Sounds Ideal

The Commodore 64's Basic doesn't match up to its remarkable sound facilities

Among the current range of home computers, the Commodore 64 is supplied with the most sophisticated sound-making facilities. These are attributable to a specialised chip called the Sound Interface Device — or SID, as it is better known.

SID provides capabilities similar to that of a commercial monophonic synthesiser. There are three oscillators with an eight-octave range (0-3900Hz in 65,536 steps); a master volume control, from 0 to 15; four waveforms for each oscillator (triangle, sawtooth, variable pulse and noise); oscillator synchronisation; and envelope generators allowing ADSR control for each oscillator. Further features include: ring modulation; programmable filter with low pass, band pass, high pass, notch output (which blocks out a narrow band of frequencies) and variable resonance; envelope filtering; two analogue-to-digital potentiometer interfaces that can be used to control SID facilities; and an external audio input, which enables additional SID chips to be linked together. Other audio signals can be input, filtered and mixed with the standard SID outputs.

It would be impossible to detail the operation of each of these features here (several good books are available), but we can explain what all these phrases mean. First of all, oscillator synchronisation causes two signals (in this case two specified voices) to be harmonically locked together, making a single, more complex tone out of the two separate signals.

Modulation is the modification of one signal by another, affecting either the frequency or amplitude (volume) of the sound. Ring modulation is the amplitude modulation of one voice by another. This results in a tone that is clear but has a jarring, discordant effect and can be used to produce bell-like sounds similar to those of steel drums. Such sounds are said to have inharmonic overtones.

Filters enable specified frequency ranges to be eliminated from a signal. The different types of filtering possible on the Commodore 64 have effects that are suggested by their names: low pass filters cut out frequencies higher than a specified frequency; band pass filters eliminate frequencies above and below a specified 'band' of frequencies; notch filters are the inverse of band pass filters — they cut out a specified band; high pass filters cut

Guiding Light

Player-Missile graphics are one of the strong points of the Atari machines

Player-Missile Graphics

Player-Missile or 'PM' graphics form an important part of the Atari's graphics capabilities. They are similar in nature to the sprite graphics available on the Commodore 64 (see page 408) and the Sord M5, allowing the programmer to design and control up to eight different high resolution shapes. These movable shapes operate independently of any background display and may be programmed to move either in front of or behind any other shapes drawn on the screen. This allows the programmer to add a third dimension to the screen effects. PM graphics can be moved smoothly, at speed, across the screen and so are ideal for fast-moving arcade games. They can also be used to create more colourful static displays than are possible using the normal graphics modes, as PM objects can be coloured independently of each other and of the background display.

As with all sprite graphics, the secret of PM graphics' facilities lies in dedicated hardware. Special registers are designed to control the movement, colour and screen display of the PM objects. All the programmer has to do is place certain values in these registers to manipulate the objects. In BASIC this is done using the POKE command. Once a number is POKEd into the relevant register then the Atari's own hardware takes over the rest of the work. This is done at machine code speed and is therefore much faster than if the process was controlled from BASIC.

Let us now look at the creation of PM objects and the registers that control them. Players are designed from a vertical strip, eight pixels wide and 128 or 256 pixels high. Each row across the strip is represented as a single byte in the computer's memory. By POKEing suitable binary codes it is possible to define the shape of a player using a similar method to that used to create user-defined characters (see page 246). Up to four players may be defined in this way, each taking up