

SC/MP ASSEMBLER REV-C 02/06/76
 SCMPKB P005235A 7/14/76

```

1          .TITLE SCMPKB, P005235A 7/14/76
2          ;*****
3          ;*
4          ;*
5          ;*          PROM #          ADDRESS          BOARD          BOARD #
6          ;*
7          ;*          460305235-001          0000          5A          9804879
8          ;*
9          ;*
10         ;*****
11         ;
12
13         0F00 RAM          =          0F00
14         0D00 DISP          =          0D00
15
16         ;          SEGMENT ASSIGNMENTS
17
18         0001 SA          =          1
19         0002 SB          =          2
20         0004 SC          =          4
21         0008 SD          =          8
22         0010 SE          =          16
23         0020 SF          =          32
24         0040 SG          =          64
25
26         ;          7 SEGMENT CONVERSION
27
28         003F N0          =          SA+SB+SC+SD+SE+SF
29         0006 N1          =          SB+SC
30         005B N2          =          SA+SB+SD+SE+SG
31         004F N3          =          SA+SB+SC+SD+SG
32         0066 N4          =          SB+SC+SF+SG
33         006D N5          =          SA+SC+SD+SF+SG
34         007D N6          =          SA+SC+SD+SE+SF+SG
35         0007 N7          =          SA+SB+SC
36         007F N8          =          SA+SB+SC+SD+SE+SF+SG
37         0067 N9          =          SA+SB+SC+SF+SG
38         0077 NA          =          SA+SB+SC+SE+SF+SG
39         007C NB          =          SC+SD+SE+SF+SG
40         0039 NC          =          SA+SD+SE+SF
41         005E ND          =          SB+SC+SD+SE+SG
42         0079 NE          =          SA+SD+SE+SF+SG
43         0071 NF          =          SA+SE+SF+SG
44         0040 DASH          =          SG
45         0079 KE          =          NE
46         0050 KR          =          SE+SG
47         005C KO          =          SC+SD+SE+SG
48
49
50         .PAGE          ' HARDWARE FOR KEYBOARD '
51
52
53         ;          FUNCTION          DATA          KYB          FUNCTION
54
55         ;          0          080          0
56         ;          1          081          1
57         ;          2          082          2
58         ;          3          083          3
59         ;          4          084          4
60         ;          5          085          5
61         ;          6          086          6
62         ;          7          087          7
    
```

```

63      ;          8          040          8
64      ;          9          041          9
65      ;          A          010          +
66      ;          B          011          -
67      ;          C          012          MUL
68      ;          D          013          DIV
69      ;          E          016          SQUARE
70      ;          F          017          SQRT
71      ;          GO         022          %
72      ;          MEM        023          =
73      ;          ABORT      024          CE/C
74      ;          TERM       027          .
75
76      ;          RAM POINTERS USED BY KITBUG,P3 IS SAVED ELSEWHERE
77
78
79      0FF9 P1H      =          0FF9
80      0FFA P1L      =          0FFA
81      0FFB P2H      =          0FFB
82      0FFC P2L      =          0FFC
83      0FFD A        =          0FFD
84      0FFE E        =          0FFE
85      0FFF S        =          0FFF
86
87      ;          COMMANDS
88
89      ; ABORT:
90      ;          THIS ABORTS THE PRESENT OPERATION. DISPLAYS - .
91
92      ;MEM:
93      ;          ALLOWS USER TO READ/MODIFY MEMORY.
94      ;          ADDRESS IS ENTERED UNTIL TERM THEN DATA IS ENTERED.
95      ;          TO WRITE DATA IN MEMORY TERM IS PUSHED.
96      ;          DATA IS READ TO CHECK IF IT GOT WRITTEN IN RAM.
97
98      ; GO:
99      ;          ADDRESS IS ENTERED UNTIL TERM.
100     ;          THE REGISTERS ARE LOADED FROM RAM AND PROGRAM
101     ;          IS TRANSFERED USING XPPC P3.
102     ;          TO GET BACK DO A XPPC P3.
103
104     .PAGE      ' INITIALIZE '
105     0000 08      NOP
106     0001      INIT:      JMP      START
107     0001 901D
108
109     ;          DEBUG EXIT
110     ;          RESTORE ENVIROMENT
111
112     0003      GOOUT:     LD        ADH(2)      ;GET GO ADDRESS.
113     0003 C20E     XPAH      3
114     0005 37      LD        ADL(2)
115     0006 C20C     XPAL      3
116     0008 33      LD        @-1(3)      ;FIX GO ADDRESS.
117     0009 C7FF     LD        E          ;RESTORE REGISTERS.
118     000B C0F2
119     000D 01      XAE
120     000E C0EB     LD        P1L
121     0010 31      XPAL      1
122     0011 C0E7     LD        P1H
123     0013 35      XPAH      1
124     0014 C0E7     LD        P2L
125     0016 32      XPAL      2
126     0017 C0E3     LD        P2H
127     0019 36      XPAH      2
128     001A C0E4     LD        S

