# CHECK MATE

**One of the features of the Commodore 64 is the ease with which the location of screen data can be relocated in memory. In this part of the machine code course, we look at a routine that makes use of this facility to allow the design and storage of up to eight alternative screen displays.**

Screen display and sprite handling are controlled by a special chip inside the Commodore 64 called the VIC II chip. The VIC chip accesses various sections of memory to obtain information from which it creates the display we see on the screen. These areas include the character ROM, where character shapes are held; the colour RAM, where colour information for the screen is held; and the screen RAM. The latter holds information about the characters to be displayed at any one of the 1,000 locations (25 rows by 40 columns) that go to make up the screen.

When the Commodore 64 is switched on, the VIC chip assumes that the screen is located in the 1,000 bytes starting at location 1024 ($0400), and it accesses this area to obtain its initial screen information. However, by altering the value of a register within the VIC chip, we can redirect the VIC chip to another area of memory — normally the first 16 Kbytes in memory. The upper four bits of the VIC control register at location 53272 ($D018) determine which one Kbyte block, out of the 16 Kbyte area in view, is interpreted as the screen. The following table shows the bit values that correspond to each of the 16 possible screen positions:

| Bit Pattern | Start Of Screen | |
|---|---|---|
| 0000XXXX | 0 | $0000 |
| 0001XXXX | 1024 | $0400 * |
| 0010XXXX | 2048 | $0800 |
| 0011XXXX | 3072 | $0C00 |
| 0100XXXX | 4096 | $1000 |
| 0101XXXX | 5120 | $1400 |
| 0110XXXX | 6144 | $1800 |
| 0111XXXX | 7168 | $1C00 |
| 1000XXXX | 8192 | $2000 |
| 1001XXXX | 9216 | $2400 |
| 1010XXXX | 10240 | $2800 |
| 1011XXXX | 11264 | $2C00 |
| 1100XXXX | 12288 | $3000 |
| 1101XXXX | 13312 | $3400 |
| 1110XXXX | 14336 | $3800 |
| 1111XXXX | 15360 | $3C00 |

* = Default Position

To make the VIC chip move the screen to another area, we have to change the upper four bits of location 53272 ($D018) to the values shown in the table. However, we must not disturb the lower four bits (the XXXX part of the bit pattern in the table) as they control another function. To zero the upper four bits without changing the value of the lower four bits, we must AND the contents of the register with 15 (00001111 in binary). Having done this, we can then OR the new contents of the register with the value we require. To position the screen in the last area that the VIC chip can see — that is starting at 15360 ($3C00) — we would OR the contents of the register with 240 (11110000 in binary). In BASIC, the following POKE statement would do this:

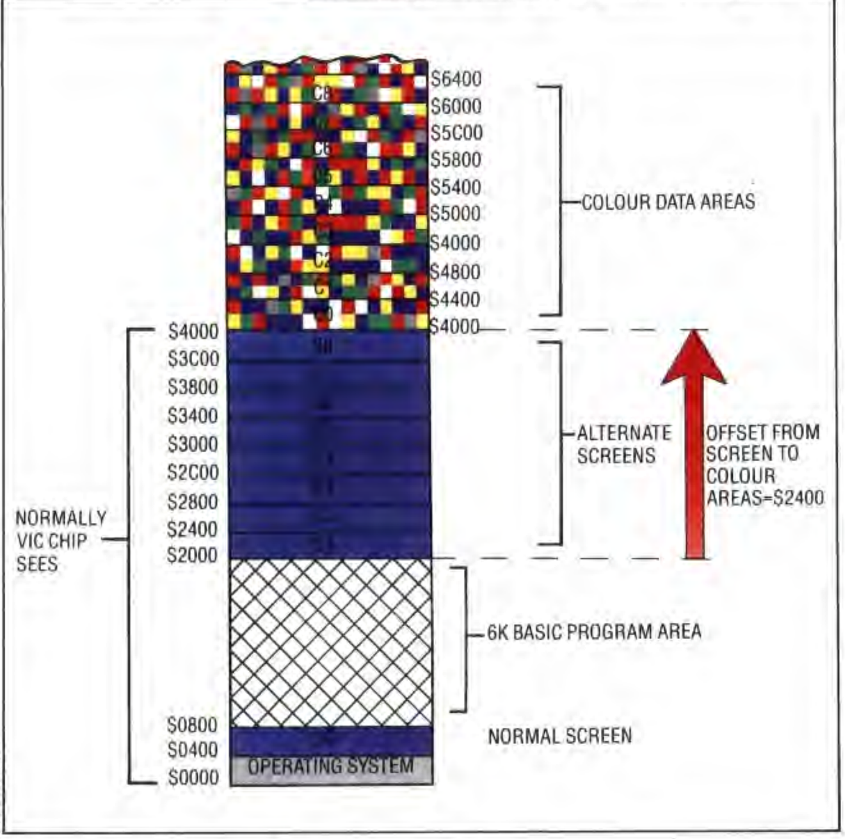
```
POKE 53272,(PEEK(53272)AND15)OR240
```

Before we could write anything on our new screen we would also have to tell the Commodore 64's operating system that the position of the screen had changed. This is done by placing the HI-byte of the new screen's start address in location 648 ($0288). For the highest screen this is $3C, and is easily worked out in BASIC by dividing the screen start address by 256.

Having changed the contents of these two

**VIC View**

The Commodore 64's video controller (VIC) chip can 'see' 16K of memory. Normally this is the first 16K, $0000 to $3FFF, but it can be made to look at any of the other three 16K blocks by altering the contents of one of the VIC control registers. The 'Alternate Screens' program sets up nine alternate screen maps within the normal 16K area seen by the VIC chip. Corresponding colour maps for each screen are held in the RAM area just above the area seen by VIC, each screen except screen 0 having a constant offset of $2400 to its colour map



KEVIN JONES