RAM beneath 00 and 01

On an unexpanded Commodore 64, how does one read the RAM locations $00 and $01?

Well, you cannot do so with the CPU directly, since it resolves these locations into internal addresses. However, the VIC II can see these addresses as external memory. So, just make one sprite with the first bit in the sprite set, and move it over the first two bytes, pretending they are part of a bitmap. By checking the sprite-to-background collision register, you can tell if the bit in the byte is set. Email me for a more complete description.

Sven Goldt and Marko Makela get credit for this answer and the next.

On an unexpanded Commodore 64, how does one write the same locations?

It seems the 6510 generates a valid R/W signal any time it does an internal read or write. This is to be expected, since the 6510 internal registers were grafted onto a 6502 core processor. Howevere, the address lines are also valid during any internal read or write, since failure to do so may write the data on the data bus to some invalid address. The data on the bus, however, comes not from the CPU, but from residual effects of the data last read of written by the VIC chip. Thus, by programming the VIC chip to read data from some known location, and by placing relevant data in that location, a write to location $00 or $01 will place the data from that last read VIC location into $00 or $01. This is usually accomplished by placing the data to be written out into location $3fff, which the VIC fetches during the time the border is being displayed. By triggering a routine when the raster hits the bottom border, you can copy location $3fff to $00 or $01.

(taken from commodore hacking)

A NOTE BY FRANTIC: See the original discussion of this in c64doc written by John West and Marko Mäkelä instead.