```

```

129 001C 07          CAS
130 001D C0DF       LD      A
131 001F 3F         XPPC   3
132
133                 ;      ENTRY POINT FOR DEBUG      ;TO GET BACK.
134
135 0020           START:
136 0020 C8DC       ST      A      ;SAVE STATUS.
137 0022 40        LDE
138 0023 C8DA       ST      E
139 0025 06        CSA
140 0026 C8D8       ST      S
141 0028 35        XPAH   1
142 0029 C8CF       ST      P1H
143 002B 31        XPAL   1
144 002C C8CD       ST      P1L
145 002E C40F       LDI     H(RAM)      ;SET P2 TO POINT TO RAM.
146 0030 36        XPAH   2
147 0031 C8C9       ST      P2H
148 0033 C400       LDI     L(RAM)
149 0035 32        XPAL   2
150 0036 C8C5       ST      P2L
151 0038 C701       LD      @1(3)      ;BUMP P3 FOR RETURN.
152 003A 33        XPAL   3      ;SAVE P3.
153 003B CA0C       ST      ADL(2)
154 003D 37        XPAH   3
155 003E CA0E       ST      ADH(2)

156                 .PAGE
157
158
159                 ;      ABORT SEQUENCE
160
161 0040           ABORT:
162 0040 C400       LDI     0
163 0042 CA02       ST      D3(2)
164 0044 CA03       ST      D4(2)
165 0046 CA08       ST      D9(2)
166 0048 C440       LDI     DASH      ;SET SEGMENTS TO -.
167 004A CA00       ST      DL(2)
168 004C CA01       ST      DH(2)
169 004E CA04       ST      ADLL(2)
170 0050 CA05       ST      ADLH(2)
171 0052 CA06       ST      ADHL(2)
172 0054 CA07       ST      ADHH(2)
173 0056           WAIT:
174 0056 C401       JS      3,KYBD      ;DISPLAY AND READ KEYBOARD.
      0058 37C4
      005A 8433
      005C 3F
175 005D 9002       JMP     WCK
176 005F 90DF       JMP     ABORT      ;COMMAND RETURN.
177                 ;RETURN FOR NUMBER.
178 0061           WCK:
179 0061 E407       XRI     07      ;CHECK IF MEM.
180 0063 9856       JZ      MEM
181 0065 E401       XRI     01      ;CHECK IF GO.
182 0067 9CD7       JNZ     ABORT

183                 .PAGE      ' GO TO '
184
185                 ;      GO WAS PUSHED
186                 ;      GO TO USER PROGRAM
187 0069           GO:
188 0069 C4FF       LDI     -1      ;SET FIRST FLAG.
189 006B CA0F       ST      DDTA(2)

