



**THE POCKET
PROFESSOR**

COMPUTER DICTIONARY

**THE GUIDE TO
COMPUTING TERMS
AND CONCEPTS**

MICRO

MINI

MAINFRAME

SOFTWARE

HARDWARE

PERIPHERALS

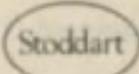
 THE POCKET
PROFESSOR

COMPUTER DICTIONARY

The guide to
computing terms and concepts

written by
Ian Scales and Geof Wheelwright

UK & USA
College Lane
Publishers

Canada
 Stoddart

Australia
& New Zealand
Methuen

© College Lane Publishers Limited and Studio Editoriale s.r.l. 1984

All rights reserved.

No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photography, recording, or any information storage or retrieval system, without permission in writing from the publisher.

First published in the United Kingdom and the United States of America, its territories and the Philippines in 1984 by College Lane Publishers Limited, 132 College Lane, Hove, East Sussex BN6 9AJ, England in association with Studio Editoriale s.r.l., Via Cerva 8, Milan, Italy.

UK ISBN 0 946913 14 5

USA ISBN 0 946913 15 3

First published in 1984 in Australia by Methuen Australia Pty Ltd, 44-50 Waterloo Road, North Ryde 2113, Australia.

First published in 1984 in New Zealand by Methuen New Zealand, 61 Beach Road, Auckland, New Zealand.

AUST/NZ ISBN 0 454 00768 X

First published in Canada in 1984 by Stoddart Publishing, A Division of General Publishing Co. Limited, 30 Lesmill Road, Toronto, Ontario, Canada.

CANADIAN CATALOGUING IN PUBLICATION DATA

Main entry under title:

Computer dictionary

(The Pocket Professor)

Includes index.

ISBN 0-7737-5015-0

1. Computers - Dictionaries I. Series.

QA76.15.C65 1984 001.64'03'21 C84-099260-2

Printed and bound in Italy

PREFACE

The widespread introduction of micro-computers in business and in the home seems to have divided the world into two camps. Those who 'know' about computers and those who don't.

Our ancestors had it easier. When the internal combustion engine was invented its method of working could easily be extrapolated from experience of steam engines. The other technological advances were similarly physical – even electricity is easily understood as a physical process. Computers don't provide this sort of intellectual luxury.

Instead of moving pieces of machinery back and forth they manipulate data or pieces of information. Using this information they can perform calculations. And they do it all invisibly.

However, it would be a mistake to think that to use computers efficiently you have to become a computer aficionado.

The principle behind this book is that how computers work is not crucial – how to gain a useful understanding of what they are capable of achieving is.

COMPUTERS AS CONSUMER ITEMS

When the motor car was first invented it was a very finicky device – prone to breaking down at the slightest excuse. It required much money to purchase, much technical know-how and creativity on the part of the owner to keep it ticking over, and not a little courage to actually drive along inadequate roads dominated by unpredictable livestock.

The first motorists were definitely a breed apart – too much money and too little sense. But, bit by bit (no pun intended) motoring became a much more popular activity, as the machines broke down less often and less cash was needed to purchase them.

The same process is happening with computers. Ten years ago the word 'com-

puter' would have caused the average person to immediately visualise a lot of people in white coats and national health spectacles tending a gigantic piece of machinery.

Nowadays most of us are more likely to think of a child monopolising the domestic TV to do battle with a hoard of space invaders – the microcomputer has almost reached the Model T Ford stage.

Today's motorists would never dream of taking a course in mechanics before buying a car. In reality there is no longer a need to do the equivalent with a computer but you do have to know how to drive one.

Like car knowledge, computer knowledge can be broken into several categories. At the top of the scale is the basic working knowledge and terminology. These are the terms and concepts you'll need to under-

stand how to actually make good use of a computer and to relate to people who are conversant with computers.

The next level is slightly more technical – more the nuts and bolts stuff which can, nevertheless, be quite useful.

The third (bottom) level is really divorced from everyday computer use. These components and processes should remain totally invisible to anyone using a computer. But if you really have the need to know, they're there.

If you're not convinced, read on.

At the very bottom of the concept of the car (the motive force) is the science of thermal physics – cars, after all, are driven by the expansion of ignited gases. The average driver doesn't need to know a thing about thermo-dynamics to buy, drive, or even fix a car. He/she may find

the subject interesting and it may help a lot in understanding the principles of car operation, but it's not necessary to know the exact terms or processes.

At the very bottom of the concept of the computer (the motive force) is the science of electronics. The computer is physically driven by electronic circuits. The average computer user may not need to know much about electronics to buy, operate, or even fix a computer. He/she may find subject interesting and it may help a lot in understanding the principles of computer operation, but it's not necessary to know the exact terms or processes.

The next level up is the component level – a car is put together by applying the sciences of thermal dynamics and motor-engineering (plus a lot of other sciences like aerodynamics, ergonomics, plastics

and soldering). To fix a car you have to know how it has been put together and how all the components function. If you own a car, however, you are almost certain to delegate this specialist activity to a professional motor mechanic while you content yourself with the chores of cleaning the windscreen, topping up the oil and water and being careful not to crash it. You don't have to know how a car fits together or how to fix it to own and operate one.

With a computer, the next level up is also the component level – a computer is put together by applying the sciences of electronics and software engineering (plus a lot of other sciences like styling, ergonomics, plastics and soldering). To fix a computer you have to know how it's been put together and how all the compo-

nents function. If you own a computer, however, you are almost certain to delegate this specialist activity to a professional computer maintenance engineer, while you content yourself with the chores of cleaning the screen, protecting the storage media and being careful not to crash it. You don't have to know how a computer fits together or how to fix it to own and operate one. And you don't have to know how to program — that is a specialist job for the professional software writer.

The top level is understanding a car, and the most important, involve the terms, concepts and skills related to owning and driving one. A good, confident and well-trained driver and a well-matched car (one that has enough space for storage, has enough power for the job you intend it to do and is priced to suit your needs) is the

end result. The car industry and its customers realised this truth long ago and 99 per cent of cars are designed with this imperative in mind. The best thing of all is that thermo-physicists, car designers, motor engineers and mechanics are almost certainly to be as statistically good or bad at driving a car as the average man in the seat.

The top level in understanding a computer, and the most important, are the terms, concepts and skills related to owning and operating one. A good, confident and well-trained user and a well-matched computer (one that has enough space for your storage, has enough power to do the job you intend to use it for and is priced to suit your needs) should be the end result. The computer industry and its customers still haven't quite

realised this basic fact and the majority of computer products are not yet designed with this imperative in mind, but they are getting there. The best thing of all is that the electronics engineer, the hardware designer, the software writer and the maintenance engineer are almost certainly to be as statistically good or bad at actually using a computer as the average man in the seat.

INTRODUCTION

Because a computer is a machine of sorts we usually try to understand its processes in a machine-type way. It's common to hear people talking of the computer doing this or performing that.

The whole subject might be easier if we could just forget about the computer for a moment, for it is really just a vehicle for a system called Boolean logic and it's the application of this logic which really concerns us.

Boolean logic simply says that everything can be determined ultimately as a series of yes/no decisions, i.e. if this (yes) then that, if not this (no) then something else. It simply breaks logic down to its bare bones.

We could use this system ourselves for doing mental calculations. As a useful exercise, think of the number of yes/no decisions you need to perform to open a

door. You advance toward the door. First you have to decide whether the door handle is on the right of the door or the left, if it's on the left, how high is it? Which way do you turn it? Does the door open inward or outward?

In fact we're actually skipping over dozens of yes/no decisions in this exercise. If we had no prior knowledge of doors our yes/no decisions might number thousands.

When we do want to go through a door we do all this calculating without being aware of it. Instead of consciously making a gigantic stream of yes/no decisions, we take short cuts of our own devising. If we had to perform the series of yes/no operations necessary for a computer we would be standing outside the door all day while we made up our minds.

The computer however, can be programmed to open a door using Boolean logic and what's more it can perform the calculations at far greater speed than we can.

Because it doesn't take short cuts like we do, but goes through every yes/no step, it doesn't make mistakes and because we've managed to make computers very fast, a computer could probably determine the steps necessary to open a door as fast as we could do it without thinking (should anyone wish to program it to do so). Presently, computers are only smart enough to understand simple yes/no questions, but man has been smart enough to configure these yes/no decisions into complex patterns and make the computer fast enough to execute the patterns and to act as a useful tool.

The key to understanding computers lies in grasping the first few concepts. Once you have these, the rest of the subject and all its terms will fall naturally into place.

How to use this book:

This book has been written to serve as both a quick reference guide to computer terms and as an introduction to the topic in general. Because computing is such an alien topic, a mountain of jargon has built up to define its processes and concepts. Much of this jargon, however is necessary. People who deal with computers have, after all, to have labels for what they're doing. But inevitably jargon also serves to mystify and make its subject matter privy only to those who understand the magic words.

As most computer terms are contingent

on others the structure of this book supports two types of entry: we have called the first the keyword category. The keywords run the length of the book and can be distinguished by this symbol: **K**. These definitions are detailed and take nothing for granted. They run in a logical order, introducing and explaining each contingent principle in a serial fashion. If you know little about computing read the keywords first — starting with **Application** and finishing with **Volatile**. The keywords are tagged to reference words and other keywords, though every attempt has been made to make the keywords stand as a self-contained introduction to computing. You only have to refer to other words if you're particularly interested.

Most of the book is made up of shorter definitions called reference words and

these make up the second type of entry. These definitions are for instant reference and liberally cross-refer to contingent concepts in the keyword section or associated terms in the other reference words.

Each entry falls into one of the two levels described above.

The entire keyword section naturally falls into the top 'drivers' level of definitions, as do most of the reference words. Driver level words are tagged with ●.

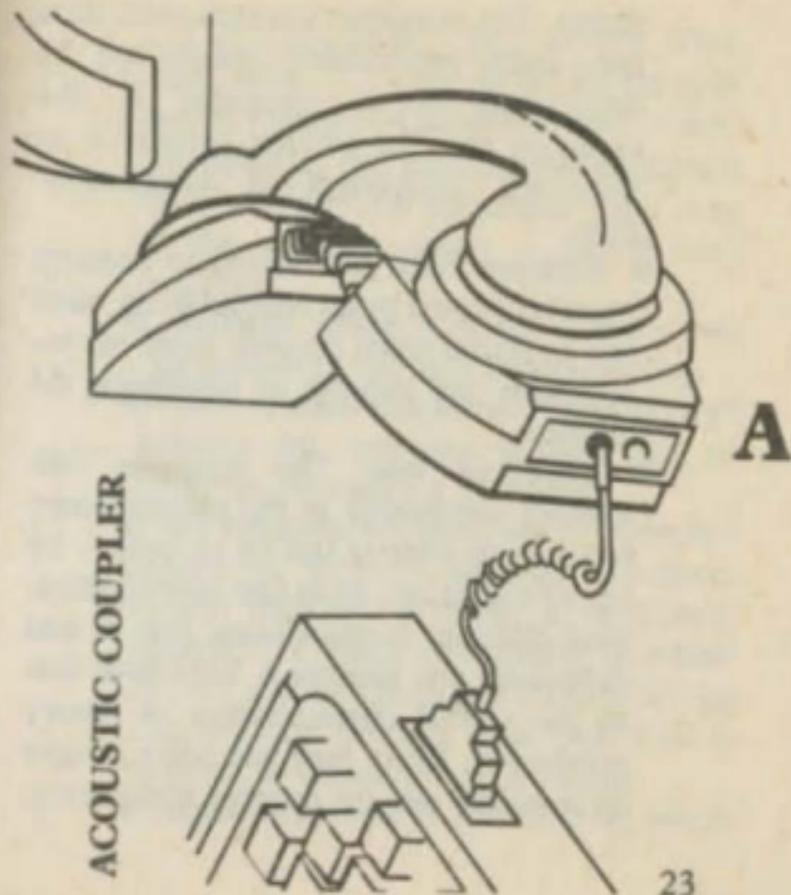
Many reference words are more at home in the component level and will, most of the time, be of little importance to the average user, though they may rear their ugly heads at the wrong moment and need to be referenced. They are tagged with ▲.

The last category is the strictly background nuts and bolts terms – the com-

puting equivalent of thermo-physics. These words are there for the adventurous or the scrabble enthusiasts – you never know when you might need one. They're tagged with ◆.

A

- **Abort** When *gremlins* cause the operator to prematurely *terminate* the *execution* of a *program*. When you abort it's usually because some sort of *error* has made further progress impossible on the program. A successful abort will at least *save* the *file*. The worst scenario is a *crash*.
- **Access** To 'access a *disk*' refers to the computer successfully finding and *loading* or *writing* a piece of *data* to or from the *storage system*. The efficiency of this operation is often termed *access time*.
- **Access time** The time it takes for a computer *device* to transfer information from its *storage system* to the computer's *screen* or *memory*.
- **Acoustic coupler** A *device* which acts as a reverse telephone handset, enabling *data* to be sent or received over the tele-



phone. The computer converts *serial data* into audio *modulations* which can be interpreted by a computer at the receiving end. An acoustic coupler is an important component in *communications*.

● **Address** Each *byte* stored in *memory* must be located by an address so its value can be found when needed. Most *micro-computers* are capable of addressing 64 *kilobytes*.

● **Address bus** The common 64k *memory* mentioned in the *address* entry requires an address bus of 16 lines – 16 on-off conditions totals just over 65 thousand different combinations. Just to add confusion, one *kilobyte* is 1024 *bytes* due to the special characteristics of *binary* numbering. Every time you add a nought *bit* to binary one the number doubles e.g.

1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096 and so on. By the time you've reached 64 kilobytes that 24 byte remainder at one kilobyte has multiplied itself to about 1.5 thousand. Take note of the numbers above – they'll keep occurring.

● **Address space** A range of space you can *address* or use in a computer *chip*.

● **Addressable memory** The amount of *memory* that you can actually 'get at' when *programming* a computer.

● **All singing, all dancing** A computer marketing term of doubtful precision. Often used to describe a marketing approach of which the speaker or writer doesn't approve, probably because he/she suspects the product itself will find it difficult to match up to its claims.

▲ **Alpha-numeric** A *character set* which

includes both numerals and alphabetical characters. An alpha-numeric keyboard, for instance, could be expected to be a traditional *qwerty*.

▲ **Analog** In many ways, analog is the opposite of *digital*. An analog device represents a value in a physical way. A thermometer is a good example – it simply indicates a value for temperature by pushing fluid up a glass tube. It's up to the viewer to determine what the temperature is by determining the rise against the marks on the glass. A digital device determines an abstract for the value and expresses it in numeric form. Digital devices, by the way, can be less accurate than analog ones simply because the digital device eventually has to arbitrarily settle on a decimal fraction to avoid some gigantic fractionalised number.

▲ **Analog to digital converter** A device which converts *analog* information into *digital* form.

● **And** A conditional term in the *BASIC* computer *programming language*. It can be used, for example when you want to test for the existence of two conditions and have the computer take a certain action when they occur. In the statement:

```
IF X = 20 and Y = 3 THEN PRINT  
    'You've got it'
```

the computer would only print the words 'You've got it' if, and only if, the value of *variable X* was equal to 20 and the value of *variable Y* was equal to three.

● **Apple** The US-based *microcomputer* company which pioneered the concept of a low-priced *personal computer* with the introduction of the Apple II in 1977.

What made the Apple II a real success was its own popularity. This is not a truism, for the II created the first commercially viable market for personal computer *applications software*. So many *programs* have been developed for use on the II over the years that the original *architecture* lives on several of Apple's latest products. Apple, however, has not been content to rest on the laurels of the Apple II. Although the arrival of the Apple III was a disappointment, the *Lisa*, launched in 1983 saw Apple again in the forefront of the *micro* industry. The *Lisa* used *32-bit microprocessor* technology, not to particularly increase the power of the programs but to make the standard applications much easier to use for the new breed of computer *user* who did not necessarily want to spend weeks learning

about the way a computer worked. *Friendliness* was the keynote. Apple calls this new approach to the user *Interface Lisa Technology*.

K ● **APPLICATION** This is a good place to start our 'keyword' definitions. Computing makes use of two elements. The *hardware* is obviously the computer and its associated components (a *printer* or a *device* for permanently storing *data*). The *software* is the intangible part but in a way the most important. Software is the *programs* or sets of *instructions* which make the computer operate specific tasks. Some software is designed to assist *programmers* or provide a system for the computer to interact with its *display* screen or *data storage*. This is called *system software*.

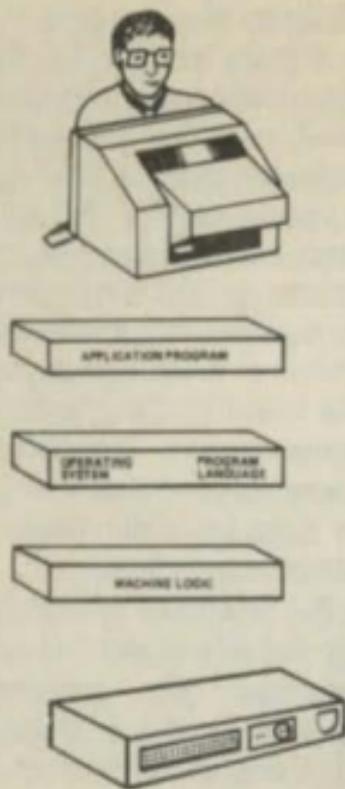
For *users*, the operation of this software

should be invisible. The other sort of software is applications software, which gets the computer to apply itself to a specific set of tasks useful to humans. A good applications program can be used with little technical knowledge. It has been designed to be used by the computer operator or user and will do a specific set of tasks; such as allowing the *entry* of account information to generate totals, balances and reports.

Application software is more or less the opposite of system software which the instruction system programmers use to generate the useful programs. We can think of computer software as a series of levels. Each level interprets between the level above and the level below. The basis is machine logic – the ones and noughts which provide the yeses and nos: the nuts

and bolts of *Boolean logic* (see *introduction*). People, however, are not very good at interpreting and manipulating a great series of yeses and nos and this is where the software starts to come in. The software takes a pattern of yeses and nos to perform a special *function*. Let's say it is a function to take two numbers and add them together. It's much easier for the programmer if this sort of yes/no pattern can be stored by the computer and called up when needed. The alternative is for the programmer to enter the yes/no logic every time he or she wants to add two numbers in a program.

So the first layer of software is a *language* which interprets between the computer and the programmer. The programmer's task is to insert the next level. Using the programming language,



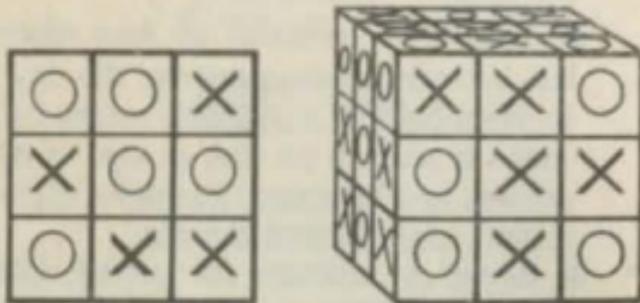
he/she creates the applications program which is then inserted between the user and the computer. The application program is a piece of software to *run* an accounting system or a *word-processing* system, for instance. Go to **BASIC** **K**

● **Architecture** This describes the physical and conceptual interconnections between the various components in a computer.

● **Archive** A collection of *data* which has become unnecessary for the day-to-day running of a computer *system*, but which is retained on *disk* or *tape* to be referred to if necessary. Examples might include accounts or letters sent.

▲ **Array** A structure for storing *data* within a *program*. In the *BASIC* computer language space must be reserved in a program for the insertion of data. The

data can be thought of '123456789'. Each element in the array can be taken out and identified according to its position. Arrays can also be two-dimensional, where each item of data in the array occupies a both a column and a row – like a game of noughts and crosses,



ARRAY

or even three-dimensional games where data occupies three relative positions (think of a rubik's cube).

- **Artificial intelligence** the ability (hotly debated) of a machine to think, reason or learn. Artificial intelligence is the ultimate goal of computer science. The Japanese in particular are investing heavily in the so-called *fifth generation* of computers which will combine high-speed *processing* with *software* which makes use of inference and analysis as well as hard logic.
- **Ascender** the portion on a *character* like k,d,h or b which sticks up above the body of the character.
- **ASCII** stands for American Standard Code for Information Interchange and is the standard way of representing *characters* in *binary* numbers – crucial when

computers of different makes *communicate*.

▲ **Assembler** A programming language which generates *machine code instructions*.

● **Asynchronous** A method of *communication* between two computer devices that doesn't require them to have the same timing signals. As long as the information between two asynchronous devices is being sent back and forth at a pre-determined rate, the *clock speed* or timing of each individual device is irrelevant.

● **Attribute** This usually refers to a *code* which delineates a certain *function* for the following item or items of *data*. For instance a screen attribute may refer to a code inserted in the middle of a *string* of text characters which changes the colour

of all the characters following it as the *program* puts the characters on the screen.

● **Auto-answer** This is a term normally associated with telephone-based computer *communications devices* called *modems*. It means simply that a modem with this facility (auto-answer), when hooked up to a computer, can answer the phone. This is particularly important when developing any form of *public-access database*, as you require the ability for people to use their computers to dial another computer and expect the receiving computer to answer.

● **Auto-dial** The ability of a *hardwired modem* to automatically dial a number.

● **Auto-repeat** A feature usually associated with computer *keyboards*. It means simply that when you hold down a *key*,

the *character* associated with that key will continue to be sent to the screen. One of the most common uses of auto-repeat is in computer games, where you can continuously 'fire' at something by holding down the space bar or *enter* key.

▲ **Azerty** European-style *keyboard* where the letters are arranged azerty on the top left of the board as opposed to the standard English *qwerty*.

B

● **Babbage** The last name of Charles Babbage, a computing pioneer. This English inventor and mathematician (born 1792 died 1871) is commonly recognised as having developed the world's first general-purpose computer – the analytical engine. But like many computers in this century, the machine had a brilliant specification, looked good on paper, but no fully working models of the thing were ever built. It was to have been a steam-driven device *programmed* by means of *punched cards*, but barely got off the drawing boards before old age and lack of finances took their toll on Mr Babbage.

● **Back-up** Something which duplicates the *function* of an active component and is kept on standby in case of disaster. It

B

is also common to talk of 'backing up' *disks* or *files* i.e. duplicating them, in case they are destroyed or *corrupted*.

▲ **Bank switching** The process of changing from using one set or bank of computer *memory chips* to another. In many computer *systems*, memory is not 'mapped out' as one large chunk, but rather as sections of 16 kilobytes, 32 kilobytes and so on. This means that these different 'banks' in an overall memory system can be set up so that they each hold completely independent sets of *data*. The action of moving from using one bank to another is known as bank-switching.

▲ **Bar chart** A form of information presentation commonly used on computers that involves employing horizontal or vertical bars of differing heights and

widths to represent certain values.

▲ **Bar code** A line of bars and spaces which is read by an *optical scanner*. Often used in stock control where goods are labelled by a small bar code sticker or the code is printed on the packaging.

● **Bar graph** A graphic representation of figures that uses solid bars of varying lengths and widths for purposes of comparison.

■ ● **BASIC** (Beginner's All purpose Symbolic Instruction Code). This is a high level *interpreter programming language*. BASIC is the most widely used programming language because it's easy to learn so it can be used after a little practice by the average computer *user* to *generate* simple programs.

As its name suggests, it is also usually pretty limiting, especially if the program

is a long one. A high level language like BASIC attempts to turn the computer's instructions into English-like *commands*. To do this it takes a whole *list* of yes/no instructions and runs them together. Some languages *compile* the *code* before the program is *run* but an interpretive language like BASIC keeps the translation at hand in the computer as it goes along. The computer examines the English-like *instruction* and then goes off and performs the series of yes/no *functions* designated to that BASIC command. Then it comes back for the next command and so on. Obviously there is a limit to the number of commands BASIC can contain, so to the computer the language appears rather clumsy. Also, because it has to switch its attention between the interpreter and the program and *data*,

BASIC tends to be slower than others languages when it's used to generate a program.

A compiled language, on the other hand, does the interpreting – from English-like commands to instructions the computer can understand, just once. The compiled version of the program is then run on the computer directly. Go to *binary* 

- ▲ **Batch** A term often used in relation to large *mainframe* computers. A batch is a collection of related *data* which is processed in one go.
- **Baud** Relates to the speed of a *data transmission*. Essentially it means the number of *bits* which can be sent per second. Common baud rates include 75, 300 and 1200 baud.
- **BBC Micro** An eight-bit 6502-based home computer developed by the British

Broadcasting Corporation in conjunction with Acorn Computers of Cambridge, England. The BBC Micro comes standard with a variety of *expansion interfaces* including plug-ins for *disk drives*, *printers*, a *second processor*, a *teletext* adaptor and a number of *machine-control devices*. The machine was used extensively as the 'benchtest' machine for two series of computer programs filmed by the BBC in the early 1980s and subsequently became a standard *micro* for schools in the UK as well as a popular home micro.

- **Bells and whistles** 'Trendy', innovative features claimed for a *hardware* or *software* product, usually by its producer's advertising agency.
- **Bespoke software** This will have been designed and written for a specific *application* as opposed to a standard applica-

tion *package*, in other words, the difference between 'tailor made' and 'off the peg'.

- **Bidirectional** A computer *printer* is often capable of printing alternately from left to right and right to left, saving the printing mechanism from returning to the left-hand margin on every line and so speeding the printing time.

K ● **BINARY** The computer's method of storing and manipulating numbers and *characters*. Our normal numbering uses base ten, in other words we count to nine then add a one (delineating one group of ten) and a nought. Therefore, 11 is one group of ten plus one. Binary uses base two and each binary number is known as a *bit*. Go to *bit* **K**

- **BIOS** Stands for Basic Input/Output System and used in the *CP/M business*

operating system for managing *serial devices* such as the *keyboard*, *screen* and some *printers*.

K ● **Bit** A computer can understand only two conditions – on or off. This can be further developed to mean yes or no or one or nought, so to use numbers it counts in base two or *binary*. You count nought then one then ten (one group of two plus nought), then 11 (one group of two plus one, which makes three), then 100 (one group of four, no group of two and no group of one) and so on. To make things manageable a computer usually deals with eight-bit numbers – numbers which consist of eight ones and noughts. Go to *bus* **K**

● **Block** A fixed length of space for the *storage* of *data* on magnetic media. Data is divided into blocks so that it can be

manipulated easily and flexibly on to the *storage system*.

▲ **Block move** A term usually associated with *word-processing* on a computer. A block move is the action of 'moving' a block of *on-screen* text from one place to another. This is commonly accomplished by placing some form of 'marker' at the beginning and end of the text you want to move, positioning the on-screen *cursor* where you want the text to be moved to and then hitting a button to *execute* the move.

● **Bold** Also known as emboldened and commonly used to describe a darker and deeper typeface facility used to highlight certain *characters*, words or phrases on a computer *screen* or *printer*.

● **Boolean logic** The basis of computer system where *functions* and *data* are rep-

resented as a series of yes/no, one/nought, combinations. See *introduction*.

● **Boot** Nothing violent – just the act of *loading* the *operating system* into the computer as a prelude to loading the *applications program* and then the *file* or files.

● **Bootstrap** Rather like a starter motor to get the system going. How do you *load* a *Disk Operating System (DOS)* off a *disk* without a DOS controlling the action? The bootstrap is kept in *ROM* and is capable of loading just enough of the *Operating System* to enable it to carry on and load itself.

● **bps** Stands for *bits per second*.

● **Bubble memory** A clever, but thus far expensive, method of *storing data*. Magnetic bubbles are created in a tiny sliver of material. By applying just the right sort of magnetic fields the 'bubbles' can be

made to 'circulate' and the polarity of the bubbles can be read in a *serial* fashion. Bubble memory is very reliable, compact and *non-volatile*.

● **Buffer** A segment of *memory* where *data* is held temporarily before it's transferred somewhere else. There is often a buffer for the *keyboard* to store key-strokes if the computer can't keep up while it's performing some complicated task. *Printers* are sometimes buffered so the computer can quickly send down the text to be printed and then get on with another task while the buffer *feeds* the printer the *characters* as it needs them.

● **Bulletin board** A computerised version of CB radio. A computer bulletin board uses the *communications* facilities of the *microcomputer* in conjunction with the ordinary phone system to let com-

puter *users* 'talk' to each other and exchange information and messages by phoning a central computer on which the messages and information can be left.

● **Bundled** The increasingly popular habit of computer manufacturers to 'throw in' free *software* on the sale of a *machine*. Bundled software usually includes an *operating system*, a *word-processing program* and perhaps *spreadsheet*, accounting or *database packages*.

