Appendix B

KITBUG PROGRAM LISTING

```
                        .TITLE  KITBUG,' P00937A  12/1/75'
                    ;*****************************************************************
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*
                    ;*****************************************************************
                    ;
            0001    P1      =       1
            0002    P2      =       2
            0003    P3      =       3
                    ;
            FFFF    EXOFF   =       -1


                        .PAGE   'STACK ASSIGNMENTS'
                        .LOCAL
                    ;
                    ; FIXED STACK ASSIGNMENTS
                    ;
0000                    .=0FFF
0FFF                STACK:
            0000    SR      =       .-STACK
0FFF                        .=.-1
            FFFF    EX      =       .-STACK
0FFE                        .=.-1
            FFFE    AC      =       .-STACK
0FFD                        .=.-2
            FFFC    PT2     =       .-STACK
0FFB                        .=.-2
            FFFA    PT1     =       .-STACK
0FF9                        .=.-2
            FFF8    PC      =       .-STACK
                    ;
            0FF6    P2ADR   =       .-1


                        .PAGE   'DEBUG ENTRY AND EXIT'
                        .LOCAL
                    ;
                    ; ON A SOFTWARE HALT, HARDWARE USES THE FOLLOWING WORDS
                    ; TO SAVE THE ENVIROMENT.
                    ;
0FF7                        .=0
0000 08                     NOP
0001 901D           START:  JMP     ENTER
                    ;
                    ; DEBUG EXIT - RESTORE ENVIROMENT AND GO.
                    ;
0003 C0FA           EXIT:   LD      STACK+EX        ; RESTORE E REG
0005 01                     XAE
0006 C0F2                   LD      STACK+PT1       ; RESTORE P1
0008 35                     XPAH    P1
0009 C0F0                   LD      STACK+PT1+1
000B 31                     XPAL    P1
000C C0EE                   LD      STACK+PT2       ; RESTORE P2
000E 36                     XPAH    P2
000F C0EC                   LD      STACK+PT2+1
```

```
 61 0011 32            XPAL    P2
 62 0012 C0E4          LD      STACK+PC        ; PUT DESIRED PC IN P3
 63 0014 37            XPAH    P3
 64 0015 C0E2          LD      STACK+PC+1
 65 0017 33            XPAL    P3
 66 0018 C7FF          LD      @EXOFF(P3)      ; ADD EXIT OFFSET TO PC
 67 001A C0E4          LD      STACK+SR        ; RESTORE SR
 68 001C 07            CAS
 69 001D C0DF          LD      STACK+AC
 70 001F 3F            XPPC    P3
 71                ;
 72                ; DEBUG ENTRY POINT
 73                ;
 74 0020 C8DC  ENTER:  ST      STACK+AC
 75 0022 06            CSA
 76 0023 C8DB          ST      STACK+SR
 77 0025 01            XAE                     ; SAVE EXTENSION REGISTER
 78 0026 C8D7          ST      STACK+EX
 79 0028 36            XPAH    P2              ;   POINTER
 80 0029 C8D1          ST      STACK+PT2
 81 002B 32            XPAL    P2
 82 002C C8CF          ST      STACK+PT2+1
 83 002E 35            XPAH    P1              ;   STACK
 84 002F C8C9          ST      STACK+PT1
 85 0031 31            XPAL    P1
 86 0032 C8C7          ST      STACK+PT1+1
 87 0034 37            XPAH    P3
 88 0035 C8C1          ST      STACK+PC
 89 0037 33            XPAL    P3
 90 0038 C8BF          ST      STACK+PC+1


 91                    .PAGE   'MAIN COMMAND LOOP'
 92                    .LOCAL
 93                ;
 94                ; THIS CODE INITIALIZES POINTER REGISTERS AND
 95                ; PROMPTS FOR AND GETS THE NEXT COMMAND.
