

!A

LLOAD SW16TEST.L,A\$4000

*** End of Pass 1

LLOAD SW16CODE.L,A\$4000

LLOAD SW16TEST.L,A\$4000

*** End of Pass 2

```
0800      1          ttl "SW16TEST Source Code, SW16TEST.L"
0800      2          src "SW16TEST.L"
0800      3      ;
0800      4      ;
0800      5      ; SW16TEST.L
0800      6      ;
0800      7      ;
0800      8      ; These routines test the various SWEET16 instructions and
0800      9      ; registers.
0800     10      ;
0800     11      ; 2024 February 14
0800     12      ;
0800     13      ;
0800     14      ; DOS 4.5, Build 06
0800     15      ;
0800     16      ; 2024 February 14
0800     17      ;
0800     18      ;
0800     19      ; Start of Source Code:  0x4000
0800     20      ; Start of Symbol List:  0x7800
0800     21      ;
0800     22      ;
0800     23      ; Copyright (c) 2024 February 14 by
0800     24      ; Walland Philip Vrbancic Jr
0800     25      ;
0800     26      ; 6223 East Peabody Street
0800     27      ; Long Beach, California  90808
0800     28      ; Unitied States of America
0800     29      ;
0800     30      ; All Rights Reserved
0800     31      ;
0800     32      ; This software is the confidential and
0800     33      ; proprietary intellectual property of
0800     34      ; Walland Philip Vrbancic Jr
0800     35      ;
0800     36      ;
0800     37      ;
0800     38      ; This version has been revised from the original.
0800     39      ;
0800     40      ;
0000     41      R0L      epz $00
0001     42      R0H      epz $01
0018     43      R12L     epz $18
0019     44      R12H     epz $19
001C     45      R14L     epz $1C
001D     46      R14H     epz $1D
001E     47      R15L     epz $1E
001F     48      R15H     epz $1F
0800     49      ;
0800     50      enz
0800     51      ;
0000     52      ZERO     equ $00
00FF     53      NEGONE   equ $FF
0800     54      ;
0800     55      ;
0800     56      ; Registers
0800     57      ;
0000     58      R0      equ $00          ; accumulator
0001     59      R1      equ $01          ; general purpose register
0002     60      R2      equ $02          ; general purpose register
```

```

0003      61  R3      equ $03      ; general purpose register
0004      62  R4      equ $04      ; general purpose register
0005      63  R5      equ $05      ; general purpose register
0006      64  R6      equ $06      ; general purpose register
0007      65  R7      equ $07      ; general purpose register
0008      66  R8      equ $08      ; general purpose register
0009      67  R9      equ $09      ; general purpose register
000A      68  R10     equ $0A      ; general purpose register
000B      69  R11     equ $0B      ; general purpose register
000C      70  R12     equ $0C      ; return address stack reg
000D      71  R13     equ $0D      ; comparands difference reg
000E      72  R14     equ $0E      ; status register
000F      73  R15     equ $0F      ; program counter register
0800      74  ;
0800      75  ;
0800      76  ; Non-register Opcodes (PR = prior register)
0800      77  ;
0000      78  RTN      equ $00      ; return to 6502 mode
0001      79  BR       equ $01      ; branch always
0002      80  BNC      equ $02      ; branch if PR carry clear
0003      81  BC       equ $03      ; branch if PR carry set
0004      82  BP       equ $04      ; branch if PR postive
0005      83  BM       equ $05      ; branch if PR negative
0006      84  BZ       equ $06      ; branch if PR zero
0007      85  BNZ      equ $07      ; branch if PR not zero
0008      86  BM1      equ $08      ; branch if PR minus one
0009      87  BNM1     equ $09      ; branch if PR not minus one
000A      88  SOUT     equ $0A      ; send "chr" to COUT
000B      89  RS       equ $0B      ; return from subroutine
000C      90  BS       equ $0C      ; branch to subroutine
000D      91  RSNS     equ $0D      ; return from subrtn, no stack
000E      92  BSNS     equ $0E      ; branch to subrtn, no stack
000F      93  SJMP     equ $0F      ; jump to address, PC=adr
0800      94  ;
0800      95  ;
0800      96  ; Register Opcodes (ACC = accumulator)
0800      97  ;
0010      98  SET      equ $10      ; value -> Rn
0020      99  LD       equ $20      ; Rn -> ACC
0030     100  ST       equ $30      ; ACC -> Rn
0040     101  LD@      equ $40      ; (Rn) -> LO ACC
0050     102  ST@      equ $50      ; LO ACC -> (Rn)
0060     103  LDD@     equ $60      ; (Rn) -> ACC
0070     104  STD@     equ $70      ; ACC -> (Rn)
0080     105  POP@     equ $80      ; (Rn-1) -> LO ACC
0090     106  STP@     equ $90      ; LO ACC -> (Rn-1)
00A0     107  ADD      equ $A0      ; ACC + Rn -> ACC
0800     108  ;          carry -> LO R14, bit 0
00B0     109  SUB      equ $B0      ; ACC - Rn -> ACC
0800     110  ;          carry -> LO R14, bit 0
00C0     111  POPD@   equ $C0      ; (Rn-1) -> HO ACC
0800     112  ;          (Rn-2) -> LO ACC
00D0     113  CPR      equ $D0      ; ACC - Rn -> R13
0800     114  ;          carry -> LO R14, bit 0
00E0     115  INR      equ $E0      ; Rn + 1 -> Rn
00F0     116  DCR      equ $F0      ; Rn - 1 -> Rn
0800     117  ;
0800     118  ;
0100     119  PAGESIZE equ $100
0800     120  ;
FD00     121  COUT     equ $FD00