K ● **BUS** These *eight-bit* numbers are dealt with by the computer in *parallel*. Instead of having to examine each nought and one in a *sequential* fashion they are sent around the computer in ranks of eight (some computers deal with 16 bits at a time or even 32 bits) over what's known as a bus. The bus is simply like an eight-



lane motorway and is usually etched on the *printed circuit board* (pcb). Its job is to deliver the information to the various components in the computer. Go to *byte* **K**

● **Business operating system** An *operating system* especially designed to run *software* appropriate for business *applications*. Popular business operating systems include *CP/M* and *MS DOS*.

● **Business system** A microcomputer *system* which is sold and *supported* as a system to be used in a business environment – to do accounting, stock control *word-processing* and so on. The business system contrasts with the *personal computer* which is designed to assist an individual *user*. In terms of the actual *hardware* there is often no difference between the two. It's more a question of the *appli-*

cations to which the computer is put. In general terms, if a computer is bought to perform a specific set of tasks in a business – run a sales ledger, for instance, then it's a business computer. If it has been bought either by or for the use of a particular individual within an organisation, then it's a personal computer.

K ● **BYTE** The eight-bit numbers are called bytes. Each byte represents a combination of on/off conditions or, as explained in the *binary* and bit entries, ones and noughts. Thus eight on/off conditions can represent 255 combinations. Count these numbers up in binary and you'll see how it works. Nought in binary equals nought in decimal and $1 = 1$, but $10 = 2$, $11 = 3$, $100 = 4$, $101 = 5$, $110 = 6$, $111 = 7$, $1000 = 8$, ... $10000 = 16$,

...100000 = 32, ...1000000 = 64,
...10000000 = 128, ...11111111 = 255. If
we'd added another bit to make
100000000 we would have got 256. This
is because 255 in binary is rather like 999
in decimal, it's the last number available
before you have to increase the number's
length.

So a byte can be expressed in 255 ways
(256 if we count 00000000). Each of
these expressions is known as a *character*.
Go to *cassette* **K**

C

- **Cable** Sheathed wire linkage, usually with plugs at either end, and which carry power from the mains supply or send information between *devices*.
- ▲ **CAD** (Computer Aided Design) A type of *application software* used by engineers, architects and graphic designers. Features can include three dimensional *graphics*.
- ▲ **Call** A *command* which brings (calls) a separately located series of *instructions* into operation during the *execution* of a *program*. The process often involves the *storing* of the *address* of the point of execution when the call was made. After the process resulting from the call has been completed the program resumes operation from the point at which it was called.
- ▲ **CAM** (Computer Aided Manufacturing) This is often associated with *CAD*

C

(Computer Aided Design) as CAD/CAM. CAM is employed where a computer *system* is used to control and monitor an assembly-line production process.

▲ **Cambridge Ring** A *local area network* system developed at a well-known university, which uses cheap, common twisted-pair *cables*. A *packet switching* method is used to transfer information between *devices* around a closed ring (each device is directly connected to two other devices). Features include a high *baud* rate and the receiving *machine* sending the packets back to the sender as an *error check*.

◆ **Capacitor** An electronic component which acts as a sort of dam: releasing electrons when they are needed and storing them up when they aren't.

● **Card** This is fast becoming a fossil. The punched card was a means of *storing digital* information using rows and columns of holes. Card is more likely to refer, these days, to an additional *printed circuit board* and associated components which can be inserted into a *microcomputer* to provide extra *hardware* features, i.e. more *memory*, *high resolution graphics* etc.

● **Cardbox program** An *application program* which replicates the activity associated with maintaining and using a simple *card-file system*. These programs usually enable the *entering* of short *records* with about six *fields* to each record. The essence of the programs is simplicity, and they are indeed easy to use – but rather limited in *function*.

● **Carriage** The roller assembly which holds and moves the paper on a type-

writer, or sometimes on a *printer*. On a computer printer the equivalent mechanism is known as a *platen* though this has not stopped the word 'carriage' creeping into computer parlance as Carriage Return (the key on an electric or electronic typewriter which sends the carriage back to its starting position for the next line). On a computer *keyboard* the *Return* key is usually used to mark a paragraph end or the end point of a typed *command*. When the computer has its Return key depressed it usually rushes off to see if it can *execute* whatever command preceded it.

K ● **CASSETTE** The domestic cassette recorder has proved a very popular method of *mass storage* for low-cost *micros*. As with the TV, however, it's been necessary to convert an *analog*

device for use with a *digital* one and this often causes problems.

The computer has to produce and interpret an audible *modulated* carrier wave through a special *interface* so that it can *save* and then *load programs* and *data*. Obviously, there is no computer-controlled *file* handling – if you save something on cassette you just have to note where on the tape it resides, manually wind the tape to the right spot and then attempt to load the program. The method is usually fraught with problems – it's often necessary to fiddle about with volume and tone controls on the recorder to get the computer to accept the data. Loading and saving times are also relatively slow.

There are a number of *systems* which attempt to provide the cost benefits of

tape with some of the facilities of a conventional *disk* system. There are several digital microcassette systems available, plus *Sinclair Research's* endless loop *microdrive*.

Digital cassette systems are also used as *streamer back-up* for *fixed disk systems* – these are efficient, *dedicated* devices and usually expensive.

The advantage of the analog cassette tape is its standardisation, both for recorders and cassettes, and for its cheapness. The method will probably remain popular for some time yet. Go to *character* **K**

- **Catalog** The list of *files* held on a *storage device* – usually a *disk drive*. To obtain a catalog, a simple *command* such as *catalog* or *directory* is typed into the computer.

- **Cathode Ray Tube** See *CRT*.

- ▲ **Cell** The physical *memory storage* area for one unit of *data*.

- **Centring** A *text formatting mode* where each line is positioned so that its mid-point is at the middle of the *screen*.

- **Centronics** The company which pioneered and championed the use of *parallel printer interfacing*.

- ▲ **Chain** Where items of *data* are *linked* together and sign-posted in such a way that the *system* knows to jump from one to the next in the set.

- ▲ **Channel** A conceptual path along which information can be sent. The term channel is often applied to *networking systems* where the routing of *data* is an important factor and cannot be totally discrete (it requires some degree of *user* intervention and understanding to properly manipulate the system).

K ● **CHARACTER** The 255 numbers available in a *byte* give more than enough space to allocate a byte to each letter of the alphabet in upper and lower case, each decimal number, and plenty of special symbols like “#!*%+?” and so on. All these can be displayed on the screen and are known as characters. With all these included only half the available 255 numbers are used. The other 128 are usually used as the computer's own special instructions for the processor, to enable it to perform calculations and *functions*.

Characters are displayed on a computer screen or on a *printer* as a *dot matrix* of eight rows by eight columns. Go to *default* **K**

● **Character codes** The numeric value the computer assigns to each letter, number and symbol the computer is

capable of producing. Character codes are also assigned to certain *cursor* and editing *functions* such as back-spacing, line feeds and tabulating. Upper and lower case versions of the same letter have different character codes.

● **Character printer** A *printer* which forms one *character* at a time — just as you or I would. *Line printers* form entire lines at once.

▲ **Character recognition** Just as men once dreamed of how it would be nice to fly, many now dream how nice it would be if a computer could automatically *read characters* from a printed page and save the bother of typing them all in. Several systems have been made to work, though never faultlessly. Some systems cheat by identifying patterns of *magnetic ink*. However, character recognition, like

voice input, is getting better and will probably be available at low cost.

● **Character set** The different *characters* able to be displayed on the *screen* by a computer or written on paper by a *printer*. Sometimes further defined as the *native character set*: the one that comes with the computer or printer.

● **Character string** A linear collection of *character bytes* which can be understood by the computer as an entity as well as individual characters. Character string manipulation plays an important part when a computer searches for words or *files* in a *database*.

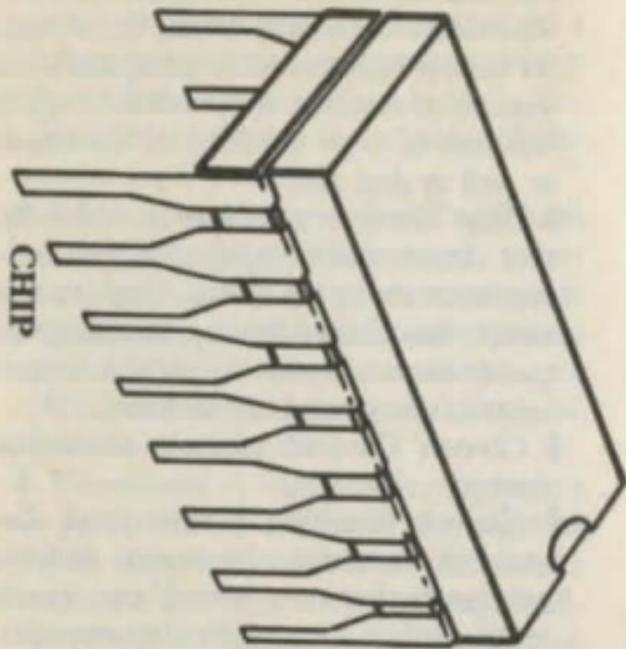
◆ **Checksum** A mathematical operation is performed on a *block of data* and its result (the checksum) is associated with that data and sent off to a *storage* unit. When the information is *read* back into

the computer it can perform the same sum again, check the new sum against the old, and determine whether the data has maintained its *integrity* during the voyage. As textual information is manipulated as a series of numbers it's possible to apply this sort of *error* checking to document as well as *data files*.

● **Chip** The sliver of *silicon* on which the tiny *circuits* and transistors of the *semiconductor device* are etched. Chip can also mean the entire device, including its plastic case and *pins* with which it connects to the *printed circuit board*.

◆ **Circuit** The path taken by an electric current.

◆ **Circuit diagram** A conceptual diagram of electronic components and the relationship between them. Uses a special set of symbols to simplify otherwise complicated arrangements.



- ◆ **Clock** The clock generates the computer's *pulses* and accurately records their number so the system can coordinate the activities of its various components.
- ◆ **Clock speed** In the phrase 'clock speed', clock is used merely as a determinate of whether a computer's on-board clock is running 'quickly' or 'slowly'. Machines with a slow clock speed are characterised by slow operation and slow execution of programs.
- **CMOS** (Complimentary Metal Oxide Silicon). CMOS *memory* and *processors* have the advantage of having an ultra-low power requirement. This means a CMOS *system* can be easily run from small batteries and heat dissipation is not a problem. An ideal combination for *lap-held portable* computers.

▲ **Coaxial** Usually refers to a type of cable where an inner wire is sheathed by insulative material upon which an outer braided tube of wire is meshed. This wire is further enclosed by a tube of insulative material. Coaxial cable is commonly seen connecting video equipment.

▲ **COBOL** (COmmon Business Oriented Language) A high level (easy to learn) *programming language*, mostly used in commerce and industry.

▲ **Code** A list of *program instructions*. Can also be active – putting a program idea into a workable form in the computer is to code it.

● **Cold start** After a computer has run into severe difficulties and has lost all its *memory*, including the *operating system*, you have to start from scratch and *load* it all back up again.

● **Command** As it sounds – a sequence of *characters* usually entered from the *keyboard* that causes the computer to undertake a series of *instructions*.

● **Comms** See *communications*.

● **Communicate** The verb form of *communications*. To communicate usually means to transfer *data* between two computers or computer *devices*. The act of communicating is known as *communications*.

● **Communicating word-processor** One able to send and receive text from *local* or *remote communication* facilities.

● **Communications** Catch-all term for the variety of communications technologies available to computers – *teletext*, *videotex*, *viewdata*. Also broadly covers *hardware* – *modems*, *serial interfaces*, *teletext adaptors*, plus *terminal software*.

It has become common for 'communications' to be abbreviated to 'Comms'. Comms cards, for instance, can be purchased for the *IBM PC* and will enable the *user* to configure additional *serial hardware* for *local* and *remote* communications.

● **Compile** A verb meaning to translate a computer *program* in *high level source code* into machine instructions.

▲ **Compiler** A high level *language* which converts the English-like *instructions* into the computer's language prior to its *running*, unlike *BASIC* which is *interpretive*.

● **Computer animation** Computers are particularly adept at *storing* three-dimensional picture images and then displaying them from a variety of perspectives. These techniques are becoming

very popular in advertising or special effect cinematography, for example, in films such as 'Star Wars'.

● **Computer graphics** Where a computer is used to produce pictures instead of words. Computer graphics are often used to illustrate *numeric data* in *bar charts*, *histograms*, *pie charts* and so on. *High resolution* graphics can also be generated to provide an enhancement to the *user interface* as with *Apple's Lisa Technology*. Less mundane uses include computer games, or point-of-sale displays.

● **Concurrency** This is usually understood to be the ability to get a computer to do several things at once. *Digital Research* manufactures an *operating system* called *Concurrent CP/M* which gives the *IBM personal computer* this capability. On a system such as *Concur-*

rent CP/M you can display several *programs* operating on-screen all at once in what are known as *windows* on one physical screen.

- **Condensed** Some *printers* offer a condensed *typeface*. The *characters* are squashed up resulting in more being printed across the line.
- **Configuration** A particular arrangement of components or a particular setting of *programming* options from within the *program*.
- **Connect time** The time which elapses between successfully *logging on* to an information service to disconnecting. On most *systems* connect time is usually charged at a set rate to the *user*.
- **Console** The *hardware* you look at when you're operating a computer – the *interface* between the *user* and the *machine*.

- **Continuous stationery** Perforated paper sheets joined together and stacked in a concertina arrangement. The *printer* can draw paper sheets through as they are needed without constant human intervention or an expensive *form feeder*. The paper can be torn into separate sheets after the printing process is completed.
- **Control character** A *character* which gives a direct *instruction* to the computer to take some action.
- **Control key** A special *key* on most computers, which when used in conjunction with a letter, is often used for issuing *commands* and *instructions* without having to type them out. On many computers, for example, the action of holding down the Control key and the 'P' (Print) key causes subsequent *output* to be sent to the *printer*.

- **Conversational** Where the *software* interrogates the *operator*, usually offering a series of options or yes/no decisions.
- **Co-processor** A *processor* which works in conjunction with the main processor in a computer. Also known as a *second processor*.
- **Corrupted** Refers commonly to information that has become garbled or destroyed. This can be caused by any one of a number of faults in either the *hardware*, *software* or *storage media* in a computer system.
- **Corrupting** the act of *data corruption*.
- **CP/M** (or Control Program Master) The first *business operating system* to gain wide currency on a *microcomputer*. The system was developed by Gary Kildall's *Digital Research* company and based

around the eight-bit *Zilog Z-80 microprocessor*. More advanced versions of the operating system have since been released for the *16-bit microprocessor* used in the *IBM PC* and *PC compatibles* in the form of *CP/M-86* and *Concurrent CP/M*.

- **Crash** Where the *hardware* or *software* suffers a serious accident – usually fatal.
- ▲ **Cross compiler** A *compiler run* on one computer which produces a *program* for use on another computer.
- **CRT** (Cathode Ray Tube) Same system as the domestic TV and still the most common way of providing a *display* for the computer. A stream of electrons are focussed into a thin beam and played across the inside of a phosphor-coated *screen*. The beam produces spots of light on the screen and its varying intensity and positioning builds up the picture. All

this happens in a vacuumous glass tube (a CRT) which is why television sets and monitors are apt to violently implode if handled too roughly.

● **Cursor** A small, sometimes flashing, symbol on the screen of a computer to indicate the spot where the next *character* typed will appear. The cursor seems to run along in front of the text as it's typed. It is most essential when the *user* wants to go back and edit earlier entries.

● **Cursor home** A *command* available in many *programs* which enables you to put the *cursor* back at the top left-hand corner of the *screen* with one *key* depression.

D

● **Daisy chain** where one *peripheral* is attached to another and so on in a *serial* fashion, with only one *cabl*e connection to the computer.

● **Daisy wheel printer** A *letter quality printer* which acts in much the same way as a typewriter, stamping fully-formed *characters* on to the paper through an ink ribbon. The characters are situated at the end of spokes radiating from a central wheel hub. The wheel is rotated so the correct sequence of characters are struck with a hammer as they pass through the 12 o'clock position. Daisy wheel printers tend to be slow and somewhat noisy, but their output is of very high quality. They are not usually capable of producing *graphics*.

● **Data** Such a well-used term that a strict

D

working definition is almost impossible. Means the information which is manipulated by the *program* to produce more information and so on. Data is often used as an opposing concept to text. *Text, files* for instance, contain numerically valueless *character codes* while data files often hold lots of numeric information.

▲ **Data breaks** A break in the flow of *data* from one computer to another. They occur whenever information in one computer *system* has to be manually entered into another. Data breaks occur most frequently in large firms which use several types of computer technology and have no facility for *linking* or transferring information between them. The frequency of data break occurrences is likely to decrease in proportion to the success of *Local Area Networks* (LANs) and the

continued development of high-speed data transfer systems. The biggest problem with *micros* has been that reliably transferring data between two machines has often meant using an RS-232 *serial* link, which doesn't often go much faster than a good office typist.

● **Data communications package** A piece of computer *software* that allows a computer to talk to another computer or computer *device*. Also known as a *terminal emulation package*.

▲ **Data compression** The process of removing unnecessary *characters* and gaps as a prelude to *storage* or *data* transmission. Also relates to complex methods of reducing the bulk of, especially, textual data. Normally, a computer uses a full *byte* to store every character, despite the fact that eight *bits* provides 256 letters in

the alphabet. In addition, a small percentage of these letters – a, e, i, o, u, b, c, r, t, h, g, l, s make up about 80 per cent of the characters in a document. Some characters i.e. *,?,@,\$, and so on are used hardly at all.

Compression techniques can assign, say, five-bit bytes to the common characters and 12-bit bytes to the uncommon, with, perhaps the midrange characters and their capitals sticking at eight. Even greater savings can be made if common words like 'and', 'the', 'are', etc. are given their own bytes.

● **Data corruption** When the *data*, either residing on a *medium* like a *floppy disk*, or perhaps being transmitted from a *remote* location, is physically and involuntarily changed – either by damage, in the case of a sensitive disk surface, or by

line noise as often happens when data is sent over the telephone system.

● **Data disk** A *floppy disk* set aside for the sole purpose of holding *data files* and containing no *program* files whatsoever.

▲ **Data integrity** That which is lost by *data corruption*.

● **Data transmission** The sending of *data* from one computer or *device* to another. In normal circumstances, however, the term is likely to apply specifically to the sending of data over the telephone system by means of a *modem*.

● **Database** A collection of information retained on *magnetic* media and updated and manipulated by a computer under the control of an *applications program*. The program can be configured to provide the *user* of the database with *reports* on the status of selected *records* or combi-

nations of records within the *system*.

▲ **DataBase Management System**

(DBMS) A *program* (or set of programs) which enables you to construct, manage and maintain a *database* on a computer.

● **Date-stamping** Used by many popular computer *operating systems* and is the process of recording the times and dates of when computer files are *accessed* modified and created. These times and dates are displayed when the computer's operating system calls for a *directory* or *catalog*.

● **dBase II** Most of today's popular computer *databases* are, in one way or another, modelled after dBase II. Although it was one of the first databases developed for *microcomputers*, it has also turned out to be one of the hardest and biggest selling. It was originally devel-

oped for use on *minicomputers* at the Jet Propulsion Laboratory in Pasadena, California as part of the Mariner space program. One of dBase II's biggest strengths has traditionally been its flexibility, but that flexibility is costly as dBase II is difficult to learn and hard to use – but once learned, it is very powerful.

▲ **Debug** To eliminate *errors* from a *program*.

▲ **Debugging aids** Special features in a *high level interpretive language* or special *program routines* in *low level* ones which assist the *programmer* to rewrite the *errors* introduced in a developing program.

● **DEC** (Digital Equipment Corporation). A well-known manufacturer of computers. Was the first company to market a mini computer in the 1970s.

● **Decimal** The human (Roman) num-

bering system which involves a base ten numbering system – you count with the numbers nought to nine add a digit and start again. In computing, this is opposed to *binary* (count to one and start again) or *hexadecimal* (count to 15 and start again).

● **Dedicated** Where an item of *hardware* is used for just one *application* or task. Where a computer has more than one *processor*, the second processor may be dedicated to maintaining the screen. It is usually called a dedicated *graphics processor*.

K ● **DEFAULT** An important universal concept in computing. Default is the procedure a *program* or a piece of *hardware* will carry out unless the *operator* specifically intervenes. For instance, most *applications programs* carry a set of

default settings for the *printing* procedure. When you specify to the *system* that you want to print out a *file* there are obviously a lot of questions that the system may need to ask in return, for example: how wide do you want the lines? how many spaces to the margin? how long is the paper in the printer? do you want double or single line spacing? do you want to show the *headers* and *footers* on the page?

Rather than have you go through every question every time you want to print something out, a system will have settings for each one of these variables. If, however, for some special reason, you want to change one of the printing parameters you can intervene and overrule the default setting before the printing is started.

The system, in this case, will also allow you to define or change the default setting itself. Go to DOS **3**.

- **Demodulation** The opposite of *modulation*. This is the process carried out at the receiving end of modulated *data* (often sent through a *modem*).
- **Demultiplexer** A device which carries out the receiving operation of converting *multiplexed* information to its (respective) *data streams*.
- **Density** This usually refers to the number of *data bits* over a given length of recording media. *Double density* indicates the data has been written on the *tracks* at twice the normal rate.
- **Desk-top computer** A *microcomputer* designed for office use on a desk.
- **Device** An item of computer *hardware* which communicates – either one-way or

both-ways – with a computer. Examples of common devices include *printers*, *VDUs*, *modems*, *disk drives*, *plotters*, *speakers* and *cassette* recorders.

- **Diagnostics** Where the computer has the ability to assist in detecting and locating *hardware* faults.
- **Dialcom** One of the larger companies involved in the international *Packet Switching Service (PSS)*
- **Digital** That which uses *binary* digits to represent text or *data*. Just about everything a computer does is digital. In computing digital is opposed to *analog*.
- ▲ **Digital Equipment** see *DEC*.
- ▲ **Digital Research** The company that pioneered and popularised the eight-bit *CP/M business operating system*. Digital has since gone on to develop versions of *CP/M* for the new range of *CP/M*

machines, including a version that allows the *concurrent* use of several *programs* at once.

▲ **DIN plug** A round, multi-pinned *plug* used on either end of a computer *device* connection.

● **Direct connect** See *hardwired*.

● **Directory** Usually refers to information maintained on each *disk* by the *operating system* which tells the system on which *tracks* and in which *sectors* the information can be found if it's about to *read data*, or which tracks and sectors have free space if it's about to *write*.

● **Disc** see *disk*.

● **Disk** Often used as an abbreviation for either *floppy disk* or *fixed disk*.

● **Disk drive** The mechanical *device* which holds and rotates a *floppy disk* or *fixed disk* and comprises an *access arm*

and *magnetic head* assembly to *read* and *write* to the *disk*. The disk drive also has a full complement of electronics to enable it to send and receive *data* to and from the computer.

● **Disk Operating System (DOS)** A *program* which controls the *saving* and *loading* of *files* or *programs*. In microcomputers, *operating system* and *disk operating system* are interchangeable terms. The most popular operating systems are *CP/M* and *MS DOS/PC DOS*.

● **Display** The picture (*text* or *graphics*) generated by a *program* or *operating system*. For example, a 40 column *display* shows the user lines of 40 characters. A colour display obviously features colour. Many computers have the ability to show a variety of displays or *display modes*.

● **Distributed processing** Where several

computers and their associated *peripherals* are *run* as a *network*.

● **Documentation** The *instructions*, tutorials, trouble-shooting guides and information which should accompany commercial *hardware* and *software* products.

● **Dongle** A piece of *hardware* containing one or more *Read Only Memory* devices (ROMs). It is inserted in a computer's *expansion slot* and is usually associated with a piece of *software*. The software will only *run* with the dongle in place, and so preventing nasty program *piracy*.

◆ **Doping** Introducing impurities (in a very precise way) during the manufacturing stage of a *semiconductor device* to turn a blank slice of *silicon* into a useful *chip* complete with the necessary *circuits*, *transistors* etc.

K ● **DOS** The computer has to have some

way of organising the *programs* and *data* in a system. The DOS (Disk Operating System) is the crucial component. It's a piece of *software* which provides a consistent set of rules for organising and *storing* information. In the *microcomputer* market manufacturers design their computers to be compatible with one or two specific *operating systems* and the *application* software manufacturers do the same with the software. In theory this means that the industry has a tendency to 'settle' on a handful of standard disk operating systems so that their programs will be usable on a wide variety of machines. The computer manufacturers do the same so their machines will have a wide variety of applications programs to be *run* on them. A computer system comprises three vital components: central

processor, memory and storage.

The processor is the brains of the system. It takes information from the memory and performs the necessary calculations, putting the results back in the memory again. The memory is also used by the processor to display text on the screen.

The system's memory is bound by two constraints. First, the contents will disappear when the power is disconnected from the system. Second, it is only of a finite size. In the case of a normal microcomputer, the memory is usually capable of holding about 64,000 *characters* of information – about 10,000 words. But it also has to hold the program which may take up more than half the available space.

On larger computers the figures are different but the result is the same – to

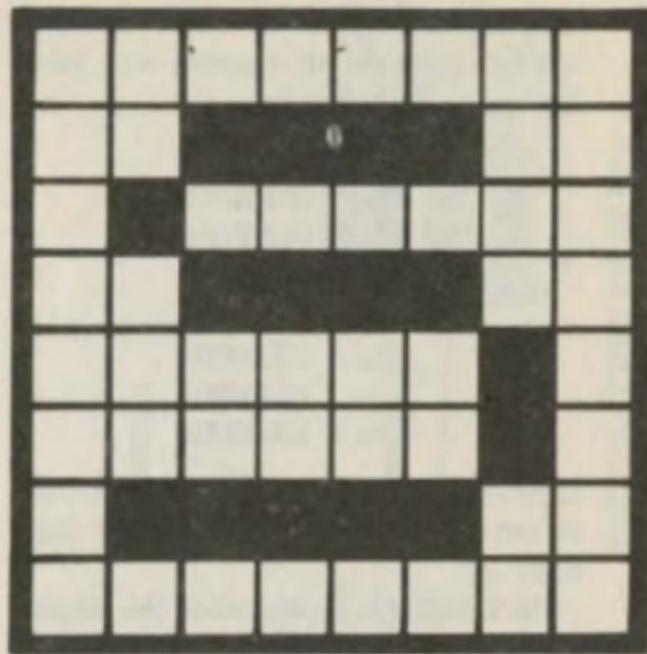
make the computer useful it has to have a consistent way of transferring the contents of the memory to some form of permanent *storage*, preferably one which is not bound by finite space constraints. So the third component in the system is magnetic storage. The *bytes* dealt with by the processor and memory have to be turned into individual *bits* and recorded on to the surface of a *magnetic disk* or *tape*. Most microcomputers use removable magnetic disks to store information, so you can store as much information as you have removable disks.

The job of the DOS, therefore, is to organise this process. The information in the computer must be designated into a *file* which is then *saved* to the disk in a coherent fashion. The DOS works out where on the disk the information should

be stored. It then organises the *writing* of the magnetic impulses on the storage medium and creates a *directory* on the disk so it has directions to find the file when the computer requires the information back again.

The DOS also handles the storage of the applications programs. Because of the limited memory size, many programs are split up into pieces. These pieces are only loaded into memory when the *user* requires the particular function they generate. Go to *dot matrix* **K**

K ● **DOT MATRIX** The basis of displaying textual computer information. A computer can understand on or off – so it finds it very difficult to understand curves, two-dimensional shapes and intensities. These are not impossible tasks for it but they do take up lots of *memory*



DOT MATRIX

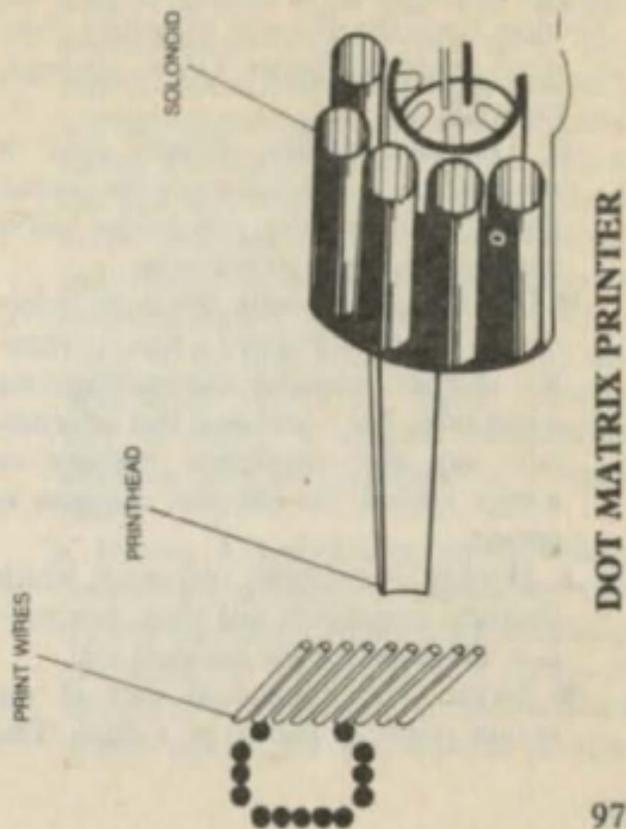
and would therefore be difficult to quickly generate on a screen.