 96                ;
 97                ; ON EXIT, E HOLDS THE COMMAND CHARACTER
 98                ;
 99 003A C4F6  CMDLP:  LDI     L(P2ADR)
100 003C 32            XPAL    P2
101 003D C40F          LDI     H(P2ADR)
102 003F 36            XPAH    P2
103 0040 C401          LDI     H(PUTC)         ; PRINT CR-LF
104 0042 37            XPAH    P3
105 0043 C4C4          LDI     L(PUTC)-1
106 0045 33            XPAL    P3
107 0046 C40D          LDI     0D
108 0048 3F            XPPC    P3              ; PRINT CR
109 0049 C40A          LDI     0A
110 004B 3F            XPPC    P3              ; PRINT LF
111 004C C42D          LDI     '-'
112 004E 3F            XPPC    P3
113 004F C401          JS      P3,GECO         ; GET COMMAND CHARACTER
    0051 37C4
    0053 8533
    0055 3F


114                    .PAGE   'GO'
115                    .LOCAL
116                ;
117                ; RESTORE MACHINE STATE AND TRANSFER CONTROL
118                ; TO SPECIFIED ADDRESS.
119                ;
120                ; G ADDRESS
121                ;
```

```
122 0056 40     GO:     LDE
123 0057 E447           XRI     'G'
124 0059 9C07           JNZ     $SKIP
125 005B 3F             XPPC    P3              ; CALL GECO
126 005C E40D           XRI     0D
127 005E 98A3           JZ      EXIT
128 0060 906A           JMP     ERROR
129 0062        $SKIP:


130                     .PAGE   'TYPE'
131                     .LOCAL
132             ;
133             ; TYPE OR MODIFY MEMORY.
134             ;
135 0062 40     TYPE:   LDE                     ; CHECK FOR TYPE COMMAND, IF
136 0063 E454           XRI     'T'             ;   NOT 'T', SKIP COMMAND.
137 0065 9809           JZ      $2
138 0067 40     MOD:    LDE
139 0068 E44D           XRI     'M'
140 006A 9C60           JNZ     $SKIP
141 006C C400           LDI     0
142 006E 9002           JMP     $1
143 0070 C401   $2:     LDI     1
144 0072 CEFF   $1:     ST      @-1(P2)         ; SAVE FLAG FOR TYPE OR MODIFY
145 0074 C400           JS      P3,GHEX         ; GET ADDRESS
    0076 37C4
    0078 DF33
    007A 3F
146 007B E40D           XRI     0D              ; CHECK TERMINATOR
147 007D 9C4D           JNZ     ERROR
148 007F C601           LD      @1(P2)          ; PUT STARTING ADDRESS IN STACK
149 0081 35             XPAH    P1
150 0082 C601           LD      @1(P2)
151 0084 31             XPAL    P1
152 0085 C401   $4:     LDI     H(PUTC)         ; PRINT CR-LF
153 0087 37             XPAH    P3
154 0088 C4C4           LDI     L(PUTC)-1
155 008A 33             XPAL    P3
156 008B C40D           LDI     0D
157 008D 3F             XPPC    P3              ; PRINT CR
158 008E C40A           LDI     0A
159 0090 3F             XPPC    P3              ; PRINT LF
160 0091 35             XPAH    P1              ; PRINT HIGH BYTE
161 0092 01             XAE                     ; READ AND RESTORE BYTE FROM P1
162 0093 40             LDE
163 0094 35             XPAH    P1
164 0095 C401           LDI     H(PHEX2)
165 0097 37             XPAH    P3
166 0098 C443           LDI     L(PHEX2)-1
167 009A 33             XPAL    P3
168 009B 40             LDE
169 009C 3F             XPPC    P3              ; CALL PHEX2
170 009D 31             XPAL    P1              ; PRINT LOW BYTE
171 009E 01             XAE
172 009F 40             LDE
173 00A0 31             XPAL    P1
174 00A1 40             LDE
175 00A2 3F             XPPC    P3              ; CALL PHEX1
176 00A3 C501           LD      @1(P1)
177 00A5 3F             XPPC    P3              ; PRINT 2-DIGIT HEX FOLLOWED BY
178                                             ;   BLANK  (PHEX1)
179 00A6 C200           LD      (P2)            ; CHECK TYPE OR MODIFY FLAG
180 00A8 9CDB           JNZ     $4
181 00AA C401           JS      P3,GECO
    00AC 37C4
    00AE 8533
    00B0 3F
```

```
00B1 E40D          XRI     0D
00B3 98D0          JZ      $4
00B5 E415          XRI     015             ; 0D XOR 018 (CAN)
00B7 9881  LOOP1:  JZ      CMDLP
00B9 C400          JS      P3,GHEX2
00BB 37C4
00BD DB33
00BF 3F
00C0 E40D          XRI     0D
00C2 9C08          JNZ     ERROR
00C4 C601          LD      @1(P2)
00C6 C601          LD      @1(P2)
00C8 C9FF          ST      -1(P1)
00CA 90B9          JMP     $4
00CC       $SKIP:


                   .PAGE   'ERROR PROCESSING'
                   .LOCAL
                ;
                ; PRINT CARRAIGE RETURN , LINE FEED AND LOOP
                ; TO THE TOP OF THE COMMAND LOOP.