```

```
0800          122 ;
FF3F          123 RESTORE equ $FF3F
FF4A          124 SAVE     equ $FF4A
0800          125 ;
0800          126 ;
0900          127         org $900
0900          128         obj $900
0900          129         usr
0900          130 ;
0900          131 ;
0900          132 ; Test COUT and RTN opcodes.
0900          133 ;
0900 20 70 0C  134 SOUTMAIN jsr SW16
0903 0A C1     135         byt SOUT,"A"
0905 0A 8D     136         byt SOUT,$8D
0907 00        137         byt RTN
0908          138 ;
0908 60        139         rts
0909          140 ;
0909          141 ;
0909          142 ; Test SET, ST@, LD, ST, DCR, LD@, and INR opcodes.
0909          143 ;
0909 20 70 0C  144 SETMAIN  jsr SW16
090C 11 00 03  145         .da #SET+R1,$300
090F 12 10 03  146         .da #SET+R2,$310
0912 10 00 00  147         .da #SET+R0,0
0915 13 34 12  148         .da #SET+R3,$1234
0918 14 76 98  149         .da #SET+R4,$9876
091B 3E        150         byt ST+R14
091C 23        151         byt LD+R3
091D 52        152         byt ST@+R2
091E 24        153         byt LD+R4
091F 52        154         byt ST@+R2
0920 F2        155         byt DCR+R2
0921 F2        156         byt DCR+R2
0922 42        157         byt LD@+R2
0923 E2        158         byt INR+R2
0924 52        159         byt ST@+R2
0925 2E        160         byt LD+R14
0926 71        161         byt STD@+R1
0927 00        162         byt RTN
0928          163 ;
0928 60        164         rts
0929          165 ;
0929          166 ;
0929          167 ; Test STD@ and LDD@.
0929          168 ;
0929 20 70 0C  169 STDMAIN  jsr SW16
092C 11 00 03  170         .da #SET+R1,$300
092F 12 10 03  171         .da #SET+R2,$310
0932 10 00 00  172         .da #SET+R0,0
0935 13 34 12  173         .da #SET+R3,$1234
0938 14 76 98  174         .da #SET+R4,$9876
093B 3E        175         byt ST+R14
093C 23        176         byt LD+R3
093D 72        177         byt STD@+R2
093E 24        178         byt LD+R4
093F 72        179         byt STD@+R2
0940 F2        180         byt DCR+R2
0941 F2        181         byt DCR+R2
0942 F2        182         byt DCR+R2
```

```

0943 F2          183          byt DCR+R2
0944 62          184          byt LDD@+R2
0945 E2          185          byt INR+R2
0946 E2          186          byt INR+R2
0947 72          187          byt STD@+R2
0948 2E          188          byt LD+R14
0949 71          189          byt STD@+R1
094A 00          190          byt RTN
094B            191          ;
094B 60          192          rts
094C            193          ;
094C            194          ;
094C            195          ; Test POP@, POPD@, and STP@.
094C            196          ;
094C 20 70 0C    197  POP@MAIN jsr SW16
094F 11 00 03    198          .da #SET+R1,$300
0952 12 10 03    199          .da #SET+R2,$310
0955 10 00 00    200          .da #SET+R0,0
0958 13 34 12    201          .da #SET+R3,$1234
095B 14 76 98    202          .da #SET+R4,$9876
095E 3E          203          byt ST+R14
095F 23          204          byt LD+R3
0960 52          205          byt ST@+R2
0961 22          206          byt LD+R2
0962 35          207          byt ST+R5
0963 24          208          byt LD+R4
0964 52          209          byt ST@+R2
0965 22          210          byt LD+R2
0966 36          211          byt ST+R6
0967 85          212          byt POP@+R5
0968 52          213          byt ST@+R2
0969 C6          214          byt POPD@+R6
096A 72          215          byt STD@+R2
096B 72          216          byt STD@+R2
096C 10 FF 00    217          .da #SET+R0,$FF
096F 92          218          byt STP@+R2
0970 2E          219          byt LD+R14
0971 71          220          byt STD@+R1
0972 00          221          byt RTN
0973            222          ;
0973 60          223          rts
0974            224          ;
0974            225          ;
0974            226          ; Test ADD, SUB, and CPR.
0974            227          ;
0974 20 70 0C    228  ADDMAIN  jsr SW16
0977 11 00 03    229          .da #SET+R1,$300
097A 12 10 03    230          .da #SET+R2,$310
097D 10 00 00    231          .da #SET+R0,0
0980 13 34 12    232          .da #SET+R3,$1234
0983 14 76 98    233          .da #SET+R4,$9876
0986 3E          234          byt ST+R14
0987 A3          235          byt ADD+R3
0988 72          236          byt STD@+R2
0989 A4          237          byt ADD+R4
098A 72          238          byt STD@+R2
098B B3          239          byt SUB+R3
098C 72          240          byt STD@+R2
098D B4          241          byt SUB+R4
098E 72          242          byt STD@+R2
098F 24          243          byt LD+R4

```

```

0990 D3          244          byt CPR+R3
0991 2D          245          byt LD+R13
0992 71          246          byt STD@+R1
0993 23          247          byt LD+R3
0994 D4          248          byt CPR+R4
0995 2D          249          byt LD+R13
0996 71          250          byt STD@+R1
0997 2E          251          byt LD+R14
0998 71          252          byt STD@+R1
0999 00          253          byt RTN
099A            254          ;
099A 60          255          rts
099B            256          ;
099B            257          ;
099B            258          ; Test BR.
099B            259          ;
099B 20 70 0C    260  BRMAIN   jsr SW16
099E 11 00 03    261          .da #SET+R1,$300
09A1 12 10 03    262          .da #SET+R2,$310
09A4 10 00 00    263          .da #SET+R0,0
09A7 13 34 12    264          .da #SET+R3,$1234
09AA 14 76 98    265          .da #SET+R4,$9876
09AD 15 A5 A5    266          .da #SET+R5,$A5A5
09B0 3E          267          byt ST+R14
09B1 23          268          byt LD+R3
09B2 01 01       269          byt BR,MYBR-*+1
09B4 25          270          byt LD+R5
09B5 72          271  MYBR     byt STD@+R2
09B6 2E          272          byt LD+R14
09B7 71          273          byt STD@+R1
09B8 00          274          byt RTN
09B9            275          ;
09B9 60          276          rts
09BA            277          ;
09BA            278          ;
09BA            279          ; Test BNC.
09BA            280          ;
09BA 20 70 0C    281  BNCMAIN   jsr SW16
09BD 11 00 03    282          .da #SET+R1,$300
09C0 12 10 03    283          .da #SET+R2,$310
09C3 10 00 00    284          .da #SET+R0,0
09C6 13 34 12    285          .da #SET+R3,$1234
09C9 14 76 98    286          .da #SET+R4,$9876
09CC 15 A5 A5    287          .da #SET+R5,$A5A5
09CF 3E          288          byt ST+R14
09D0 23          289          byt LD+R3
09D1 D4          290          byt CPR+R4
09D2 02 01       291          byt BNC,MYBNC-*+1
09D4 25          292          byt LD+R5
09D5 72          293  MYBNC     byt STD@+R2
09D6 2E          294          byt LD+R14
09D7 71          295          byt STD@+R1
09D8 00          296          byt RTN
09D9            297          ;
09D9 60          298          rts
09DA            299          ;
09DA            300          ;
09DA            301          ; Test BC.
09DA            302          ;
09DA 20 70 0C    303  BCMAIN     jsr SW16
09DD 11 00 03    304          .da #SET+R1,$300

```

```

09E0 12 10 03      305      .da #SET+R2,$310
09E3 10 00 00      306      .da #SET+R0,0
09E6 13 34 12      307      .da #SET+R3,$1234
09E9 14 76 98      308      .da #SET+R4,$9876
09EC 15 A5 A5      309      .da #SET+R5,$A5A5
09EF 3E            310      byt ST+R14
09F0 24            311      byt LD+R4
09F1 D3            312      byt CPR+R3
09F2 03 01         313      byt BC,MYBC-*+1
09F4 25            314      byt LD+R5
09F5 72            315      MYBC byt STD@+R2
09F6 2E            316      byt LD+R14
09F7 71            317      byt STD@+R1
09F8 00            318      byt RTN
09F9              319      ;
09F9 60            320      rts
09FA              321      ;
09FA              322      ;
09FA              323      ; Test BP.
09FA              324      ;
09FA 20 70 0C      325      BPMMAIN jsr SW16
09FD 11 00 03      326      .da #SET+R1,$300
0A00 12 10 03      327      .da #SET+R2,$310
0A03 10 00 00      328      .da #SET+R0,0
0A06 13 34 12      329      .da #SET+R3,$1234
0A09 14 76 98      330      .da #SET+R4,$9876
0A0C 15 A5 A5      331      .da #SET+R5,$A5A5
0A0F 3E            332      byt ST+R14
0A10 24            333      byt LD+R4
0A11 B3            334      byt SUB+R3
0A12 71            335      byt STD@+R1
0A13 B3            336      byt SUB+R3
0A14 71            337      byt STD@+R1
0A15 04 01         338      byt BP,MYBP-*+1
0A17 25            339      byt LD+R5
0A18 72            340      MYBP byt STD@+R2
0A19 2E            341      byt LD+R14
0A1A 71            342      byt STD@+R1
0A1B 00            343      byt RTN
0A1C              344      ;
0A1C 60            345      rts
0A1D              346      ;
0A1D              347      ;
0A1D              348      ; Test BM.
0A1D              349      ;
0A1D 20 70 0C      350      BMMAIN jsr SW16
0A20 11 00 03      351      .da #SET+R1,$300
0A23 12 10 03      352      .da #SET+R2,$310
0A26 10 00 00      353      .da #SET+R0,0
0A29 13 34 12      354      .da #SET+R3,$1234
0A2C 14 76 98      355      .da #SET+R4,$9876
0A2F 15 A5 A5      356      .da #SET+R5,$A5A5
0A32 3E            357      byt ST+R14
0A33 B3            358      byt SUB+R3
0A34 71            359      byt STD@+R1
0A35 05 01         360      byt BM,MYBM-*+1
0A37 25            361      byt LD+R5
0A38 72            362      MYBM byt STD@+R2
0A39 2E            363      byt LD+R14
0A3A 71            364      byt STD@+R1
0A3B 00            365      byt RTN

