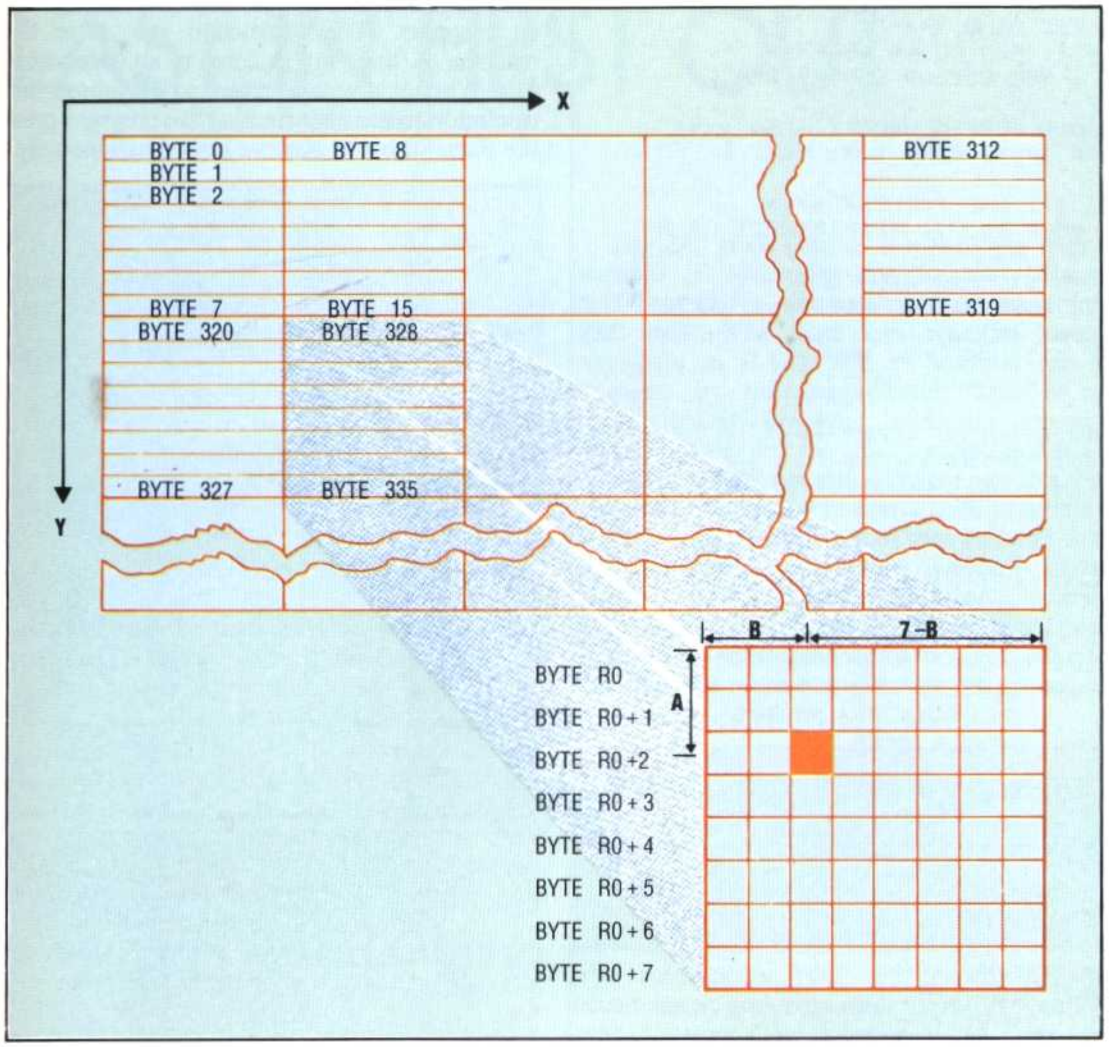

Point To Point
The dot pixels that make up the Commodore 64 hi-res screen cannot be accessed directly: the 40 × 25 text screen is mapped onto 8,000 bytes of RAM, each text position being described by eight bytes. A dot pixel's position is described in hi-res by X, its distance (in pixels) from the left of the screen, and Y, its distance from the top of the screen. These numbers must be converted into the address of the byte that contains the pixel, and the number of the relevant bit in that byte

which corresponding bit in the 8,000-byte memory map is to be set to one or zero.

The horizontal byte position can be found from the X co-ordinate by the following command:

```
HB = INT(X/8)
```

Similarly, the required vertical byte can be found from the Y co-ordinate:

```
VB = INT(Y/8)
```

The first byte of the character cell that contains the required bit, R0, can be calculated from HB and VB:

```
R0 = VB*320 + HB*8
```

The byte that contains the required bit will be R0, plus the remainder when Y has been divided by eight. This remainder can easily be found from the right-most three bits of the value of Y. If A = YAND7 and BASE is the address of the first byte in the 8,000-byte block, then the address of the byte, BY, that contains the bit we require can be found:

```
BY = BASE + R0 + A
```

The bit within the byte BY can be found by calculating the remainder when the X co-ordinate is divided by eight. If B = XAND7 then the following POKE will set to one the bit that corresponds to the pixel with co-ordinates X and Y:

```
POKE BY, PEEK(BY)OR(2↑(7-B))
```

Now that each pixel can be individually turned on, routines can be designed to draw shapes on the screen. The following program shows how straight lines can be drawn from one point (X1,Y1) to another (X2,Y2). A circle may be plotted by specifying the co-ordinates of its centre (CX,CY) and the radius RA. There is also a subroutine that will draw a triangle given the co-ordinates of its three corners (XA,YA), (XB,YB) and (XC,YC). You may wish to experiment by entering co-ordinates other than those given in the program.

It is interesting to note that the structure of this program consists of a series of tiered subroutines. The lowest level routine is the one that plots a single point on the screen. This subroutine is used by a higher level routine that draws a straight line. At a higher level still, the PLOT TRIANGLE routine uses the PLOT LINE routine three times to construct its three sides. This approach to programming has many advantages. It is flexible, since it would be very easy to design a routine to draw, say, regular hexagons. Such a routine would call the PLOT LINE routine, which would in turn call the PLOT POINT routine. Alternatively, the DRAW TRIANGLE routine could be used to construct hexagons from equilateral triangles. In this case the DRAW HEXAGON routine would form a fourth tier to the program structure.