                ;
00CC C401  ERROR:  LDI     H(PUTC)         ; PRINT LINE FEED
00CE 37            XPAH    P3
00CF C4C4          LDI     L(PUTC)-1
00D1 33            XPAL    P3
00D2 C40A          LDI     0A
00D4 3F            XPPC    P3
00D5 C43F          LDI     '?'
00D7 3F            XPPC    P3
00D8 C400          LDI     0
00DA 90DB          JMP     LOOP1


                   .PAGE     'HEX NUMBER INPUT'
                   .LOCAL
                ;
                ; GHEX GETS A 16-BIT VALUE AND PUSHES IT TO THE STACK.
                ; GHEX2 ASSUMES THE FIRST CHAR IS IN THE E REGISTER.
                ; ONLY THE LAST 4 INPUT DIGITS ARE SAVED.
                ;
                ; RETURNS VALUE IN TOP 2 WORDS OF STACK AND TERMINATOR
                ; IN THE AC AND EX REGISTERS.
                ;
00DC C401  GHEX2:  LDI     1
00DE 9002          JMP     $6
00E0 C400  GHEX:   LDI     0               ; RESET GHEX2 FLAG
00E2 CAFB  $6:     ST      -5(P2)
00E4 C485          LDI     L(GECO)-1       ; SAVE RETURN ADDRESS AND SET UP
00E6 33            XPAL    P3              ;   TO GECO
00E7 CEFD          ST      @-3(P2)         ; STORE RETURN ADDRESS TO LEAVE ROOM
00E9 C401          LDI     H(GECO)         ;   FOR RESULT
00EB 37            XPAH    P3
00EC CEFF          ST      @-1(P2)
00EE C2FF          LD      -1(P2)
00F0 9C01          JNZ     $1
00F2 3F            XPPC    P3
00F3 C400  $1:     LDI     0               ; INITIALIZE RESULT TO 0
00F5 CA03          ST      3(P2)
00F7 CA02          ST      2(P2)
00F9 40    $LOOP:  LDE
00FA 03            SCL
00FB FC3A          CAI     '9'+1           ; CHECK FOR 0-9
00FD 940F          JP      $2              ; NOT 0-9, TOO LARGE
00FF 03            SCL
0100 FCF6          CAI     '0'-'9'-1       ; CHECK FOR 0-9
0102 9419          JP      $3              ; IF POSITIVE, NUMBER IS
```

```
243                                                 ;   IN RANGE AND CONVERTED.
244 0104 C601  $RET:    LD      @1(P2)              ; NUMBER IS NOT A HEX DIGIT,
245 0106 37             XPAH    P3                  ;   RETURN
246 0107 C601           LD      @1(P2)
247 0109 33             XPAL    P3
248 010A 40             LDE
249 010B 3F             XPPC    P3
250 010C 90D2           JMP     GHEX
251 010E 03     $2:     SCL
252 010F FC0D           CAI     'F'+1-'9'-1         ; CHECK FOR DIGITS A-F.
253 0111 94F1           JP      $RET                ; NUMBER TOO LARGE
254 0113 03             SCL
255 0114 FCFA           CAI     'A'-'F'-1
256 0116 9402           JP      $4                  ; DIGIT BETWEEN A&F
257 0118 90EA           JMP     $RET
258 011A 02     $4:     CCL
259 011B F40A           ADI     10                  ; ADJUST DIGIT VALUE FOR 10-16
260 011D CAFF   $3:     ST      -1(P2)              ; SAVE ADJUSTED DIGIT
261 011F C404           LDI     4                   ; SET UP BIT COUNTER FOR
262 0121 CAFE           ST      -2(P2)              ;   SHIFT.
263 0123 02     $5:     CCL                         ; SHIFT HEX DIGIT LEFT ONE
264 0124 C203           LD      3(P2)               ;   DIGIT, ONE BIT EACH
265 0126 F203           ADD     3(P2)               ;   TIME THROUGH LOOP.