```

```

0A3C          366 ;
0A3C 60       367          rts
0A3D          368 ;
0A3D          369 ;
0A3D          370 ; Test BZ.
0A3D          371 ;
0A3D 20 70 0C 372 BZMAIN    jsr SW16
0A40 11 00 03 373          .da #SET+R1,$300
0A43 12 10 03 374          .da #SET+R2,$310
0A46 10 00 00 375          .da #SET+R0,0
0A49 13 34 12 376          .da #SET+R3,$1234
0A4C 14 76 98 377          .da #SET+R4,$9876
0A4F 15 A5 A5 378          .da #SET+R5,$A5A5
0A52 3E       379          byt ST+R14
0A53 24       380          byt LD+R4
0A54 B4       381          byt SUB+R4
0A55 71       382          byt STD@+R1
0A56 06 01    383          byt BZ,MYBZ-#+1
0A58 25       384          byt LD+R5
0A59 72       385 MYBZ     byt STD@+R2
0A5A 2E       386          byt LD+R14
0A5B 71       387          byt STD@+R1
0A5C 00       388          byt RTN
0A5D          389 ;
0A5D 60       390          rts
0A5E          391 ;
0A5E          392 ;
0A5E          393 ; Test BNZ.
0A5E          394 ;
0A5E 20 70 0C 395 BNZMAIN   jsr SW16
0A61 11 00 03 396          .da #SET+R1,$300
0A64 12 10 03 397          .da #SET+R2,$310
0A67 10 00 00 398          .da #SET+R0,0
0A6A 13 34 12 399          .da #SET+R3,$1234
0A6D 14 76 98 400          .da #SET+R4,$9876
0A70 15 A5 A5 401          .da #SET+R5,$A5A5
0A73 3E       402          byt ST+R14
0A74 24       403          byt LD+R4
0A75 B3       404          byt SUB+R3
0A76 71       405          byt STD@+R1
0A77 07 01    406          byt BNZ,MYBNZ-#+1
0A79 25       407          byt LD+R5
0A7A 72       408 MYBNZ    byt STD@+R2
0A7B 2E       409          byt LD+R14
0A7C 71       410          byt STD@+R1
0A7D 00       411          byt RTN
0A7E          412 ;
0A7E 60       413          rts
0A7F          414 ;
0A7F          415 ;
0A7F          416 ; Test BM1.
0A7F          417 ;
0A7F 20 70 0C 418 BM1MAIN   jsr SW16
0A82 11 00 03 419          .da #SET+R1,$300
0A85 12 10 03 420          .da #SET+R2,$310
0A88 10 00 00 421          .da #SET+R0,0
0A8B 13 34 12 422          .da #SET+R3,$1234
0A8E 14 76 98 423          .da #SET+R4,$9876
0A91 15 A5 A5 424          .da #SET+R5,$A5A5
0A94 3E       425          byt ST+R14
0A95 24       426          byt LD+R4

```



```

0A96 B4          427          byt SUB+R4
0A97 36          428          byt ST+R6
0A98 F6          429          byt DCR+R6
0A99 26          430          byt LD+R6
0A9A 71          431          byt STD@+R1
0A9B 08 01       432          byt BM1,MYBM1-#+1
0A9D 25          433          byt LD+R5
0A9E 72          434 MYBM1    byt STD@+R2
0A9F 2E          435          byt LD+R14
0AA0 71          436          byt STD@+R1
0AA1 00          437          byt RTN
0AA2             438          ;
0AA2 60          439          rts
0AA3             440          ;
0AA3             441          ;
0AA3             442          ; Test BNM1.
0AA3             443          ;
0AA3 20 70 0C    444 BNM1MAIN jsr SW16
0AA6 11 00 03    445          .da #SET+R1,$300
0AA9 12 10 03    446          .da #SET+R2,$310
0AAC 10 00 00    447          .da #SET+R0,0
0AAF 13 34 12    448          .da #SET+R3,$1234
0AB2 14 76 98    449          .da #SET+R4,$9876
0AB5 15 A5 A5    450          .da #SET+R5,$A5A5
0AB8 3E          451          byt ST+R14
0AB9 24          452          byt LD+R4
0ABA B4          453          byt SUB+R4
0ABB 36          454          byt ST+R6
0ABC E6          455          byt INR+R6
0ABD 26          456          byt LD+R6
0ABE 71          457          byt STD@+R1
0ABF 09 01       458          byt BNM1,MYBNM1-#+1
0AC1 25          459          byt LD+R5
0AC2 72          460 MYBNM1   byt STD@+R2
0AC3 2E          461          byt LD+R14
0AC4 71          462          byt STD@+R1
0AC5 00          463          byt RTN
0AC6             464          ;
0AC6 60          465          rts
0AC7             466          ;
0AC7             467          ;
0AC7             468          ; Test BS and RS opcodes.
0AC7             469          ;
0AC7             470 PRT1SUB:
0AC7 2C          471          byt LD+R12
0AC8 72          472          byt STD@+R2
0AC9 0A C1       473          byt SOUT,"A"
0ACB 0A C2       474          byt SOUT,"B"
0ACD 0A C3       475          byt SOUT,"C"
0ACF 0A 8D       476          byt SOUT,$8D
0AD1 0B          477          byt RS
0AD2             478          ;
0AD2             479 SUB2:
0AD2 2C          480          byt LD+R12
0AD3 72          481          byt STD@+R2
0AD4 F3          482          byt DCR+R3
0AD5 F4          483          byt DCR+R4
0AD6 F5          484          byt DCR+R5
0AD7 0B          485          byt RS
0AD8             486          ;
0AD8 20 70 0C    487 BSM1AIN  jsr SW16

```

```

0ADB 10 00 00    488      .da #SET+R0,0
0ADE 3E          489      byt ST+R14
0ADF 11 10 03    490      .da #SET+R1,$310
0AE2 12 20 03    491      .da #SET+R2,$320
0AE5 13 34 12    492      .da #SET+R3,$1234
0AE8 14 76 98    493      .da #SET+R4,$9876
0AEB 15 A5 A5    494      .da #SET+R5,$A5A5
0AEE 1C 00 03    495      .da #SET+R12,$300      ; stack
0AF1 0C D4       496      byt BS,PRT1SUB-+1
0AF3 B0          497      byt SUB+R0
0AF4 F0          498      byt DCR+R0
0AF5 71          499      byt STD@+R1
0AF6 0C 0C       500      byt BS,SUB1-+1
0AF8 23          501      byt LD+R3
0AF9 71          502      byt STD@+R1
0AFA 24          503      byt LD+R4
0AFB 71          504      byt STD@+R1
0AFC 25          505      byt LD+R5
0AFD 71          506      byt STD@+R1
0AFE 0C C7       507      byt BS,PRT1SUB-+1
0B00 2E          508      byt LD+R14
0B01 71          509      byt STD@+R1
0B02 00          510      byt RTN
0B03             511      ;
0B03 60          512      rts
0B04             513      ;
0B04             514      SUB1:
0B04 2C          515      byt LD+R12
0B05 72          516      byt STD@+R2
0B06 0C 09       517      byt BS,PRT2SUB-+1
0B08 F3          518      byt DCR+R3
0B09 F4          519      byt DCR+R4
0B0A F5          520      byt DCR+R5
0B0B 0C C5       521      byt BS,SUB2-+1
0B0D B0          522      byt SUB+R0
0B0E F0          523      byt DCR+R0
0B0F 71          524      byt STD@+R1
0B10 0B          525      byt RS
0B11             526      ;
0B11             527      PRT2SUB:
0B11 2C          528      byt LD+R12
0B12 72          529      byt STD@+R2
0B13 0C B2       530      byt BS,PRT1SUB-+1
0B15 0A F8       531      byt SOUT,"x"
0B17 0A F9       532      byt SOUT,"y"
0B19 0A FA       533      byt SOUT,"z"
0B1B 0A 8D       534      byt SOUT,$8D
0B1D 0B          535      byt RS
0B1E             536      ;
0B1E             537      ;
0B1E             538      ; Test BSNS and RSNX opcodes.
0B1E             539      ;
0B1E             540      PRT3SUB:
0B1E 2C          541      byt LD+R12
0B1F 72          542      byt STD@+R2
0B20 0A C1       543      byt SOUT,"A"
0B22 0A C2       544      byt SOUT,"B"
0B24 0A C3       545      byt SOUT,"C"
0B26 0A 8D       546      byt SOUT,$8D
0B28 0D          547      byt RSNS
0B29             548      ;

