

Moving data to auxiliary memory

In your assembly-language programs, you can use the built-in subroutine named AUXMOVE to copy blocks of data from main memory to auxiliary memory or from auxiliary memory to main memory. Before calling this routine, you must put the data addresses into byte pairs in page zero and set the carry bit to select the direction of the move—main to auxiliary or auxiliary to main.

Warning Don't try to use AUXMOVE to copy data in page zero or page one (the 65C02 stack) or in the bank-switched memory (\$D000-\$FFFF). AUXMOVE uses page zero all during the copy, so it can't handle moves in the memory space switched by ALTZP.

The pairs of bytes you use for passing addresses to this subroutine are called A1, A2, and A4, and they are used for parameter passing by several of the Apple IIe's built-in routines. The addresses of these byte pairs are shown in Table 4-9.

Table 4-9
Parameters for AUXMOVE routine

Name	Location	Parameter passed
Carry		1 = Move from main to auxiliary memory 0 = Move from auxiliary to main memory
A1L	\$3C	Source starting address, low-order byte
A1H	\$3D	Source starting address, high-order byte
A2L	\$3E	Source ending address, low-order byte
A2H	\$3F	Source ending address, high-order byte
A4L	\$42	Destination starting address, low-order byte
A4H	\$43	Destination starting address, high-order byte

Note: The X, Y, and A registers are preserved by AUXMOVE.

Put the addresses of the first and last bytes of the block of memory you want to copy into A1 and A2. Put the starting address of the block of memory you want to copy the data to into A4.

The AUXMOVE routine uses the carry bit to select the direction to copy the data. To copy data from main memory to auxiliary memory, set the carry bit; to copy data from auxiliary memory to main memory, clear the carry bit.

When you make the subroutine call to AUXMOVE, the subroutine copies the block of data as specified by the A byte pairs and the carry bit. When it is finished, the accumulator and the X and Y registers are just as they were when you called AUXMOVE.