266 0128 CA03           ST      3(P2)
267 012A C202           LD      2(P2)
268 012C F202           ADD     2(P2)
269 012E CA02           ST      2(P2)
270 0130 BAFE           DLD     -2(P2)
271 0132 9CEF           JNZ     $5
272 0134 02             CCL
273 0135 C203           LD      3(P2)               ; ADD CURRENT DIGIT INTO
274 0137 F2FF           ADD     -1(P2)              ;   NUMBER
275 0139 CA03           ST      3(P2)
276 013B 3F             XPPC    P3                  ; GET NEXT CHAR
277 013C 90BB           JMP     $LOOP               ; AND LOOP


278                     .PAGE   'HEX NUMBER OUTPUT'
279                     .LOCAL
280                 ;
281                 ; PRINT HEX NUMBER WITH TRAILING BLANK (PHEX1) OR
282                 ; WITHOUT IT (PHEX2).  NUMBER TO BE PRINTED IS
283                 ; IN AC.
284                 ;
285 013E CEFF  PHEX1:  ST      @-1(P2)             ; SAVE AC
286 0140 C420           LDI     020                 ; SET FLAG TO PRINT BLANK AFTER
287 0142 9004           JMP     $1                  ;   NUMBER
288 0144 CEFF  PHEX2:  ST      @-1(P2)             ; SAVE AC
289 0146 C400           LDI     0                   ; CLEAR FLAG TO PRINT BLANK
290 0148 CEFF  $1:      ST      @-1(P2)             ;   AFTER NUMBER
291 014A C4C4           LDI     L(PUTC)-1           ; LOAD ADDRESS OF PUTC TO P3
292 014C 33             XPAL    P3                  ;   AND SAVE RETURN ADDRESS
293 014D CEFF           ST      @-1(P2)
294 014F C401           LDI     H(PUTC)
295 0151 37             XPAH    P3
296 0152 CEFF           ST      @-1(R2)
297 0154 C402           LDI     2                   ; SET FLAG FOR 1ST NUMBER
298 0156 CEFF           ST      @-1(P2)
299 0158 C204           LD      4(P2)               ; GET ORIGINAL VALUE
300 015A 1C             SR                          ; SHIFT TO LOW 4 BITS
301 015B 1C             SR
302 015C 1C             SR
303 015D 1C             SR
304 015E 02     $5:     CCL                         ; CONVERT TO ASCII
305 015F F4F6           ADI     -10
306 0161 9404           JP      $2                  ; NUMBER IS A THRU F
307 0163 F43A           ADI     '0'+10
308 0165 9002           JMP     $3
```

```
309 0167 F440  $2:     ADI     'A'-1           ; THE -1 TAKES CARE OF CARRY IN
310 0169 3F    $3:     XPPC    P3              ; PRINT NUMBER
311 016A BA00          DLD     (P2)
312 016C 9806          JZ      $4
313 016E C204          LD      4(P2)           ; GET ORIGINAL NUMBER
314 0170 D40F          ANI     0F              ; MASK 2ND DIGIT
315 0172 90EA          JMP     $5
316 0174 C203  $4:     LD      3(P2)           ; CHECK FOR PRINTING BLANK
317 0176 9801          JZ      $6
318 0178 3F            XPPC    P3              ; IF NOT 0, PRINT BLANK
319 0179 C201  $6:     LD      1(P2)           ; RESTORE RETURN ADDRESS
320 017B 37            XPAH    P3
321 017C C202          LD      2(P2)
322 017E 33            XPAL    P3
323 017F C604          LD      @4(P2)          ; RESTORE STACK AND AC
324 0181 C601          LD      @1(P2)
325 0183 3F            XPPC    P3              ; RETURN
326 0184 90B8          JMP     PHEX1


327                    .PAGE   'GECO'
328                    .LOCAL
329            ;
330            ; GECO IS USED FOR KEYBOARD INPUT SO IT ECHOS THE
331            ; CHARACTER BUT DOES NOT ENABLE THE READER RELAY.
332            ;
333 0186 C408  GECO:   LDI     8               ; SET COUNT = 8
334 0188 CAFF          ST      -1(P2)
335 018A 06    $2:     CSA                     ; WAIT FOR START BIT
336 018B D420          ANI     020
337 018D 9CFB          JNZ     $2              ; NOT FOUND
338 018F C457          LDI     87              ; DELAY 1/2 BIT TIME
339 0191 8F04          DLY     4
340 0193 06            CSA                     ; IS START BIT STILL THERE?