```

```

0B29 20 70 0C    549  BSNSMAIN  jsr  SW16
0B2C 10 00 00    550          .da  #SET+R0,0
0B2F 3E          551          byt  ST+R14
0B30 11 10 03    552          .da  #SET+R1,$310
0B33 12 20 03    553          .da  #SET+R2,$320
0B36 13 34 12    554          .da  #SET+R3,$1234
0B39 14 76 98    555          .da  #SET+R4,$9876
0B3C 15 A5 A5    556          .da  #SET+R5,$A5A5
0B3F 1C 00 03    557          .da  #SET+R12,$300    ; stack
0B42 0E DA       558          byt  BSNS,PRT3SUB-#+1
0B44 B0          559          byt  SUB+R0
0B45 F0          560          byt  DCR+R0
0B46 71          561          byt  STD@+R1
0B47 0E 0C       562          byt  BSNS,SUB3-#+1
0B49 23          563          byt  LD+R3
0B4A 71          564          byt  STD@+R1
0B4B 24          565          byt  LD+R4
0B4C 71          566          byt  STD@+R1
0B4D 25          567          byt  LD+R5
0B4E 71          568          byt  STD@+R1
0B4F 0E CD       569          byt  BSNS,PRT3SUB-#+1
0B51 2E          570          byt  LD+R14
0B52 71          571          byt  STD@+R1
0B53 00          572          byt  RTN
0B54            573          ;
0B54 60          574          rts
0B55            575          ;
0B55            576  SUB3:
0B55 2C          577          byt  LD+R12
0B56 72          578          byt  STD@+R2
0B57 F3          579          byt  DCR+R3
0B58 F4          580          byt  DCR+R4
0B59 F5          581          byt  DCR+R5
0B5A B0          582          byt  SUB+R0
0B5B F0          583          byt  DCR+R0
0B5C 71          584          byt  STD@+R1
0B5D 0D          585          byt  RSNS
0B5E            586          ;
0B5E            587          ;
0B5E            588          ; Test SJMP opcode.
0B5E            589          ;
0B5E            590  PART2:
0B5E 2F          591          byt  LD+R15
0B5F 72          592          byt  STD@+R2
0B60 0A C1       593          byt  SOUT,"A"
0B62 0A C2       594          byt  SOUT,"B"
0B64 0A C3       595          byt  SOUT,"C"
0B66 0A 8D       596          byt  SOUT,$8D
0B68 0F 9A 0B    597          .da  #SJMP,PART3
0B6B            598          ;
0B6B 20 70 0C    599  SJMPMAIN  jsr  SW16
0B6E 10 00 00    600          .da  #SET+R0,0
0B71 3E          601          byt  ST+R14
0B72 11 10 03    602          .da  #SET+R1,$310
0B75 12 20 03    603          .da  #SET+R2,$320
0B78 13 34 12    604          .da  #SET+R3,$1234
0B7B 14 76 98    605          .da  #SET+R4,$9876
0B7E 15 A5 A5    606          .da  #SET+R5,$A5A5
0B81 1C 00 03    607          .da  #SET+R12,$300    ; stack
0B84 0E 98       608          byt  BSNS,PRT3SUB-#+1
0B86 B0          609          byt  SUB+R0

```

```
0B87 F0          610          byt DCR+R0
0B88 71          611          byt STD@+R1
0B89 0E CA       612          byt BSNS,SUB3-++1
0B8B 23          613          byt LD+R3
0B8C 71          614          byt STD@+R1
0B8D 24          615          byt LD+R4
0B8E 71          616          byt STD@+R1
0B8F 25          617          byt LD+R5
0B90 71          618          byt STD@+R1
0B91 0F 5E 0B    619          .da #SJMP,PART2
0B94            620          ;
0B94            621          PART4:
0B94 2F          622          byt LD+R15
0B95 72          623          byt STD@+R2
0B96 2E          624          byt LD+R14
0B97 71          625          byt STD@+R1
0B98 00          626          byt RTN
0B99            627          ;
0B99 60          628          rts
0B9A            629          ;
0B9A            630          PART3:
0B9A 2F          631          byt LD+R15
0B9B 72          632          byt STD@+R2
0B9C F3          633          byt DCR+R3
0B9D F4          634          byt DCR+R4
0B9E F5          635          byt DCR+R5
0B9F B0          636          byt SUB+R0
0BA0 F0          637          byt DCR+R0
0BA1 71          638          byt STD@+R1
0BA2 0F 94 0B    639          .da #SJMP,PART4
0BA5            640          ;
0BA5            641          ;
0BA5            642          icl "SW16CODE.L"
```

```
LLOAD SW16CODE.L,A$4000
```

```

0BA5          1          ttl "SW16CODE Source Code, SW16CODE.L"
0BA5          2          ;
0BA5          3          ;
0BA5          4          ; SW16CODE.L
0BA5          5          ;
0BA5          6          ;
0BA5          7          ; This is the SWEET16 processing code.
0BA5          8          ;
0BA5          9          ;
0BA5         10          dfs PAGESIZE-*)&NEGONE,ZERO
0C00         11          dfs $70-*)&NEGONE,ZERO
0C70         12          ;
0C70         13          ;
0C70         14          ; Preserve 6502 registers and init SW16 program counter.
0C70         15          ;
0C70 20 4A FF    16  SW16      jsr SAVE
0C73         17          ;
0C73 68         18          pla
0C74 85 1E      19          sta R15L
0C76         20          ;
0C76 68         21          pla
0C77 85 1F      22          sta R15H
0C79         23          ;
0C79         24          ;
0C79         25          ; Interpret a single SW16 opcode, then fetch the next SW16
0C79         26          ; opcode.
0C79         27          ;
0C79 20 7F 0C   28  SW16B     jsr SW16C
0C7C         29          ;
0C7C 4C 79 0C   30          jmp SW16B
0C7F         31          ;
0C7F         32          ;
0C7F         33          ; Increment the SW16 program counter.
0C7F         34          ;
0C7F E6 1E      35  SW16C     inc R15L
0C81 D0 02      36          bne SW16D
0C83         37          ;
0C83 E6 1F      38          inc R15H
0C85         39          ;
0C85         40          ;
0C85         41          ; Push the common H0 routine address byte onto the stack
0C85         42          ; and process the SW16 opcode.
0C85         43          ;
0C85 A9 0D      44  SW16D     lda /SETCMD
0C87 48         45          pha
0C88         46          ;
0C88 A0 00      47          ldy #ZERO
0C8A         48          ;
0C8A         49          ;
0C8A         50          ; Mask the specified SW16 register and double it for
0C8A         51          ; non-register address table indexing or prior register.
0C8A         52          ;
0C8A B1 1E      53          lda (R15L),Y
0C8C 29 0F      54          and #$0F
0C8E         55          ;
0C8E 0A         56          asl
0C8F AA        57          tax
0C90         58          ;
0C90         59          ;
0C90         60          ; Extract opcode.  If it is zero then process a

