Sweet 16 for Bibo Assembler#

General Information

Author: Steve "Woz" Wozniak
Assembler: BiboAssembler
Published: 1977

see also Original Sweet 16

- The ATARI uses the zero page registers at $E0-$FF instead of $00-$1F
- the "save" and "restore" routines are not in ROM, but in the source

Download: Sweet16Bibo/sweet16.atr

```
00000   .LI OFF
00010 ************************
00020 *                       *
00030 * APPLE II PSEUDO MACHINE *
00040 *       INTERPRETER     *
00050 *                       *
00060 * COPYRIGHT (C) 1977    *
00070 * APPLE COMPUTER INC.     *
00080 *                       *
00090 * STEVE WOZNIAK          *
00100 *                       *
00110 ************************
00120 * TITLE: SWEET 16 INTERPRETER
00130 ;
00140 R0L      =  $E0
00150 R0H      =  $E1
00160 R14H    =   $FD
00170 R15L    =   $FE
00180 R15H    =   $FF
00190 ;
00200        .OR $7889
00210 ;       .OF D:SWEET16.COM
00220 ;
00230 ; PRESERVE 6502 REG CONTENT
00240 ;
00250 SW16   JSR SAVE
00260 ;
00270 ; INIT SWEET16 PC
00280 ; FROM RETURN ADDRESS
00290 ;
00300 SW16A  PLA
00310        STA R15L
00320        PLA
00330        STA R15H
00340 ;
00350 ; INTERPRET AND EXECUTE
00360 ; ONE SWEET16 INSTRUCTION
00370 ;
00380 SW16B  JSR SW16C
00390        JMP SW16B
00400 ;
00410 ; INCREMENT SWEET16 PC
00420 ; FOR FETCH
```

```
00430 ;
00440 SW16C  INC R15L
00450        BNE SW16D
00460        INC R15H
00470 ;
00480 ; COMMON HIG BZTE FOR ALL
00490 ; ROUTINES, PUSH ON STACK FOR
00500 ; RTS
00510 SW16D  LDA /SET
00520        PHA
00530 ;
00540 ; FETCH INSTRUCTION
00550 ;
00560        LDY #$0
00570        LDA (R15L),Y
00580 ;
00590 ; MASK REGISTER SPECIFICATION
00600 ;
00610        AND #$F
00620 ;
00630 ; DOUBLE FOR TWO BYTE REGISTERS
00640 ;
00650        ASL
00660 ;
00670 ; TO X REGISTER FOR INDEXING
00680 ;
00690        TAX
00700        LSR
00710 ;
00720 ; NOW HAVE OPCODE
00730 ;
00740        EOR (R15L),Y
00750 ;
00760 ; IF ZERO THEN NO REG OPCODE
00770 ;
00780        BEQ TOBR
00790 ;
00800 ; INDICATE PRIOR RESULT REG
00810 ;
00820        STX R14H
00830 ;
00840 ; OPCODE * 2 TO LSB'S
00850 ;
00860        LSR
00870        LSR
00880        LSR
00890 ;
00900 ; TO Y REGISTER FOR INDEXING
00910 ;
00920        TAY
00930 ;
00940 ; LOW ORDER ADR BYTE ON STACK
00950 ;
00960        LDA OPTBL-2,Y
00970        PHA
00980 ;
00990 ; GOTO REG-OPCODE ROUTINE
01000 ;
01010        RTS
```

```
01020 ;
01030 ; INCREMENT PC
01040 ;
01050 TOBR    INC R15L
01060         BNE TOBR2
01070         INC R15H
01080 ;
01090 ; LOW ORDER ADR BYTE ONTO
01100 ; STACK FOR NON-REG OPCODE
01110 ;
01120 TOBR2   LDA BRTBL,X
01130         PHA
01140 ;
01150 ; PRIOR RESULT REG INDEX
01160 ;
01170         LDA R14H
01180 ;
01190 ; PREPARE CARRY FOR BC. BNC.