341 0194 D420          ANI     020
342 0196 9CF2          JNZ     $2              ; NO
343 0198 06            CSA                     ; SEND START BIT (NOTE THAT
344 0199 DC01          ORI     1               ;   OUTPUT IS INVERTED)
345 019B 07            CAS
346 019C C47E  $LOOP:  LDI     126             ; DELAY 1 BIT TIME
347 019E 8F08          DLY     8
348 01A0 06            CSA                     ; GET BIT (SENSEB)
349 01A1 D420          ANI     020
350 01A3 9802          JZ      $3
351 01A5 C401          LDI     1
352 01A7 CAFE  $3:     ST      -2(P2)          ; SAVE BIT VALUE (0 OR 1)
353 01A9 1F            RRL                     ; ROTATE INTO LINK
354 01AA 01            XAE
355 01AB 1D            SRL                     ; SHIFT INTO CHARACTER
356 01AC 01            XAE                     ; RETURN CHAR TO E
357 01AD 06            CSA                     ; ECHO BIT TO OUTPUT
358 01AE DC01          ORI     1
359 01B0 E2FE          XOR     -2(P2)
360 01B2 07            CAS
361 01B3 BAFF          DLD     -1(P2)          ; DECREMENT BIT COUNT
362 01B5 9CE5          JNZ     $LOOP           ; LOOP UNTIL 0
363 01B7 06            CSA                     ; SET STOP BIT
364 01B8 D4FE          ANI     0FE
365 01BA 07            CAS
366 01BB 8F08          DLY     8
367 01BD 40            LDE                     ; AC HAS INPUT CHARACTER
368 01BE D47F          ANI     07F
369 01C0 01            XAE
370 01C1 40            LDE
371 01C2 3F            XPPC    P3              ; RETURN
372 01C3 90C1          JMP     GECO
```

```
373                     .PAGE   'PUTC'
374                     .LOCAL
375             ;
376             ; PUT CHARACTER  IN AC TO TTY.  ALL REGS SAVED.
377             ; IF INPUT DETECTED, CONTROL PASSES TO PROMPT.
378             ; NOTE: TTY LOGIC LEVELS ARE INVERTED FOR OUTPUT
379             ;
380 01C5 01    PUTC:   XAE
381 01C6 C4FF          LDI     255
382 01C8 8F17          DLY     23
383 01CA 06            CSA                     ; SET OUTPUT BIT TO LOGIC 0
384 01CB DC01          ORI     1               ;   FOR START BIT. (NOTE INVERSION)
385 01CD 07            CAS
386 01CE C409          LDI     9               ; INITIALIZE BIT COUNT
387 01D0 CAFF          ST      -1(P2)
388 01D2 C48A  $1:     LDI     138             ; DELAY 1 BIT TIME
389 01D4 8F08          DLY     8
390 01D6 BAFF          DLD     -1(P2)          ; DECREMENT BIT COUNT.
391 01D8 9810          JZ      $EXIT
392 01DA 40            LDE                     ; PREPARE NEXT BIT
393 01DB D401          ANI     1
394 01DD CAFE          ST      -2(P2)
395 01DF 01            XAE                     ; SHIF DATA RIGHT 1 BIT
396 01E0 1C            SR
397 01E1 01            XAE
398 01E2 06            CSA                     ; SET UP OUTPUT BIT
399 01E3 DC01          ORI     1
400 01E5 E2FE          XOR     -2(P2)