```

```

190 006D C440      LDI      DASH                ;SET DATA TO DASH.
191 006F CA00      ST       DL(2)
192 0071 CA01      ST       DH(2)
193 0073          GOL:
194 0073 C459      LDI      L(DISPA)-1          ;FIX ADDRESS SEG.
195 0075 33        XPAL     3
196 0076 3F        XPPC     3                    ;DO DISPLAY AND KEYBRD.
197 0077 9006      JMP      GOCK                ;COMMAND RETURN.
198 0079 C41A      LDI      L(ADR)-1           ;SET ADDRESS.
199 007B 33        XPAL     3
200 007C 3F        XPPC     3
201 007D 90F4      JMP      GOL                  ;NOT DONE.
202 007F          GOCK:
203 007F E403      XRI      03                  ;CHECK FOR TERM.
204 0081 9880      JZ       GOOUT              ;ERROR IF NO TERM.
205
206
207                ;      INCORRECT SEQUENCE
208                ;      DISPLAY ERROR WAIT FOR NEW INPUT
209
210
211 0083          ERROR:
212 0083 C479      LDI      KE                  ;FILL WITH ERROR.
213 0085 CA07      ST       ADHH(2)
214 0087 C450      LDI      KR
215 0089 CA06      ST       ADHL(2)
216 008B CA05      ST       ADLH(2)
217 008D CA03      ST       D4(2)
218 008F C45C      LDI      KO
219 0091 CA04      ST       ADLL(2)
220 0093 C400      LDI      0
221 0095 CA02      ST       D3(2)
222 0097 CA01      ST       DH(2)
223 0099 CA00      ST       DL(2)
224 009B 90B9      JMP      WAIT

225                .PAGE      ' MEMORY TRANACTIONS '
226
227 009D          DTACK:
228 009D C211      LD       NEXT(2)            ;CHECK IF DATA FIELD.
229 009F 9C36      JNZ     DATA              ;ADDRESS DONE.
230
231
232 00A1          MEMDN:
233 00A1 C20E      LD       ADH(2)            ;PUT WORD IN MEM.
234 00A3 35        XPAH     1
235 00A4 C20C      LD       ADL(2)
236 00A6 31        XPAL     1
237 00A7 C20D      LD       WORD(2)
238 00A9 C900      ST       (1)
239 00AB 900E      JMP      MEM
240
241 00AD          MEMCK:
242 00AD E406      XRI      06                ;CHECK FOR GO.
243 00AF 98D2      JZ       ERROR            ;CAN NOT GO NOW.
244 00B1 E405      XRI      05                ;CHECK FOR TERM.
245 00B3 98E8      JZ       DTACK            ;CHECK IF DONE.
246 00B5 AA0C      ILD     ADL(2)            ;UPDATE ADDRESS LOW.
247 00B7 9C02      JNZ     MEM               ;CHECK IF UPDATE HI.
248 00B9 AA0E      ILD     ADH(2)
249
250                ;      MEM KEY PUSHED
251 00BB          MEM:
252 00BB C4FF      LDI      -1                ;SET FIRST FLAG.
253 00BD CA11      ST       NEXT(2)          ;SET FLAG FOR ADDRESS NOW.
254 00BF CA0F      ST       DDTA(2)
255 00C1          MEML:

```

```

256 00C1 C20E LD ADH(2)
257 00C3 35 XPAH 1 ;SET P1 FOR MEM ADDRESS.
258 00C4 C20C LD ADL(2)
259 00C6 31 XPAL 1
260 00C7 C100 LD (1)
261 00C9 CA0D ST WORD(2) ;SAVE MEM DATA.
262 00CB C43F LDI L(DISP D)-1 ;FIX DATA SEG.
263 00CD 33 XPAL 3
264 00CE 3F XPPC 3 ;GO TO DISPD SET SEG FOR DATA.
265 00CF 90DC JMP MEMCK ;COMMAND RETURN.
266 00D1 C41A LDI L(ADR)-1 ;MAKE ADDRESS.
267 00D3 33 XPAL 3
268 00D4 3F XPPC 3
269 00D5 90EA JMP MEML ;GET NEXT CHAR.
270 00D7 DATA:
271 00D7 C4FF LDI -1 ;SET FIRST FLAG.
272 00D9 CA0F ST DDTA(2)
273 00DB C20E LD ADH(2) ;SET P1 TO MEMORY ADDRESS.
274 00DD 35 XPAH 1
275 00DE C20C LD ADL(2)
276 00E0 31 XPAL 1
277 00E1 C100 LD (1) ;READ DATA WORD.
278 00E3 CA0D ST WORD(2) ;SAVE FOR DISPLY.

279 .PAGE
280 00E5 DATAL:
281 00E5 C43F LDI L(DISP D)-1 ;FIX DATA SEG.
282 00E7 33 XPAL 3
283 00E8 3F XPPC 3 ;FIX DATA SEG-GO TO DISPD.
284 00E9 90C2 JMP MEMCK ;CHAR RETURN.
285 00EB C404 LDI 4 ;SET COUNTER FOR NUMBER OF SHIFTS
286 00ED CA09 ST CNT(2)
287 00EF AA0F ILD DDTA(2) ;CHECK IF FIRST.
288 00F1 9C06 JNZ DNFST
289 00F3 C400 LDI 0 ;ZERO WORD IF FIRST.
290 00F5 CA0D ST WORD(2)
291 00F7 CA11 ST NEXT(2) ;SET FLAG FOR ADDRESS DONE.
292 00F9 DNFST:
293 00F9 02 CCL
294 00FA C20D LD WORD(2) ;SHIFT LEFT.
295 00FC F20D ADD WORD(2)
296 00FE CA0D ST WORD(2)
297 0100 BA09 DLD CNT(2) ;CHECK FOR 4 SHIFTS.
298 0102 9CF5 JNZ DNFST
299 0104 C20D LD WORD(2) ;ADD NEW DATA.
300 0106 58 ORE
301 0107 CA0D ST WORD(2)
302 0109 90DA JMP DATAL

303 .PAGE 'HEX NUMBER TO SEGMENT TABLE '
304
305
306 ; HEX NUMBER TO SEVEN SEGMENT TABLE
307
308
309 010B CROM:
310 010B 3F .BYTE N0
311 010C 06 .BYTE N1
312 010D 5B .BYTE N2
313 010E 4F .BYTE N3
314 010F 66 .BYTE N4
315 0110 6D .BYTE N5
316 0111 7D .BYTE N6
317 0112 07 .BYTE N7
318 0113 7F .BYTE N8
319 0114 67 .BYTE N9

```

```

320 0115 77      .BYTE  NA
321 0116 7C      .BYTE  NB
322 0117 39      .BYTE  NC
323 0118 5E      .BYTE  ND
324 0119 79      .BYTE  NE
325 011A 71      .BYTE  NF

```

```

326                .PAGE  ' MAKE 4 DIGIT ADDRESS '
327 011B          ADR:
328
329
330                ;      SHIFT ADDRESS LEFT ONE DIGIT THEN
331                ;      ADD NEW LOW HEX DIGIT.
332                ;      HEX DIGIT IN E REGISTER.
333                ;      P2 POINTS TO RAM.
334
335 011B C404      LDI      4                ;SET NUMBER OF SHIFTS.
336 011D CA09      ST       CNT(2)
337 011F AA0F      ILD      DDTA(2)         ;CHECK IF FIRST.
338 0121 9C06      JNZ      NOTFST         ;JMP IF NO.
339 0123 C400      LDI      0                ;ZERO ADDRESS.
340 0125 CA0E      ST       ADH(2)
341 0127 CA0C      ST       ADL(2)
342 0129          NOTFST:
343 0129 02        CCL                ;CLEAR LINK.
344 012A C20C      LD       ADL(2)         ;SHIFT ADDRESS LEFT 4 TIMES.
345 012C F20C      ADD      ADL(2)
346 012E CA0C      ST       ADL(2)         ;SAVE IT.
347 0130 C20E      LD       ADH(2)         ;NOW SHIFT HIGH.
348 0132 F20E      ADD      ADH(2)
349 0134 CA0E      ST       ADH(2)
350 0136 BA09      DLD      CNT(2)         ;CHECK IF SHIFTED 4 TIMES.
351 0138 9CEF      JNZ      NOTFST         ;JMP IF NOT DONE.
352 013A C20C      LD       ADL(2)         ;NOW ADD NEW NUMBER.
353 013C 58        ORE
354 013D CA0C      ST       ADL(2)         ;NUMBER IS NOW UP DATED.
355 013F 3F        XPPC      3
356