The dot matrix is therefore an ideal alternative because an entire *character* can be displayed on a screen with just a few *bytes*, for example:

```
byte 1 00011000
byte 2 00100100
byte 3 01000010
byte 4 10000001
byte 5 11111111
byte 6 10000001
byte 7 10000001
byte 8 10000001
```

If you look closely you'll see that the ones or ons in this matrix displays the character 'A'.

The same logic is used when this display information is required to be printed. A



dot matrix *printer* is still the simplest and most cost-effective way of getting computer marks on paper. Go to *electronic mail* **K**

▲ **Double density** Where *data* is recorded on to a *disk* at twice the normal *density*. Usually using a technique called *modified frequency modulation*.

● **Download** To obtain computer information or *programs* from a source – usually another computer transmitting over a telephone line – and load that information into the computer's *memory* or *storage* system. Its effective opposite is *upload*.

▲ **D-plug** A multipin connector which connects computers and their *devices* – it is 'D'-shaped at the business end.

● **Drive** The mechanical part of the *storage system* is known as a drive. The

most common type for the average *microcomputer* is a *floppy disk* drive. The floppy disk is a flexible plastic disk with a magnetic coating encased in a protective jacket. The floppy disk drive clamps over the disk and rotates it. A magnetic head is then positioned and *reads* or *writes* the information. A *Winchester drive* rotates a non-removable rigid disk. The immobility of the disk is compensated by the Winchester drive's far higher capacity and its ability to transfer information between drive and computer at a faster rate.

▲ **Driver** A *program* or *routine* which handles the computer's relationship with a *peripheral device*. Many *word-processing* programs, for instance, have a variety of drivers to accommodate all the standard *printer* types. Once the driver is brought

into operation it will literally drive the printer in the way appropriate for *output* from the program.

● **Drum plotter** Where the *plotter* mechanism consists of a paper-*feed* mechanism (much like that found in a *printer*) and a pen moving back and forth across the face of the paper as it is drawn through the *device*. Drum plotters have been largely superseded by *flatbed plotters*.

● **Dumb** A *device* dependent for its useful operation on the presence of another device, usually a computer. The terms *dumb*, *smart* and *intelligent* usually refer to computer *terminals*. A dumb terminal is one with no intelligence – it is only used to send *commands* and receive display information from a computer.

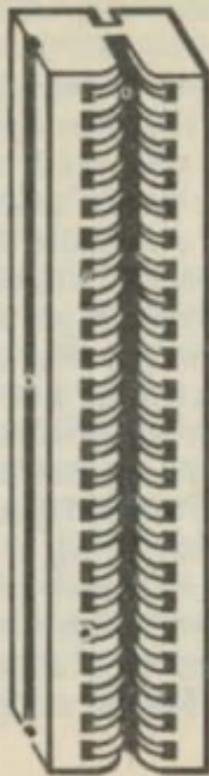
● **Dump** The non-selective transference of *data* from one device to another. For

instance, it is common to hear 'dumping the screen to the *printer*'. This means that the contents of the screen are printed directly on to the paper as they appear.

● **Dynamic RAM** Not as good as it sounds – a form of *Random Access Memory* which needs constant *memory refresh* to keep the contents intact. *Static RAM* is far less *volatile*.

E

- **Echo** A computer echo works in much the same way as a human echo. When you yell out your name in a big, empty room you often get the sound echoed back to you, and it confirms to you that the sound has travelled somewhere and come back. It's very common in computers, when using a *device* like a *printer* or a *modem* that you need to get some kind of an echo sent back from that device to confirm for the computer that the information sent out has been received. There are some special circumstances under which the sending back of that echo is unnecessary, and an echo suppressor is used to eliminate the echo.
- **Edge connector** An exposed portion of the computer's *printed circuit board* usually at the back or edge of the com-



EDGE CONNECTOR

E

puter. It serves to connect the computer to the outside world, often through standard *interfaces* that let the computer transmit signals to *printers*, TV sets, *monitors*, other computers and *modems*.

● **Editor** That which allows you to change or modify information you've typed into a computer. On most computers, you are given at least a *line editor*, which allows you to edit one specific line of a *program* at a time. The next grade up from this is the *screen editor*, which lets you edit any line of the program that appears on your current screen, thus enabling you to modify big chunks of a program without having to call the line number up every time. A third, and even better option, is the *text editor* which lets you treat the whole program as a block of text and lets you manipulate it exten-

sively without having to have all of it on-screen. There are also *word-processing* packages which allow you to edit computer programs, using all the powerful *commands* available to most word-processors. Once the program *file* is created under the word-processor, you can then *run* the file in whatever *programming language* it is designed for (*BASIC* - in the case of most home computers). The need for a good editor depends on what you want your computer to do. If you're not likely to be doing a great deal of programming, but rather hope to simply use *applications packages*, then the quality of the machine's built-in editor shouldn't be of great concern.

● **Eight-bit processor** The first form of *microcomputer processor* to gain wide acceptance. Although it has since been

superseded as the most popular processor in business equipment by the *16-bit processor* (as well as some forms of the *32-bit processor*), the eight-bit processor is still used in inexpensive home computers and lots of cut-price business machines. The 6502 processor in the *Apple II*, *BBC Micro*, Atari and Commodore computers (on some of these machines a 6502 variant was used, but it was still a chip from the same family) was one of the two major eight-bit processors to achieve wide acceptance. The other was the *Zilog Z-80* processor, which became the base around which the *CP/M business operating system* was first developed by Gary Kildall and his *Digital Research* company. For several years, these two eight-bit processors straddled the home and personal computer market until *IBM* declared its

intention to go for 16-bits in the *IBM PC*.

- **Electronic funds transfer** An important phrase to know since it is likely to change your financial future. The Electronic Funds Transfer System (EFTS) is a new paperless method of transferring credit information via computers. The system is based on the premise that one day everyone will have bank accounts which issue them with 'debit' rather than 'credit' cards. Instead of writing a cheque, using cash or using a credit card to do the weekly shopping, you would instead present this debit card. When presented at a store's check-out counter, the store's attendant would run your card through a computer. The computer would then check if there was enough money in your account to pay for whatever you were buying, then debit your account by the

amount of your purchase and credit the store's account for that same amount.

K ● **ELECTRONIC MAIL** One of the many services that can be provided to computer users hooked up to a *telecommunications network*. Electronic mail is a *system* under which a computer handles the delivering and sending of messages to people over a phone line. It usually works like this: you and a friend are both registered with an electronic mail service and both own computers with *communications software*, *modems* and *communications interfaces* and you want to send messages to each other. To send a message (otherwise known as a piece of electronic mail) to your friend, you type the message into your computer using any one of a number of *word-processors*. You then hook your computer and telephone

together through your *modem*, so the computer can send information down the phone line. When you dial up the phone number of the electronic mail service, a computer will answer at the other end and, if your computer is hooked up correctly, it will answer back. Once communication is firmly established between the computer at the electronic mail service and your *micro*, you simply answer a few questions regarding the 'number' of your mailbox, press a button on your computer and it 'sends' the message you wrote earlier. The message is then stored on the electronic mail service's computer and the service will tell your friend when he next calls up the service that a message from you is waiting. He can then read it, store it on his computer, print it out or do whatever he likes with it. Go to *error message* **K**

● **Electrosensitive matrix printer** A computer printing machine that uses specially coated paper and a 'burning-in' method to print numbers, letters and *graphics* on it. Electrosensitive matrix printers form their *characters* in much the same way as *dot matrix* printers. Rather than the printing *head* coming in contact with ink and then paper, the printing mechanism is electrically charged, so that where it comes in contact with the electrosensitive paper, the paper turns from its usual silver to a blue or black colour. This produces the effect of looking like printed text, although the *density* of the matrix on electrosensitive printers is often poor. Such printers are, however, usually inexpensive as they have few moving parts.

▲ **Emulation** The act of making one

thing act like another. The most common emulation task in computing is known as terminal emulation, which means that you make a computer (usually a small *micro*) act like it is a *terminal* hooked up to a larger computer (usually a *mainframe* or *minicomputer*). You can also buy emulation *cards* and *programs* which make one computer act like another so that it can use that computer's *software* and *peripherals*.

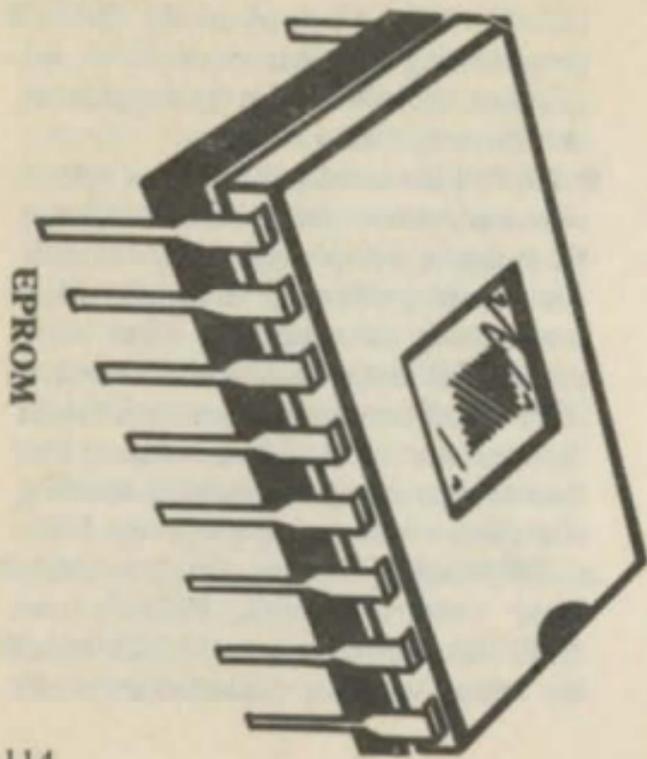
● **End user** Any person who uses a computer with *software*. *Programmers* are not usually considered to be end users because they are not usually at the 'end' of the line of people to use a computer. *Hardware* designers design the computers, *software* engineers and *programmers* make the *programs* that allow us regular folk to use the computers. We are

considered the 'end users' because we are at the end of this chain of production.

- **Enter** This usually refers to a certain *key* on a computer *keyboard*; the 'Enter' or *carriage return* key. This key is usually used as a confirmation of a given action within a *program*. In a *word-processor*, you hit the Enter key to confirm that you've come to the end of a paragraph or block of text. In a *BASIC* program, the Enter key is often required to confirm a response to a question posed by the computer. Enter can also be used as a verb, with people referring to having 'entered' or keyed-in a program (meaning that they typed a program into their computer).
- **Entry** A single unit of information entered into a computer system.
- **Envelope** In computing, this often deals with sound, and we speak of

defining the 'envelope' of a given sound. On a number of *micros*, there is an *ENVELOPE command* in the *BASIC* programming *language* which allows you to adjust and tinker with the frequencies and the nature of a sound.

- ◆ **EOF** This usually refers to a special *character, code* or marker that designates the end of a *file* on *magnetic tape* or *disk*. End of file markers are important when transmitting information to other computers, *disk drives* and *cassette recorders*. They tell the computer when a particular file has come to an end and confirm that the computer can go on and do something else (like *execute a program* in the file).
- ▲ **EPROM** (Erasable, Programmable Read Only Memory). PROMs and EPROMs are do-it-yourself *ROMs* and are often used by manufacturers for



storing early version ROM programs (e.g. the *BASIC* which usually comes in Read Only Memory with the machine). The EPROM is bought as a blank and inserted in an EPROM Programmer, or EPROM Blower. The *memory* is then set. However, the EPROM can be 'reblown' by exposing the *chip* inside to ultra-violet light (it has a small window for this purpose, though the window is usually covered with some sort of label to prevent any light getting in and *corrupting* the program. The light makes the EPROM blank again.

▲ **Erasable** A method of *storing* information that allows you to erase that same information. *Magnetic tape* and *floppy disks* are the most common erasable storage media, while the ROM (Read Only Memory) silicon chip is the most

common non-erasable media. Simply put, it describes a form of *data* storage from which unwanted data can be deleted.

▲ **Ergonomics** The science of office and office equipment design. An ergonomic office takes into consideration factors like the intensity and glare of lights, the proper angle at which you should be sitting to work at a computer *terminal*, how the *keys* should be configured on your computer or typewriter to get maximum efficiency out of it, and the size and type of letters on a computer screen.

● **Error** A sequence of *bits* which, when manipulated by the system, cause it to malfunction. An error is usually communicated to the *user* as an *error message* when he has entered a meaningless command. It may also occur when *data* is loaded into the computer from an

external *device* like a *disk drive*. Physical problems on the magnetic *medium* can cause the sequence of bits to be distorted and this will be picked up by the error checking system. Often the system will try again a few times. If it still can't get satisfaction it will give up and signal an error.

● **Error-checking** See *parity*.

■ **ERROR MESSAGE** Tells you when you're doing something wrong with a computer. Error messages vary in their depth, with the best *languages* and *operating systems* giving the most detailed explanations in their error messages, and the worst giving either a simple *numeric* code or just the word 'Error' in big letters. Error messages are very important in *debugging programs* as they can make the difference between finding a problem in

a program quickly and getting it corrected in a hurry, and fiddling around for days at a time trying to fix a single 'bug'.

They are also important when you have to deal with the operating system. When something goes wrong with, say a *floppy disk*, you're likely to get messages like — 'Can't read disk. (A)bort? (T)ry again?' This might sound a little terse, but is an improvement on earlier operating systems which were capable of throwing out messages like 'Error 6' when something went wrong.

Old *mainframe* computers often required programmers to put masses of paper *punchcards* through a large *cardreader*, only to find the computer telling them simply that there was some form of error somewhere in what were often huge programs. Even the smallest

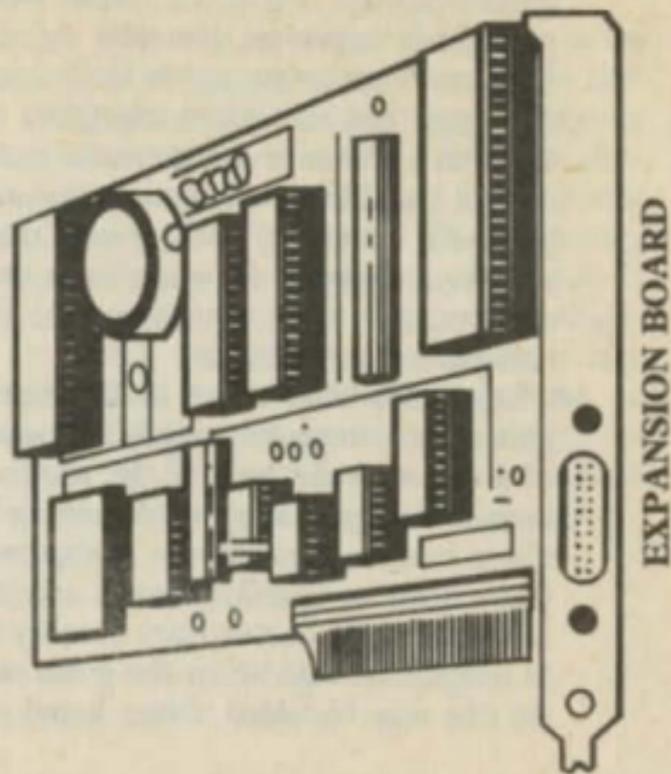
inexpensive *micro* now tells you at least which *line number* your error comes from and offers some sort of numeric code to describe the nature of that error. Go to *fixed disk* **K**

● **ESC** (Escape) Usually the *key* at the top left-hand corner of the computer *keyboard*. It is most often used in computer *software* as a *toggle* between one state and another or as a way of giving computer keys another value. By giving what is known as an 'escape sequence' (hitting the escape key in conjunction with some other key) you can generate values from the keyboard which will cause the computer or a *peripheral* attached to it to carry out special functions. Escape sequences are often used with *printers*, to allow the computer to specify where text should be *condensed*, emboldened,

italicised and underlined.

▲ **Ethernet** A specific type of *networking system* for *micros*. This network was developed by *Intel*, *Xerox* and *Digital Equipment Co.* using *coaxial cabling* to allow computer *terminals* to communicate between one another, quickly and efficiently. The system can transmit information at ten *megabits* per second and support more than 100 terminals or *workstations*.

- **Execute** To carry out a *command* or series of commands (program).
- **Execution** To *execute* a program. To carry out a procedure.
- **Expansion board** A *printed circuit board* which, when installed in a computer, gives the computer greater capabilities. Common expansion boards allow a computer to use *printers*, *modems*, *disk*



drives, colour *monitors*, *networks* and 80-column displays. Expansion boards were particularly important for older *micro-computers* such as the *Apple II*, because the Apple had no built-in capability to use even a printer or disk drive. As such, it was considered an *open computer* because it constituted nothing more than a shell and *keyboard* for whatever expansion boards or *cards* were important for each particular *application*.

● **Expansion interface** A facility incorporated in a computer, usually by means of a *socket* on the back of the machine accessible to the *user*, by which extra electronic componentry may be configured. The expansion interface usually accepts cartridges which contain extra memory or an *interface* through which *disk drives* and the like may be added. Often found on

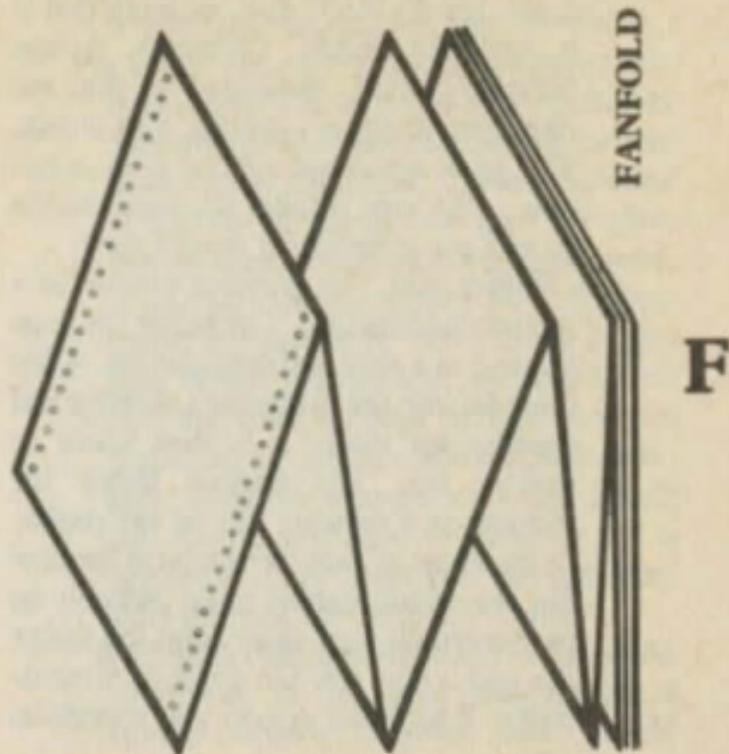
home/hobby machines or *lap-held portable*.

▲ **Export** The 'shipping' of information from one *program* to another. This is usually done by developing a common *format* under which information from one program can be exported, and then *imported* to another program. This exporting facility is becoming increasingly important in *integrated software* packages where, for example, the ability to take figures and tables prepared in a *spreadsheet* and export them so they can be incorporated into a *word-processed* business report, is well worth using.

F

● **Fanfold** A particular type of computer paper also known as *continuous stationery*. It is distinguished by holes at the side of the paper which fit on a *tractor feed printer* and perforations which allow the paper to be torn off in page-size pieces when it comes out of the printer. The holes at the side of the paper allow the sprockets on a tractor feed printer to 'pull' the paper through the printer and pass it in front of the *print head* where the text is printed on the paper and it is then fed line-by-line out of the other end. The sprocket holes and tractor feed are very important, as they ensure the smooth movement of the paper through the printer.

▲ **Fast access memory** Memory which mimics a *disk drive* or other *storage*



device. Fast access memory is also sometimes called a *RAM disk*, meaning that it is ordinary RAM (Random Access Memory) being configured so that the computer thinks it's dealing with a disk. The great advantage of this is in *access time* – the time it takes the computer to go and get information from a *file*.

▲ **Father file** The previous version of a recently updated *file*. The father file is an element in a *data* updating process. A file is *loaded* into the computer's *memory* and changes are made. It is then *saved* as another file. The original father file remains as a separate file on the *system*. In the event of *data corruption* in the new file, the recent updates could probably be performed all over again from the father file and a new son file created. Alternatively, if for some reason you wanted to

go back and see what the file was like without the updates, you have the option.

● **Feed** The movement of computer text by one line. The inclusion of feeds in text *output* to *printers* ensures that the printer moves down a line after it prints each line of text on the paper. *Line feeds* are often used in conjunction with *carriage* returns. A carriage return sent from the computer causes the printer to move to the beginning of the current line; when followed by a line feed, it brings the printer to the beginning of the line below. So a computer must often output a carriage return and then a line feed in order to get a printer to move down one line and then start printing the next line of text.

◆ **Field** A common element in a collection of *records*. If you were manipulating a simple *cardbox* program there would

probably be a separate field in each record for surname, Christian name, post code, telephone number. So if you wanted to search the *file* to see how many Smiths were in there you would instruct the system to search through all the records for Smith – a surname. The system would then examine all the surname fields in each record.

◆ **FIFO** Short for First In First Out and describes the order in which a computer handles or *stacks* certain types of information. It is the computer version of a bus queue – the first item of *data* to be entered into a FIFO system will be the first to be considered.

● **Fifth generation** A name describing the generation of computers that will operate using some form of *artificial intelligence* system. Although no computer yet has

been developed around anything more complicated than the principle of *Boolean logic* (the reduction of all problems – in their most simple form – to a series of noughts and ones), the fifth generation computer would move several steps beyond this. It will move to the point where the complexity of decisions about whether a piece of *data* is to be represented as a nought or one, is arrived at by some form of rudimentary logic and intelligence. Intelligence is often considered to be not only the ability to learn things and retain that knowledge, but also the ability to 'reason'. Artificially intelligent computers might 'reason' by identifying who it is they are working with and tailoring their method of operation to that person. If, for example, a blind person was using a fifth generation computer

that perhaps had a braille *keyboard*, a *speech synthesis* device and a *speech recognition* system a blind person could enter his identity and password and the computer system would from then on take commands from speech and 'answer' or carry out its work using the speech synthesis system. The 'intelligence' of this fifth-level system would have thus worked out who was using the system and taken action to respond to that.

● **File** The analogy to paper systems is fairly accurate. The file is the entity in which an individually accessible body of information is *stored* and manipulated by the computer. For instance, if you created a *financial model* on a *spreadsheet* program you would instruct the system to store the model as a specific file. That file can then be read into the computer

and manipulated by the spreadsheet program.

● **File access** The process of locating and *loading* a file into *memory*.

● **File creation** Naming and defining a *file* for the system. Under most *operating systems* there can be a variety of different types of file – document files, *ASCII* files, and so on.

● **Financial model** A business plan or economic projection commonly prepared using a financial modelling or *spreadsheet* computer *program*. A financial model is usually more detailed and complex than a simple spreadsheet, but it is trying to accomplish much the same sort of thing: projecting what will happen to a given financial enterprise according to the behaviour of certain unknowns.

▲ **Firmware** Software implemented and

sold on *hardware*. Firmware relates to a *program* which has been stored on a Read Only Memory device (*ROM*). Many games are sold as firmware in *cartridge* form. There is almost no likelihood of widespread *piracy* and the cartridge is a much more reliable medium than *tape* or *disk*. It is also faster to *load*. This method is, however, much more expensive.

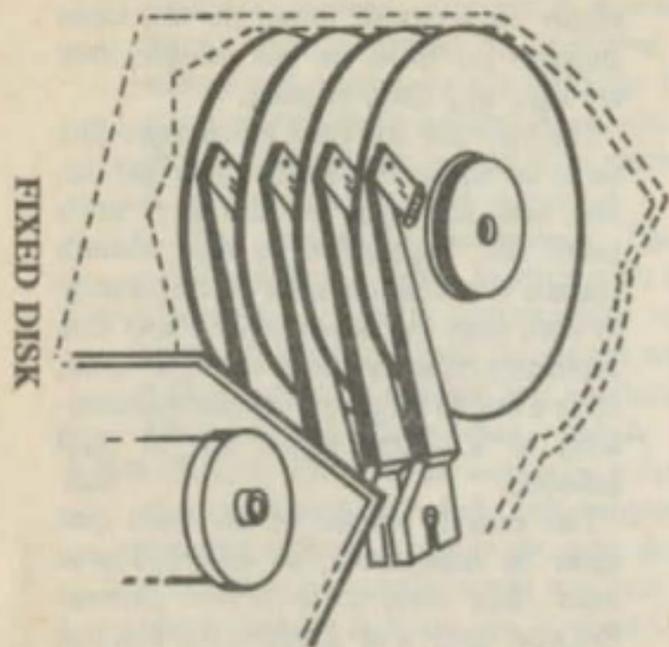
▲ **First generation computer** Those produced in the 1950s, starting with the Univac 1 – the first commercially successful computer. These computers were bulky, unreliable and had tiny memories. They ran on vacuum tubes – slowly.

■ ● **FIXED DISK** Also known as a *hard disk* or *Winchester disk*, this *magnetic storage* system is probably the most popular method of storing large amounts of *microcomputer data on line*. Instead of a

single circular thin *wafer* of magnetic material (in the form of a *floppy* or *flexible disk*) the hard disk system comprises a *platter* or series of disks. As the name suggests the disks are not flexible, they are rigid and fixed in place.

The hard disk can pack much more data on to an equivalent surface area and can also *read* and *write* its data at a much faster rate. Because of its hard, smooth surface it can handle a far greater *density* of data than the floppy disk. Fixed disk capacities typically run to three, five, ten, 20 or even 40 megabytes (million characters) to a floppy disk's 300 to 1000 *kilobytes*.

The smooth surface of the fixed disk must be maintained, so the *read/write head* 'flies' over it at a low altitude. Because there is no contact, the disk can



rotate at high speed without damage, so the rate at which the data is transferred to and from the computer is faster than that attained by a floppy disk.

But the fine tolerances which enable this performance are delicate so the disks must be kept sheltered from dust. This usually means that the disks are immovable, though there are some removable hard disk products available. These disks are completely sealed in a *cartridge*.

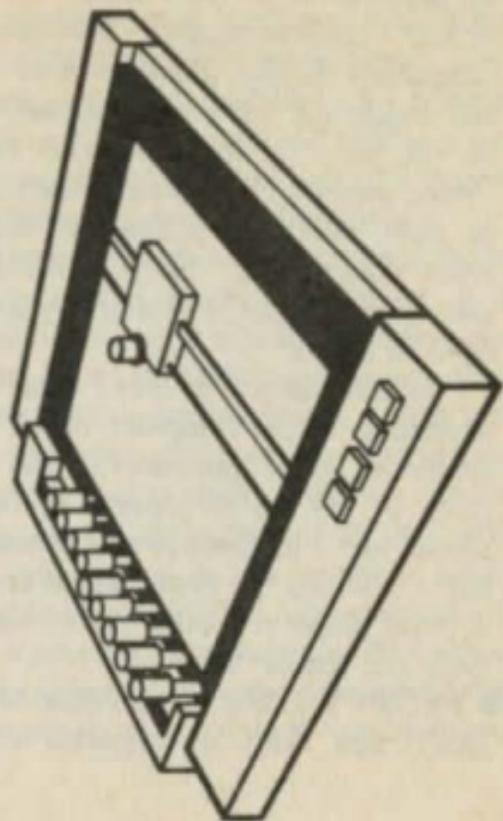
The disadvantage of the fixed disk as a competitor to the floppy is its price. Because of the precision of the components there's a limit to the downward drift in prices. Another problem is *back-up*. A fixed disk is just as reliable or unreliable as any other storage *device* and the information contained on it needs copying to avoid total disaster should it

crash. Therefore a system with a fixed disk also needs a floppy disk drive to provide a removable *medium*.

An alternative is *tape streamer* back-up – a 20 *megabyte* fixed disk could require about 60 floppy disks to duplicate its information. Obviously backing up with floppies here would be very time-consuming. One tape from a tape streamer can duplicate the contents of a fixed disk. The capacity and speed of a streamer system makes it a viable alternative. Go to *floppy disk* **K**

◆ **Flag** In programming a flag is an indicator in the *program* to mark an unusual event or show the existence of a certain condition. A flag is often a single *bit* which is set to on or off as the program *executes*.

