Waiting Room

Computers transfer information at a rate much faster than mechanical devices like printers can handle. This problem is solved by using a short-term memory called a buffer

The buffers used by trains are designed to cushion impact by absorbing energy in springs or damped pistons. Computers have buffers, too, and in some ways they function like the train's buffers by helping different parts of the computer system to 'get on together'.

The term is used somewhat loosely in the computer world and covers two quite distinct things. To the programmer 'buffer' means a specialised use of computer memory, while to the circuit designer it means a type of electrical signal amplifier. The second type, which we shall call signal buffers, is dealt with in the panel.

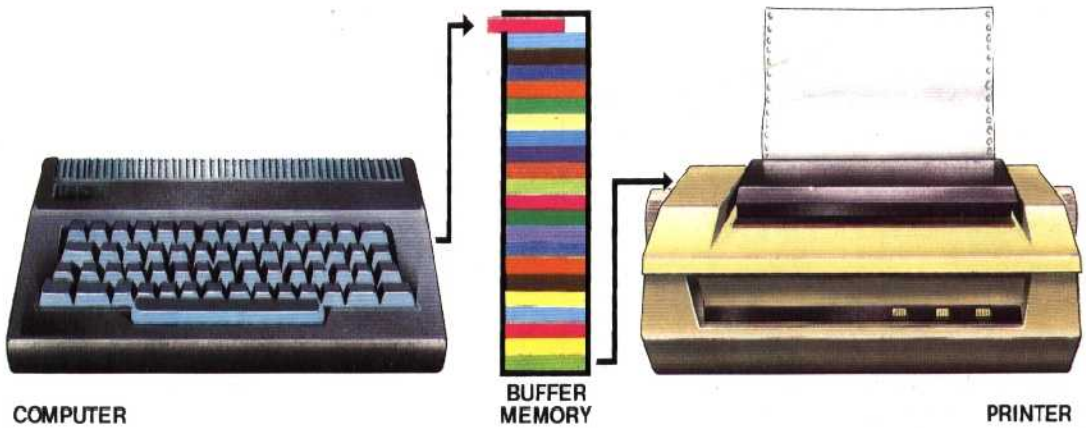
Memory Buffers

Imagine a word processor program that, among other things, can move a block of text from one part of a 'document' in the computer's memory to another part. The text consists of printable characters and spaces and certain 'unprintable' characters such as the Carriage Return. All these are represented in the computer's memory as ASCII codes in binary. One byte of memory is needed for each character. To move the characters in the block from their old positions in memory to their new ones means that another part of the computer's memory has to be set aside as a temporary text storage area. Such an area of memory specially set aside for a specific task is known as a buffer.

As a second example, consider the problem of printing a document created on a word processor. The document might consist of 15,000 separate characters, but they clearly could not all be sent to the printer for printing at once — most printers cannot print faster than about 80 characters a second. To overcome this, part of the computer's memory will be set aside, under the control of the word processor software, as a 'print buffer'. The software will first fill up this buffer with characters to be printed, and then send them out to be printed at a speed appropriate for the printer.

The print buffer may not be very big, perhaps only 128 or 256 bytes in capacity, but the principle remains the same however big it is. First a 'block' of ASCII characters is written into it, and then these are despatched out again, one byte at a time. The first byte to be written into the buffer will also be the first byte to be read out from it (obviously we will want the characters to be printed in the same order they were typed in). This type of buffer is called a 'First In First Out', or FIFO buffer. When all the characters have been read out from the buffer, the software will fill it with the next block of characters destined for the printer.

FIFO buffers of this type are an extremely common feature of most types of computer software. They are used wherever incompatibilities of speed exist, not only between computers and printers, but also between computers and floppy disk drives and between computers and computer keyboards. Although the phenomenal processing speed of computers means that they can usually identify which keys have been pressed faster than typists can type them, there may be times when the computer cannot identify the keys and display the corresponding characters fast enough. This can happen if the computer is momentarily busy doing something else (accessing a disk, for example). When this happens, it is common to incorporate into the computer's operating system a 'type ahead buffer'. This buffer 'remembers' which keys have been pressed and the computer displays them as soon as it can. Normally, the user will not notice



Temporary Stopover
One of the most common uses for a buffer is between the computer and a printer because the printer cannot output characters at the same speed that the computer sends them. Characters are consequently stored in the temporary memory until this buffer is full, whereupon a 'busy' signal is sent to the computer to stop it transmitting. The contents of the buffer memory are then sent to the printer in the same order that they were received, but at a much slower rate. When this has finished, the process begins again until the whole text has been printed