```

```

357                .PAGE  ' DATA TO SEGMENTS '
358
359
360
361                ;      CONVERT HEX DATA TO SEGMENTS.
362                ;      P2 POINTS TO RAM.
363                ;      DROPS THRU TO HEX ADDRESS CONVERSION.
364
365
366 0140          DISPD:
367 0140 C401      10 LDI      H(CROM)         ;SET ADDRESS OF TABLE.
368 0142 35        10 XPAH      1
369 0143 C40B      10 LDI      L(CROM)
370 0145 31        10 XPAL      1          P1 = 010B
371 0146 C20D      10 LD       WORD(2)         (P2 = 0B00) (P1 = 0E01)
372 0148 D40F      10 ANI      0F                ;GET MEMORY WORD. #EA = 01
373 014A 01        7  XAE
374 014B C180      10 LD       -128(1)         ;GET SEGMENT DISP.
375 014D CA00      10 ST       DL(2)         ;SAVE AT DATA LOW.
376 014F C20D      10 LD       WORD(2)         ;FIX HI.
377 0151 1C        10 SR
378 0152 1C        10 SR                ;SHIFT HI TO LOW.
379 0153 1C        10 SR
380 0154 1C        10 SR
381 0155 01        10 XAE
382 0156 C180      10 LD       -128(1)         ;GET SEGMENTS.
383 0158 CA01      10 ST       DH(2)         ;SAVE IN DATA HI.

```

Totd 123 0K

384
385
386

```

387          .PAGE      ADDRESS TO SEGMENTS
388
389
390
391          ;          CONVERT HEX ADDRESS TO SEGMENTS.
392          ;          P2 POINTS TO RAM.
393          ;          DROPS THRU TO KEYBOARD AND DISPLAY.
394
395
396 015A      DISPA:
397 015A 03          SCL
398 015B C401      10 LDI      H(CROM)          ;SET ADDRESS OF TABLE.
399 015D 35          XPAH      1
400 015E C40B      10 LDI      L(CROM)          P1 = 010B
401 0160 31          XPAL      1
402 0161      LOOPD:
403 0161 C20C      10 LD        ADL(2)          ;GET ADDRESS.
404 0163 D40F      10 ANI      0F
405 0165 01          XAE
406 0166 C180      10 LD        -128(1)          ;GET SEGMENTS.
407 0168 CA04      10 ST        ADLL(2)          ;SAVE SEG OF ADR LL.
408 016A C20C      10 LD        ADL(2)
409 016C 1C          SR
410 016D 1C          SR          ;SHIFT HI DIGIT TO LOW.
411 016E 1C          SR
412 016F 1C          SR
413 0170 01          XAE
414 0171 C180      10 LD        -128(1)          ;GET SEGMENTS.
415 0173 CA05      10 ST        ADLH(2)
416 0175 06          CSA          ;CHECK IF DONE.
417 0176 D480      10 ANI      080
418 0178 9809      10 JZ        DONE
419 017A 02          CCL          ;CLEAR FLAG.
420 017B C400      10 LDI      0
421 017D CA03      10 ST        D4(2)          ;ZERO DIGIT 4.
422 017F C602      10 LD        @2(2)          ;FIX P2 FOR NEXT LOOP.
423 0181 90DE      11 JMP        LOOPD
424 0183      DONE:
425 0183 C6FE      10 LD        @-2(2)          ;FIX P2.
426
427

```

```

428          .PAGE      DISPLAY AND KEYBOARD INPUT
429
430          ;          CALL      XPPC      3
431
432          ;          JMP        COMMAND IN A GO=6, MEM=7, TERM=3
433          ;          IN E GO=22, MEM=23, TERM=27.
434          ;          NUMBER RETURN      HEX NUMBER IN E REG.
435
436          ;          ABORT KEY GOES TO ABORT.
437
438          ;          ALL REGISTERS ARE USED.
439
440          ;          P2 MUST POINT TO RAM. ADDRESS MUST BE XXX0.
441
442          ;          TO RE-EXECUTE ROUTINE DO XPPC P3.
443
444
445 0185      KYBD:
446 0185 C400      10 LDI      0          ;ZERO CHAR.
447 0187 CA0B      10 ST        CHAR(2)