● **Flat screen display** A generic term



FLATBED PLOTTER

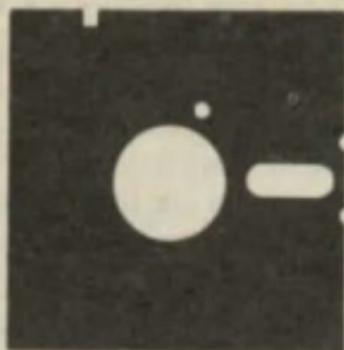
for alternatives to the *cathode ray tube* which conventionally provides a computer's *display*. Because of its bulk and weight the CRT is often unsuitable for use on a *lap-held computer*. *Liquid Crystal Display* (LCD) for instance, can be more expensive and usually display fewer lines of text. They do, however, take up little space, requiring only an inch or so in depth.

● **Flatbed plotter** A *device* fitted as a *peripheral* to the computer which plots geometric shapes generated by the computer by pen on to paper. The flatbed plotter uses a mechanical arm to manipulate a pen over the paper which lies flat. A *drum plotter* is another device for carrying out similar tasks.

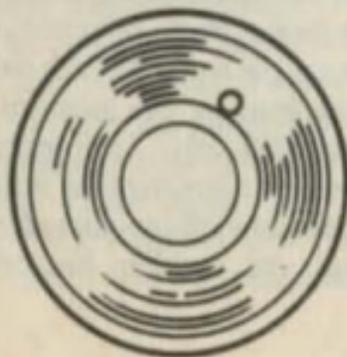
■ ● **FLOPPY DISK** A thin (like camera film) *disk* with a magnetic coating

enclosed in plastic jacket. The floppy disk is combined with the floppy disk drive to provide a means of permanently *storing programs* and information. Magnetic information is *written* and *read* as a series of magnetic impulses on the magnetic oxide coating on the disk surface.

Floppy disks come in four sizes and two types. The earliest and largest is the eight-inch and is rapidly becoming obsolete. It and the standard 5.25-inch floppy disk have a flexible plastic jacket which protects the disk's surface. The standard flexible jacket does, however, expose a *window* on the surface of the disk so that it can be read by the *magnetic head* in the floppy disk drive. Unfortunately, this is a possible entry-point for dust and dirt which may *corrupt* the *data* stored there. However, the problem has been over-



FLOPPY DISK



come by the latest version of the floppy disk – the three- and 3.5- inch floppy disks. Although the disk itself is still flexible, it is completely encased in a hard plastic shell. An aluminium shutter covers the magnetic head window while the disk is out of the *drive*. The standard 5.25- and eight-inch floppy disk is exposed at its centre where the drive clamps directly to the disk itself.

The hard-shelled floppy, however, has a central hub which is engaged by a short elongated shaft in the drive. Because they're smaller, the three-inch and 3.5-inch floppy disks offer several advantages. Less power is needed so there is less of a heat dissipation problem. Smaller distances are travelled by the *read/write head* so *access times* have the potential to be improved. The disks themselves are

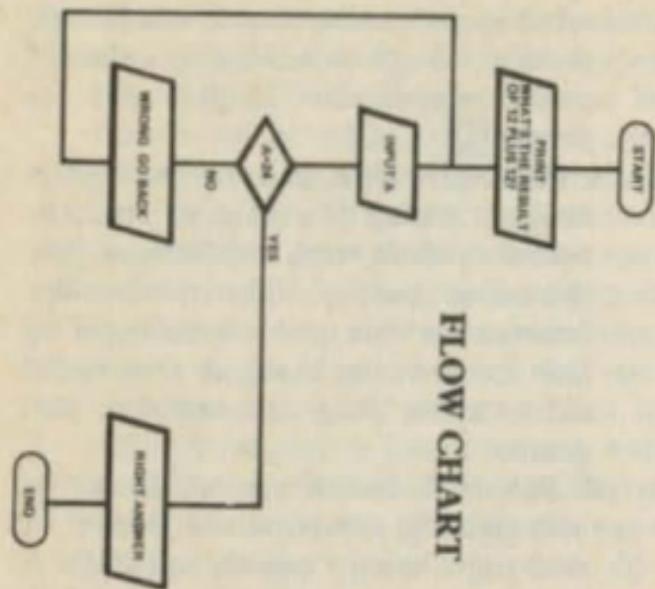
well-protected and likely to throw up fewer *corruption* problems.

Floppy disk capacities can vary greatly. While the eight-inch disk format has always remained rather standardised (being introduced by *IBM* has helped matters), 5.25-inch floppy disks can be used in drives which operate to all sorts of specifications. Generally, floppy disks are manufactured by the same process but different grade production runs earmark the disks in that run for different capacities. Some drives use both sides, with a magnetic head for each disk surface. Some write and read to the disk at different *densities* or *format* the disk with a different number of *tracks*. The floppy disks themselves are generally the same physical design; it's the drive that's different.

Manufacturers usually grade the disks and allocate each grade to be used reliably with a certain type of drive. The specifications usually comprise single-sided/single density, double-sided/single density, single-sided/double density, double-sided/double density. Go to *format* **K**

▲ **Flowchart** Used as an aid to *program* design. Consists of a series of interconnected symbols, each indicating a programming *function*. The relationships between the functions are indicated by lines between the boxes. A very useful aid in doing long and complex programs.

● **Footer** A line of text which can be automatically printed at the bottom of each page during a printing operation. A footer may say 'page 1'. On each occur-



rence the numbering is naturally increased, though the word 'page' will remain the same. A *header* performs the same function except it's at the top of a page.

● **Footprint** The surface area taken up by a computer or *peripheral* – a particularly important statistic for *desk-top systems*.

▲ **Font** An old printing term for the type of type – for instance some *typefaces* include the serifs (the little feet on the bottom of the letters), some don't. Variations like these are designed into a computer's typeface (the artistic design of its *character set*) and a set of *alphanumeric characters* with the same designs is known as a font. *Typestyles* can be further imposed over the individual fonts. Fonts may often be *boldened*, *italicised* or

condensed. Some computers, like *Apple's* Macintosh, have the ability to *display* and *print* out a variety of fonts, each in a variety of styles.

▲ **Foreground** Relates mostly to large computers. A foreground program is given a high priority on a *multi-user system* – in other words its instructions are *executed* before those of a background program. Especially important if the *user* is waiting at a *terminal* somewhere for the information to come back. Less time-critical tasks are consigned to the background and executed when there's time and space.

● **Form feed** A facility found on many *printers* where the printer feeds the paper through to the top of the next form or page – usually after the document just printed has *terminated* halfway through a

page. Form feed can be executed under *program control* or under *user-control* through a button on the printer.

▲ **Form feeder** The mechanical device which automatically feeds pieces of paper into a *printer* – usually a *daisy wheel printer* where the job requires the use of letterhead paper.

● **Form letter** A standard letter usually sent out to a list of people in a *batch*. A computerised *word-processing system* is capable of producing a standard letter and making it look as if it has been written especially for the target. It does this by automatically inserting the correct name and address during the *printing* process. The finished result contrasts favourably with form letters produced on a photocopier with the names inserted on a typewriter or with a pen.

K ● **FORMAT** The way the *hard disk* or *floppy disk* is physically organised into *tracks* and *sectors*. Can also be used as an active verb. You can 'format' floppy disks, for instance, before they are used to *store* information. The formatting process involves the *system* marking off the tracks into sectors and testing the sectors to ensure they are physically capable of taking the information without the risk of *data corruption*.

Because of the different formatting characteristics required by each *drive*, floppy disks are usually formatted by the *user* through a *utility program* which comes with the *Disk Operating System* being used with the computer. The operating system then works out where the individual *files* are to be held on disk according to the format it has already laid

down on its surface. As the disk starts to fill up with information and some of the files are deleted by the operating system (at the user's instruction) the sectors thus freed for further use can be rewritten with new information. Splitting the storage area into lots of sectors enables the system to manage the placement of continually changing numbers and sizes of files with the optimum in space economy. Go to help **K**

● **Forth** A computer *language* known for its speed of *execution*, use of *Reverse Polish Notation* and *LIFO* (Last In First Out) *stacking* system.

▲ **FORTRAN** A *high-level programming language* standing for FORMula TRANslation. Mostly used for scientific and mathematical work and probably not for the interested bystander.

- ▲ **Four-bit byte** A half sized *byte* – sometimes called a *nibble*.
- **Friction feed** A feeding mechanism for a computer *printer* which relies on rubber rollers (like a typewriter) to move the paper through the system, so contrasting with the more usual *pin* or *tractor feed* methods which ensure paper alignment.
- **Friendly** A very important concept in computing which means just as it sounds. *Software* and *hardware* are friendly if the way they operate and the way they are controlled can be communicated to the *user* in simple language and if the knowledge needed to operated the item is easily learnt. A friendly system must also prompt the user and expect commands back in plain English.
- **Front end** The *device* or part of a *program* through which the *user* interacts

with the system. The *front end* is the aspect of the system by which the user understands what's going on.

- **Full duplex** In *communications*, where information can be sent and received at the same time. See *half duplex*, *simplex*.
- ▲ **Function** Properly, it is anything which is done or performed, though in computing it will usually refer specifically to a *programming* function. In *BASIC* for instance, there is usually a *DEFine FUNction* (DEF FN) *command* where the programmer can define a mathematical or logical function, assign it a name and then call it into the program whenever it is needed.
- **Function keys** A cluster of *keys* on a common *microcomputer keyboard*, whose function is defined entirely by the *program* under which it is working. Any

given function key can easily be programmed to carry out a wide variety of tasks when depressed. The function keys on a number of computers can have more than one value per key, with second and third values being provided by depressing the function key in conjunction with the *control* key and the shift key respectively.

G

- ◆ **Gap** A space between two *records* on a *disk* or *tape* recording medium. The gap is set as a series of zeros and is simply there to overcome some of the physical problems of a mechanical system. It allows for speed fluctuations and so forth.
- ▲ **Garbage collection** A process relating mostly to *virtual storage systems* where redundant information is identified and deleted.
- ▲ **Gas plasma panel** A type of *flat screen display* where a *matrix* of cells is filled with neon gas and an electrical charge, applied to the individual cells, illuminates them to form the *characters*. It has the advantage of not requiring continual *memory refresh* as once the cell has been turned on it stays that way until changed.
- ▲ **Gate** The physical junction in a *tran-*

G

istor controlling the on/off condition. It's the complex series of gate conditions which performs the necessary *Boolean logic* functions to process information or store items in *memory*.

● **Generate** To create or cause to be created. You might speak of a computer system 'generating' a *line feed* – meaning that the computer system would cause an *output device* (such as a *CRT* or *printer*) to move down one line before putting its next piece of information on-screen.

● **Generation** The history of the computer's technological development can thus far be split into three distinct phases: first-generation computers were those running on valves (these used a lot of electricity); second-generation computers were upgraded to run on *transistors* and third generation computers are the

ones with *semiconductors* we use at present. There is some controversy over whether we are now in the third or fourth generations due to the advances made over the past few years with *Very Large Scale Integration*. The Japanese, for instance, now claim to be working on the *fifth generation* of computers which will use the huge *VLSI memories* and powerful *processors* (possibly *transputors* running in parallel) to generate programs with *artificial intelligence*.

▲ **Generator** A special *program* which creates another program or adapts a loosely-defined *applications program*. A generator is often used with loosely structured *database* programs to create *report* formats to extract useful information from the database.

● **Gigabyte** One thousand million (bil-

lion) *bytes*. Sometimes expressed as *gbyte*. The coming generation of *optical disk storage systems* are capable of dealing in gigabytes of *data*.

● **Glitch** A momentary cut-out in voltage from the mains which will usually result in a loss of the *memory*. A glitch can occur when there's a sudden increase in power consumption on the local circuit – it can even be induced by a refrigerator turning itself on. There is a wide variety of power conditioners available to cure (or not cure) the problem. The opposite to a glitch is a *spike*.

▲ **Global** With respect to total rather than partial. It's common to talk of a *global search* in a *program* where the computer goes off and looks through the entire *file* for a word, words or *data*.

● **Golfball** The printing element on a

golfball printer. The *alpha-numeric characters* are moulded on a single sphere. The print mechanism rotates the sphere to select a character and the formed character is struck through a ribbon on to the paper.

● **Graphics** Generally taken to mean almost any non-text representations produced by a computer, although it commonly means pictures, drawings and graphs produced by a computer. The best computer graphics are made by separately *addressing* each *pixel* on the computer's *screen*.

● **Gremlin** Elusive or mystifying fault in a *program* or item of *hardware*.

▲ **Gun** The part of the *cathode ray tube* which generates the electrons for the beam.

H

- **Half-duplex** —Totally misleading semantics. Actually means the ability to send information on a *modem link* in both directions but only in one direction at a time. The two-way *communication* does not occur simultaneously as in *full duplex*.
- ▲ **Hamming code** A system of receiving *data* transmissions where enough *error checking* information is included with the data so that the receiving system can automatically correct single *bit* mistakes without asking the sender to send the information again.
- ▲ **Hand-shaking** A set exchange of signals between two pieces of equipment as a prelude to the sending of information. The signals ensure that either item is performing to the satisfaction of the other. Handshaking occurs between *modems*.

- **Hang up** Where a fault in the *program* or the *hardware* causes the *system* to jam. When a system 'hangs up' everything on the *screen* freezes and it's usually impossible to *enter commands* through the *keyboard*. The only solution is to *reset* the computer and start again.
- **Hard copy** A computer *output* usually *printed* on paper. This is different from a *soft copy* display as seen on a *VDU* screen.
- **Hard disk** See *fixed disk*.
- **Hardware** A general term describing all of the physical components of a computer, from *keyboards*, *monitors* and *disk drives* to *chips* and wires. *Software*, on the other hand, describes the *programs* which may *run* on a computer.
- **Hardwired** This usually refers to a type of *modem*. Instead of sending

H

impulses through an *acoustic coupler* and thence through the telephone handset, the hardwired modem sends its signals directly down the telephone line. It has the advantage of easier set-up, less chance of background noise *corrupting* the data and is capable of *auto-dial* and *auto-answer*. Disadvantages include less portability (it usually has to be hooked into a special wall socket). Also known as a *direct connect* modem.

- **Head** The assembly containing the electromagnets which *read* and *write* to a *tape*, *floppy disk*, or *Winchester* disk.
- **Header Programs** which have the ability to *print* material to paper will often automatically print an identifying line at the top of each page. It could include the title of the document, name of the author, date and so on. It will also usually auto-

matically number each of the pages. Header can also mean an identifying tag at the beginning of a *data communications packet*. This provides the *packet switching system* with the necessary information to direct the packet to its destination.

- **Helix printer** An *impact printer* which produces a similar result to a *dot matrix printer*. Hammers strike a revolving cylinder with ridges. The intersection point of the vertical wedge-shaped hammer on one side of the paper and ribbon and the horizontal ridge on the cylinder on the other produces the dots to form the *characters*.

- K** ● **HELP** Usually describes the information assistance portion of a computer *program*. Help in a computer program can take many forms – from the know-it-all

who sits over your shoulder and endlessly offers 'constructive' criticism to the trouble-shooting section of the program *documentation* – but the most popular form of help is the *help menu*.

The help menu is a batch of information that can be 'called up' while you're using a program to remind you how to move the *cursor keys*, how to *save* a document, how to change your *printer configuration* or simply how to get out of the program without losing all your work. The quality and usefulness of help menus varies in almost direct proportion to the price of the piece of *software*. A simple and cheap piece of software isn't likely to have good help menus for two main reasons:

1) If the help menus are too good, and the software house hasn't got a lot of money to 'protect' their programs, it

would be quite easy for people to copy their programs and use them for free without the documentation. With good enough help menus, you almost never have to look at a *manual*.

2) It takes a good deal of time and skill to build proper help menus into a program – to say nothing of the room that the help files will take up on your *storage medium*.

But good help menus are well worth having. On Micropro's Wordstar *word-processor*, for example, the help menus are achieved by what are known as 'overlays' which are 'pulled in' from files on disk as they are needed. This means that information on a given subject is available quickly, without having to sacrifice large amounts of computer *RAM*. Go to *hexadecimal* **K**

◆ **Heuristic** A trial-and-error approach to problem solving – *fifth generation* computing will probably make much of heuristic processes to arrive at *intelligent output*. Heuristics involve inference and most likely choices rather than the manipulation of absolute logic.

● **Hewlett Packard** A US-based multinational computer company which until the early 1980s concentrated on the development of larger and more specialised computer systems than those in the mainstream *personal computer* market. As the personal computer market grew, however, the long term scientific instrument makers turned to the development of an *MS-DOS* PC with a *touch-screen* alternative *input* system (simply called the HP-150) and a portable with *ROM-based applications software* and a

flat-screen (known as the HP-110).

■ ● **HEXADECIMAL** A numbering system often used in computing. As we saw earlier, decimal numbers and computers don't really go well together – as a computer works in *binary* it tends to end up with totals like 1,024 and so on. The alternative is for us to understand it through binary numbers – great strings of ones and noughts, but of course, those aren't suited to human needs. Hexadecimal is the compromise. Hexadecimal (often known as hex) is a base 16 numbering system. You count 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, 10, 11. . . . This is very useful because it enables you (once you have the hang of it) to nominate any *byte* with just two numbers, since the decimal equivalent of ff is 255 – the highest value represented

by a byte (11111111).

Many technical diagrams, for instance *memory maps*, find it similarly helpful to use hexadecimal numbering – ffff, for instance, expresses exactly 64K. Go to *information technology* **K**

- ▲ **Hierarchical files** Computer *files* arranged in a predefined order or pattern.
- **High level interpretive language** Most common type is *BASIC* where each instruction in the *program* takes the form of understandable but terse English e.g. IF X = 90 THEN GO TO 100. The program is *stored* in the computer in this form so that it may be easily tested and modified by the programmer. The commands in the program, however, must be interpreted by the computer into *machine code* instructions as the program is *executed*.

- **High resolution** Well-defined and clearly readable, high resolution often refers to the quality of computer *monitors*, as well as the *density* of the dots in the *letter quality mode* high-resolution *dot-matrix printer*.
- **Histogram** Rather like a *bar graph* in appearance, except each bar plots the frequency of occurrences of certain phenomena in a given time frame. A collection of bars into a histogram give a graphic account of the occurrence over several time frames.
- **Host** A system which prepares *data* or other *programs* for another computer or *system*.
- **Host computer** When a computer *communicates* with another, usually larger, computer it essentially takes on the characteristics of a computer *terminal*

— or goes through the process of *terminal emulation*. The passive computer (the one which doesn't have a *user* attached to the end of it) becomes the host computer. The computer acting as the host computer's terminal sends it *commands* as if it were its terminal and receives back information as if it were the host computer's *screen*.

- **Housekeeping** It is often necessary to delete *redundant files* from a *system* or update or rationalise. On some systems it is necessary to rationalise the way the files have been *stored* on the *disk*. After many *readings* and *writings* the files can become scattered across many *sectors* of the disk, impairing *access time*. Thus housekeeping organises the files back into a coherent structure.

I

- **IBM** (International Business Machines) It is big — one of the largest and most successful companies in the world. It also succeeded in less than three years in making the IBM PC a worldwide industry standard machine. Need we say much more. . . .

- **IBM-compatible** Any computer that can use *programs, peripherals* and *expansion boards* designed for the IBM Personal Computer. There are many machines which claim to be compatible with the IBM PC, but only those that can use both IBM PC *software* AND *hardware* can really be said to be compatible. There are some machines which can claim a measure of equivalence with the PC by using the *MS-DOS operating system* upon which IBM's own operating

I

system – *PC-DOS* – is based, but they could not be said to be fully compatible. These fully compatible machines are also sometimes known as ‘IBMables’.

● **Icon** The picture symbols used in the *Apple Lisa* environment to help choose various *functions*. If, for example, you want to *scroll* up the *screen*, you need only position the on-screen arrow over an up-arrow icon on the screen and press a button the *Lisa's mouse*. The *Lisa* environment is not the only one to use the icon ‘picture-symbol’ system, but Apple helped to pioneer it in the *Lisa* and subsequently the *Macintosh* computer.

● **Ideas processor** A *program* to help you sort ideas and thoughts. This type of program spots relationships between ideas and notes jotted down on a computer and tries to help impose on the living disorder

that is the human brain.

▲ **Identifier** A *character* which identifies a body of *data*.

● **Illegal** An *error message* which occurs when an unrecognisable *command* is entered into a *program*.

● **Impact printer** A *printer* which forms *characters* by striking an element on an ink-bearing ribbon and thereby printing a character on the paper. *Dot matrix* and *daisy wheel* printers are impact printers. A *thermostatic* or *ink jet* printer is not.

▲ **Implementation** The state of attaining a working *system*. It is common to hear of a *program* written for one *machine* being implemented on another, in other words the program has been adapted and made to work.

● **Import** The opposite of *export*. In a computer context, it is most often used

to describe the *loading* of *data files* from one *program* to another in a common *format*.

▲ **Impulse dialling** The way a rotary dial telephone achieves a connection, as opposed to a push-button telephone or *auto-dial modem*.

▲ **Infonet** An international *network system* that can hook subscribers into a range of *databases* world-wide.

▲ **Information bits** When a *byte* of information is *communicated* between two items of computer equipment extra *bits* are normally added as an *error check*. The bits that contain the actual information communicated by the byte are the information bits.

● **Information provider** The term to describe someone who contributes information to a *public access database*.

■ ● INFORMATION TECHNOLOGY

(IT) A term defining the entire field of computers and telecommunications. The concept of information technology is becoming increasingly important as the various technologies converge. One overriding need is to develop common standards of communication interchange so that some of the possibilities of exchanging textual information can be exploited.

As such, Information Technology has been much associated with government initiatives in this area, especially in the European context where there is much discussion towards developing common European Community standards enabling the European industries to compete effectively in the international IT market.

The possibilities of IT are undoubtedly

immense. The ultimate goal of the concept is to generate extensive exchange of textual information over computer links – a development which would ultimately revolutionise the working practises of a very large percentage of the working population. Go to *integrated software* **K**

◆ **Initialise** Most programming structures require the *program* to set up its *system parameters* at the start of each *program execution*. Initialisation usually sets up the ground rules for the execution of the program, defining starting values for a variety of variables and setting aside sections of *memory* for various tasks otherwise not apparent at the beginning of the program.

● **Ink-jet printer** A *non-impact printer* which forms *characters* on paper by directing ultra-precise miniature jets of

ink in the right place at the right time. It has the advantage of being very quiet and often very precise, but it does tend to be a more costly technology than impact *dot matrix* though it has great potential for providing low-cost, flexible printers when it is further developed.

▲ **Input device** Anything which allows you to put information into a computer. Examples include *punch card readers*, *keyboards*, *light pens*, *mice*, and your own finger (in the case of *touch screen* computers).

● **Input/Output (I/O)** A general term to describe the area of operations concerned with sending and receiving information to and from the *processing unit*. For instance the components and *software* responsible for sending text to the *screen* or interpreting the *keys* on the *keyboard*

are part of the I/O.

An input/output channel is a *device* which handles the *data* between the *processor* and another component, while an input/output device is the thing at the other end which, like a keyboard, sends the information into the computer, or like a *monitor*, accepts and manipulates information as computer output.

- **Instruction** A single *command* within a computer *program* which specifies which operation the computer should carry out. This means that there are actually several levels of instruction, for example, the command PRINT is an instruction though this is a *high level* instruction which is then decoded by the computer and executed as a series of *machine code* instructions. A set of instructions make up a computer program.

▲ **Instruction set** The individual *instructions* which can be understood by the *processor* itself with the need for any interpretation by a *language interpreter*. The computer simply decodes the instruction (one instruction for each processor function).

● **Integrated circuit** The basis of cheap computers. A large number of related components on a unified substrata, in other words, when one component on the *circuit* goes, you throw the whole *device* out – the integrated circuit is physically a single device.

● **Integrated modem** A telephone-based computer communications *device* (or *modem*) that is built into the machine rather than added as an additional item after purchase.

■ ● **INTEGRATED SOFTWARE** The

ability of *programs* to share *data files* or at least *pipeline* information between programs. *Applications* programs are generally written as one-offs, but it soon became apparent that when computers are used to run a business the various activities don't fit neatly into the application categories dictated by the applications programs. It was realised that most *users* required a set of common *applications* — generally a *word-processor*, *spreadsheet*, *database system*, and often some sort of *graphics* program to represent figures and reports in *bar charts* and so on.

Some companies therefore offer *suits* of such programs which share characteristics and are identifiably from the same stable. True integration, however, must offer more than this. The best enable the user

to *window* one application with another and easily transfer information from one to the other (from a database package to a word-processing package, for instance, where there is a requirement for *reading* a list of names and addresses as *headers* to a form letter).

For integration to be used to its best advantage, this windowing should be as instantaneous and as flexible as possible. A good integrated software package will let you hit only one button and manipulate information as either spreadsheet data, database record or word-processor document. The only problem with this kind of a properly-working, fully-implemented, integrated software is that it requires lots of *memory* and/or *disk storage*. In order to move quickly between one program and another, those programs

have to be easily and quickly loadable from a disk *drive* or *hard disk* OR they all need to reside in the computer's memory. Either way, you need to spend a good deal on either storage or memory to get a system that will be able to handle properly full integration. Go to *integration* **K**

K ● **INTEGRATION** As it sounds – the ability of *software* or *hardware* to work in conjunction with hitherto incompatible entities. Integration in computers tends to be an ongoing process. What is considered to be integrated one year has been surpassed by something which is 'truly' integrated the next: and so on up the track. This applies especially to *integrated software*. Go to *interface* **K**

● **Integrity** A measure of how accurate, valid and free from *corruption* a piece of

stored data or computer system is considered to be.

● **Intel** Manufacturers and designers of two of the world's most popular computer *processors*: the 6502 and the 8088/8086. Intel's 6502 was used in the most popular *microcomputer* in the late 1970s and early 1980s (the *Apple II*), while their 16-bit *chips* – the 8088 and 8086 – were used in the *IBM PC* and the many *IBM PC* clones that sprung from the success of the *PC*. Intel can also claim to be among the very first people to produce the first *four-bit microprocessors* in the early 1970s.

● **Intelligence** The capacity of a *device* to perform operations or modify the way operations are performed through a set of *instructions* contained in the device's own *memory*. The Commodore 64 *disk drive*,

for example, is said to have 'intelligence' because all the *commands* for handling the *input/output* to the actual disk media are held in the disk drive itself rather than the computer.

● **Interactive** A term which applies to a *program* in which the *user* and the computer perform a function jointly. Every user's entry will cause the computer to respond accordingly. Examples of this are computer games and *Prestel*.

K ● **INTERFACE** A *device* which converts signals generated in one device into a form suitable for another. Most interfaces convert signals into a standard *format*, they are then transmitted to another interface on the receiving equipment and converted back into signals suitable for its use.

A common interface is the *Centronics*

interface, usually used for sending information to a *printer*. A printer is really a single-purpose computer with its own *processor* and small *memory*. The computer sends the text in small *batches*. The interface is configured so that the printer sends out a 'ready' signal when it's ready to receive another batch. Interfaces are also situated between the computer and *disk drive* unit.

Interface standardisation is an important concept as it enables a certain amount of compatibility between specialist manufacturers. A computer manufacturer can concentrate on producing a good computer and leave printer manufacturers to what they do best, by providing a standard printer interface on the machine. The *user* then has a wide choice of printers to choose from with the

knowledge that any printer purchased with the appropriate interface will be able to *run* on his machine. The Centronics is a *parallel* interface. The RS 232 is a *serial* interface. Go to *keyboard* **K**

● **Interpreter** A programming *language* which converts the human *source code* into computer *instructions* as it *runs* through the *program*. *BASIC* is an interpreter.

● **Interpretive** A term used to describe a computer *program* intended to be run through an *interpreter*.

● **Interrupt** A system whereby the computer will stop its current *program*, *save* it and continue on to a new task. After the task has been completed, the computer will return to the original program and resume its *execution*.

● **Italic** A typeface which can be pro-

duced on most computer *printers* and on-screen by some *word-processing programs*.

J

- ▲ **Jack** The male partner in a jack and *socket* arrangement. The jack is usually the piece that hangs off the *cable* while the socket is the thing it plugs into.
- ▲ **Job** A unit of work performed by a computer.
- **Journal** Inherited from accounting jargon. It is a chronological record of entries made in a *file* or *block* of *data*. If the active version of a *record* is *corrupted* before a conventional *back-up* is made, the journal provides a trail of entries which can be used to reconstruct the changes made in the record or file since the last back-up.
- **Joystick** A responsive alternative to the *keyboard* and especially useful in controlling *real-time* action on a screen. Most joysticks act like *auto-repeat keys* on a

keyboard. The stick can be manipulated so that a simple mechanism in the base closes one of four *circuits* (North, South, East, West).

The information is usually interpreted by the computer as a specific *character code* and the computer *executes* whatever instruction the particular code calls for in the *program* – usually to move an object in a particular direction. A joystick will have at least one ‘fire’ button as well. As well as their obvious applications in game playing, joysticks can be used to position the *cursor* or generate on-screen *graphics* in more mundane programs.