01200 ;
01210         LSR
01220 ;
01230 ; GOTO NON-REG OPCODE ROUTINE
01240 ;
01250         RTS
01260 ;
01270 ; RETURN TO 6502 MODE ROUTINE
01280 ;
01290 ; POP RETURN ADDRESS
01300 ;
01310 RTNZ    PLA
01320         PLA
01330 ;
01340 ; RESTORE 6502 REG CONTENT
01350 ;
01360         JSR RESTORE
01370 ;
01380 ; RETURN TO 6502 CODE VIA PC
01390 ;
01400         JMP (R15L)
01410 ;
01420 ; SET REGISTER ROUTINE
01430 ;
01440 ; GET HIGH BYTE OF CONSTANT
01450 ;
01460 SETZ    LDA (R15L),Y
01470         STA R0H,X
01480         DEY
01490 ;
01500 ; GET LOW BYTE OF CONSTANT
01510 ;
01520         LDA (R15L),Y
01530         STA R0L,X
01540 ;
01550 ; Y REG CONTAINS 1
01560 ;
01570         TYA
01580 ;
01590 ; ADD 2 TO PC
01600 ;
```

```
01610        SEC
01620        ADC R15L
01630        STA R15L
01640        BCC SET2
01650        INC R15H
01660 SET2    RTS
01670 ;
01680 ; OPCODE TABLE
01690 ;
01700 OPTBL  .DA #SET-1 ; 1X
01710 BRTBL  .DA #RTN-1 ; 0
01720        .DA #LD-1   ; 2X
01730        .DA #BR-1   ; 1
01740        .DA #ST-1   ; 3X
01750        .DA #BNC-1 ; 2
01760        .DA #LDAT-1  ; 4X
01770        .DA #BC-1   ; 3
01780        .DA #STAT-1  ; 5X
01790        .DA #BP-1   ; 4
01800        .DA #LDDAT-1 ; 6X
01810        .DA #BM-1   ; 5
01820        .DA #STDAT-1 ; 7X
01830        .DA #BZ-1   ; 6
01840        .DA #POP-1 ; 8X
01850        .DA #BNZ-1 ; 7
01860        .DA #STPAT-1 ; 9X
01870        .DA #BM1-1 ; 8
01880        .DA #ADD-1 ; AX
01890        .DA #BNM1-1  ; 9
01900        .DA #SUB-1 ; BX
01910        .DA #BK-1   ; A
01920        .DA #POPD-1  ; CX
01930        .DA #RS-1   ; B
01940        .DA #CPR-1 ; DX
01950        .DA #BS-1   ; C
01960        .DA #INR-1 ; EX
01970        .DA #NUL-1 ; D
01980        .DA #DCR-1 ; FX
01990        .DA #NUL-1 ; E
02000        .DA #NUL-1 ; UNUSED
02010        .DA #NUL-1 ; F
02020 ;
02030 ; THE FOLLOWING CODE MUST
02040 ; BE CONTAINED IN A SINGLE PAGE
02050 ;
02060 SET      JMP SETZ  ; ALWAYS
02070 LD     LDA R0L,X
02080 BK     = LD+1
02090        STA R0L
02100 ;
02110 ; MOV RX TO R0
02120 ;
02130        LDA R0H,X
02140        STA R0H
02150        RTS
02160 ;
02170 ST     LDA R0L
02180 ;
02190 ; MOV R0 TO RX
```

```
02200 ;
02210        STA R0L,X
02220        LDA R0H
02230        STA R0H,X
02240        RTS
02250 ;
02260 ; STORE BYTE INDEIRECT
02270 ;
02280 STAT    LDA R0L
02290 STAT2  STA (R0L,X)
02300        LDY #$0