```

```

0C90          61 ; non-register opcode routine.
0C90          62 ;
0C90 4A       63 ; lsr
0C91          64 ;
0C91 51 1E    65 ; eor (R15L),Y
0C93 F0 0D    66 ; beq TOBR
0C95          67 ;
0C95          68 ;
0C95          69 ; Save register*2 in LO status byte and clear HO status
0C95          70 ; byte. Form opcode*2 for register address table indexing.
0C95          71 ;
0C95 86 1C    72 ; stx R14L
0C97 84 1D    73 ; sty R14H
0C99          74 ;
0C99 4A       75 ; lsr
0C9A 4A       76 ; lsr
0C9B 4A       77 ; lsr
0C9C          78 ;
0C9C A8       79 ; tay
0C9D          80 ;
0C9D          81 ;
0C9D          82 ; Save the LO address onto the stack to process this SW16
0C9D          83 ; register opcode. Also C-flag = 0; N-flag = 0 for SETCMD.
0C9D          84 ;
0C9D B9 E1 0C 85 ; lda OPTBL-2,Y
0CA0 48       86 ; pha
0CA1          87 ;
0CA1 60       88 ; rts
0CA2          89 ;
0CA2          90 ;
0CA2          91 ; Increment the SW16 program counter for the expected
0CA2          92 ; branch address. Save the LO address onto the stack to
0CA2          93 ; process this SW16 non-register opcode.
0CA2          94 ;
0CA2 E6 1E    95 TOBR inc R15L
0CA4 D0 02    96 ; bne TOBR2
0CA6          97 ;
0CA6 E6 1F    98 ; inc R15H
0CA8          99 ;
0CA8 BD E2 0C 100 TOBR2 lda BRTBL,X
0CAB 48       101 ; pha
0CAC          102 ;
0CAC          103 ;
0CAC          104 ; Recall prior register*2 in LO status byte and set up
0CAC          105 ; carry flag from HO status byte for BC and BNC opcodes.
0CAC          106 ; Also N-flag = 0 and Y-reg = 0.
0CAC          107 ;
0CAC A6 1C    108 ; ldx R14L
0CAE          109 ;
0CAE A5 1D    110 ; lda R14H
0CB0 4A       111 ; lsr
0CB1          112 ;
0CB1 60       113 ; rts
0CB2          114 ;
0CB2          115 ;
0CB2          116 ; Y-reg set to 2 by SET opcode index*2. First get HO byte,
0CB2          117 ; then LO byte for designated register.
0CB2          118 ;
0CB2 B1 1E    119 SETZ lda (R15L),Y
0CB4 95 01    120 ; sta R0H,X
0CB6          121 ;

```

```

0CB6 88          122          dey
0CB7            123          ;
0CB7 B1 1E      124          lda (R15L),Y
0CB9 95 00      125          sta R0L,X
0CBB            126          ;
0CBB            127          ;
0CBB            128          ; Increment the SW16 program counter by 2.  C-flag = 0.
0CBB            129          ;
0CBB A5 1E      130          lda R15L
0CBD 69 02      131          adc #2
0CBF 85 1E      132          sta R15L
0CC1            133          ;
0CC1 90 02      134          bcc SET2
0CC3            135          ;
0CC3 E6 1F      136          inc R15H
0CC5            137          ;
0CC5 60          138 SET2    rts
0CC6            139          ;
0CC6            140          ;
0CC6            141          ; R12 must contain the stack address where to pop the
0CC6            142          ; return SW16 program counter.  Pop the HO byte, then
0CC6            143          ; the LO byte.  Then R12 - 2 -> R12.
0CC6            144          ;
0CC6            145          ;
0CC6 A2 18      146 RSZ     ldx #R12L
0CC8            147          ;
0CC8 20 F7 0D    148          jsr DCRCMD
0CCB            149          ;
0CCB A1 00      150          lda (R0L,X)
0CCD 85 1F      151          sta R15H
0CCF            152          ;
0CCF 20 F7 0D    153          jsr DCRCMD
0CD2            154          ;
0CD2 A1 00      155          lda (R0L,X)
0CD4 85 1E      156          sta R15L
0CD6            157          ;
0CD6 60          158          rts
0CD7            159          ;
0CD7            160          ;
0CD7            161          ; Copy the current SW16 program counter in R15 to R12.
0CD7            162          ;
0CD7 A5 1E      163 BSNSZ   lda R15L
0CD9 85 18      164          sta R12L
0CDB            165          ;
0CDB A5 1F      166          lda R15H
0CDD 85 19      167          sta R12H
0CDF            168          ;
0CDF 4C 9C 0D    169          jmp BRCMD
0CE2            170          ;
0CE2            171          ;
0CE2            172          ; The byte-pairs of this table have been swapped to remove
0CE2            173          ; the unused Fn+1 entry.
0CE2            174          ;
0CE2 00          175 BRTBL   byt RTNCMD-1          ; 0
0CE3 08          176 OPTBL   byt SETCMD-1          ; 1n
0CE4 9B          177          byt BRCMD-1          ; 1
0CE5 0E          178          byt LDCMD-1          ; 2n
0CE6 9C          179          byt BNCCMD-1         ; 2
0CE7 17          180          byt STCMD-1          ; 3n
0CE8 AD          181          byt BCCMD-1          ; 3
0CE9 20          182          byt LD@CMD-1         ; 4n

```

```

0CEA B0      183      byt BPCMD-1      ; 4
0CEB 2A      184      byt ST@CMD-1      ; 5n
0CEC B5      185      byt BMCMD-1      ; 5
0CED 39      186      byt LDD@CMD-1      ; 6n
0CEE BA      187      byt BZCMD-1      ; 6
0CEF 42      188      byt STD@CMD-1      ; 7n
0CF0 C1      189      byt BNZCMD-1      ; 7
0CF1 52      190      byt POP@CMD-1      ; 8n
0CF2 C8      191      byt BM1CMD-1      ; 8
0CF3 62      192      byt STP@CMD-1      ; 9n
0CF4 D1      193      byt BNM1CMD-1      ; 9
0CF5 6B      194      byt ADDCMD-1      ; An
0CF6 DA      195      byt SOUTCMD-1      ; A
0CF7 79      196      byt SUBCMD-1      ; Bn
0CF8 0A      197      byt RSCMD-1      ; B
0CF9 4B      198      byt POPD@CMD-1      ; Cn
0CFA 8F      199      byt BSCMD-1      ; C
0CFB 7B      200      byt CPRCMD-1      ; Dn
0CFC DF      201      byt RSNSCMD-1      ; D
0CFD 32      202      byt INRCMD-1      ; En
0CFE 0C      203      byt BSNSCMD-1      ; E
0CFF F6      204      byt DCRCMD-1      ; Fn
0D00 E8      205      byt SJMPCMD-1      ; F
0D01      206      ;
0D01      207      ;
0D01      208      ; These routines must reside on the same page.
0D01      209      ;
0D01      210      ;
0D01      211      ; RTN opcode returns to 6502 processing mode. Pop the
0D01      212      ; SW16C return address, restore the 6502 registers, and
0D01      213      ; return to 6502 mode using the SW16 program counter.
0D01      214      ;
0D01 68      215      RTNCMD    pla
0D02 68      216      pla
0D03      217      ;
0D03 20 3F FF 218      jsr RESTORE
0D06      219      ;
0D06 6C 1E 00 220      jmp (R15L)
0D09      221      ;
0D09      222      ;
0D09      223      ; SET opcode gets a 2-byte immediate constant.
0D09      224      ;
0D09 10 A7    225      SETCMD    bpl SETZ      ; always taken
0D0B      226      ;
0D0B      227      ;
0D0B      228      ; RS opcode is return from a BS subroutine call.
0D0B      229      ;
0D0B 10 B9    230      RSCMD     bpl RSZ      ; always taken
0D0D      231      ;
0D0D      232      ;
0D0D      233      ; BSNS opcode is branch to subroutine using no address
0D0D      234      ; stack.
0D0D      235      ;
0D0D 10 C8    236      BSNSCMD   bpl BSNSZ      ; always taken
0D0F      237      ;
0D0F      238      ;
0D0F      239      ; LD opcode moves the contents of Rn to ACC.
0D0F      240      ;
0D0F B5 00    241      LDCMD     lda R0L,X
0D11 85 00    242      sta R0L
0D13      243      ;

