Tech Info Library

Apple II+ Mini-Assembler (2 of 4)

Revised: 3/2/88
Security: Everyone

Apple II+ Mini-Assembler (2 of 4)

==========

This article last reviewed: 21 September 1984

The mnemonics and formats accepted by the mini assembler are the same as those listed by the 6502 Programmers Manual, with the following exceptions and differences:

1. All imbedded blanks are ignored, except inside addresses.

2. All addresses typed in are assumed to be in hex (rather than decimal or symbolic). A preceding "$" (indicating hex rather than decimal or symbolic) is therefore optional, except that it should not precede the instruction address).

3. Instructions that operate on the accumulator have a blank operand field instead of "A".

4. When entering a branch instruction, the argument of the branch mnemonic should be the address of the target of the branch. If the destination address is not known at the time the instruction is entered, simply enter an address that is in the neighborhood, and later re-enter the branch instruction with the correct target address. NOTE: If a branch target is specified that is out of range, the mini-assembler will flag the address as being in error.

5. The operand field of an instruction can only be followed by a comment field, which starts with a semicolon (";"). Obviously, the mini-assembler ignores the field and in fact will type over it when the line is typed over in disassembler format.

6. Any page zero references will generate page zero instruction formats if such a mode exists. There is no way to force a page zero address to be two bytes, even if the address has leading zeroes.

In general, to specify an addressing type, simply enter it as it would be listed in the disassembly. For information on the disassembler, see page 49

of the Apple II Reference Manual.

```
0000:                          **************************
0000:                          *                        *
0000:                          *       APPLE II         *
0000:                          *   MINI-ASSEMBLER       *
0000:                          *                        *
0000:                          **************************
002E:           FORMAT  EQU    $2E
002F:           LENGTH  EQU    $2F
0031:           MODE    EQU    $31
0033:           PROMPT  EQU    $33
0034:           YSAV    EQU    $34
0035:           L       EQU    $35
003A:           PCL     EQU    $3A
003B:           PLH     EQU    $3B
003D:           A1H     EQU    $3D
003E:           A2L     EQU    $3E
003F:           A2H     EQU    $3F
0042:           A4L     EQU    $42
0043:           A4H     EQU    $43
0044:           FMT     EQU    $44
0200:           IN      EQU    $200
D64B:           NEW     EQU    $D64B
F88E:           INSDS2  EQU    $F88E
F8D0:           INSTDSP EQU    $F8D0
F94A:           PRBL2   EQU    $F94A
F953:           PCADJ   EQU    $F953
F9B4:           CHAR1   EQU    $F9B4
F9BA:           CHAR2   EQU    $F9BA
F9C0:           MNEML   EQU    $F9C0
FA00:           MNEMR   EQU    $FA00
FC1A:           CURSUP  EQU    $FC1A
FD67:           GETLNZ  EQU    $FD67
FDED:           COUT    EQU    $FDED
FE00:           BL1     EQU    $FE00
FE78:           A1PCLP  EQU    $FE78
FF3A:           BELL    EQU    $FF3A
FFA7:           GETNUM  EQU    $FFA7
FFBE:           TOSUB   EQU    $FFBE
FFC7:           ZMODE   EQU    $FFC7
FFCC:           CHRTBL  EQU    $FFCC
```

Tech Info Library Article Number:6