COUNT IN
474 03C3 CD 04 0D          STRA.R1     TEMP
475 03C6 3F 02 24          BSTA.UN     BIN
476 03C9 CD 04 0E          STRA.R1     TEMP+1
477 03CC 3F 02 24          BSTA.UN     BIN
478 03CF 59 03             BRNR.R1     ALOA          CNT = 0 MEANS EOF
479 03D1 1F 84 0D          BCTA.UN     *TEMP
480 03D4 CD 04 28    ALOA  STRA.R1     MCNT
481 03D7 3F 02 24          BSTA.UN     BIN           CHECK BCC ON INFORMATI
482 03DA 0C 04 2C          LODA.R0     BCC
483 03DD 9C 00 1D          BCFA.Z      EBUG
484 03E0 C3               STRZ         R3            READ DATA
485 03E1 CF 04 29    BLOA  STRA.R3     CNT
486 03E4 3F 02 24          BSTA.UN     BIN
487 03E7 0F 04 29          LODA.R3     CNT
488 03EA EF 04 28          COMA.R3     MCNT
489 03ED 18 06             BCTR.EQ     CLOA          HAVE READ BCC
490 03EF 01               LODZ         R1
491 03F0 CF E4 0D          STRA.R0     *TEMP,R3      STORE DATA
492 03F3 DB 6C             BIRR.R3     BLOA
493 03F5 0C 04 2C    CLOA  LODA.R0     BCC
494 03F8 9C 00 1D          BCFA.Z      EBUG
495 03FB 1F 03 B5          BCTA.UN     LOAD
496                    *
497                       ORG          H'400'
498                    ******    RAM DEFINITIONS
499 0400             COM   RES          9
500 0409 77 00       XGOT  PPSL         0
501 040B 1B 80             BCTR.UN     *$+2          MUST PREDEED THE TEMP
502 040D             TEMP  RES          2
503 040F             TEMQ  RES          2
504 0411             TEMR  RES          1
505 0412             TEMS  RES          1
506 0413             BUFF  RES          BLEN
507 0427             BPTR  RES          1
508 0428             MCNT  RES          1
509 0429             CNT   RES          1
510 042A             CODE  RES          1
511 042B             OKGO  RES    1
512 042C             BCC   RES    1
513 042D             MARK  RES    BMAX+1
514 042F             HDAT  RES    BMAX+1
515 0431             LDAT  RES    BMAX+1
516 0433             HADR  RES    BMAX+1
517 0435             LADR  RES    BMAX+1
```