02310 ;
02320 ; INDICATE R0 IS RESULT NEG
02330 ;
02340 STAT3  STY R14H
02350 INR      INC R0L,X
02360        BNE INR2    ; INC RX
02370        INC R0H,X
02380 INR2    RTS
02390 ;
02400 ; LOAD INDIRECT (RX) TO R0
02410 ;
02420 LDAT    LDA (R0L,X)
02430        STA R0L
02440 ;
02450 ; ZERO HIGH ORDER R0 BYTE
02460 ;
02470        LDY #$0
02480        STY R0H
02490        BEQ STAT3  ; ALWAYS
02500 ;
02510 ; HIGH ORDER BYTE = 0
02520 ;
02530 POP      LDY #$0
02540        BEQ POP2    ; ALWAYS
02550 POPD    JSR DCR     ; DECR RX
02560 ;
02570 ; POP HIGH ORDER BYTE @RX
02580 ; AND SAV IN Y REG
02590 ;
02600        LDA (R0L,X)
02610        TAY
02620 POP2    JSR DCR     ; DECR RX
02630 ;
02640 ; LOW ORDER BYTE TO R0
02650 ;
02660        LDA (R0L,X)
02670        STA R0L
02680        STY R0H
02690 ;
02700 ; INDICATE R0 AS LAST RES. REG
02710 ;
02720 POP3    LDY #$0
02730        STY R14H
02740        RTS
02750 ;
02760 ; LOW ORDER BYTE TO R0, INC RX
02770 ;
02780 LDDAT  JSR LDAT
```

```
02790 ;
02800 ; HIGH ORDER BYTE TO R0
02810        LDA (R0L,X)
02820        STA R0H
02830        JMP INR   ; INC RX
02840 ;
02850 ; STORE INDIRECT LOW ORDER
02860 ; BYTE AND INC RX THEN
02870 ; STORE HIGH  ORDER BYTE,
02880 ; INC RX AND RETURN
02890 ;
02900 STDAT  JSR LDAT
02910        LDA R0H
02920        STA (R0L,X)
02930        JMP INR
02940 STPAT  JSR DCR   ; DEC RX
02950        LDA R0L
02960 ;
02970 ; STORE R0 LOW BYTE @RX
02980 ;
02990        STA (R0L,X)
03000 ;
03010 ; INDICATE R0 AS LAST RES REG
03020 ;
03030        JMP POP3
03040 ;
03050 DCR      LDA R0L,X
03060        BNE DCR2    ; DEC RX
03070        DEC R0H,X
03080 DCR2    DEC R0L,X
03090        RTS
03100 ;
03110 ; RESULT TO R0
03120 ;
03130 SUB      LDY #$0
03140 ;
03150 ; NOTE Y REG = 13 * 2 FOR CPR
03160 ;
03170 CPR      SEC
03180        LDA R0L
03190        SBC R0L,X
03200        STA R0L,Y  ; RY=R0-RX
03210        LDA R0H
03220        SBC R0H,X
03230 SUB2    STA R0H,Y
03240 ;
03250 ; LAST RESULT REG * 2
03260 ;
03270        TYA
03280 ;
03290 ; CARRY TO LSB
03300 ;
03310        ADC #$0
03320        STA R14H
03330        RTS
03340 ;
03350 ADD      LDA R0L
03360        ADC R0L,X
03370        STA R0L   ; R0=RX+R0
```

```
03380         LDA R0H
03390         ADC R0H,X
03400 ;
03410 ; R0 FOR RESULT
03420 ;
03430         LDY #$0
03440 ; FINISH ADD
03450         BEQ SUB2
03460 ;
03470 ; NOTE X REG IS 12 * 2 !