```



```

0D13 B5 01      244      lda R0H,X
0D15 85 01      245      sta R0H
0D17            246      ;
0D17 60          247      rts
0D18            248      ;
0D18            249      ;
0D18            250      ; ST opcode moves the contents of ACC to Rn.
0D18            251      ;
0D18 A5 00      252 STCMD   lda R0L
0D1A 95 00      253      sta R0L,X
0D1C            254      ;
0D1C A5 01      255      lda R0H
0D1E 95 01      256      sta R0H,X
0D20            257      ;
0D20 60          258      rts
0D21            259      ;
0D21            260      ;
0D21            261      ; LD@ opcode loads the LO ACC indirectly from memory
0D21            262      ; using the address in Rn. HO ACC is cleared. Make R0
0D21            263      ; the prior register.
0D21            264      ;
0D21 A1 00      265 LD@CMD   lda (R0L,X)
0D23 85 00      266      sta R0L
0D25            267      ;
0D25 A0 00      268      ldy #ZERO
0D27 84 01      269      sty R0H
0D29            270      ;
0D29 F0 06      271      beq ST@3          ; always taken
0D2B            272      ;
0D2B            273      ;
0D2B            274      ; ST@ opcode stores the contents of LO ACC indirectly to
0D2B            275      ; memory using the address in Rn. Make R0 the prior
0D2B            276      ; register. Fall into INRCMD.
0D2B            277      ;
0D2B A5 00      278 ST@CMD   lda R0L
0D2D            279      ;
0D2D 81 00      280 ST@2     sta (R0L,X)
0D2F            281      ;
0D2F A0 00      282      ldy #ZERO
0D31            283      ;
0D31 84 1C      284 ST@3     sty R14L
0D33            285      ;
0D33            286      ;
0D33            287      ; INR opcode increments the register Rn.
0D33            288      ;
0D33 F6 00      289 INRCMD   inc R0L,X
0D35 D0 02      290      bne INR2
0D37            291      ;
0D37 F6 01      292      inc R0H,X
0D39            293      ;
0D39 60          294 INR2     rts
0D3A            295      ;
0D3A            296      ;
0D3A            297      ; LDD@ opcode loads the ACC indirectly from memory using
0D3A            298      ; the address in Rn. The LO byte is loaded first. Make
0D3A            299      ; R0 the prior register.
0D3A            300      ;
0D3A 20 21 0D   301 LDD@CMD   jsr LD@CMD
0D3D            302      ;
0D3D A1 00      303      lda (R0L,X)
0D3F 85 01      304      sta R0H

```

```

0D41          305 ;
0D41 90 F0    306          bcc INRCMD          ; always taken
0D43          307 ;
0D43          308 ;
0D43          309 ; STD@ opcode stores the contents of the ACC indirectly to
0D43          310 ; memory using the address in Rn. LO ACC is stored first.
0D43          311 ; Make R0 the prior register.
0D43          312 ;
0D43 20 2B 0D 313 STD@CMD  jsr ST@CMD
0D46          314 ;
0D46 A5 01    315          lda R0H
0D48 81 00    316          sta (R0L,X)
0D4A          317 ;
0D4A 90 E7    318          bcc INRCMD          ; always taken
0D4C          319 ;
0D4C          320 ;
0D4C          321 ; POPD@ opcode decrements Rn and gets the H0 byte
0D4C          322 ; indirectly from memory using the address in Rn.
0D4C          323 ; Fall into POP@CMD.
0D4C          324 ;
0D4C 20 F7 0D 325 POPD@CMD jsr DCRCMD
0D4F          326 ;
0D4F A1 00    327          lda (R0L,X)
0D51 A8       328          tay
0D52          329 ;
0D52 2C 00 00 330          bit *-*
0D55          331          dfs !-2
0D53          332 ;
0D53          333 ;
0D53          334 ; POP@ opcode sets the H0 byte to zero, then decrements Rn
0D53          335 ; and gets the LO byte indirectly from memory using the
0D53          336 ; address in Rn. Store the H0 and LO bytes into the ACC.
0D53          337 ; Make R0 the prior register.
0D53          338 ;
0D53 A0 00    339 POP@CMD  ldy #ZERO
0D55          340 ;
0D55 20 F7 0D 341          jsr DCRCMD
0D58          342 ;
0D58 A1 00    343          lda (R0L,X)
0D5A 85 00    344          sta R0L
0D5C          345 ;
0D5C 84 01    346          sty R0H
0D5E          347 ;
0D5E A0 00    348 POP@3    ldy #ZERO
0D60 84 1C    349          sty R14L
0D62          350 ;
0D62 60       351          rts
0D63          352 ;
0D63          353 ;
0D63          354 ; STP@ opcode decrements Rn and stores the LO ACC
0D63          355 ; indirectly to memory using the address in Rn. Make R0
0D63          356 ; the prior register.
0D63          357 ;
0D63 20 F7 0D 358 STP@CMD  jsr DCRCMD
0D66          359 ;
0D66 A5 00    360          lda R0L
0D68 81 00    361          sta (R0L,X)
0D6A          362 ;
0D6A 90 F2    363          bcc POP@3          ; always taken
0D6C          364 ;
0D6C          365 ;

```

```

0D6C      366 ; ADD opcode sets Y-reg to 0 so ACC + Rn -> ACC, carry
0D6C      367 ; saved to HO status byte. Make R0 the prior register.
0D6C      368 ;
0D6C A5 00 369 ADDCMD   lda R0L
0D6E 75 00 370          adc R0L,X
0D70 85 00 371          sta R0L
0D72      372 ;
0D72 A5 01 373          lda R0H
0D74 75 01 374          adc R0H,X
0D76      375 ;
0D76 A0 00 376          ldy #ZERO
0D78 F0 0E 377          beq CPR2          ; always taken
0D7A      378 ;
0D7A      379 ;
0D7A      380 ; SUB opcode sets Y-reg to 0 so ACC - Rn -> ACC, carry
0D7A      381 ; saved to HO status byte. Make R0 the prior register.
0D7A      382 ;
0D7A A0 00 383 SUBCMD   ldy #ZERO
0D7C      384 ;
0D7C      385 ;
0D7C      386 ; CPR opcode leaves Y-reg set to 13*2 (SET opcode index*2)
0D7C      387 ; so ACC - Rn -> R13, carry saved to HO status byte. Make
0D7C      388 ; R13 the prior register.
0D7C      389 ;
0D7C 38    390 CPRCMD   sec
0D7D      391 ;
0D7D A5 00 392          lda R0L
0D7F F5 00 393          sbc R0L,X
0D81 99 00 00 394          sta R0L,Y
0D84      395 ;
0D84 A5 01 396          lda R0H
0D86 F5 01 397          sbc R0H,X
0D88      398 ;
0D88 99 01 00 399 CPR2   sta R0H,Y
0D8B      400 ;
0D8B 84 1C 401          sty R14L
0D8D      402 ;
0D8D      403 ;
0D8D      404 ; Save carry out bit.
0D8D      405 ;
0D8D 26 1D 406          rol R14H
0D8F      407 ;
0D8F 60    408          rts
0D90      409 ;
0D90      410 ;
0D90      411 ; BS opcode is branch to subroutine. Save the current
0D90      412 ; SW16 program counter indirectly to memory using the
0D90      413 ; address in R12. R15L is saved first, then R15H; finally
0D90      414 ; R12 + 2 -> R12. Fall into BRCMD.
0D90      415 ;
0D90 A2 18 416 BSCMD   ldx #R12L
0D92      417 ;
0D92 A5 1E 418          lda R15L
0D94 20 2D 0D 419          jsr ST@2
0D97      420 ;
0D97 A5 1F 421          lda R15H
0D99 20 2D 0D 422          jsr ST@2
0D9C      423 ;
0D9C      424 ;
0D9C      425 ; BR opcode is branch always. Fall into BNCCMD.
0D9C      426 ;