J

K

- **Key** A single key on a *keyboard*, but also often used as a verb – it's common to 'key-in' some text or *data* i.e. to enter it into the computer from the keyboard.
- **Key in** To enter information into a computer system through its *keyboard*.

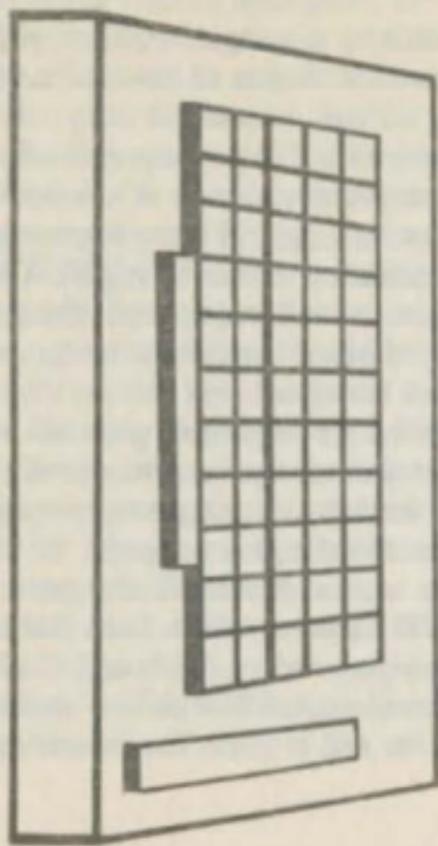
K ● **KEYBOARD** The traditional *input device* for the computer, usually replicates the typewriter keyboard and provides a variety of extra keys for special functions. For instance, there is usually a *control key* which acts like a Capitals shift key on a typewriter, when it's depressed and another key is pressed, the *character* will be interpreted by the system as a *control character* and will cause some *function* to take place within the *program*. For instance a CONTROL S(ave) will often *save* the current *file* to a *disk*.

The computer actually reads the keyboard by running a small *concurrent* program which tests all the keys many times a second to see if they are being depressed. The keys on the keyboard are each positioned over a junction on two lines on a grid. When a key is depressed it creates a *circuit* on a grid. The computer can tell which key is being pressed by which of the circuits in the grid have been activated.

Although keyboard grids all work in pretty much the same way, a wide variety of keyboard configurations have been researched and developed.

A unique keyboard alternative is the PCB Maltron, which looks like an ordinary typewriter keyboard that's been hacked up, shuffled around and then left in the sun to melt. But in truth, there is

K



a certain logic the way the keyboard is arranged. A *numeric keypad* sits in the centre of the contoured keyboard and the keys are colour-coded. Aside from the numeric pad in the centre there are four other clusters: a left-hand segment of 12 character keys, a right-hand segment of an additional 13 and two small clusters in the middle that handle *protocols*, *line feeds* and some punctuation.

Another keyboard alternative is the microwriter configuration which has only six keys arranged on the front of a handheld *micro* such that they naturally fit the hand. Numbers and characters are selected by multiple depressions according to character shapes. Each finger is dedicated to one key, while the thumb, being more nimble, has responsibility for an extra key which acts as a

control to switch the *user* in and out of different 'modes' such as caps lock, numerals and so on. Go to *Lisa Technology* **K**

● **Keypad** A convenience *keyboard* often used when a large amount of *numeric data* has to be entered manually into a computer *system*. Keypads (as opposed to full-sized boards) are also used with *videotext* systems. Nearly all the *commands* in videotext are numeric.

▲ **Keypunch** Now ancient history. Keypunches were used to punch holes representing information into *cards*, prior to that information being *read* into the computer.

● **Keyword** A word in a *programming language* which acts as a *high level command* e.g. in *BASIC* keywords include PRINT, NEXT, STOP, FOR etc.

● **Kilobit** 1024 *bits*, or 1 Kbit.

● **Kilobyte** 1024 *bytes*, or 1k, or 1Kbyte.

● **Knowledge engineering** Rather a lofty term for the work involved in designing and constructing an information *system* or complex *database*.

L

- **LAN** Local Area Network.
- **Language** In order for you to communicate with a computer, you have to find a common set of *instructions* or a language that both you and the computer can understand. Just as two people speaking a common language facilitates their exchange of ideas, a computer and *user* must use the same precise language. Among the more common languages are *pascal* and *BASIC*.

Most home computers speak one dialect, or another, of the *BASIC programming* language, although an increasing number of machines are gaining the capability to speak a variety of languages in addition to *BASIC*. These extra languages can either take the form of a *ROM* cartridge which plugs into the *Printed Circuit*

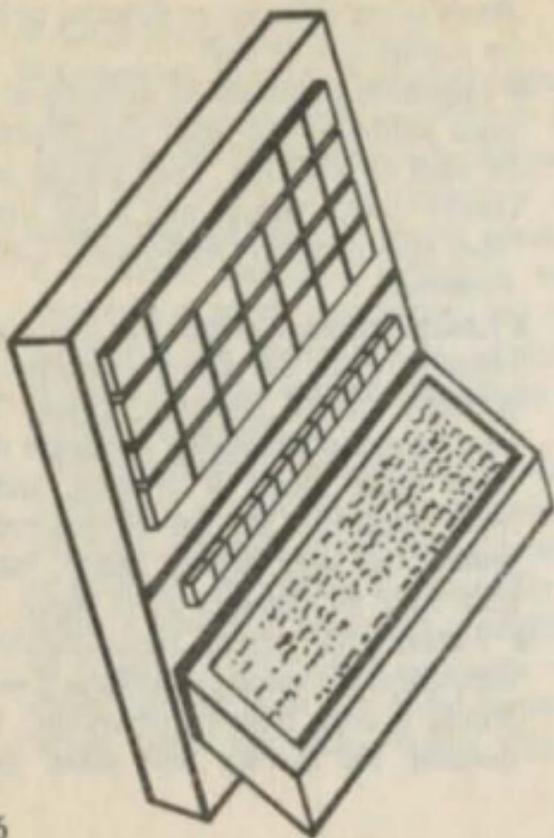
Board of the machine, or they can be held on *cassette tape* or *disk*.

- **Lap-held portable** As opposed to the plain *portable*. Lap-helds can obviously be used on your lap. They usually have battery power supply and a *flat screen*. They also often have *non-volatile CMOS memories*.

▲ **Laser Printer** A *printer* that uses a laser beam to write *dot matrix characters* on specially treated computer paper. The laser printer offers higher *resolution* than many other forms of printing, and is much quicker. Like the *electrosensitive printer*, the laser printer 'burns' rather than 'inks' its messages on to paper.

▲ **Latency** A general term describing some form of (usually) mechanical delay during a *data* transfer procedure. For instance, the average time taken for a

L



block of data on a given *track* to rotate to the position held by the *read/write head*. This latent time must be accounted for when calculating the actual speed of a *storage device* and added to the *transfer speed* itself.

◆ **Leader** At the unused section at the beginning of a magnetic *cassette tape* which ensures that you don't start recording over that part of the tape which is most prone to stretching and distortion. The most important thing about a tape leader is that you remember it's there and don't try to record over it.

◆ **Least significant bit** The one or nought at the end (the right) of a *binary number*. Least significant, because it can only change the value of the number by one. The *most significant bit*, on the other hand (the left-most one or nought), could

change it anywhere from 50 to 100 per cent. Consider a least significant bit change from 100000 to 100001. Now look at a most significant bit change from 100000 to 000000.

◆ **Lens** That which focuses the random elements in a *cathode ray tube (CRT)* so that they turn into a cohesive electron beam when they finally hit the screen. Cathode ray tubes form the basis of displays for most computers as they are at the very heart of television sets and computer *monitors*.

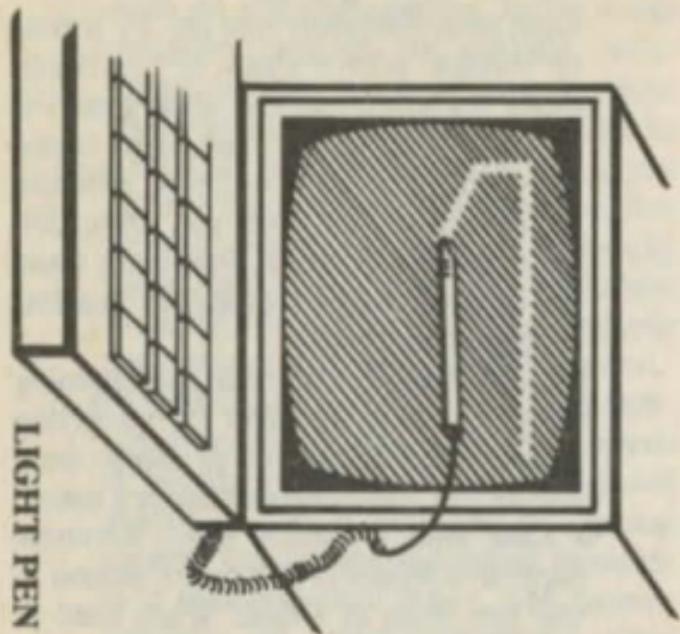
● **Letter-quality printer** A computer printing *device* which yields a print quality dense enough and legible enough to be used in business letters. This phrase is commonly used interchangeably with *daisy wheel printer* or *golfball printer* as those are two most popular forms of

printer to produce this high-quality and high-density type.

▲ **Library** A collection of *programs* stored together in an organised fashion. Although storing programs might seem to defeat the whole purpose of having them, the fact is that most programs suggest that you *back-up* or take a security copy of the program. You then store the original copy in a safe place and only use the security copy for your day-to-day program work. Some *software* houses don't allow you to make security copies in this way for fear that people will duplicate the program en masse and distribute it at a cut-down price as a *pirate* piece of software. It is with these manufacturers that you might write away for a new *disk* (at a nominal fee) if you lose or destroy your original.

- ▲ **LIFO** (Last In First Out) It is a term to describe how some computers *store* or *stack* up their information. In the *Forth* computer language, for instance, information or *data* is put on a *stack* where the computer deals with it on a Last In First Out basis. This means that if you entered the numbers two, three and five (in that order), and decided to multiply them together – the computer would first multiply five and three and then multiply the result of that multiplication by two.
- **Light Emitting Diode (LED)** A technology sometimes used in small *flat screen* or secondary displays. A (usually) red *matrix* of dots is produced by passing current through the individual elements in a *semiconductive device*. The current produces an electroluminescent glow from each element to form the matrix pattern.

- **Light pen** A pen-shaped *device* which, when hooked up to a computer allows you to 'draw' pictures on the screen. Light pens effectively turn the TV screen or *monitor* into a piece of 'electronic paper' on which you can draw whatever you like. The *software* that often comes with light pens lets you choose different thicknesses for the lines you draw, different colours, as well as draw easily some pre-defined shapes (commonly squares, triangles and circles).
- **Line editor** A computer programming term describing a system for modifying *programs* that allows you to change them line by line – and only one line at a time.
- **Line feed** A control which advances paper in a *printer* in an upward motion at one line space at a time. When used in conjunction with *carriage* return or *enter*,



a line feed performs much the same function as a simple 'return' on a typewriter. The inclusion or exclusion of a line feed can also be important in setting up *communications protocols* with other computers or computer information services.

- ▲ **Line loss** The termination of *data* transmission, usually over a telephone line. Simply, they've hung up on you.
- **Line noise** Telephone line noise. The crackling that makes conversation on the telephone difficult for humans can have a devastating effect on a computer which is trying to understand hundreds of *bits* every second. Line noise can often cause a total *crash* or generate lots of spurious *characters* on the screen of a computer which is trying to receive information over the line through a *modem*.

▲ **Line number** Many *programming languages* make use of line numbering where each command or sequence of commands forms a line. All the lines are ordered by number. This enables the *program* to look back to a particular line during its *execution* or jump forward to a line (and then to the subsequent lines) to perform an often-used function required in the program. The *BASIC* language, for instance can accept the command 'GOTO (line) 100'. When the function has been performed, a further command will instruct the system to go back to the line it left and continue executing the program in its numbered order.

● **Line printer** A term commonly used to describe high-speed, low-quality *dot matrix printers* that print one line at a time. Until the advent of *high density* dot

matrix printers, line printers were only used for 'rough copy' and draft documents, but never for finished correspondence or reports. This increase on quality of dot matrix printers has also been accompanied by a corresponding increase in the variety of typefaces available on such printers, so that underlining, *subscripts*, *superscripts*, *boldening* and some foreign *characters* are standard on even the cheapest printers.

▲ **Link** Either a bit of *programming* or a piece of computer *hardware* which joins pieces of the *system* together. A linking piece of programming or *software* would allow different programs to 'talk' to one another so that they could exchange information, while a linking piece of hardware lets two computers or computer *peripherals* exchange information in a

common *code*.

● **Liquid crystal display** A display that employs liquid crystal technology to produce segmented *alpha-numeric characters*. Such displays are typically used in watches and portable lap-held computers and consist of a 'sandwich' which uses polarised glass as the bread and liquid crystal as the filling. When electricity is passed through the thin wires embedded in the glass, the liquid crystal either reflects or absorbs light.

● **Lisa** A significant *32-bit microcomputer* introduced by *Apple* in 1983. The Lisa was and is very powerful and very expensive but it was the first microcomputer to feature truly *integrated software*. This software was called Lisa Technology.

K ● **LISA TECHNOLOGY** A *user interface* developed by *Apple Computer* for

the Lisa *microcomputer*. The basis of the approach was the use of an alternative *input device*. Instead of relying on the *qwerty keyboard* Apple reasoned that most newcomers to computing were used to understanding information and its *storage* and organisation in terms of files, folders, paper documents – accessible to the user when they were arrayed on a desk-top. Neither were most of the potential computer buyers used to using keyboards – a task traditionally associated with secretaries and clerical workers. The Lisa Technology environment involved the use of a hand-held pointing device called a *mouse* which could be moved around the real desk-top (the one that used to have all the paper records on it) and cause the *cursor* on the screen of the Lisa to move in an equivalent direc-

tion and distance. On the screen are *icons* (graphic representations of items or functions). Icons include pictures of named documents and a trash can for throwing away unwanted *files*. The user can use the mouse which has one button to indicate *DO commands*, to specify the documents, functions or *programs* by positioning the cursor over the appropriate icon and pressing the button. The icons or the documents loaded up can be dragged about the screen by use of the mouse and positioned where the cursor wants them, just as he or she would operate on a normal desk.

The idea is to make the Lisa Technology a dominant feature of the **system** so that third party **software** writers will write their programs to conform to Lisa Technology conventions. This means that

when a program is bought for use on the machine its mode of operation will be similar to those already well used and understood. Perhaps what is surprising about the advent of Lisa Technology, is why it took so long and why, as yet Apple is the only company which seems to have thought coherently about the directions computers should be moving in the late 1980s. Go to *modem* 

▲ **LISP** (LISt Processing) LISP is an *interpretive* computer *language* most commonly used for developing and modifying *high level languages*. At one time, LISP was only available for use on larger computers, but it has since been developed for even the smallest of home *micros*.

▲ **List** A sequence of computer statements, such as those produced by a com-

puter *running* the *BASIC* computer language, given when the *LIST command* is invoked. Lists can be produced in one of two ways: as screen *output* or output to a *printer*. Depending on the type of built-in *editor* the computer has, the *program* can be easily modified or changed when it is listed on-screen. List can also refer to an ordered set of *data* items in a program.

- **Listing** A *program print-out*.
- **Load** Commonly used to describe the operation of taking information from a *magnetic tape* or *disk drive* and transferring that information to a computer's *memory*. In the *BASIC* computer language, the word *LOAD* has a very specific meaning in that it *commands* the computer to prepare itself for *input* from one form of *device* or another. When a

program has been loaded, it can be called up or *run* by the computer. In order to use any computer program, that program must be loaded in one way or another.

You can also speak of loading *data* or *files*. In a *word-processor*, for example, you first load the actual program and can then load any file of text you've previously been working on and modify it. The practical opposite of loading is *saving*, where you take the information you have placed in the computer's memory and transfer that information to a disk drive or *cassette recorder*, by means of invoking some form of save.

- **Local** An element or *routine* in a *program* which is confined to one part of a program – contrasts with global. Can also relate to distance – a *Local Area Network* is usually housed within one building or

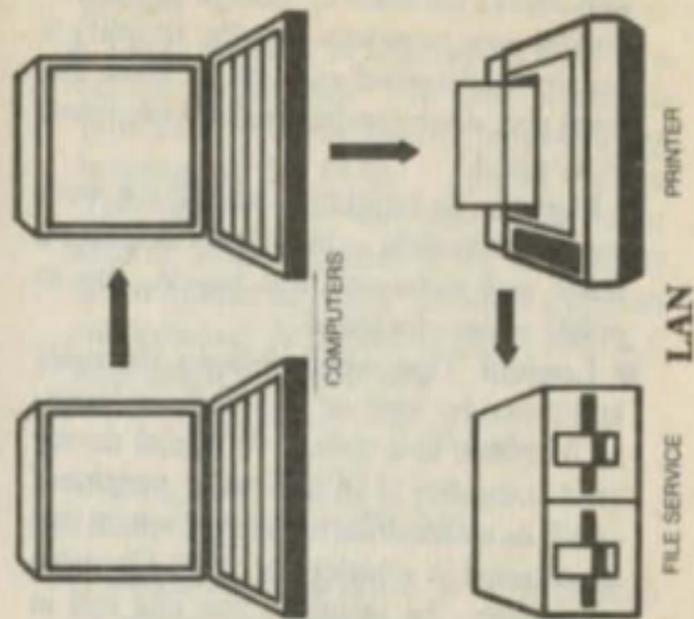
even one room.

● **Local Area Network** or LAN is a system devised to allow a number of computers or computer *terminals* to exchange information and/or share a central *storage device*. The power of a LAN is usually limited to a single office or building, although upwards of 100 terminals or computers can often be present on the *system*.

● **Locate** The action of finding *data stored* in a computer. All data is addressed by a chosen name and its position on a *disk* is referred to its location.

▲ **Lockout** This condition refers to a *program* which has exclusive use of a *file* until the job is completed.

▲ **Logical drive** Where a single large capacity *fixed disk* system is understood by the *operating system* as being divided



into two or more *drives*. With such a configuration, for instance, the *floppy disk drive* (for *back-up storage* and introducing new programs into the system) is usually understood as drive A while the fixed disk drive can be divided into drives B, C and D.

● **Log-on** To register a connection with computer system – usually by offering a name and *password*. You log-on, also to *public access databases*.

● **Logical** That which follows the rules laid down by logic or, in computer terms, as it relates to a *device*. A logical device on a computer is an add-on or *peripheral* (such as a *disk drive* or *printer*) which can be *accessed* in a variety of ways. On some computers, for instance, you can call in the use of certain peripherals by giving them logical 'device numbers' so that

when you give the computer a certain number it will send the *data* to a certain device.

▲ **Loop** A series of repeated actions in a computer *program*. The most common type of loop in the *BASIC* programming language is the FOR...NEXT loop. This allows you to cause a series of actions to be repeated a set number of times before anything else in a program can proceed. A typical FOR...NEXT loop might look like this:

```
10 FOR A = 1 TO 10
20 PRINT A
30 NEXT A
```

It would cause the letter 'A' to be printed on screen ten times before anything else in the program could proceed. A series of FOR...NEXT loops can be constructed

within one another, but they must be properly *nested*.

- **Lower case** The small, non-capital letters. *Upper case* is capitals.
- ▲ **Low level language** One that simply provides symbols for each of the *processor's instructions*. As always it's the eternal battle between the advantages of speed, low *memory* use and flexibility in *programming* as represented by a low level *assembly language*, and the difficulty of learning and using it. A *compiler* is often held to provide the best of both worlds.
- **Low resolution** The crudest level of picture or *character* definition on a computer screen or *printer*. It can also be one of a number of different screen 'modes' which a computer can display. The only great advantages to generating low

resolution characters and pictures with a computer is that they tend to use up less of the computer's *memory* and are much easier to develop.

- **Luggable** Endearing name for a rather heavy portable computer. Portable computers which feature *disk* drives, *CRT*, and all the items normally found on a desk-top computer tend to weight somewhere between 20 and 30 pounds - luggable rather than portable.

M

- **Machine** In general a *device* that has a power source and performs a reasonably complex task. In computer trade jargon, it refers to either a computer or a processor.
- ▲ **Machine Code** *Object code compiled or assembled* for a given computer. Although the term is often used interchangeably with object code, machine code usually means little sections of *program* written in the *native machine language* of a given processor. The advantage of writing programs in this way is that they don't have to spend time being *interpreted*, but can instead be directly understood by the computer and carried out that much faster. If you write a program in the *BASIC* computer language, it must be first translated into the machine code

your computer understands before the computer can do anything with the *BASIC commands* you've typed in.

- **Machine instructions** A *machine code* instruction.
- **Macintosh** Important product-offering from *Apple*. Really a scaled down version of its *Lisa computer*, with smaller *memory*, smaller *disk* storage and smaller price. The Macintosh uses a *mouse* to make command selections in the *applications* packages which run on it and it uses the *Lisa* system of *icons* and on-screen *windows* to display data and helpful information.
- ▲ **Magnetic card** A coated plastic card to *store* information on. The coating is magnetisable, thus allowing information to be magnetically *written* to and from it.
- ▲ **Magnetic disk** A removable *floppy* or

M

fixed Winchester disk with a magnetic coating to *store data recordings*. One or both surfaces of the disk can be magnetically encoded with a series of concentric *tracks* of data. The data on each track is organised into *sectors* to provide the dependent *system* with manageable chunks of data.

▲ **Magnetic ink character recognition**

A system for electronically *reading* information, printed with special magnetic ink. As the ink is magnetised, it lends itself very easily to being read by a *read head* consisting of a transducer that induces flux reversals of fixed spacing in the deposited magnetic ink and a slit scanner which reads the *characters* in much the same way as *magnetic tape* is read.

● **Magnetic tape** An electronically

charged ribbon on which are recorded various electrical pulses representing *bits* of information, reproducible as audio sound or computer *code*. Subject to deterioration.

● **Mailbox** The theoretical location to which you send and from which you can receive *electronic mail*. When you sign on to an electronic mail service, you are assigned a mailbox and a mailbox number. Other *users* on the electronic mail *system* can then send a message to that mailbox number and the message will actually be stored on a section of computer *storage* space set aside for messages in your mailbox. On most electronic mail systems, you will also incur 'storage charges' if you allow messages to sit uncollected and unread in your mailbox. The idea of the electronic mailbox is that

you collect your messages regularly, store the ones that you want to keep on your own *disk* system and delete the messages from your electronic mailbox so they aren't tying up computer space with the electronic mail service.

- **Mailshot** A series of almost-identical computer-generated letters. If you use a *word-processor* to send out one letter, printed many times with a different name and address on each copy, you are said to be doing a mailshot. This mailshot is usually accomplished with the aid of some form of *merging program* that works in conjunction with a word-processing *package*. You type the letter into the word-processor in much the same way as you might type a normal letter into a typewriter, but when it comes to putting the addresses and names in the letter, you

instead type in special *characters* or symbols. Once you've got this master letter done, you create what's known as an '*address file*' which contains the names and addresses of all the people who are to receive the letters. When you then print out the letters for your mailshot, the program first goes to the address file, gets all the addresses it needs for the first letter, inserts them in place of the special symbols and prints the first letter. After that, it goes on and does the same with each subsequent address until it reaches the end of the address file.

- **Mainframe** A large computer which acts as the central processing *device* in a very large computer installation. This is the sort of computer system which is tended by men in white coats and is usually operated by very large companies,

banks and government departments. Mainframe also acts as a generic term to cover the large *data processing* environment. It is common to speak of mainframe computing, or mainframe installations etc.

● **Maintenance** Keeping a computer *up and running*. Maintenance contracts are often part of the deal when computer equipment is purchased, though sometimes it is available at extra cost. It serves basically the same purpose as a warranty on a car. If it breaks down while under a maintenance contract 'they' rather than 'you' have to fix (or pay for) it. Maintenance can also relate to *programs* where the program is *debugged* as *bugs* are discovered or updated as *applications* change.

● **Manual** The *handbook*, item of docu-

mentation, or user's guide with an item of *hardware* or *software*.

● **Massage** The last loving tampering with a *program* after it's been *debugged*, usually to get it going faster or performing a little better. Massaging is a ridiculous analogy.

● **Mass storage** *Memory* used for the accumulation of large numbers of *files*. This is usually stored on *floppy disks*, *cassette tapes*, etc.

● **Matrix** A grid or criss-cross of dots which together make up letters, numbers or symbols that are readable on a computer *screen* or *printer*. See *dot matrix*.

● **Mbit** One megabit or one million *bits*.

▲ **Medium** The physical *device* (*floppy disk*) or *tape* on which information can be stored by the computer.

● **Megabit** One million *bits*. Often

expressed as Mbit.

● **Memory** A general term applied to the *input*, retaining and *output of data*. A computer has a fast electronic memory while such *devices* as *disk drives* have a slower external memory.

▲ **Memory map** A diagrammatic representation of what section of *memory* in a *system* is reserved for what set of *functions*. *Addressable memory* space may be allocated to entities like the *operating system*, *interpretive language*, *BIOS* (Basic Input Output System) as well as the *program display*, and *data*.

▲ **Memory refresh** Where the computer constantly recharges *dynamic RAM memory chips* to keep the contents of the memory in place. This contrasts with the requirements of static *RAM* which only needs the presence of mains power to

keep everything in place.

▲ **Memory workspace** The amount of *user memory* required to *run* and use a *program* effectively after the program itself has been *loaded*.

● **Menu** A selection of options displayed by the computer. Just as in a restaurant where a customer chooses an item from a menu, the *user* also chooses what *function* he or she desires, and the computer carries out the request.

▲ **Metal Oxide Semiconductor (MOS)** A species of *semiconductor device* which enables Large Scale Integration (LSI) i.e. the relative miniaturisation of the various components on the *chip*. So LSI and VLSI, the current forms of semiconductor technology used in *microcomputers*, enable the squeezing of more components on to the chip.

- **Micro Pet** abbreviation for the ubiquitous *microcomputer*.
- **Microcomputer** What the whole thing's all about. A micro as opposed to a *minicomputer* or *mainframe* is a small stand-alone computing *device* with usually no more than two *processors* and not more than about one *megabyte* of *RAM memory*. Its distinguishing factor is that its intelligence is provided by a *micro-processor*.
- **Microdrive** A magnetic tape-based computer *storage device* developed by Sinclair Research for their Sinclair Spectrum and QL microcomputers.
- **Microprocessors** A single *semiconductor device* which carries out the *processing* tasks in a *digital system*. It was its development which made possible the *microcomputer*. A microprocessor incor-

porates both the *arithmetic logic unit* and the *control unit* – components previously requiring separate dedicated devices. It normally handles eight- or 16-bit *words* – there are also semi-32-bit *microprocessors* like the *Motorola 68000*.

- ▲ **Microsecond** A millionth of a second.
- **Minicomputer** Halfway between a *mainframe* and a *microcomputer* – at least in size. The minicomputer was originally developed as an affordable mainframe for medium- to small-sized installations in the 1970s. It is usually the size of a large refrigerator and is often *multi-user*. In terms of power, application and price the boundaries between the lower-cost minis and up-market microcomputers have become blurred.
- **Mode** A particular state of operation. For instance some computers are capable

of generating different sized *characters*, different *display resolutions*, or colour or non-colour on the screen. These options can be called modes. The *user* can choose which type of display or display mode he wants. The pay-off is the reclamation of more *memory* for the *program* or *data*.

A *high resolution* colour screen mode, for instance, eats into the available memory. Non-colour, low resolution screens which present larger (and therefore, fewer) characters at any one time require very little memory overhead. Modes may also be used to describe many other states of operation – *full duplex*, *half-duplex*.

K ● **MODEM** (MOdulator-DEModulator)
A *hardware* device which converts *digital* information from a computer into *modulations* on a carrier wave which can be sent down the telephone line by another

modem and transferred to another piece of *hardware* at the other end. Modems are either *acoustically coupled* or *hardwired*.

Because the telephone system was designed to take *analog* voice signals, modem *transfer rates* tend to be slow in computer terms. There is also a variety of variables concerned with *transmission* – the speed the modulations are sent, the details concerning *full* or *half-duplex*, the way the *error checking parity bits* are set up in each *byte*, and whether transmission is *synchronous* or *asynchronous*.

Communication speeds can also vary between what are deemed sending and receiving machines. *Prestel*, for instance, is set so the subscriber sends information at 75 *baud* and receives information at a much faster 1200 *baud*. This is obviously

very suitable for an information service like Prestel because most of the traffic is going in one direction. The Prestel computer sends a screen-load of text at 1200 baud and the *user* sends back a single byte *command*, or at the most, a handful of bytes, at a time, so it matters little at what speed the subscriber sends his text. Unfortunately, there are occasions when a subscriber on the system may want to send a large amount of text at a time — it is deathly slow.

