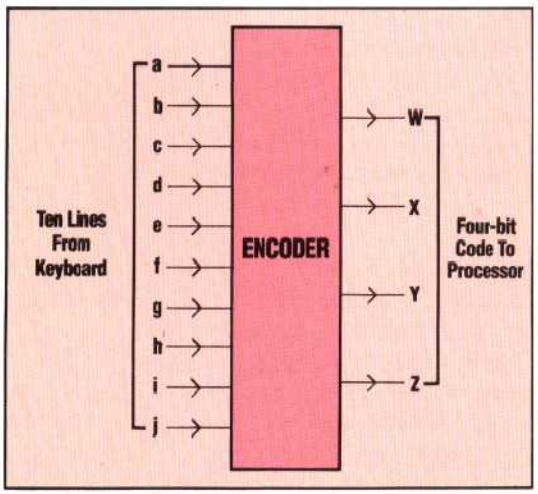
CHANGING LINES

The CPU of a home computer carries out its work by sending instructions, in the form of electrical pulses, down channels of communication to internal or peripheral devices. We look at the special logic circuits — known as encoders and decoders — which translate the instructions into electrical signals, and vice versa.

The processor sends instructions to both internal devices (the accumulator, ALU, etc.) and peripheral pieces of equipment (such as a printer). Often the number of lines used by a peripheral device may be reduced to provide a simplified input to the processor. This is known as *encoding*. An example of an encoder is a circuit used in conjunction with a keyboard. This may have 64 output lines, one of which produces a signal when the corresponding key is depressed. As only one key is usually pressed at any one time, each of the 64 possible output signals can be coded as a six-bit binary number. This means that only six lines are required to carry the information to the processor about which key was pressed. The device that converts 64 lines to six lines is an encoder. In practice, a further two lines are added, one to be used as a parity check and the other to signal that a Shift or Control key has been used in conjunction with another key.

To demonstrate this principle let's consider a much simpler keyboard that has only 10 keys, which would allow a user to type in a number from 0 to 9 inclusive. A three-bit binary code would give us only eight possible combinations, so we must design our encoder to accept 10 lines of input and produce four lines of output.



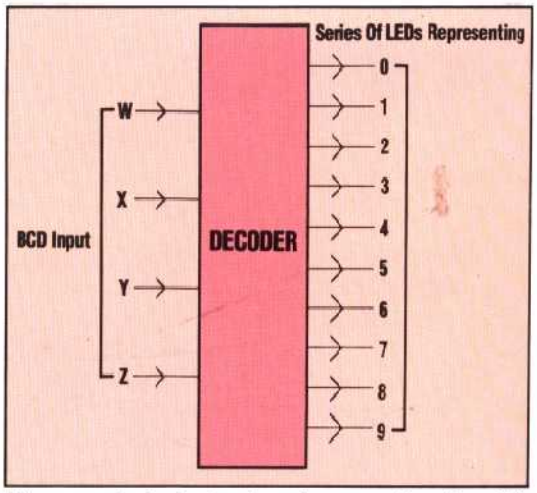
As only one of the 10 lines can be activated (or *set high*) at any one time, the truth table for the encoder will be:

Decimal	Inputs										Outputs			
	a	b	c	d	e	f	g	h	i	j	W	X	Y	Z
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	0	0	0	1	0	1
6	0	0	0	0	0	0	1	0	0	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	0	0	1	1	1
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1

Decoding is the reverse of encoding. Rather than accepting a large number of input lines to produce a small number of output lines, a decoder accepts a small number of inputs (usually in the form of binary codes from the processor) and from this selects one of a larger number of output lines, which control the activity of an output device such as a printer or x-y plotter. Encoders and decoders are also used in the control of disk head movements and selecting output channels from device numbers.

DECODER DESIGN

Let us now look at how a simple decoder may be designed using AND, OR and NOT gates, by considering the following problem. A decoder is required to convert binary-coded decimal (BCD) codes into a form that will switch on one of 10 LED lights corresponding to the decimal value of the code. Here we are really dealing with a circuit that will produce the reverse of the encoder example given earlier.



Binary-coded decimal codes are the four-bit binary representations of the decimal digits 0 to 9, and therefore the decoder will have four input lines. As any combination of the four lines can be set high, there are 16 possible inputs. We are only interested in the first 10 of these combinations and