```

```

0D9C 18          427 BRCMD      clc
0D9D            428 ;
0D9D            429 ;
0D9D            430 ; BNC opcode is branch if carry is clear from prior opcode.
0D9D            431 ;
0D9D B0 0E       432 BNCCMD     bcs BRTS
0D9F            433 ;
0D9F            434 ;
0D9F            435 ; Get displacement byte. If it is negative set Y-reg to
0D9F            436 ; minus one for 2's compliment addition. Y-reg = 0.
0D9F            437 ;
0D9F B1 1E       438             lda (R15L),Y
0DA1 10 01       439             bpl BR2
0DA3            440 ;
0DA3 88          441             dey
0DA4            442 ;
0DA4            443 ;
0DA4            444 ; Add displacement to program counter.
0DA4            445 ;
0DA4 65 1E       446 BR2        adc R15L
0DA6 85 1E       447             sta R15L
0DA8            448 ;
0DA8 98          449             tya
0DA9            450 ;
0DA9 65 1F       451             adc R15H
0DAB 85 1F       452             sta R15H
0DAD            453 ;
0DAD 60          454 BRTS       rts
0DAE            455 ;
0DAE            456 ;
0DAE            457 ; BC opcode is branch if carry is set from prior opcode.
0DAE            458 ;
0DAE B0 EC       459 BCCMD     bcs BRCMD
0DB0            460 ;
0DB0 60          461             rts
0DB1            462 ;
0DB1            463 ;
0DB1            464 ; BP opcode is branch if prior register's value is
0DB1            465 ; postive.
0DB1            466 ;
0DB1 B5 01       467 BPCMD     lda R0H,X
0DB3 10 E7       468             bpl BRCMD
0DB5            469 ;
0DB5 60          470             rts
0DB6            471 ;
0DB6            472 ;
0DB6            473 ; BM opcode is branch if prior register's value is
0DB6            474 ; negative.
0DB6            475 ;
0DB6 B5 01       476 BMCMD     lda R0H,X
0DB8 30 E2       477             bmi BRCMD
0DBA            478 ;
0DBA 60          479             rts
0DBB            480 ;
0DBB            481 ;
0DBB            482 ; BZ opcode is branch if prior register's value is zero.
0DBB            483 ;
0DBB B5 00       484 BZCMD     lda R0L,X
0DBD 15 01       485             ora R0H,X
0DBF F0 DB       486             beq BRCMD
0DC1            487 ;

```

```

0DC1 60          488          rts
0DC2          489          ;
0DC2          490          ;
0DC2          491          ; BNZ opcode is branch if prior register's value is not
0DC2          492          ; zero.
0DC2          493          ;
0DC2 B5 00      494 BNZCMD    lda R0L,X
0DC4 15 01      495          ora R0H,X
0DC6 D0 D4      496          bne BRCMD
0DC8          497          ;
0DC8 60        498          rts
0DC9          499          ;
0DC9          500          ;
0DC9          501          ; BM1 opcode is branch if prior register's value is
0DC9          502          ; negative one.
0DC9          503          ;
0DC9 B5 00      504 BM1CMD    lda R0L,X
0DCB 35 01      505          and R0H,X
0DCD          506          ;
0DCD 49 FF      507          eor #NEGONE
0DCF F0 CB      508          beq BRCMD
0DD1          509          ;
0DD1 60        510          rts
0DD2          511          ;
0DD2          512          ;
0DD2          513          ; BNM1 opcode is branch if prior register's value is not
0DD2          514          ; negative one.
0DD2          515          ;
0DD2 B5 00      516 BNM1CMD   lda R0L,X
0DD4 35 01      517          and R0H,X
0DD6          518          ;
0DD6 49 FF      519          eor #NEGONE
0DD8 D0 C2      520          bne BRCMD
0DDA          521          ;
0DDA 60        522          rts
0ddb          523          ;
0ddb          524          ;
0ddb          525          ; SOUT opcode sends the "chr" value to COUT.
0ddb          526          ;
0ddb B1 1E      527 SOUTCMD   lda (R15L),Y
0DDD          528          ;
0DDD 4C ED FD   529          jmp COUT
0DE0          530          ;
0DE0          531          ;
0DE0          532          ; RSNS opcode is return from a BSNS subroutine call using
0DE0          533          ; no address stack. Copy the saved SW16 program counter
0DE0          534          ; in R12 to R15.
0DE0          535          ;
0DE0 A5 18      536 RSNSCMD   lda R12L
0DE2 85 1E      537          sta R15L
0DE4          538          ;
0DE4 A5 19      539          lda R12H
0DE6 85 1F      540          sta R15H
0DE8          541          ;
0DE8 60        542          rts
0DE9          543          ;
0DE9          544          ;
0DE9          545          ; SJMP opcode gets a 2-byte immediate address for the SW16
0DE9          546          ; program counter. Get the LO byte, then the HO byte, and
0DE9          547          ; save the address to R15. Make R15 the prior register.
0DE9          548          ; Fall into DCRCMD.

```

```
0DE9          549      ;
0DE9 B1 1E    550  SJMPCMD  lda (R15L),Y
0DEB AA       551          tax
0DEC          552      ;
0DEC C8       553          iny
0DED          554      ;
0DED B1 1E    555          lda (R15L),Y
0DEF          556      ;
0DEF 86 1E    557          stx R15L
0DF1 85 1F    558          sta R15H
0DF3          559      ;
0DF3 A2 1E    560          ldx #R15L
0DF5 86 1C    561          stx R14L
0DF7          562      ;
0DF7          563      ;
0DF7          564      ; DCR opcode decrements the register Rn.
0DF7          565      ;
0DF7 B5 00    566  DCRCMD   lda R0L,X
0DF9 D0 02    567          bne DCR2
0DFB          568      ;
0DFB D6 01    569          dec R0H,X
0DFD          570      ;
0DFD D6 00    571  DCR2     dec R0L,X
0DFF          572      ;
0DFF 60       573          rts
0E00          574      ;
0E00          575      ;
```

BSAVE SW16TEST,A\$0900,B,L\$0500

```
0E00          576          usr SW16TEST
0E00          577      ;
0E00          578      ;
0E00          579          stt "SW16TEST Symbol Table"
0E00          580      ;
0E00          581      ;
0E00          582          end 111
```

*** End of Assembly

Symbol List starts at 0x7800, ends at 0x7DD2, used 0x05D2, remaining 0x39DE

Symbols unsorted:

R0L	0000	R0H	0001	R12L	0018	R12H	0019	R14L	001C
R14H	001D	R15L	001E	R15H	001F	ZERO	0000	NEGONE	00FF
R0	0000	R1	0001	R2	0002	R3	0003	R4	0004
R5	0005	R6	0006	R7	0007	R8	0008	R9	0009
R10	000A	R11	000B	R12	000C	R13	000D	R14	000E
R15	000F	RTN	0000	BR	0001	BNC	0002	BC	0003
BP	0004	BM	0005	BZ	0006	BNZ	0007	BM1	0008
BNM1	0009	SOUT	000A	RS	000B	BS	000C	RSNS	000D
BSNS	000E	SJMP	000F	SET	0010	LD	0020	ST	0030
LD@	0040	ST@	0050	LDD@	0060	STD@	0070	POP@	0080
STP@	0090	ADD	00A0	SUB	00B0	POPD@	00C0	CPR	00D0
INR	00E0	DCR	00F0	PAGESIZE	0100	COUT	FEDE	RESTORE	FF3F
SAVE	FF4A	SOUTMAIN	0900	SETMAIN	0909	STDMAIN	0929	POP@MAIN	094C
ADDMAIN	0974	BRMAIN	099B	MYBR	09B5	BNCMAIN	09BA	MYBNC	09D5
BCMAIN	09DA	MYBC	09F5	BPMAIN	09FA	MYBP	0A18	BMMAIN	0A1D
MYBM	0A38	BZMAIN	0A3D	MYBZ	0A59	BNZMAIN	0A5E	MYBNZ	0A7A
BM1MAIN	0A7F	MYBM1	0A9E	BNM1MAIN	0AA3	MYBNM1	0AC2	PRT1SUB	0AC7
SUB2	0AD2	BSMAIN	0AD8	SUB1	0B04	PRT2SUB	0B11	PRT3SUB	0B1E
BSNSMAIN	0B29	SUB3	0B55	PART2	0B5E	SJMPMAIN	0B6B	PART4	0B94
PART3	0B9A	SW16	0C70	SW16B	0C79	SW16C	0C7F	SW16D	0C85
TOBR	0CA2	TOBR2	0CA8	SETZ	0CB2	SET2	0CC5	RSZ	0CC6
BSNSZ	0CD7	BRTBL	0CE2	OPTBL	0CE3	RTNCMD	0D01	SETCMD	0D09
RSCMD	0D0B	BSNSCMD	0D0D	LDCMD	0D0F	STCMD	0D18	LD@CMD	0D21
ST@CMD	0D2B	ST@2	0D2D	ST@3	0D31	INRCMD	0D33	INR2	0D39
LDD@CMD	0D3A	STD@CMD	0D43	POPD@CMD	0D4C	POP@CMD	0D53	POP@3	0D5E
STP@CMD	0D63	ADDCMD	0D6C	SUBCMD	0D7A	CPRCMD	0D7C	CPR2	0D88
BSCMD	0D90	BRCMD	0D9C	BNCCMD	0D9D	BR2	0DA4	BRTS	0DAD
BCCMD	0DAE	BPCMD	0DB1	BMCMD	0DB6	BZCMD	0DBB	BNZCMD	0DC2
BM1CMD	0DC9	BNM1CMD	0DD2	SOUTCMD	0DDB	RSNSCMD	0DE0	SJMPCMD	0DE9
DCRCMD	0DF7	DCR2	0DFD						