The most popular standard where the communication involves an equal amount of sending and receiving for both parties is 300/300 baud. A 1200/1200 standard is also gaining ground and would seem to give the best of both worlds.

Modems are capable of sending and receiving information at a variety of baud

rates and the speeds can be associated with a variety of special *modes* (echo, synchronous, odd/even parity, etc.) though it's usually a case of paying extra for more speeds and more feature options. Go to *packet switching* **K**

● **Modulation** Turning *digital* information into *analog*, real world signals. A *modem* is used to modulate and *demodulate* signals for the telephone line. A different sort of modulator is used to convert the display information generated by a computer into signals suitable for reception by a domestic television set.

● **Monitor** A *Visual Display Unit* (VDU) using a *Cathode Ray Tube* (CRT), i.e. it monitors what's happening in the *program* for the person using the *system*.

● **MOS** See *metal oxide semiconductor*.

● **Motherboard** A collection of female

slots, usually inside the cabinet of the computer, into which *cards* (mini *circuit boards*) can be inserted (usually vertically) to provide additional memory or features to the *system*. The slots on the motherboard are served by the *bus*.

The **Apple 2** and **IBM PC** microcomputers are both examples of systems which make use of a motherboard.

▲ **Motorola** Developers of the 68000 16/32-bit processor used on the *Sinclair QL*, the *Apple Lisa* and the *Apple Macintosh computers*. The 68000 processor supports the *Unix operating system* and claims the capability to *multi-task* properly (do several things at once).

▲ **Mouse** A computer *input device* pioneered by the *Xerox Corporation* and popularised on the *desk-top computer* by *Apple Computer* in its *Lisa* and

Macintosh computers. The mouse is a handheld plastic box with a ball embedded in its underside. By rolling the mouse across a desk-top, you control the position of the computer's *cursor*, or pointer, on your screen. As you slide a mouse forward, for instance, the cursor moves forward to the top of the screen. With many *programs*, the computer conveys your *commands* to the computer via on-screen pictures or *icons*. On Apple's *Macintosh*, for instance, the picture of a hand holding a paintbrush represents a drawing program. To *run* that program, you use the mouse to move the pointer over the picture and press the button on top of the mouse twice.

● **MS-DOS** Microsoft's 16-bit *operating system*, which is used in various forms on the *IBM PC* and all its lookalikes.

MS-DOS was the first major alternative operating system to *Digital Research's* eight-bit *CP/M system* and borrows a good deal from CP/M. However, MS-DOS in general is not that similar to CP/M and when it came on the scene offered simplified *commands* structures as well as little extras such as *file date-stamping* in an attempt to dim the lure of the then all-powerful CP/M. Newer versions of CP/M have since adopted much of what it thought was good in MS-DOS and neither operating system is unique any more.

▲ **Multiplexer** A *device* which gets several sets of *communications* simultaneously going down the same line during computer to computer communication. This is multiplexing and can be achieved by having several machine to machine

communications running on different frequencies or slicing up the time available on the line between different machines – each communication process gets a small amount of time to send a section of *data*, then the next, and so on. A device called *demultiplexer* waits at the other end to descramble the signals. The idea is to make better use of *dedicated* data carrying lines over long distances.

● **Multi-tasking** When a computer is able to perform more than one task, or *execute* more than one *program* at a time. A much-maligned term as most computers multi-task to some extent. An example is a program concurrently maintaining the screen display while the computer executes its other instructions.

Real multi-tasking is usually taken to mean getting the computer to do two or

more complex tasks at the one time. A popular and useful example would be sending information to the *printer* while at the same time enabling the computer to be used for another task – writing more letters to print out, for instance.

● **Multi-user system** A computer system with more than one *terminal* hooked into it.

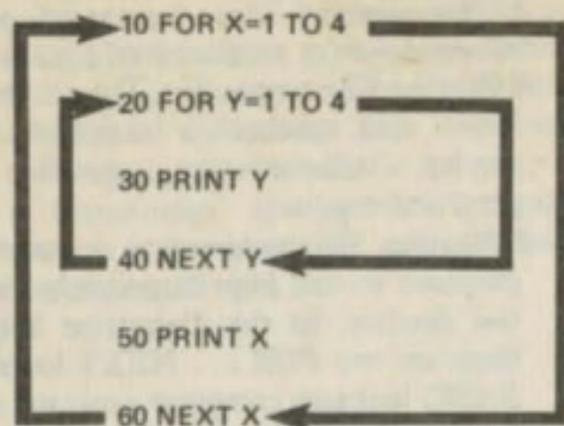
N

▲ **Nanosecond** One thousandth of a *microsecond* (or a billionth of a second).

● **Native Character Set** The numbers, letters and symbols a computer can display without any special re-programming.

▲ **Nesting** The requirement to structure *programs* so that *loops* sit properly inside one another. In the illustration below, there are two FOR . . . NEXT loops in *BASIC* language computer program, and they are properly nested. The outer loop which starts at line ten, can only be completed after the completion of the inner loop, and its corresponding NEXT statement doesn't turn up until line 60.

It's very important that these loops stay nested otherwise the program would not work. The crucial element in the looping procedure is the NEXT statement.



= YYYYYYYYYX
 YYYYYYYYYX

NESTING

Whenever the program encounters a NEXT it jumps back to the *variable* expressed in the statement and the program counts this as a step.

In this case the program counts line 10 as one step for the variable X, it then moves on to line 20 and counts one step on the variable Y, prints Y (which will be the number 1), moves on to line 40 where it jumps back to line 20 and continues the process. Once it has done this four times it goes on and prints X (which is also 1 at this stage). The printed *output* would therefore appear as 12341123421234312344 once this procedure had been completed four times - the number of times we go back to X.

- **Net** See *network*.
- **Network** In computer terms, this usually refers to some *system* of connecting computers together. Also known as a

Local Area Network.

- **Newline** A *key* on a computer which performs the same *function* as Return or Enter.
- ◆ **Nibble** A *four-bit half-byte*.
- ◆ **N-Key rollover** The ability of a given computer *keyboard* to obtain the value of one *key* while another key is still being depressed. This ability is particularly important to computers that are used to play games as games often require the constant depression of one key (for 'firing') while pressing others (for moving on-screen objects back and forth).
- ▲ **Noise** Besides its obvious meaning, noise can also refer to distortion on a line between two *devices*. This is known as line noise. The most common form of line noise occurs when you try to transmit

data over the telephone line. This noise is usually represented as either a 'crackling' on the line or an 'echo' in which signals feedback on themselves (the latter happens more frequently on trans-Atlantic calls than local ones).

- **Non-impact printer** Any *printer* that works without using the mechanism of an inked ribbon being struck against paper by some sort of printing head. The *electrosensitive matrix printer* is one such *device*.
- **Non-trivial** A *program* or *programming routine* deemed worthy of serious attention because it's either hard to *write* or taxing to use.
- **Non-volatile memory** *Memory* which keeps its contents without external power supply. *CMOS* memories, which require very low currents can be maintained with

tiny rechargeable batteries. These are replenished during normal operation. Some *systems* are actually recharged by domestic batteries which, when the computer is being operated, provide power to a flat screen.

- **Non-volatile storage** A *storage device* which manages to keep its contents without mains power. Would obviously include *magnetic disks* and *tapes*, and *bubble memory*.
- **Number crunching** A *programming* task which requires much arithmetical calculation. This sort of work is usually associated with scientific *applications*.
- **Numeric** That which consists of digits or numbers. In computer *programming*, you often speak of numeric *variables*. These are simply letters or a combination of letters and numbers which represent one number or series of numbers.

O

▲ **Odd parity** An *error-checking system* where an extra *bit* is added to a *byte* to make the number of ones in the byte odd – if the one bits already make an odd number the odd parity byte will be nought. If even it will be one. This way, if there is a transmission error and the value of one of the bits is changed, the number represented by the byte will be even and the transmission fault will be detected by the receiving system. Even parity makes up an even number.

▲ **OEM** (Original Equipment Manufacturer) A term generally used by manufacturers for a customer who uses one of their products in their own manufacturing processes.

● **Off-line** Not under the control of the computer. With a *microcomputer* you are

O

most likely to come across the concept when dealing with a *printer*. To put a printer off-line is to put it under manual control. Most printers have an on-line/off-line button. When the printer is off-line it can be controlled from its own panel – you can perform *line* or *form feeds*, for instance. When it's on-line it only obeys instructions from the computer itself.

● **Off-line storage** Relates to some form of *storage* which is not under the control of the *operating system*. The term mostly applies to large systems which may still use *paper tape* or *punched cards*. These media are manually inserted and read into the computer when the information contained on them is required.

In another sense, off-line storage also includes any information stored on a

floppy disk which doesn't happen to be in the *drive* at a particular time – it's *off-line*.

● **Olivetti** A *microcomputer* manufacturer, previously concerned principally with typewriters. Produces a range of equipment for office use including a *lap-held portable* and *IBM-compatible 16-bit* business microcomputers.

● **One-off** Unlikely to be repeated action. May also refer to a computer or *software* produced to a customer's specification.

● **On-line** The term to describe a *system* that employs a number of *input/output terminals* which can be geographically separated from the computer doing the main *processing* by great distances. On-line is also used sometimes to mean 'in communication with', so that if your *microcomputer* is 'on-line' to a *mainframe* computer it means that your micro is in

communication with the mainframe in one way or another.

● **Open** Commonly a computer *command* meaning to open a line of communication. You can open a *channel* of communication between computers, open a *file* on a *disk drive* so that it is ready to receive information and open a line to a *printer* so that you can send information to it.

Open can also refer to computers themselves. Older computers such as the *Apple II* were considered open because they contained no real *interface* that allowed you to do anything with the machine. Whatever you wanted the computer to do, you catered for by buying the appropriate *expansion cards* or *printed circuit boards*. These *systems* often worked on the 'S-100' *bus* system, which allowed you to interchangeably insert cards to control

disks, printers, *monitors* and *communications devices* in any one of a number of bus 'slots'.

▲ **Open computer** A computer whose technical information has entered the public domain (it's available legally to those who wish to know). This enables *third party* manufacturers to supply add-ons which require detailed knowledge of the *host* system's innermost workings to perform properly. An open system will also enable *programmers* to better understand the way the computer goes about doing things so they can make use of its particular features.

● **Open system** Usually refers to a computer which has been designed in such a way that other *software* or *hardware* producers will find it possible to sell items which are compatible with it, hence

increasing its appeal for the *user*, who will be able to find a wide range of products to configure with it. Making a *system* open will usually involve the publication of specifications to assist or even encourage third parties in this endeavour. A closed system has been designed to do the opposite.

● **Operating system** The *program* which goes between the *application program* and the computer. Application programs are written to work with specific operating systems, this enables *files* from one application program to be housed on a *disk* with other application program files. The operating system usually controls the disk's operation. See *Disk Operating System*.

● **Operator** A person who performs tasks, rather than *writes programs*, on a computer.

● **Optical bar reader** A *device* which reads bar codes.

▲ **Optical character reader** A *device* which scans printed characters and compares them with characters stored in its *memory* to provide information for a computer-based *reading system*. When an optical character reader has gathered information from scanning a series of characters, those characters could be put on-screen or put out through a voice-synthesis system.

Developments in this technology are still at early stages, but advances in the computer field are usually so rapid that by the time you read this the price, size and technology behind such a device could well be within the reach of large numbers of people.

● **Optical disk** An optical *mass-storage*

system using the latest laser-disk technology for holding vast amounts of information which can be quickly called up when needed.

● **Optical scanner** A device which scans data optically and transmits the information received in that scan to a computer or computer *device*.

▲ **Originate mode** Describes the state of a computer *modem*. When using a modem to allow your computer to communicate with other computers over the phone line, you have to know whether your modem has to be set to either the 'answer' state or the 'originate' state.

In most cases, the originate state will be the one you want as you'll be phoning up the other computer and it will be waiting to 'answer' you. Originate simply means that the call is originating from you and

you have to be the first one to send out a 'tone' over the phone line so the modem knows there's a computer, rather than a person at the other end. On some of the more sophisticated modems, the answering modem will detect whether the caller is a computer or a person, and if it's a person, then will hang up and let you talk to whoever is phoning. If your computer is going to be the one receiving the phone calls, however, it will have to be on 'answer' – but unless the modem is what's known as an *auto-answer*, it won't be able to answer calls from other computers without human help.

● **Output** The end result of processing information. May also be information that is produced during *processing*.

▲ **Overlay** A *file* – usually on *disk* – that brings information into another *program*.

One of the most popular computer programs to use overlays is Micropro's *Wordstar word-processing* program. This program uses the 'overlays' for areas of text at the top of the screen that contain 'help' information. The overlays are only displayed on the screen when you call them into the program by pressing a 'control' sequence of *keys*. In order to get the overlay with information about printing, for example, you would hit the *control key* and the P key at the same time. This *instruction* would cause the program to go and get the appropriate information from the overlay file and display it on the screen.

P

- **PABX** (Private Automatic Branch Exchange) A telephone switching system that allows internal lines (say a company's) to be connected to the telephone network. PABX and computing are likely to become increasingly inter-related as companies in one field utilise technology in the other. The increasing importance of *modems* as gateways to *public information services* also offers the opportunity to provide low cost *networking* systems inside an organisation where a sophisticated *digital PABX* is already installed.
- **Package** Usually related to a *software applications program* where the program has been *written* to undertake non-specific but common tasks. The package can be sold, installed and operated by the *user* without any more *programming* or

P

adaption by the vendor. It is basically an item of off-the-shelf software. It should include complete, professional *documentation* and, increasingly, a set of learning tutorials.

K ● **PACKET SWITCHING** An economical way of sending long data over long (expensive) distances. The data is assembled into a series of self-contained, standard-length 'packets' – each packet is equipped with data to identify its destination through the data *network*. Instead of a single *communication* process between two *systems* tying up a line from the time the communication starts to the time it finishes, several communications can be seemingly simultaneously in progress.

This optimises the use of the lines in the network and reduces the cost. Packet

switching also gets round the different world standards in telephone exchanges. The world-wide packet-switching network allows someone in Japan, for example, to send a message or piece of *electronic mail* to someone in the UK by simply dialling the number of a local computer service and transferring the information to that service.

The information is then translated by the Japanese packet-switching service into 'packets' of standard-format data and relayed to the UK in phone tones that can be understood by the UK telephone system. When received at a UK computer telephone service, the data is then transferred to the recipient's *electronic mailbox* ready for collection next time he calls his electronic mail service.

Packet-switching services usually work

on annual and monthly charges and often come as part of a larger telephone/computer information exchange network like Britain's *Telecom Gold* service or the *Dialcom* service in the US. Go to *personal computer* **K**

▲ **Packet switching exchange** The computer facilities responsible for assembling and 'switching' packets of *data* in a *packet switching* process.

● **Packet switching system** The collective name (often abbreviated as PSS) for the international *packet switching* service.

● **Page** A unit of *storage* in virtual storage *systems* or in some variations of the *BASIC* computer *language*, a *variable* which gives the address in the computer's *memory* where the *BASIC* has stored your *program*. It is usually set to be the

lowest available address in the computer's *RAM* (Random Access Memory), but on computers like the Acorn *BBC* Micro can be changed with the *BASIC keyword* page.

● **Page break** During the printing process a *printer* can usually be instructed to allow for the change of page in continuous stationery. It advances the paper from before the end of one page into a starting position on the next where it resumes, often printing a page *header* and *footer* as part of the process.

● **Paging** Same as *bank-switching*.

▲ **Paper tape** Early form of *data storage*. A continuous ribbon of paper into which a pattern of holes is punched. The tape is read either by a light system reading the holes, or by electrical brushes which make contact through the holes.

● **Parallel** See *parallel printer interfacing*.

● **Parallel printer interfacing** A system of sending a *data* in a computer. Most commonly refers to a method of communication between a computer and its *printer*. The *hardware* which allows a computer to communicate with a printer working on the parallel data transfer system is known as a parallel or *Centronics interface* (Centronics being one of the companies to pioneer this type of data transfer system).

● **Parameter** A setting or constraint within a *program* or *system*. Parameters can relate to just about anything but can generally be assumed to be the limits or specifications within which a specific aspect of a *device* or program operates. For example, *printing* parameters will

relate to the number of features or degree of flexibility you can initiate during the printing process.

▲ **Parity** A *error-checking system* to ensure safe *data* transmission between computers and their *devices*. It works by selectively adding one *bit* to the bit patterns that make up the data so that the data either has an odd number of bits (and its parity is therefore odd) or an even number of one-bits (even parity).

This extra one-bit that's added to cause the system to operate as either even or odd is known as a parity bit. But, enough of theory. All this means is that if you are going to get devices or computers to 'talk' to one another you have to get their *baud* rates (the speed at which they are transmitting information) as well as their parity synchronised.

If you are communicating over a *modem* to a *bulletin board* that's, say, operating at 300 baud, requires odd parity, seven data bits and one stop bits, you just set your *micro's communications package* to work at 300 baud, odd parity, seven data bits and one stop bit. In most cases, the leg-work of setting up those *protocols* is done by the communications package (unless you're trying to write your own, of course).

▲ **Password** A security measure on many *operating systems*. Always found on *communications bulletin boards*. An authorised *user* retains a password which must be *entered* before *access* to the *system* can be gained.

● **PC-DOS** IBM's adaption of Microsoft's *MS-DOS operating system* for the 8088 and 8086 range of *16-bit Intel pro-*

cessor. PC-DOS is probably now the world's standard *disk operating system* for *personal computers*.

● **Peripherals** An *input/output device* which is connected to and controlled by the computer. A *printer, disk drive, plotter, monitor* or *keyboard* are all examples.

■ **PERSONAL COMPUTER** A concept that came along with the development of the *microprocessor*. For the first time it became feasible to dedicate an integral computer *system* to the use of one person – whether that use be business orientated or of a truly personal, recreational nature.

Since the late 1970s the number of people personally owning, or using, a computer of some description has skyrocketed. Personal computers can be divided into two categories.

Home computers have storage facilities of a limited nature – usually *cassette tape*. They also tend to use a television set to display information. Their uses obviously include games, both of the interactive ‘adventure’ type and action or ‘arcade’ games. In an adventure game the *user* interrogates the system from the *keyboard* and gets back prompts and information on his progress in the game. The game itself traditionally involves some sort of epic journey full of danger and mystical happenings. These games go through stages. If the adventurer makes the right decisions and takes the right actions he works his way through the journey (or through the *program* if you want to look at it that way) and eventually wins the game.

Losing the game usually involves being

killed. If killed, he usually has to start again, though this time, of course, the levels already passed can be worked through quickly.

Action games probably need no introduction – there is usually some sort of intergalactic threat which can only be neutralised by the intervention of a lone spaceship representing planet Earth. These games are usually controlled with *joystick* and fire button and the object is usually to shoot something before it shoots you.

There is, perhaps, one variation to the neat categorisation delineated above, and that is flight simulation. Many systems, including the Commodore 64 and the *IBM PC* and its compatibles have flight simulator programs available for them. Here the player's mission is to control an

aeroplane through the keyboard during take-off, flight and landing. The user is presented with a pilot's-eye view of a flight in progress, complete with a control panel and a view of the world outside the cockpit. The goal of the typical flight simulation program is to get as near to the real thing as possible. Pilots say that many of them are very good.

At the other end of the personal computer scale is the 'serious' machine. The target user is someone who works in an office and is likely to require a variety of applications programs to assist him in a wide variety of tasks. These personal computers usually run one of the standard *operating systems* (CP/M) or PC-DOS, have *disk drives*, full travel *keyboard* and a *memory* of at least 64 *kilobytes*. These personal computers will

nearly always make use of dedicated *monitor* rather than a TV screen to display text and *graphics*.

The key to the serious personal computer, though is the storage – nearly always a pair of *floppy disk* drives or perhaps, sometimes a *fixed disk*. *Storage devices* are really the dividing line and they make the difference between a flexible system which is able to manipulate large amounts of *data* and handle long and sophisticated *programs*, and one which can't.

Disk drives are also the difference between an expensive system and a cheap one, because they are principally mechanical devices, and precision devices at that. They can't benefit from the lowering costs of electronic components to the same extent that the computer itself can.

This has led to some interesting attempts at providing 'intermediate' storage technologies - those which attempt to provide some of the advantages of a disk drive - *random access* to information, speed, and an operating system which will locate and *load* the right piece of program or data when it is called for by the program - with the supreme advantage of low-price associated with less precise tape drive technologies. The two notable attempts are *Sinclair Research's* Microdrive unit and a similar product from Coleco, the US toy manufacturer. Both companies decided to use an endless wafer tape.

The Sinclair version uses very small *cartridges* which contain endless loop of micro-thin videotape. The cartridge can fit easily into a matchbox. When it is run

through the drive it unwinds from the inside of a spool and winds back on the outside of the same spool again. Thus it completes a *circuit* in about ten seconds and arrives back at its beginning point without the need to rewind. Although ten seconds is a long time in computing, the Microdrive system does mimic some of the more important features of a disk drive system. It saves data in *blocks* and can load and *save files* under the control of an operating system.

The Microdrive brings us to the third, emerging, area of computing: computers like Sinclair's QL which fall somewhere between the traditional 'upmarket' personal computers and the home micros, both in price and performance. Go to *printer* 

▲ **Pie chart** A popular diagrammatic rep-

resentation in computer *graphics programs*. The pie chart is a circular representation of a body of *data* which can be subdivided into 'slices' and assigned. Some programs are able to express three-dimensional pie charts, more for effect than illustrative value.

● **Pin** One of several connectors between an *integrated circuit device*, *semiconductor* or *chip* and the *printed circuit board*. The pins are either directly soldered to the board or are housed in *sockets*. This enables an individual component to be easily replaced.

● **Pin feed** A feeding mechanism for *printers*. The perforated paper is meshed with pins on the *platen*. As the platen rotates it draws the paper under the printing mechanism with the necessary precision to ensure, even lines of type,

proper *page breaks* and so on. Pin feed is now used interchangeably with *tractor feed*, though properly, a tractor feed uses an even firmer hand on the paper — drawing it in by means of a caterpillar tractor arrangement.

▲ **Pinout** A pinout diagram is the schematic of the *functions* of various *pins*, both with *integrated circuits* and on mysterious, everchanging *devices* like *serial interface connections*. The pinout is the subject matter. You need to know which pins in a 21 pin *D socket* do what before you start wiring it to anything.

● **Piracy** Copying a *program* and thereby depriving the authors of royalties. Piracy is commonly held to be a severe problem in the computer *software* industry. Statistics show that some programs are copied and distributed such that only one pro-

gram in ten in use has actually been bought and paid for. What is not obvious, however, is how many pirates would actually buy the programs if not given the choice of pirating. One has to suspect that collecting pirated software is more a hobby than a vice. Many people just like to have large numbers of different programs sitting on *disks* and never actually use them at all.

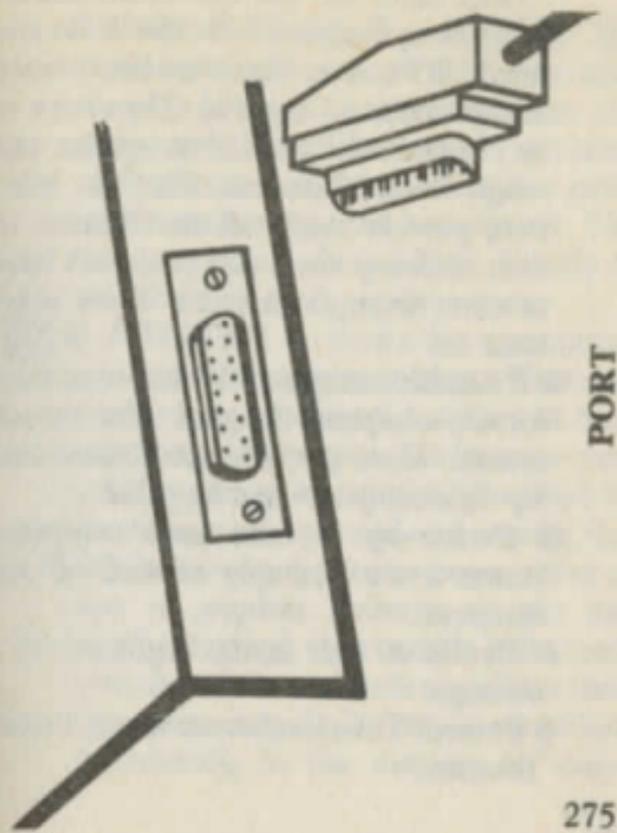
Nevertheless the software industry is prone to coming up with new ways to combat the pirates – complicated protection *routines* which make copying the program more than once impossible, *dongles* (one to each licensed *user*) without whose presence in the machine the program can't *run*. One of the best ways, perhaps, is to make the *documentation* good enough so that the user will buy

the *package* (rather than just copy the program) to get it.

- **Pixel** Stands for picture element. A pixel is a single small cell illuminated on the screen. A construction of pixels will make up a picture or *graphic*.
- ▲ **Platen** The revolving cylinder in a *printer* which guides and provides a surface against which the print mechanism (on a impact printer) can strike.
- ▲ **Platter** A single *disk* in a *fixed disk configuration*. A fixed disk is made up of a stack of platters.
- **Plotter** A computer *output* device which creates two-dimensional images, such as graphs, on paper. Plotters are usually either *drum* or *flatbed* types.
- **Plug compatible manufacturer** One who produces computers or *peripherals* which can be used with another popular

system – plugged in and used. The ‘popular’ manufacturer in question is nearly always *IBM*. The term was coined several years ago and applies mostly to the *mainframe* computer market. Many manufacturers made their names supplying products *IBM* was unable to deliver for its own systems, or supplying products *IBM* would only deliver at a higher price.

- **Port** A place (i.e. a socket) where *data* can arrive at or leave from an item of *hardware*.
- **Portability** Another much-abused term. Portable *microcomputers* have been very popular over the past few years. They break down into two types: the true portables (or *lap-helds*) which can be carried about easily and are usually independent of mains power.



Then there are the full-blown micro-computers, complete with *disk drives* and small *CRT* screen. These are best termed *transportables* or *luggables*. They have to be plugged into the wall to operate and weigh about 30 pounds. They do, however, provide nearly all the features of their desk-top rivals and you don't have to move them about much if you don't want to.

- **Portable computer** Widely-used term for any computer designed to be moved around. More properly sub-divided into *lap-held computers* and *luggables*.
- **Power-up** To turn on a computer. Power-down is simply to shut off the computer.
- ▲ **Prefix** A *code* at the beginning of a message.
- **Prestel** The *viewdata* service of British Telecom.

● **Printed Circuit Board (PCB)** The *device* which mounts and defines the relationship between all the components in a computer. The PCB is made of plastic and features a mass of conductive paths which run between the components. The components can be soldered directly to the board by their *pins* or *sockets*.

- K** ● **PRINTER** A *device* for committing computer information to paper. There are many types of printers, but they all have one thing in common: the express purpose of taking information, generated by a computer or *keyboard*, and causing that information to appear on paper, either as text or *graphics*. Printers are not just dumb 'plug-ins', they contain quite complex electronics and sometimes have their own *processors* built into them. Most importantly, in the majority of cases,

however, is what's known as the printer's *character set*. This defines what letters, numbers and symbols the printer is capable of putting on paper. The most popular sort of printer for home computers, the *dot matrix*, has a standard character set built into the *Read Only Memory chips* inside it. This character set usually contains the full alphabet, numbers one to ten, and an ability to generate *codes* that will cause letters and numbers to be printed in *bold face*, underlined or subscripted. But to print graphics, it is not this character set that is usually used. To get a computer to print screen-graphics, the dot-by-dot patterns of the screen are sent to the printer in what's known as a dot-addressable fashion so that each dot on the paper can be matched with a corresponding dot on the

computer-generated screen from which it is printed. *Daisy wheel printers*, the most popular type of business printer, don't use dots and built-in character sets, but rather wheels with numbers and letters on them. Like a typewriter, these daisy wheels have a series of spokes; on the end of each is a raised impression of the letter or numbers it's going to be printing. The printer gets these letters and numbers on to the paper by rotating the daisy wheel so that the number you want to print is aligned in front of an inked ribbon and, then, projecting that spoke forward on to the ribbon, which causes an inked impression to be left on the paper. Because this process brings forth printed type that looks a lot more like traditional typewritten work, it was naturally the one to first take its place in offices. But dot

matrix typefaces of increasing quality are challenging this supremacy and could soon cause dot matrix printers to be equally as popular in offices as daisy wheels. Go to *qwerty* **K**

▲ **Print-out** The used paper media. Print-out usually refers to the physical results of the printing process rather than the information contained on it. Can also be active e.g. 'doing a print-out'.

● **Processor** The central part of any computer system. The processor is the *device* which performs the operations on *data*. In *microcomputing* the entire processing unit is contained on one *chip*.