401 01E7 07            CAS                     ; PUT BIT TO TTY
402 01E8 90E8          JMP     $1
403 01EA 06    $EXIT:  CSA                     ; SET STOP BIT
404 01EB D4FE          ANI     0FE
405 01ED 07            CAS
406 01EE D420          ANI     020             ; CHECK FOR KEYBOARD INPUT
407 01F0 9803          JZ      $2              ; ATTEMPTED INPUT (NOTE THAT
408                                            ;   INPUT IS NOT INVERTED)
409 01F2 3F            XPPC    P3              ; RETURN
410 01F3 90D0          JMP     PUTC
411 01F5 C400  $2:     JS      P3,CMDLP
    01F7 37C4
    01F9 3933
    01FB 3F
412      0000          .END


   ******       0 ERRORS IN ASSEMBLY   ******
$1&     $1(     $1)     $1+     $2&     $2(     $2)     $2*     $2+     $3(
0072    00F3    0148    01D2    0070    010E    0167    018A    01F5    011D

$3)     $3*     $4&     $4(     $4)     $5(     $5)     $6(     $6)     $EXIT+
0169    01A7    0085    011A    0174    0123    015E    00E2    0179    01EA

$LOOP(  $LOOP*  $RET(   $SKIP%  $SKIP&  AC      CMDLP   ENTER   ERROR   EX
00F9    019C    0104    0062    00CC    FFFE    003A    0020    00CC    FFFF

EXIT    EXOFF   GECO    GHEX    GHEX2   GO      LOOP1   MOD     P1      P2
0003    FFFF    0186    00E0    00DC    0056    00B7    0067    0001    0002

P2ADR   P3      PC      PHEX1   PHEX2   PT1     PT2     PUTC    SR      STACK
0FF6    0003    FFF8    013E    0144    FFFA    FFFC    01C5    0000    0FFF

START   TYPE
0001    0062


   FCB3    08E0
```

Appendix C

APPLICATION EXAMPLE

C.1 SOFTWARE "ONE-SHOT"

The following program is intended for use with the SC/MP Kit. The program simulates a retriggerable one-shot. A momentary contact switch is used to "fire the one-shot" (begin the program). The switch is connected to the SENSE A input to SC/MP; SENSE A is the interrupt input. When the interrupt (switch closure) is detected, the FLAG 1 output from SC/MP is set to a logic '1' and is used to drive an LED indicator through a transistor. (The hardware for this demonstration circuit is shown schematically in figure C-1.) A Delay Instruction (DLY) is then used to generate a delay of approximately 4 seconds. After 4 seconds, the LED will be turned off by setting the FLAG 1 output to '0' (zero). If the switch is held down, or depressed again before the LED is turned off, the LED remains lit; it is turned off approximately 4 seconds after the last switch opening.

Table C-1 is an assembler listing for the program showing the memory locations, assembler mnemonic, and machine language format (in hexadecimal) for each instruction in the program.

Using KITBUG and the TTY, the program could be entered into memory using the Modify Command of KITBUG and then could be executed using the Go command. Table C-2 shows the printout of this program that would be obtained using the KITBUG Type Command.