```

```

448 0189 C40D      10 LDI      H(DISP)      ;SET DISPLAY ADDRESS.
449 018B 35        8 XPAH      1
450 018C          OFF:
451 018C C4FF      10 LDI      -1          ;SET ROW/DIGIT ADDRESS.
452 018E CA10      18 ST       ROW(2)     ;SAVE ROW COUNTER.
453 0190 C40A      10 LDI      10         ;SET ROW COUNT.
454 0192 CA09      18 ST       CNT(2)
455 0194 C400      10 LDI      0
456 0196 CA0A      10 ST       PUSHED(2)  ;ZERO KEYBOARD INPUT.
457 0198 31        10 XPAL      1          ;SET DISP ADDRESS LOW.
458 0199          LOOP:
459 0199 AA10      22 ILD      ROW(2)     ;UP DATE ROW ADDRESS AND LOAD
460 019B 01        7 XAE
461 019C C280      18 LD       -128(2)   ;GET SEGMENT. (-128 = XTended used for offset)
462 019E C980      18 ST       -128(1)   ;SEND IT.
463 01A0 8F00      26 3 DLY      0          ;DELAY FOR DISPLAY.
464 01A2 C180      LD       -128(1)   ;GET KEYBOARD INPUT.
465 01A4 E4FF      XRI      OFF        ;CHECK IF PUSHED.
466 01A6 9C4C      JNZ      KEY        ;JUMP IF PUSHED.
467 01A8          BACK:
468 01A8 BA09      22 DLD      CNT(2)     ;CHECK IF DONE.
469 01AA 9CED      9 au 11 JNZ      LOOP XPPC 3 ;NO IF JUMP.
470 01AC C20A      Total: 3024 LD      PUSHED(2)   ;CHECK IF KEY.
471 01AE 980A      1 xppc 3 JZ       CKMORE
472 01B0 C20B      - 3024 LD      CHAR(2)    ;WAS THERE A CHAR?
473 01B2 9CD8      3024 JNZ      OFF        ;YES WAIT FOR RELEASE.
474 01B4 C20A      3024 LD      PUSHED(2)   ;NO SET CHAR.
475 01B6 CA0B      ST       CHAR(2)
476 01B8 90D2      JMP      OFF
477 01BA          CKMORE:
478 01BA C20B      LD       CHAR(2)    ;CHECK IF THERE WAS A CHAR.
479 01BC 98CE      JZ       OFF        ;NO KEEP LOOKING.

480          .PAGE
481
482          ;      COMMAND KEY PROCESSING
483
484 01BE          COMMAND:
485 01BE 01        XAE          ;SAVE CHAR.
486 01BF 40        LDE          ;GET CHAR.
487 01C0 D420      ANI      020       ;CHECK FOR COMMAND.
488 01C2 9C28      JNZ      CMND      ;JUMP IF COMMAND. /
489 01C4 C480      LDI      080       ;FIND NUMBER.
490 01C6 50        ANE
491 01C7 9C1B      JNZ      LT7       ;0 TO 7.
492 01C9 C440      LDI      040
493 01CB 50        ANE
494 01CC 9C19      JNZ      N89       ;8 OR 9.
495 01CE C40F      LDI      0F
496 01D0 50        ANE
497 01D1 F407      ADI      7          ;MAKE OFF SET TO TABLE.
498 01D3 01        XAE          ;PUT OFF SET AWAY.
499 01D4 C080      LD       -128(0)   ;GET NUMBER.
500 01D6          KEYRTN:
501 01D6 01        XAE          ;SAVE IN E.
502 01D7 C702      LD       @2(3)     ;FIX RETURN.
503 01D9 3F        XPPC      3         ;RETURN.
504 01DA 90A9      JMP      KYBD      ;ALLOWS XPPC P3 TO RETURN.
505
506 01DC 0A0B      .BYTE     0A,0B,0C,0D,0,0,0E,0F
507 01E4          LT7:
508 01E4 60        XRE          ;KEEP LOW DIGIT.
509 01E5 90EF      JMP      KEYRTN
510 01E7          N89:

```



```

511 01E7 60      XRE      ;GET LOW.
512 01E8 F408    ADI      08      ;MAKE DIGIT 8 OR 9.
513 01EA 90EA    JMP      KEYRTN

514              .PAGE
515 01EC          CMND:
516 01EC 60      XRE
517 01ED E404    XRI      04      ;CHECK IF ABORT.
518 01EF 9808    JZ       ABRT     ;ABORT.
519 01F1 3F      XPPC     3      ;IN E 23=MEM,22=GO,27=TERM.
520              ;IN A 7=MEM,6=GO,3=TERM.
521 01F2 9091    JMP      KYBD     ;ALLOWS JUST A XPPC P3 TO RETURN.
522
523
524 01F4          KEY:
525 01F4 58      ORE
526 01F5 CA0A    ST       PUSHED(2) ;MAKE CHAR.
527 01F7 90AF    JMP      BACK     ;SAVE CHAR.
528
529 01F9          ABRT:
530 01F9 C400    LDI      H(ABORT)
531 01FB 37      XPAH     3
532 01FC C43F    LDI      L(ABORT)-1
533 01FE 33      XPAL     3
534 01FF 3F      XPPC     3      ;GO TO ABORT

```

```

535              .PAGE      ' RAM OFF-SET '
536
537
538      0000 DL      =      0      ;SEGMENT FOR DIGIT 1
539      0001 DH      =      1      ;SEGMENT FOR DIGIT 2
540      0002 D3      =      2      ;SEGMENT FOR DIGIT 3
541      0003 D4      =      3      ;SEGMENT FOR DIGIT 4
542      0004 ADLL    =      4      ;SEGMENT FOR DIGIT 5
543      0005 ADLH    =      5      ;SEGMENT FOR DIGIT 6
544      0006 ADHL    =      6      ;SEGMENT FOR DIGIT 7
545      0007 ADHH    =      7      ;SEGMENT FOR DIGIT 8
546      0008 D9      =      8      ;SEGMENT FOR DIGIT 9
547      0009 CNT     =      9      ;COUNTER.
548      000A PUSHED  =     10      ;KEY PUSHED.
549      000B CHAR    =     11      ;CHAR READ.
550      000C ADL     =     12      ;MEMORY ADDRESS LOW.
551      000D WORD    =     13      ;MEMORY WORD.
552      000E ADH     =     14      ;MEMORY ADDRESS HI.
553      000F DDTA    =     15      ;FIRST FLAG.
554      0010 ROW     =     16      ;ROW COUNTER.
555      0011 NEXT    =     17      ;FLAG FOR NOW DATA.
556
557
558      0000          .END

```

```

*****      0 ERRORS IN ASSEMBLY      *****
A      ABORT  ABRT  ADH   ADHH  ADHL  ADL   ADLH  ADLL  ADR
0FFD  0040  01F9  000E  0007  0006  000C  0005  0004  011B

BACK  CHAR  CKMORE CMND  CNT   COMMAN CROM  D3   D4   D9
01A8  000B  01BA  01EC  0009  01BE  010B  0002  0003  0008

DASH  DATA  DATAL  DDTA  DH    DISP  DISPA DISPD DL   DNFST
0040  00D7  00E5  000F  0001  0D00  015A  0140  0000  00F9

DONE  DTACK  E      ERROR GO    GOCK  GOL   GOOUT INIT KE
0183  009D  0FFE  0083  0069  007F  0073  0003  0001  0079

KEY   KEYRTN KO    KR    KYBD  LOOP  LOOPD LT7   MEM  MEMCK

```

SCMPKB

01F4	01D6	005C	0050	0185	0199	0161	01E4	00BB	00AD
MEMDN	MEML	N0	N1	N2	N3	N4	N5	N6	N7
00A1	00C1	003F	0006	005B	004F	0066	006D	007D	0007
N8	N89	N9	NA	NB	NC	ND	NE	NEXT	NF
007F	01E7	0067	0077	007C	0039	005E	0079	0011	0071
NOTFST	OFF	PlH	PlL	P2H	P2L	PUSHED	RAM	ROW	S
0129	018C	0FF9	0FFA	0FFB	0FFC	000A	0F00	0010	0FFF
SA	SB	SC	SD	SE	SF	SG	START	WAIT	WCK
0001	0002	0004	0008	0010	0020	0040	0020	0056	0061
WORD									
000D									

A799 08AB