03480 ;
03490 BS      LDA R15L
03500 ;
03510 ; PUSH LOW PC BYTE VIA R12
03520 ;
03530         JSR STAT2
03540         LDA R15H
03550 ;
03560 ; PUSH HIGH PC BYTE
03570 ;
03580         JSR STAT2
03590 BR      CLC
03600 BNC       BCS BNC2  ; NO CARRY TEST
03610 ;
03620 ; DISPLACEMENT BYTE
03630 ;
03640 BR1       LDA (R15L),Y
03650         BPL BR2
03660         DEY
03670 ;
03680 ; ADD TO PC
03690 ;
03700 BR2       ADC R15L
03710         STA R15L
03720         TYA
03730         ADC R15H
03740         STA R15H
03750 BNC2     RTS
03760 BC      BCS BR
03770         RTS
03780 ;
03790 ; DOUBLE RESULT REG INDEX
03800 ;
03810 BP      ASL
03820 ;
03830 ; TO X REG FOR INDEX
03840 ;
03850         TAX
03860 ; TEST FOR PLUS, BRANCH IF SO
03870 ;
03880         LDA R0H,X
03890         BPL BR1
03900         RTS
03910 ;
03920 ; DOUBLE RESULT REG INDEX
03930 ;
03940 BM      ASL
03950         TAX
03960 ;
```

```
03970 ; TEST FOR MINUS
03980 ;
03990        LDA R0H,X
04000        BMI BR1
04010        RTS
04020 ;
04030 ; DOUBLE RESULT REG INDEX
04040 ;
04050 BZ     ASL
04060        TAX
04070 ;
04080 ; TEST FOR ZERO (BOTH BYTES)
04090 ;
04100        LDA R0L,X
04110        ORA R0H,X
04120        BEQ BR1
04130        RTS
04140 ;
04150 ; DOUBLE RESULT REG INDEX
04160 ;
04170 BNZ      ASL
04180        TAX
04190 ;
04200 ; TEST FOR NON-ZERO (BOTH)
04210 ;
04220        LDA R0L,X
04230        ORA R0H,X
04240        BNE BR1
04250        RTS
04260 ;
04270 ; DOUBLE RESULT REG INDEX
04280 ;
04290 BM1      ASL
04300        TAX
04310 ;
04320 ; TEST FOR $FF (-1) BOTH BYTES
04330 ;
04340        LDA R0L,X
04350        AND R0H,X
04360        EOR #$FF
04370        BEQ BR1
04380        RTS
04390 ;
04400 ; DOUBLE RESULT REG
04410 ;
04420 BNM1    ASL
04430        TAX
04440 ;
04450 ; TEST FOR NO $FF
04460 ;
04470        LDA R0L,X
04480        AND R0H,X
04490        EOR #$FF
04500        BNE BR1
04510 NUL      RTS
04520 ; 12*2 FOR R12 AS STACK POINTER
04530 ;
04540 RS     LDX #$18
04550 ;
```

```
04560 ; DECR STACK POINTER
04570 ;
04580        JSR DCR
04590 ;
04600 ; POP HIGHRETURN ADDRESS TO PC
04610 ;
04620        LDA (R0L,X)
04630        STA R15H
04640 ;
04650 ; SAME WITH LOW ORDER BYTE
04660 ;
04670        JSR DCR
04680        LDA (R0L,X)
04690        STA R15L
04700        RTS
04710 RTN      JMP RTNZ
04720 ------------------------------
04730 SAVE
04740        STA ACC
04750        STX XREG
04760        STY YREG
04770        PHP
04780        PLA
04790        STA STATUS
04800        CLD
04810        RTS
04820 ------------------------------
04830 RESTORE  LDA STATUS
04840        PHA
04850        LDA ACC
04860        LDX XREG
04870        LDY YREG
04880        PLP
04890        RTS
04900 ------------------------------
04910 ACC      .HX 00
04920 XREG    .HX 00
04930 YREG    .HX 00
04940 STATUS   .HX 00
04950 ------------------------------
04960 TEST
04970        JSR SW16
04980        .HX 1100F0 ; SET R1
04990        .HX 120040 ; SET R2
05000        .HX 130002 ; SET R3
05010        .HX 41    ; LD @R1
05020        .HX 52    ; ST @R2
05030        .HX F3    ; DCR R3
05040        .HX 07FB   ; BNZ -4
05050        .HX 00    ; RTN
05060        RTS
05070 ------------------------------
```