When you read or write the data byte, the address automatically increments. Some 6502 address modes cause two accesses at the same location, which in this case can cause double incrementing. To avoid double incrementing, use either absolute addressing, or an indexed form with the base in a different page. For example, with the RamFactor in slot 4, you could use:

```
LDA   $C0C3       absolute addressing
    or
LDY   #$C8        slot * 16 +$88
STA   $BFFB,Y     $C083-$88,Y
```

Whenever the lower or middle address byte changes from a value with bit 7 = 1 to one with bit 7 = 0, the next higher byte increments automatically. This means that you should always load the bytes in the order low-middle-high, and always load all three of them. (Unless, of course, you are sure of the previous contents and can be sure you will get predictably correct results.)

Finding the RamFactor Card

A program can search the slots for a RamFactor card by looking for a unique pattern in various firmware locations. The bytes at $Cs00-Cs07 are standard for both the Apple Memory Expansion Card and the RamFactor card. They are:

```
$Cs00:C9  20  C9  00  C9  03  C9  00
```

In addition, the byte at $CsFB on a card of this type has bit 0 = 1. For RamFactor, the whole byte = $01. The byte at $CsFA distinguishes between the various brands of memory expansion cards. RamFactor has $AE at this location.

Finding RamFactor's Size

There are several ways a program can find out the size of the RamFactor card, or the size of the current partition.

Finding Size via ProDOS Status Call

If you are using ProDOS, the ProDOS status call can be used. The methods for finding the ProDOS firmware entry point, and for making a status call, are explained on pages 112-115 of Apple's "ProDOS Technical Reference Manual" and on pages 6-6 through 6-11 of Quality Software's "Beneath Apple ProDOS". Briefly, the firmware entry point will be $Csxx, where s is the slot number and xx is the contents of $CsFF in the firmware.