● **Program** A self-contained set of *instructions* which will enable a computer to perform a required operation. The computer converts the list of human

instructions into a string of *binary* decisions. Programs can be written in *high* or *low level language*.

▲ **Program generator** A *program* used in order to create other programs.

▲ **Programmer** One who *programs* a computer.

▲ **Program suite** A term used to describe related *programs* necessary for a particular *application*.

● **Program support** The help a computer manufacturer gives in relation to its *software*.

● **Programmable** Where a *device* can store a sequence of *instructions* and execute them.

● **Programming** The development of a self-contained set of *instructions* which enable a computer to perform a specific task. There are a variety of programming

languages. These languages are for the benefit of humans, not computers. The computer has to take the instructions in the language and convert them into instructions it can use, either before the running of the program, as with a compiler or assembler, or during the execution of the program as with an interpretive language like BASIC.

Programming is still a very specialised task, but as computers gain more power for less money, very sophisticated high-level languages will be developed, making programming a creative non-specialist task. In other words, skilled, experienced accountants will be programming an accounting package, rather than a skilled, experienced programmer working off a specification from an accountant.

● **Programming aids** A wide variety of

programs can be obtained to help programmers create more programs. These include debugging programs, compilers, ready-made mathematical subroutine libraries.

▲ **PROM** (Programmable Read-Only Memory) *Read-Only Memory* (ROM) chips which may be easily encoded with instructions after manufacture (do-it-yourself ROMs). EPROMS may be erased as well.

● **Prompt** A message appearing on the screen when a specific action is required from the user, for example: 'insert disk in drive A'.

▲ **Protected field** An area upon the screen which cannot be modified by the operator. Usually, information such as instructions and error messages are displayed here.

▲ **Protocol** A set of rules which describe the method in which information may be transferred between two computer systems.

▲ **Public access database** One which is open, to some extent, either by subscription, or application, to the public.

● **Public Data Network (PDN)** A communications facility which provides a data transmission service to the general public.

● **Public domain** This usually applies to programs upon which there are no claims of copyright. Public domain software has been distributed and copied freely from user to user. Many public domain programs are distributed from bulletin boards where anyone logging on can down-load the program and use it, or make copies and pass it on.

◆ **Pulse** A brief (a few microseconds), upward change in electrical current.

● **Punched card** Early storage medium. Data is represented by a series of patterns of holes in a stiff card. The holes are usually created with a keypunch and each card can contain up to 80 characters.

Q

▲ **Queue** Where items in a list are treated on a First In First Out basic (FIFO).

K ● **Qwerty** A term which describes the standard *keyboard* layout of most typewriters. The first six letters of the top row of the keyboard are qwerty, hence the name. This arrangement of *keys* dates back to around 1885 when a Mr Sholes developed it to slow down the speed of typists. His problem was that typists' fingers were moving so fast that the type-arms of popular typewriters didn't have time to hit the paper and then return before the typist was pressing the next key. Although typewriters, and other machines that use keyboards, have surmounted the problem that Mr Sholes' keyboard design was developed to cure, the qwerty arrangement has exerted a

certain staying power. A number of 'more efficient' alternative key arrangements have been proposed in the last century, but has always foundered under massive pressure from typewriter manufacturers who saw no vast profit – and a great deal of risk – in trying to adopt a new standard.

Among the attempts to 'rationalise' keyboard design since the adoption of qwerty have been the Dvorak and Fitch arrangements. The Dvorak was developed by Dr Emile Dvorak in 1932 using what he called the DSK pattern. He claimed the system was easier to learn and caused a 35 per cent improvement in typing speed. The DSK divides keys into those groups used by the left and right hands. Odd numbers and vowels are kept to the left-hand side of the keyboard – on the top

Q

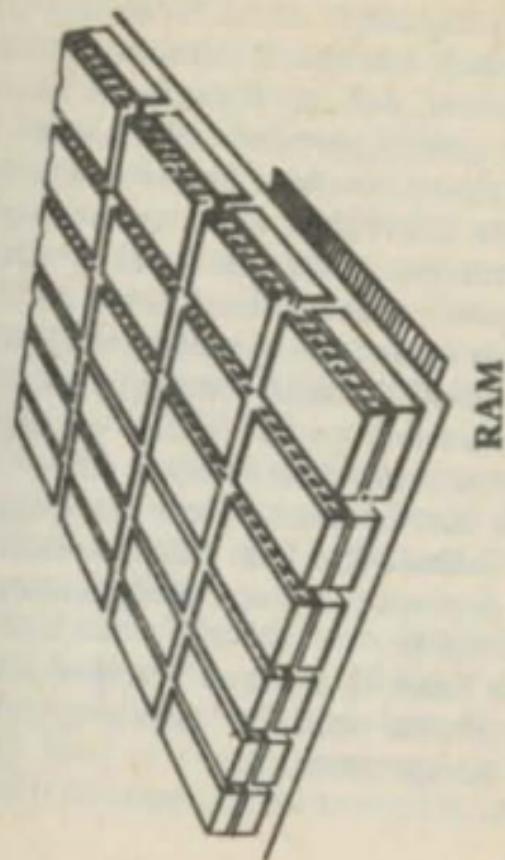
and middle rows respectively. The right-hand side of the keyboard is assigned to handling even numbers and remaining consonants. Punctuation symbols are on the left-hand side with mathematical symbols on the right.

The Fitch arrangement takes this idea of the importance of vowel and consonant arrangement and adjusts it once again. On a Fitch keyboard, the vowels are in a typists 'home' position (the middle row of keys with four fingers on either side of the two middle keys). It starts with all the vowels in this home position and then branches out to the most commonly-used consonants (which huddle just above and below the 'home' position), with the least-used consonants on the extremes. It might seem that far too much energy has been spent by both

designers on vowel and consonant positioning, but research has shown that vowels make up fully 40 per cent of normal letter usage. Go to *report* **K**

R

- **Random access** Where it's just as easy for the *system* to get to one part of whatever's being *accessed* (in most cases some sort of *storage medium*) as it is to get to any other part. Random access does not apply to *tape storage* – here the *medium* has to be wound to the right spot before a *read* operation can be performed. It does apply to *disk* systems. Recorded music represents a good analogy. A long playing record is a random access *device* – you can pick up the turn-table arm and place it immediately on any track. A *cassette* tape, however, must be wound to the right spot.
- **Random access memory (RAM)** The random element (see *random access*) explicitly applies to *read/write* memory *chips*. The feature which differentiates



R

the RAM chip is not its Randomness, but its *volatility*. A RAM *memory* is one which can have information stored in it (write) and can immediately locate and communicate the information when required to do so by the *system*. Unlike the ROM chip, whose contents are permanent, it loses its memory when the power is discontinued. Some RAM chips get around this by utilising small batteries (CMOS), which provide power after the operating power (mains or large batteries) has been switched off.

▲ **Raster scan** Where the image on a *Cathode Ray Tube* (CRT) is built up by displaying an image line by line across the *display*.

● **Read** To detect and interpret *data* from another source – usually a *disk drive storage system*.

▲ **Read/write head** The *magnetic head* which reads information from a magnetic medium – *tape, floppy disk, fixed disk*.

▲ **Read Only Memory (ROM)**. Information which is fixed into a *chip* by a manufacturer and which may not be altered. On many *microcomputers* the ROM *memory* will contain a version of the *BASIC programming language*. On smaller systems (those without *disk drives*) the BASIC well might be extended to provide the equivalent of *operating system commands*. So the ROM is used to make up for the lack of fast *storage facilities*.

▲ **Real-time** When the computer is being used to control a real world event or set of events. The most important thing is that there is a time framework within which the computer has to work to calcu-

late and control options. Real time *applications* really started with the American space program where computers were used to help keep tabs on the enormous amount of *data* that needed to be *processed* while the flight was in progress. Considering the relatively short period involved in a successful rocket launch, especially the first, critical stage, it's easy to see why a real-time computer *system* was developed. More mundane real-time applications might include manufacturing process control.

▲ **Real-time clock** A clock which you may *program* into your computer which gives you the year, month, date and time. Real-time clocks are useful for *date stamping* – where the *operating system* automatically generates a date and time stamp on the *file catalog* when a file is

opened or created.

● **Record** A collection of *data* which is treated by the *system* as an entity. Records are usually held as sub-entities within a *file*. The file simply being the receptacle for identically *formatted* records – for instance, in a simple *cardbox* application program, each record could consist of a contact's name, company, address and telephone number. The information associated with a single contact would constitute a single record.

● **Redundant files** Computer files that are either unnecessarily duplicated or are no longer useful.

▲ **Redundancy check** A system of *error checking*.

▲ **Refresh** Continual pulse sent to *dynamic RAM* to keep the contents of its *memory* intact. Can also refer to the *dis-*

play, when a change in the displayed image occurs, the entire display is often rebuilt, or refreshed. This is opposed to, perhaps, *scrolling* in a text *mode*.

▲ **Register** An important component in the *processor* – usually acts like a very fast transitory *storage* area. Registers are of fixed length – eight, 16 and sometimes 32 *bits*, and can have a variety of roles, for example, to store a running total of a calculation.

◆ **Relocatable program** A *program* which can be moved, if needs be, to a different part of the *memory*. The program is structured in such a way that the *addresses* are easily modified.

● **Remote** A term used to describe a *device* or operation geographically separated from the computer system that controls it.

■ ● **REPORT** Usually relates to a sort of intelligence gathering operation conducted on a large body of *data* – a *database*. Report *generators* are one of the key components in a database *package*. They are a *program* within a program generated by the *user* which enables a series of *searches* and operations to be carried out on the data.

Running a database may seem like a fairly simple operation – you simply *enter* lots of information into the computer and then use the power of the computer to pull the information back out again when it is required. But it's the very vagueness of a program specification like this which makes the database one of the most problematical *applications* in *microcomputing*. The main problem is that, while some applications, say *word-processing*, are

standard, information management is not. A *system* running a word-processing *package* for a grocer is liable to vary little from the one being run by a dentist. The application is a very standard one in that both parties are likely to be composing letters, perhaps in batches as *mail shots* and writing the odd *one-off* letter as the need arises. When, however, you look at the management of information you probably find that the grocer and the dentist have remarkably different needs. One stores patient records, the other monitors stock. While this information is easy enough to store in categories, the way it is used will probably be markedly different. This is where report writing comes in.

Reports are like programs which run inside the database program itself. Their

job is to select individual records or fields, extract *numeric* information or statistics from them and then display the information required in a specific *format*. While the grocer and the dentist may be using the same database package their report formats are likely to be very different. One of the grocer's reports might include calculating which lines in the shop are offering the best return once account is taken of margins, stock levels, shelf space and so forth. The report needed to extract this sort of information from the body of data making up his database will bear no resemblance to the dentist's likely requirements. These could comprise a format to select which patients were due to come in for a regular check-up.

As you can see, the steps a report program has to take to produce these two

completely different results means that the instructions to the program must be based at a very low level (as with a *programming language* where a low level language hands the control of the minute detail of the program over to the *programmer*, a high level language structures the program in large, prefabricated steps). To provide the necessary flexibility to enable the dentist to use the same program as the grocer, a database program has to avoid prefabricated slices. Go to *storage* **K**

● **Reset** To either return to zero or to a pre-determined beginning point. This applies particularly to computer *memory*. To reset it is to wipe it clean – ready to start accepting *data* again. Most computers have a reset button which *users* can use when a *system hangs up* or gets

trapped in some sort of *loop*.

● **Reset button** See *reset*.

● **Resolution** The degree of detail in a *display*. A *high resolution* display on a *CRT*, for instance, is one that uses many more individual *bits* on the screen to build up the pictures or *characters*. *Low resolution* displays are cruder in appearance but use much less *memory* to maintain the display.

● **Retrieval** To locate and *access* a particular piece of information from a *storage system*.

● **Return** The act of pushing the *carriage return key* on the *keyboard*. Also a word in the *BASIC* computer language denoting the end of a *subroutine* or sequence of computer *instructions*.

● **Reverse Polish Notation** This is the method of *writing* programs in the *Forth*

programming languages. It essentially involves the convention of entering first the *data* you wish to manipulate and then following that with the operations you want to perform on that data. If, for example, you wanted to add the number three and four in Forth under Reverse Polish Notation you would first enter three and four and then enter the operation (+) that you wanted to perform on them. The advantage of this *system* is that it allows you *stack* up large amounts of data and move through them quickly. The data is pulled down from this stacking up in what is known as a Last In First Out or *LIFO* basis.

▲ **ROM** See *Read Only Memory*.

● **Run** To *execute* a *program* on a computer.

S

● **Save** To commit a body of information contained in the *memory* of the computer to some form of *magnetic* medium – *disk* or *tape*.

● **Scan** Where a program examines all *fields* in turn, looking for a *string* match.

● **Screen editor** A system for modifying computer *programs* that allows you to change any part of a program shown on-*screen* at any one time.

▲ **Screen format** The way the *display* is organised.

● **Scroll** Where information is presented on the *screen* of a computer in such a way that it flows from the top of the screen to the bottom of the screen, e.g. a long text document. The computer then duplicates the behaviour of paper technology by simulating the unfurling of the

S

document from the bottom of the screen – just as if the text was winding off a roller and rewinding on to another at the top. The alternative is for the computer to abruptly display the next screen of continuous information. Most people are used to dealing with paper, so scrolling usually suits better.

- **Search** A process undertaken by a computer *program* to examine a body of *data* in search of a specific *string* of *characters* or numbers. Searching is a common occurrence in a *database program* for instance.
- **Second processor** Also known as a *co-processor*. This simply means the presence of a second computer processing *chip* inside a computer. Until the early 1980s most computers only had one *processor*, usually of the eight-bit type, but

then 16-bit computers (like IBM's PC) began to gain popularity. Computer manufacturers then wanted to produce machines that could take advantage of the vast range of *software* produced for eight-bit machines, while not losing out on the future advantages of having a computer which ran 16-bit software. Their answer to this was the inclusion of two processors in the computer – one to handle old programs and another to handle new. This is not the only reason for the inclusion of a second processor – they are often added to machines to make them more powerful by perhaps putting the existing processor to work handling *input/output* tasks while the second processor handles a special task like *graphics* generation or sound.

- **Sector** A subsection of a single *track*

on a *magnetic disk storage system*. Each track on the disk must be divided into sectors so that the system knows where it's going. *Floppy disks* have a little hole in the disk near the hub – when this aligns with a corresponding hole in the jacket the confluence is noted by the system which can then work out the rotational position of the disk in the drive and act accordingly.

▲ **Seek** The act of moving the *read/write head* to the correct lateral position – over the specified *track*.

● **Segment** An independently-functioning subset of a computer *program* which can be *loaded* and *executed* without the association of the rest of its program. Programs are often *written* in segments for systems with a limited *address space* of 64k. It doesn't take much

data, plus the *operating system* and display area, to leave the room for the actual program rather short. For instance, *WordStar*, perhaps still the world's most popular *word-processing package* is split up into many segments. When the user wants to do a *block move search* for a word, or print out a *file* the program goes off and loads the relevant program segments for each one of these user-functions.

● **Semiconductor** Describes the actual physical reality of a *chip* or *integrated circuit*. The *silicon* material which makes up the semiconductor *device* is capable of variable conductivity subject to conditions like temperature and impurities in the *substrate* of the device. The semiconductor is *doped* with impurities to produce a complex pattern of varying levels

of conductivity, so producing miniaturised transistors and circuits in the device.

▲ **Sequential** When *records, files* or *fields* are in a definite order and may only be manipulated or *accessed* on that basis. See *sequential access*.

▲ **Sequential access** Where *directory* information for each *record* in a *storage system* is contained at the end of the previous record. So the system is constrained to go through the entire medium to *read* or *write*. This contrasts with *random access* where a directory is kept of the positions of each *block* of data – each block can be *accessed* individually and in any order.

● **Serial** One after the other.

▲ **Serial data** Data which is sent or understood to or by a computer as a series

of *bits* – one following the other. This is opposed to it being *parallel* where the data is sent simultaneously in ranks of eight.

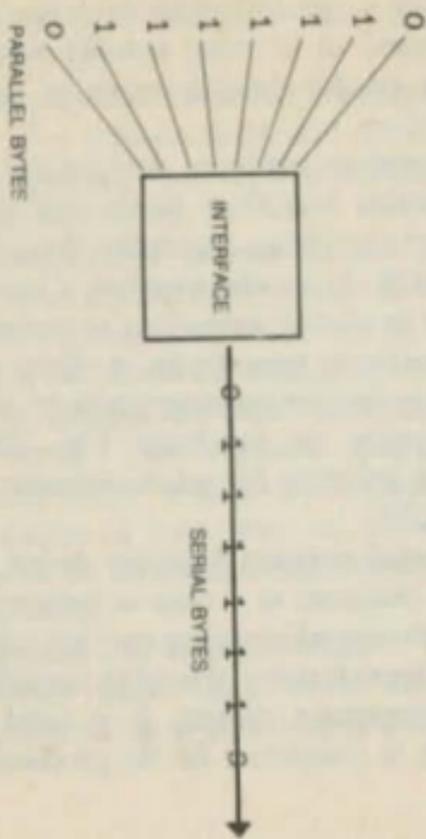
● **Serial access** Same as *sequential access*.

● **Serial interface** Sends and receives *data* in an *Indian file*, rather than in ranks as with a *parallel interface*. Converting data to a *serial format* is a necessary prerequisite to transmitting it down a telephone line for *communication* to another computer or *database*. The standard serial interface for *microcomputers* is the *RS 232*.

▲ **Serial printer** A *printer* which prints one *character* at a time as opposed to a *line printer* which prints one line at a time.

● **Silicon** A material used to manufacture *semiconductor devices*. It is used most often in computers for the production of

SERIAL INTERFACE



silicon *integrated circuits* (also known as silicon *chips*). Such chips are commonly made by building on a silicon slice about 2 mm square and 0.2 mm thick. Each layer that's built in this slice contains an electrical microcircuit. When a charge is applied to the completed chip, silicon atoms either release an electron to become positively charged or take an electron to become negatively charged. These charged regions in each layer combine to act as amplifiers and *storage* devices for current.

● **Silicon Valley** Otherwise known as the Santa Clara district of San Francisco's Bay Area, held to be the centre of the world's microelectronics industry, with literally hundreds of firms located there. Among them are the giant *Apple* and *Hewlett Packard* computer manufacturers.

● **Simplex Data** transferred in one direction only.

● **Sinclair Research** Successful UK home computer manufacturer. Achieved a 'hit' with the ZX81 and then the colour ZX Spectrum hobby *micros* which both found unexpected markets as games and educational *systems*. Sinclair's approach involves relatively low margins and mail-order marketing to make up the difference. In addition the micros were cut to the bone in terms of traditional computer features – lacking all but the most essential *interfaces* and making do with non-standard alternative *keyboard* mechanisms.

Sinclair has also moved into the lower end of the business market with the QL micro whose principle feature is its suite of *bundled applications programs* –

spreadsheet, word-processor, database and business graphics packages.

● **16-bit processor** Is largely supplanting the formerly ubiquitous eight-bit *micros*, at least in the business/personal computer area. A 16-bit processor can manipulate 16-bit *words* and perform certain functions much faster than a processor which deals in eight-bits at a time. The principle 16-bit processor is the *Intel 8088*. It has gained dominance principally because *IBM* selected it for its *IBM-PC*.

● **Smart** Usually refers to a computer *terminal* which has the ability to perform some *local processing* tasks instead of just being able to send and receive information from its *host* – as is the function of a *dumb* terminal. An intelligent terminal is usually just a smart terminal but intelli-

gent sounds better.

- **Socket** Usually taken to mean the female partner in a connection between a *cable* and a computer *port*. In most cases it is safe to assume that the *jack* or plug is the connector on the end of the *cable* and the socket is the thing it plugs into.
- **Soft copy** Information stored on *disk* or other magnetic media as opposed to 'hard copy' or information which is stored by printing it out on paper.
- ▲ **Soft keys** Function keys on a computer *keyboard* which can be *programmed* to return a given *character* to the computer – i.e. a character the *program* is expecting to receive.
- **Software** Computer *programs* produced and sold in the market often with *documentation*, learning material, sample *files*, etc. With such a broad definition

there is obviously a huge range in the types of software. Software can be divided up between programs which help the machine to use other programs, and programs which directly help the *user*. In the first category is the *operating system*. This provides a working environment for the *application program* and should, unless needed, operate almost invisibly. The second sort of software is working programs – *written* to perform specific tasks like *word-processing*, accounting, *graphics* and so on.

- ▲ **Sort** To arrange items in a predefined sequence.
- **Source code** *Programming instructions* written or entered by the *user*, prior to its being *assembled*, *compiled* or *interpreted* by the machine into *object code*.
- **Special character** Any *character* on a

keyboard that is not a letter, number or a space is a special character. This includes # \$ % ^ & * , etc.

● **Speech synthesis** The process of using a computer to generate sounds that approximate human speech. There are a number of ways of accomplishing this task, from 'recording' a series of letters and words on to a computer *disk* or *chip* or recording all the common phonemes (or parts of speech that make up the normal human sounds), to creating very primitive computer sound effects that sound vaguely like someone speaking. Increasingly it is more common that the 'sound information' in average speech is encoded on special speech chips that are actually built into computers and can be *accessed* by simple *commands* in the *BASIC* computer *language* so that the

phonetic *command* 'SAY Heh Loh' would get the computer to utter the word 'hello'.

● **Spelling checker** A *program* that works in conjunction with a *word-processor* to check the spelling (and often count the words) in a given document. Spelling checkers are not usually produced by themselves, but as an add-on module to existing word-processors. Almost all of the popular word-processing *packages* for *micros* have spelling-checker packages that you can buy to work with them. Spelling checkers work by comparing *strings* or collections of letters and numbers. A good spelling checker has a dictionary of 20,000 words or more and checks the words you've written against the words in that dictionary. If a word in your text is not in the dictionary, then the spelling checker

will mark that word as potentially misspelled. It will then let you list the words it thinks are misspelled and ask you if you want to change any of them. You can then decide which of the 'misspelled' words are actually just proper names and places that the computer's dictionary couldn't know about and which are actual true misspellings. If the names and places are going to appear commonly in your work, you can often decide to add words to the dictionary so that it will not single out such words as misspelled, but instead recognise them.

- **Spike** Momentary surge in current on a mains line. A spike can have traumatic consequences for a delicate computer. At best it may lose the contents of *memory*. At worst it may do fatal internal damage to the *hardware* as well. There are a range

of *devices* claimed to offer protection from spikes on the line. The opposite is a *glitch* – a momentary drop-out in power. The memory can go just as surely, though this occurrence is less likely to damage the hardware.

- **Spooling** To use the *storage system* as a *buffer* when *data* is being sent to a *peripheral*. Almost always used in relation to *printers*.
- **Spreadsheet** One of the most popular and obviously useful *applications* to which a *microcomputer* can be put. Enables financial information to be displayed and calculated in a grid of columns and rows. The spreadsheet illustrates financial performance over a time frame. See *Financial model*.
- **Sprocket feed** See *pin feed*.
- ▲ **Stack** Where *data* is added at the end

of a list and is retrieved from the same end. This method is known as *Last In First Out* (LIFO).

● **Stand-alone** A *device* which operates independently.

▲ **Static memory RAM** which does not require continual *memory refresh* from the *processor* – all it requires to keep its memory is an uninterrupted supply of electricity.

● **Static RAM** Stands for Static Random Access Memory.

● **Store** To cause the *system* to place information in a *storage device*.

K ● **STORAGE** *Data* being used in a *system* is manipulated and stored in two forms. The most immediate form is in *RAM*. When it's not about to be used the data in *RAM* is *saved* to a storage medium – either *tape* or *disk*. It is impor-

tant not to get storage confused with *memory*. Memory is the *volatile* data temporarily inside the computer. Storage is the *non-volatile* (usually magnetic) means of holding on to data once there is no room left for it in the computer itself. Storage media are also a means of *importing* new programs and data to the computer system. Once in the memory data and programs can be directly *accessed* by the *processor*. Most of the time, of course, the data is just sitting there, but obviously the speed of the system is greatly increased if the right data is available at the right time. It's the job of the storage system to get the right *bits* into memory and save the other bits back again when they're not immediately needed. Go to *volatile* **K**

● **Storing** The act of placing information

in a *storage device*.

- ◆ **Stream** A flow of *bits* (usually *serial*) over a particular path between one *device*, or one part of a device, and another.
- **Streamer** A high speed *digital* tape recording *device* designed as a *back-up* companion to the *fixed disk*. A streamer usually uses half inch *magnetic tape*. Like the fixed disk it is usually an expensive item in comparison with alternative *storage* combinations, for example, multiple *floppy disks* which provide a built-in back-up solution by enabling the *user* to duplicate the contents of entire disks or *files*.
- ▲ **String** A linear collection of numbers or *characters*. String handling is an important part of *programming*.
- **Structured programming** The proper way to *program*. A set way of

undertaking a programming task, starting with designing the program and splitting it up, if possible, into small parts or *sub-routines*. A well-structured program is easier to *write*, *debug* and modify, especially when these tasks have to be undertaken by someone who didn't do the original work.

- ▲ **Subroutine** A sequence of *instructions* that have to be used several times during the *execution* of the *program*. In *BASIC* the *routine* is called with a GOSUB – the program then goes off, performs the operation and returns to its leaving point. This way the subroutine can be called in many times without complicating the layout of the *listing*.
- ▲ **Subscript** A *variable* or number which identifies an element in a named set of numbers. Subscripts are most likely to be

encountered during simple *BASIC programming* where it is used to delineate the elements in an *array*. It can also refer to half-sized *character* printed on the bottom half of the print matrix. Found as an option in many *printers* and useful in printing mathematical expressions.

◆ **Substrate** As it relates to the *silicon chip*. The material which supports the circuitry – in this case the silicon sliver. The real magic of the chip is the *doped* areas, but the silicon provides the base, or environment to make all this possible.

● **Support** The after-sales service and advice available to the purchaser of a computer or piece of *software* from the vendor. The level of support (or non-support) expected from a vendor should be carefully evaluated by a potential

buyer. As products go, computers are perhaps more in need of back-up than most.

▲ **Synchronous** When two *devices* must be synchronised to undertake a certain task. Most likely to occur in relation to *communications*. Synchronous communication requires both devices to be sharing exactly the same timings. Asynchronous communication can function almost without reference to the state of the corresponding device.

● **System** Defines a complete computer installation including *software*, *peripherals* and *hardware*. That which works interdependently to produce a given result.

● **System analyst** Someone who evaluates the needs of a computing *application* and recommends *hardware*, *software* etc.

● **System software** Usually thought of

as *software* which performs tasks for the *system* itself, as opposed to an *application program* which seeks to produce usable *output*. System software can also be defined as above minus the *operating system*.

T

● **Tape** Most commonly refers to *magnetic tape* used to store information transmitted from computers. The information is encoded on the tape, such that when it is 'played back' the computer can interpret the signals it originally sent to the magnetic tape recorder as *program* information or data. Can also refer to paper tape, on which information is recorded by punching holes in the tape in strategic places, such that the positioning of the holes in the tape determines the nature of the information it holds. When used as a verb as in 'Did you tape the program?', it can mean to store or save a piece of information on tape.

▲ **Tape drive** A *magnetic storage device* which uses magnetic tape or *tape cassettes* or cartridges.

T

▲ **Telecom Gold** Business-orientated *communications* service for *microcomputer users* run by British Telecom. Subscribers can use a *modem* to dial up the service which offers telexing, *mailbox* facilities, *database* facilities and long distance *data* communication.

▲ **Telecommuting** The practice of working at home with a computer and 'sending' your work to the office via the *communications* facilities available on a computer. Telecommuters have the distinct advantage of making their work do the commuting for them – and even then the work only commutes over the telephone line to another computer at the telecommuter's place of work.

▲ **Teleshopping** Ordering goods and services through a public *communications network* e.g. *Prestel* offers teleshopping in

certain areas.

▲ **Telesoftware** Computer *programs* obtained using the telephone-based or broadcast-based *communications* facilities available on a *micro*. Telesoftware services allow you to dial up another computer which has computer programs *stored* on it, 'pull down' a copy of one of those computer programs via a system which automatically bills you for the *software* you take at the same time. Free telesoftware offered over *teletext* TV broadcasting services give you *access* to free programs if you have the appropriate adaptor hooked up to your micro. This receiving equipment picks up the telesoftware transmissions, decodes them and lets you store the decoded versions of the program on your computer.

▲ **Teletext** A business correspondence

standard for *word-processors* and electronic typewriters.

▲ **Teletext** Not to be confused with the above. Teletext is a method of broadcasting *digital* information which can be picked up and displayed on a domestic television set.

▲ **Terminal** A device which a computer operator utilizes to communicate with the computer. Usually comprises a *keyboard* and *monitor*.