Symbols alphabetically sorted:

ADD	00A0	ADDCMD	0D6C	ADDMAIN	0974	BC	0003	BCCMD	0DAE
BCMAIN	09DA	BM	0005	BM1	0008	BM1CMD	0DC9	BM1MAIN	0A7F
BMCMD	0DB6	BMMAIN	0A1D	BNC	0002	BNCCMD	0D9D	BNCMAIN	09BA
BNM1	0009	BNM1CMD	0DD2	BNM1MAIN	0AA3	BNZ	0007	BNZCMD	0DC2
BNZMAIN	0A5E	BP	0004	BPCMD	0DB1	BPMAIN	09FA	BR	0001
BR2	0DA4	BRCMD	0D9C	BRMAIN	099B	BRTBL	0CE2	BRTS	0DAD
BS	000C	BSCMD	0D90	BSMAIN	0AD8	BSNS	000E	BSNSCMD	0D0D
BSNSMAIN	0B29	BSNSZ	0CD7	BZ	0006	BZCMD	0DBB	BZMAIN	0A3D
COUT	FEDE	CPR	00D0	CPR2	0D88	CPRCMD	0D7C	DCR	00F0
DCR2	0DFD	DCRCMD	0DF7	INR	00E0	INR2	0D39	INRCMD	0D33
LD	0020	LD@	0040	LD@CMD	0D21	LDCMD	0D0F	LDD@	0060
LDD@CMD	0D3A	MYBC	09F5	MYBM	0A38	MYBM1	0A9E	MYBNC	09D5
MYBNM1	0AC2	MYBNZ	0A7A	MYBP	0A18	MYBR	09B5	MYBZ	0A59
NEGONE	00FF	OPTBL	0CE3	PAGESIZE	0100	PART2	0B5E	PART3	0B9A
PART4	0B94	POP@	0080	POP@3	0D5E	POP@CMD	0D53	POP@MAIN	094C
POPD@	00C0	POPD@CMD	0D4C	PRT1SUB	0AC7	PRT2SUB	0B11	PRT3SUB	0B1E
R0	0000	R0H	0001	R0L	0000	R1	0001	R10	000A
R11	000B	R12	000C	R12H	0019	R12L	0018	R13	000D
R14	000E	R14H	001D	R14L	001C	R15	000F	R15H	001F
R15L	001E	R2	0002	R3	0003	R4	0004	R5	0005
R6	0006	R7	0007	R8	0008	R9	0009	RESTORE	FF3F
RS	000B	RSCMD	0D0B	RSNS	000D	RSNSCMD	0DE0	RSZ	0CC6

RTN	0000	RTNCMD	0D01	SAVE	FF4A	SET	0010	SET2	0CC5
SETCMD	0D09	SETMAIN	0909	SETZ	0CB2	SJMP	000F	SJMPCMD	0DE9
SJMPMAIN	0B6B	SOUT	000A	SOUTCMD	0DDB	SOUTMAIN	0900	ST	0030
ST@	0050	ST@2	0D2D	ST@3	0D31	ST@CMD	0D2B	STCMD	0D18
STD@	0070	STD@CMD	0D43	STDMAIN	0929	STP@	0090	STP@CMD	0D63
SUB	00B0	SUB1	0B04	SUB2	0AD2	SUB3	0B55	SUBCMD	0D7A
SW16	0C70	SW16B	0C79	SW16C	0C7F	SW16D	0C85	TOBR	0CA2
TOBR2	0CA8	ZERO	0000						

Symbols numerically sorted:

ZERO	0000	RTN	0000	R0L	0000	R0	0000	R1	0001
R0H	0001	BR	0001	R2	0002	BNC	0002	R3	0003
BC	0003	R4	0004	BP	0004	R5	0005	BM	0005
R6	0006	BZ	0006	R7	0007	BNZ	0007	R8	0008
BM1	0008	R9	0009	BNM1	0009	SOUT	000A	R10	000A
RS	000B	R11	000B	R12	000C	BS	000C	RSNS	000D
R13	000D	R14	000E	BSNS	000E	SJMP	000F	R15	000F
SET	0010	R12L	0018	R12H	0019	R14L	001C	R14H	001D
R15L	001E	R15H	001F	LD	0020	ST	0030	LD@	0040
ST@	0050	LDD@	0060	STD@	0070	POP@	0080	STP@	0090
ADD	00A0	SUB	00B0	POPD@	00C0	CPR	00D0	INR	00E0
DCR	00F0	NEGONE	00FF	PAGESIZE	0100	SOUTMAIN	0900	SETMAIN	0909
STDMAIN	0929	POP@MAIN	094C	ADDMAIN	0974	BRMAIN	099B	MYBR	09B5
BNCMAIN	09BA	MYBNC	09D5	BCMAIN	09DA	MYBC	09F5	BPMAIN	09FA
MYBP	0A18	BMMAIN	0A1D	MYBM	0A38	BZMAIN	0A3D	MYBZ	0A59
BNZMAIN	0A5E	MYBNZ	0A7A	BM1MAIN	0A7F	MYBM1	0A9E	BNM1MAIN	0AA3
MYBNM1	0AC2	PRT1SUB	0AC7	SUB2	0AD2	BSMAIN	0AD8	SUB1	0B04
PRT2SUB	0B11	PRT3SUB	0B1E	BSNSMAIN	0B29	SUB3	0B55	PART2	0B5E
SJMPMAIN	0B6B	PART4	0B94	PART3	0B9A	SW16	0C70	SW16B	0C79
SW16C	0C7F	SW16D	0C85	TOBR	0CA2	TOBR2	0CA8	SETZ	0CB2
SET2	0CC5	RSZ	0CC6	BSNSZ	0CD7	BRTBL	0CE2	OPTBL	0CE3
RTNCMD	0D01	SETCMD	0D09	RSCMD	0D0B	BSNSCMD	0D0D	LDCMD	0D0F
STCMD	0D18	LD@CMD	0D21	ST@CMD	0D2B	ST@2	0D2D	ST@3	0D31
INRCMD	0D33	INR2	0D39	LDD@CMD	0D3A	STD@CMD	0D43	POPD@CMD	0D4C
POP@CMD	0D53	POP@3	0D5E	STP@CMD	0D63	ADDCMD	0D6C	SUBCMD	0D7A
CPRCMD	0D7C	CPR2	0D88	BSCMD	0D90	BRCMD	0D9C	BNCCMD	0D9D
BR2	0DA4	BRTS	0DAD	BCCMD	0DAE	BPCMD	0DB1	BMCMD	0DB6
BZCMD	0DBB	BNZCMD	0DC2	BM1CMD	0DC9	BNM1CMD	0DD2	SOUTCMD	0DDB
RSNSCMD	0DE0	SJMPCMD	0DE9	DCRCMD	0DF7	DCR2	0DFD	COUT	FDED
RESTORE	FF3F	SAVE	FF4A						