Note that this program (and any program utilizing interrupts) uses Pointer Register P3. Therefore, a program-controlled transfer back to KITBUG cannot be accomplished (see section 4.6 for a discussion of transfer of control between KITBUG and application programs).

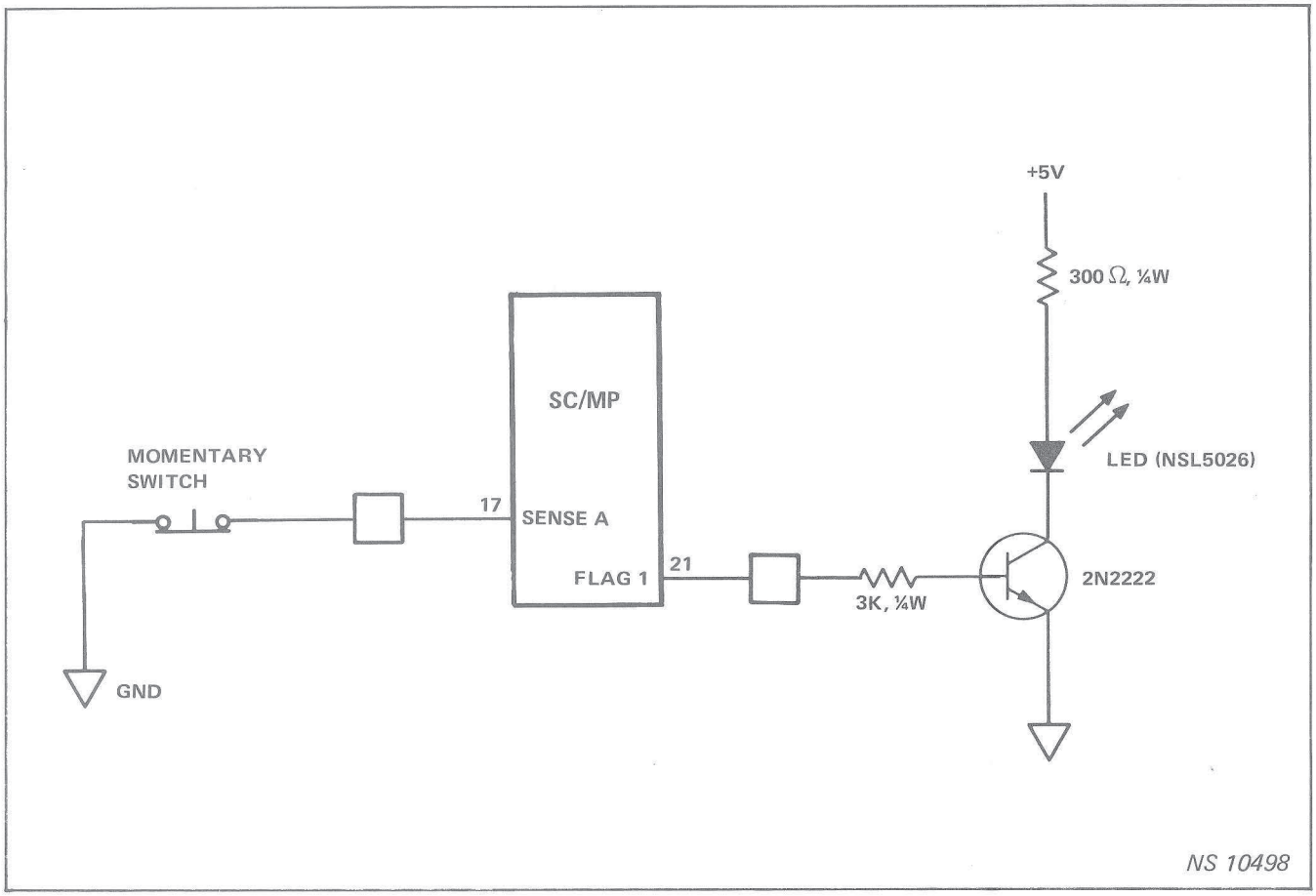

Figure C-1. 'One-Shot' Schematic

Table C-1. "One-Shot" Assembler Listing

```
                                                  Assembler Operand
Memory   Machine Language          Assembler Opcode      |        Comments
Address  Code (Hex)                Mnemonics             v

0F00     04                            DINT                       ;DISABLE INTERRUPT
0F01     C41C                          LDI      L(GO)-1
0F03     33                            XPAL     3
0F04     C40F                          LDI      H(GO)
0F06     37                            XPAH     3                 ;SET UP INTERRUPT
                                                                  ;POINTER

0F07     C400       OVER:              LDI      0
0F09     C81E                          ST       COUNT             ;ZERO LOOP CNTR
0F0B     05                            IEN                        ;ENABLE INTERRUPTS
0F0C     C01B       SEARCH:            LD       COUNT
0F0E     98FC                          JZ       SEARCH            ;WAIT FOR INTERRUPT
0F10     C4FF                          LDI      X'FF
0F12     8FFF       LOOP:              DLY      X'FF
0F14     B813                          DLD      COUNT
0F16     9CFA                          JNZ      LOOP
0F18     C400                          LDI      0
0F1A     07                            CAS                        ;TURN OFF LED
0F1B     90EA                          JMP      OVER

                                   ;INTERRUPT SERVICE

0F1D     C402       GO:                LDI      2
0F1F     07                            CAS                        ;TURN ON LED
0F20     C40F                          LDI      15
0F22     C805                          ST       COUNT
0F24     05                            IEN
0F25     3F                            XPPC     3                 ;RETURN TO MAIN PROG
0F26     90F5                          JMP      GO

                    ;DATA AREA

         0F29       COUNT:           .=.+1

         0000                        .END
```

Table C-2. Printout of "One-Shot" Program Using Type Command

```
-TF00 (CR)
0F00 04
0F01 C4
0F02 1C
0F03 33
0F04 C4
0F05 0F
0F06 37
0F07 C4
0F08 00
0F09 C8
0F0A 1E
0F0B 05
0F0C C0
0F0D 1B
0F0E 98
0F0F FC
0F10 C4
0F11 FF
0F12 8F
0F13 FF
0F14 B8
0F15 13
0F16 9C
0F17 FA
0F18 C4
0F19 00
0F1A 07
0F1B 90
0F1C EA
0F1D C4
0F1E 02
0F1F 07
0F20 C4
0F21 0F
0F22 C8
0F23 05
0F24 05
0F25 3F
0F26 90
0F27 F5
```