● **Terminal emulation** Where a computer runs a *program* which makes it act like a *terminal* to another computer. Terminal emulation allows the emulating computer to send commands to the *host* system and receive back a *display*, usually via a *modem* over the telephone system. The host system treats the connected computer just like one of its own termi-

nals. A good terminal emulation program will allow the user to save information sent by the host system to its *disks*. It may even facilitate the automatic dialling of the number and set up the necessary settings (*baud* rate, *echo*, etc) for the computer being connected.

▲ **Terminate** To end, finish or stop a *program*.

▲ **Text editor** A *program* which manipulates text – a *word-processing* program could be described as a text editor.

● **Text files** Computer-based files composed entirely of non-programming text. Text files do not usually contain special control *characters* or codes, only letters and numbers. A text file can also refer to a file produced by a *word-processor*.

● **Text formatting mode** The state in a computer *word-processing program* which

causes text to fit into a predefined line-spacing, margin and design structure.

▲ **Thermostatic** A type of thermal printer where the characters are scorched on to specially coated paper.

● **Third Party** Usually referring to a supplier of computer equipment or *software* that works with another company's computer system.

● **32-bit microprocessor** An advanced *processor chip* which can manipulate data in *32-bit words*. The term is a fairly imprecise one – some 32-bit processors are capable of processing with 32-bit words internally but still send data to other chips eight bits at a time. The number of bits a processor deals with does not necessarily indicate its speed and power, but quoting bits has become the computer equivalent of the car manufac-

turer's old standby, horsepower. What horsepower doesn't tell you, of course, is how heavy the car is, how aerodynamic its body shape and how much petrol it uses - all the things you need to know to tell you how fast or how efficiently you can get from A to B. Computers are the same - bits are not enough. True power or usefulness can only be determined by calculating a computer's *clock speed*, the efficiency of its *operating system* and the speed of its *disk drives* or *language*.

▲ **Time manager** A *program* designed to take the place of the standard business diary. This type of program can take many forms, but usually includes some form of on-screen representation of a calendar, an on-screen representation of a page in an appointments diary and some form of *search* facility. It is the search

facility that allows the electronic time manager, or diary, to score over its paper counterpart. In a normal day-to-day diary, there is only one way to search for an appointment: on the basis of when that appointment is supposed to happen. But let's say that you had an important appointment with a solicitor, but couldn't remember which day in a given month that appointment was for. In order to find that information in a diary you would have to flip through every day in the month, hoping that you would come across the solicitor's name somewhere in the diary. But with a time manager, you would just type the word 'solicitor' or the name of the person and the computer would automatically find the engagement and bring it up on-screen — that is providing you have already plugged in the

machine, *booted* up the program and happen to have it *running* at the time when you want to do the search.

▲ **Time sharing** Where several *users* or companies use the same large multi-user computer *system*. Time sharing is probably becoming less important as the price of computing drops in most *application* areas.

● **Toggle** A method of selecting from a range of options where the options (two or more) are displayed or highlighted one at a time (usually on the screen) in response to *input* from the *keyboard*.

A typical toggling operating might involve choosing between options before a printing operation in an *application program*. After the PRINT *command* has been *executed* the *user* is presented with a screen of conditions in textual form e.g.

'show header . . on'. By positioning the *cursor* at the head of this *prompt* and pushing the space bar the prompt will change to 'show header . . . off'. A condition which will cause the *header* to be ignored during the printing process. Depressing the space bar again will bring 'on' back, but a RETURN or ENTER will establish the change and the cursor will jump to the next prompt having toggled the header prompt to off. A toggling operation may involve more than two options.

● **Token** A much used concept. It's usually possible to shorten *word commands* to the point just before the shortened version can become the same as another word command. This optimum is called a token. For instance in a system which uses the commands VALIDATE and VALUE the respective tokens would

have to be VALI and VALU. If we assumed there are no commands beginning with P apart from PRINT, its token could be P. It is all the computer needs to go on.

● **Touch-screen** A computer *input system* that lets you give the computer *commands* by simply touching the screen. As popularised on *Hewlett Packard's HP-150 microcomputer*, the touch-screen works by emitting a 'grid' of infra red beams in front of the screen so that when the grid is interrupted at any point, a unique value will be generated and sent to the computer. This means that if you point to the command box at the lower left-hand corner of the screen (the 'origin' of the grid), the grid is broken at co-ordinate 0, 0 and a value - similar to a *keyboard value* - is sent to the computer.

- **Touch tablet** A computer *input device* that consists of a pen and a smooth plastic pad. Images drawn on the pad are automatically converted to computer *code* and displayed on-screen. The plastic pad becomes on-screen 'paper' and the pen – which can offer lines of different character and colour – becomes a paintbrush or shading pencil. Touch tablets also usually offer a plastic sheet over the main tablet so that you can slide pictures under it and trace over them to get reproductions on-screen.
- **Track** A path along which *data* is recorded on a *magnetic* medium – tape or *disk*. *Disk tracks* are usually divided into *sectors*.
- **Tractor feed** A mechanism which guides continuous stationery through a high-speed *printer*. A tractor feed is the

heavy-duty version of *pin feed*. It uses the same principle as a caterpillar tractor to gain a firm purchase on the paper through the *sprocket* holes in the continuous stationery.

- **Transfer speed** The speed at which *data* can be transferred from one part of a computer or computer *system* to another. It is common to talk of transfer speeds between a *disk drive* and the computer. The speed is usually expressed in *bits* or *bytes* per second.
- ◆ **Transistor** *Semiconductor device* used to control the current in a *circuit*. Transistors are the basis of modern computing.
- **Transportable** A portable computer that's not quite as portable as the manufacturer would have its potential customers believe. Although full-featured business machines with a small *Cath-*

ode-Ray Tube screen, dual *disk drives*, full *keyboard*, standard *processor*, *memory size* and *interfaces* are still called portables, the advent of real *portability* in the form of the *lap-held* computer has called for some form of differentiation. Most of the large portables weigh between 20 and 30 pounds. Not a weight you're likely to carry round for more than 100 yards at a time. *Luggable* is a far more apt description for this class of computer.

▲ **Transputor** Still under development. An advance on traditional *processor architecture*. Several transputors will be able to work in *parallel* at great speed, each one completing parts of the processing *job* and considerably speeding the time taken on a *program* over the *serial* (one task at a time) *generation* of computers currently in use. It is hoped that

the transputor will converge with the development of advanced *programming techniques* to usher forth the age of *fifth generation artificial intelligence systems*.

● **Trivial** A *program* or programming routine deemed unworthy of serious attention because it's either easy to write or fun to use (like a game).

● **Typeface** The type of *character set*. The appearance of *alpha-numeric characters* either on the *screen* or as they appear printed on paper. The difference between typeface, *font* and *typestyle* are less strict than they would be in the printing world. In computer usage they overlap considerably. However, the term will usually apply to a *printer* where it delineates the style of characters produced on the page – they may be italic, bold, etc., or be of certain artistic designs – Helvetica, New Times

Roman etc.

- **Typestyle** Functionally the same as *typeface*.

U

▲ **Unbundle** If a computer manufacturer begins charging separately for *software* that was previously available inclusively with a computer purchase, he is unbundling. See *bundled*.

▲ **Uncommitted Logic Array (ULA)** A logic *array* (a sequence of *and/or* steps) implemented on a *semiconductor device* which is uncommitted at the time of manufacture. It is then configured to a computer manufacturer's requirements at the final stage.

▲ **Unformatted capacity** Often quoted by manufacturers of *disks* and *drives*. The total amount of *data* that can go on the disk - this includes the areas in normal circumstances taken up by the *headers*, *sector* marks and so on, laid down during the *disk formatting* process. The

U

formatted capacity describes the space left after the formatting process. As this figure changes slightly, depending on what *system* is doing the formatting, disk manufacturers prefer to use a concrete unformatted figure.

- ▲ **Uninterruptable power supply** An add-on item which keeps a fully charged battery on standby in case of mains failure. Should this occur, the *system* will switch over to battery without interruption, preventing system *crash*, *memory* loss, disaster etc.
- ▲ **Unix** A computer *operating system* based around the 32-bit *Motorola* 68000 *processor*. Unix was first developed as a 'scientific' O/S in the US, but has grown in worldwide popularity considerably since then. It still, however, has yet to develop an appeal for the wider spectrum

of business and home users. Part of the reason for this has been the large amounts of *storage* required to use Unix and the lack of 68000-based computers to *run* the system on. Although a number of computers now use the 68000 *chip*, notably *Sinclair's* QL and *Apple's* Macintosh, Unix is not the standard operating system for most of those computers.

- ▲ **Up and running** When a *hardware* or *software* product actually comes off the drawing board, is *debugged* or adapted for a particular machine or *operating system* and becomes a useful, working item.
- ▲ **Upgrade** To add new products to an existing computer *system*.
- **Upload** To send computer information or *programs* to a source – usually another computer receiving over a telephone line – and have that receiving computer store

the information in its *memory* or on its *screen*. Its effective opposite is *download*.

▲ **Upper case CAPITAL LETTERS.** Lower case are normal letters or *characters*.

● **User** Anyone who uses a computer.

● **User group** *Users* of a specific computer *system* who establish a club to enable them to exchange useful information, hints, tips and *software*. User groups are often promoted by and closely associated with the vendor of the computer system itself. In some cases the user group can act as a pressure group (a sort of union) to influence the policies of the vendor. This is particularly common in *mainframe* environment where the relationship between the computer company and the user group is a very important

one since the user is often tied by the existing investment to further purchases and *upgrades*. It is very difficult and expensive to switch systems, so it is in the interests of both parties to maintain a close relationship.

● **User interface** The set of *screen displays* through which the *user* instructs the *program* and receives information on its *status* or the way the program is displayed on the *screen* to the user. You will often hear the term associated with *friendly*. A friendly user interface is one which provides an easy-to-understand set of prompts and expects a consistent and understandable set of *commands*.

▲ **Utility Programs** Supplied with an *operating system* (usually) for extra maintenance tasks on the system, for example, copying *disks*, *reading corrupted disks*.

V

- ▲ **Validation** Checking *input* against a pre-defined set of criteria to minimise *errors*.
- ▲ **Vanilla** The *default* or 'automatic' settings in a *program* or computer. A vanilla setting on a *word-processing* program, for example, refers to the assumed type of *printer* to which the program is expecting to send its information. If your printer didn't come within the list of vanilla settings predetermined by the program's manufacturers, then you would have to customise your word-processor. But the whole purpose of vanilla settings is that you DON'T have to customise your program as the vanilla setting is likely to take in the vast majority of popular printers.
- ▲ **Variable** A quantity in a *program* that is named, e.g. $x = 2$. x is therefore a

variable with a value of two. As the program progresses the value of x can be changed e.g. $x = x + 1$.

- **VDU** Visual Display Unit.
- ▲ **Verification** Entering *data* from the *keyboard* twice as a check on its *integrity*.
- ▲ **Videotex** A generic term covering two-way interactive *communication* between a computer (or dedicated adaptor) acting as a *terminal* to a remote computer running a large *database*. The essential feature of videotex is that you get to choose which information you want the *system* to send you. It sends you a list of choices and you send it a series of *commands*. It's very *menu-driven* so you don't need to know anything about databases or computers to operate it. Videotex systems have been designed to offer a wide range of general-purpose informa-

V

tion. *Prestel* is the UK example – it features up-to-date information on the weather, stock exchange, airline departures, as well as general purpose news reports and, specialist information.

▲ **Viewdata** The same as videotex.

◆ **Virtual** Something which replicates another condition so that to the *user* or the *program* they are indistinguishable.

● **Virtual Storage System** A system which uses a *virtual* type of *storage* mechanism. Because the total area of the *storage device* is available in such a system, the addressing of such storage does not have to be limited by the normal constraints of the way it is divided up.

▲ **Visicalc** (VISible CALculator) The forerunner to the present crop of *spreadsheet applications programs*. Visicalc was developed for use on the

Apple II and the program can be credited with propelling the *microcomputer* to age in the eyes of the sceptical computer industry. It was a really useful program for a wide variety of applications. See *spreadsheet*.

▲ **Visual Display Unit (VDU)** A *terminal* – provides a *keyboard* to input *data* and a screen to display it. A VDU would probably be hooked up to a large *mainframe* or *mini* computer.

▲ **VLSI** (Very Large Scale Integration) A *chip* manufacturing process which enables much cramming of *transistors* and *circuits* – typically up to 10,000 transistors per chip.

▲ **Voice input** An alternative *input device* that lets you give *commands* or send information to a computer by talking to it. The challenge facing designers of such

devices is in making them accurate and useful at the same time as being cost-efficient. Because people don't pronounce words in the same way, at the same speed and at the same volume every time they speak, it's difficult to design computer equipment that can easily identify human sounds. You can buy cheap, less accurate speech input devices now, but really accurate and affordable speech recognition is still a way off yet.

K ● **VOLATILE** Prone to take extreme exception to the slightest upset. Refers specifically to *memory* and *storage*.

One way of viewing a computer's storage and *output* is in terms of volatility versus flexibility. When *data* and *programs* are in their most flexible and useful form they are also in their most volatile. The most volatile stage is when informa-

tion has just been entered into the *system* from the *keyboard*. At this stage the slightest upset to the system can cause all the information to be lost entirely. At the same time, however, the information is at its most flexible. It can be manipulated by the program with ease, changed, calculated or deleted in microseconds. Once the information is read to a *disk* or tape system, however, it has become far less flexible in that form. Even at this stage, however, it's a simple matter (seconds) to load the *file* back into the system, do a few changes and *save* it back as an amended file. Volatility, however, is still present. Place the medium (disk or tape) near a magnetic field, dirty its surface or damage it in any way and, again, the information can be lost forever.

The most non-volatile form of storage

is probably to print the information out on paper and put it in a filing cabinet. At this stage, however, it has become almost totally inflexible, the only way to make changes is to type it all back in again. But, barring an act of God, it is almost completely non-volatile.

◆ **Volatile storage** A term for a *storage* facility that depends on a constant supply of electrical power. When the power is cut, the *data* is lost.

◆ **Von Neuman machine** The original computer design, put together by von Neuman in 1946.

W

▲ **Wafer** The starting point in the manufacture of *semiconductors*. They start life as part of a large *silicon* wafer which is then given the etching and photolithography treatment prior to having the *chips* sliced from the wafer.

● **Winchester disk** See *fixed disk*.

▲ **Winchester Technology** See *fixed disk*.

▲ **Window** A *programming* technique where a section of the screen is designated to act as a viewing area for another body of text or information. For example, when using a *word-processing package* a windowing facility might allow the *user* to create a window on the screen – say a small square in the top left-hand corner – which could display a catalog of all the *files* contained on the current *data disk*. The information contained there could be

W

scrolled through independently of the main body of text still visible on the remaining portion of the screen. Once the user has located and noted the information needed, control is passed back to the text being edited and the window disappears.

Windowing is obviously of great advantage in *integrated applications software*. Here a user could extract information from a *spreadsheet* (just one example) and enter it directly into a management report. Windowing is also used to display *help* information or *menu prompts*.

▲ **Word** The number of *bits* manipulated by a computer simultaneously in *parallel*. Most systems *processors* use eight, 16 or 32 bit words.

▲ **Word-Processing** Using a *terminal* or computer to compose, *store*, manipulate and print out documents and text.

● **Word-processor** A computer *dedicated* to *word-processing*. Some word processors, however, have acquired the ability to run other general types of software. 'Word-processor' may also sometimes be applied to the word-processing software designed to run on a conventional computer.

▲ **WordStar** A sophisticated *word-processing program* and one of the most popular *applications* programs. Originally developed to run under the *CP/M operating system* and then adapted to work with *MS-DOS* and *PC-DOS*. *WordStar* features extensive *on-screen prompting* and makes use of *CONTROL commands* (where the CONTROL key is depressed in conjunction with a mnemonic key e.g. CONTROL P to execute the printing of a file). Although not a particularly easy

program to master (books have been written on how best to use the *system*) it features just about every option required by anyone who wants to produce documents on a computer – *block move*, find and replace, line count and so on. It also has companion programs – Mailmerge (a program which enables the user to personalise form letters for *mailshots*) and SpellStar – a *spelling checking* program which will not only count all the words in a document but also tell you how many different words you've used (fascinating but useless).

● **Workstation** The physical environment used by a person operating a computer. The items in a work station are generally understood to include a *keyboard*, *screen* (perhaps a *disk storage device*) as well as the usual chair, desk

and surface work area.

▲ **Wrap-around** In *word-processing*, when the last word in line has run out of space, instead of truncating the word and starting halfway through it on the next line, a wrap-around feature takes the entire word and starts it at the first *character* position on the next line.

▲ **Write** The function of encoding a magnetic medium with *binary* information that can be stored and *read* back. Information is converted into magnetic impulses which are transferred by a magnetic *read/write head* to either a *magnetic tape* or *disk* in *serial* form.

▲ **Write head** The magnetic head in a *disk* or *tape storage* unit which applies the magnetic impulses to the surface of the media.

X

▲ **Xerox** Has felt the need to extend its product range into *microcomputers* and *word-processors* and so exploit its already considerable presence in the office equipment market. Products included *mouse input devices* – on the Xerox Star computer.

▲ **X/Y plotter** Draws lines over a flat surface with reference to a set of x/y co-ordinates. See *flatbed plotter* and *drum plotter*.

Z

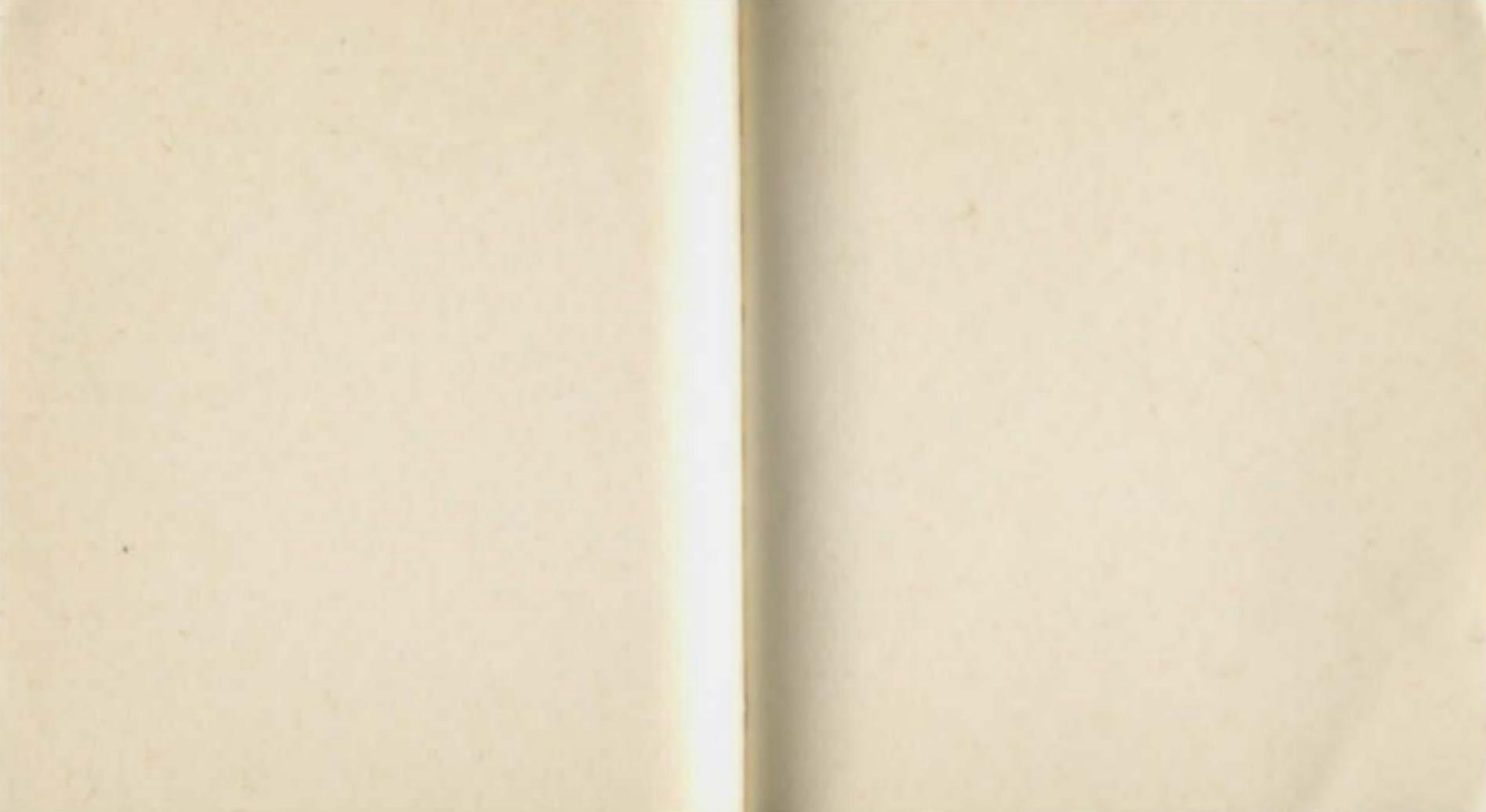
▲ **Zilog** US-based *semiconductor* manufacturer which scored a big hit with its range of *Z80 processors*. The Z80 achieved dominance in the *eight-bit desk-top microcomputer* market where the standard *CP/M disk operating system* was based on its instruction set.

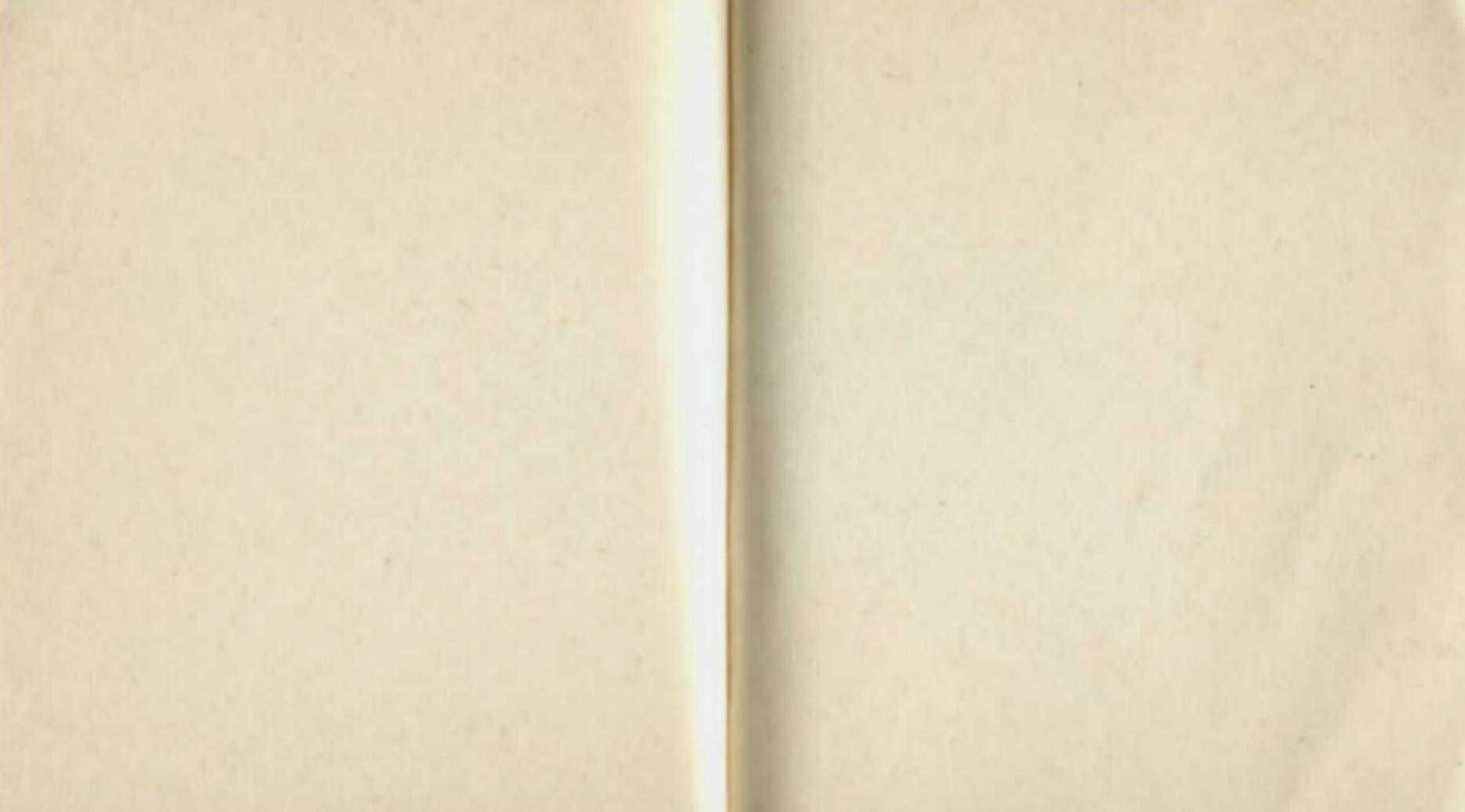
X Z

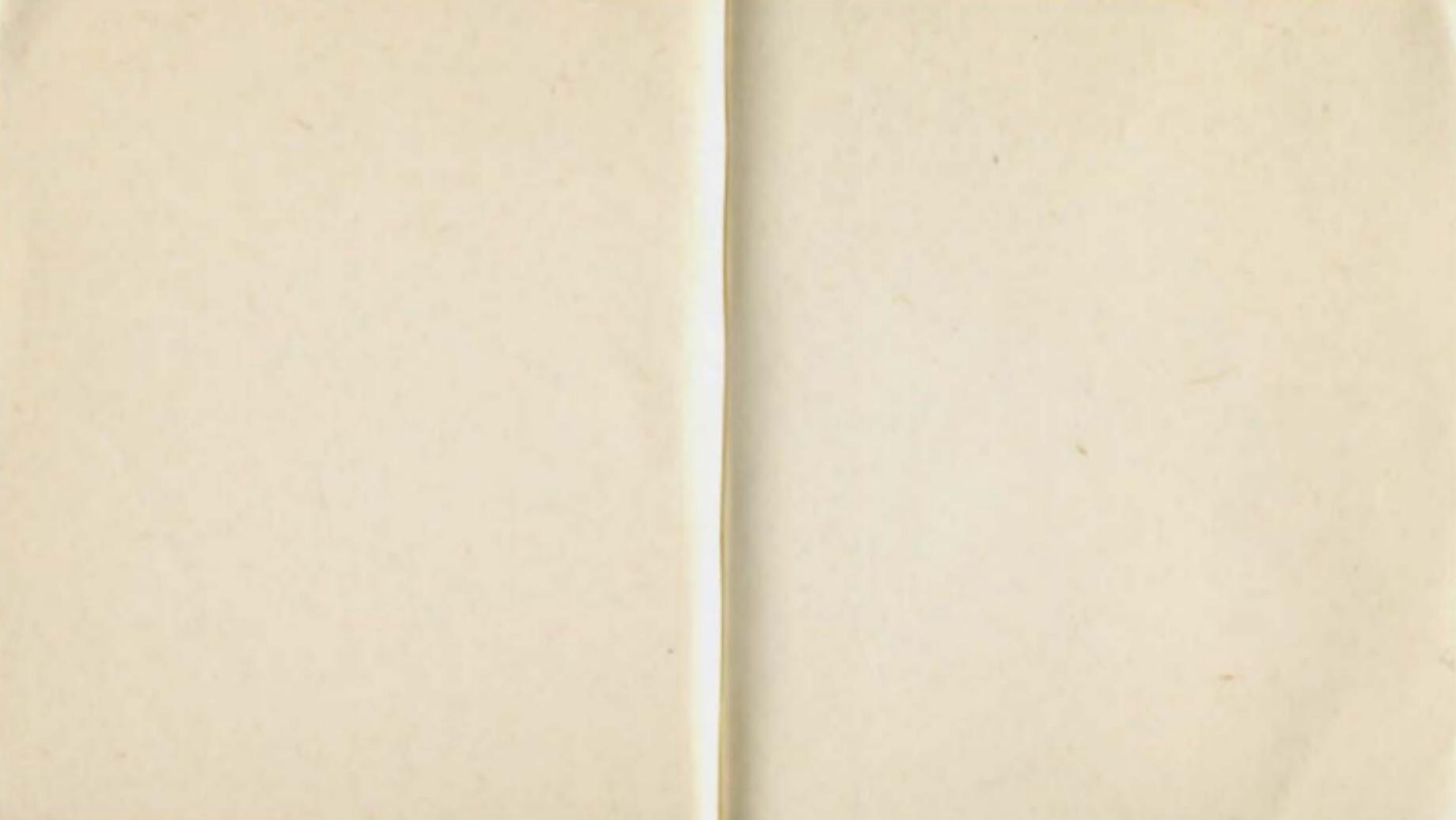
171

Faint, illegible text, possibly bleed-through from the reverse side of the page.









Stoddart

ISBN 0-7737-5015-0