50p

Practical Computing

May 1980 Volume 3 Issue 5

Gambling: double your money on the micro?

The next move in Videotex

Print out from the parallel port

Cobol: purpose-made for business

Reviewed: Z-Plus, UK 101

The best computers PLUS the best service

At MicroCentre, we're concentrating our resources on what we genuinely believe are the very best computers available today. . . . Cromemco computers, naturally. This way we can offer you the best deal possible.

What we don't do

What we don't do is spread our expertise thinly amongst umpteen different systems, or try to stock every S100 product on the market. We don't claim to offer "impartial" advice on the best buy. And we don't sell from price lists or catalogues.

The MicroCentre approach

Some micro-computer suppliers work like that, but we don't. Because we realise that when you're buying a computer you want more than the "brochures and boxes" approach. You want to see computers running; to try them out with different software products; to study the documentation; above all, you want expert answers to your most searching questions.

Cromemco specialists

That's why we've specialised in Cromemco systems. Not simply because we think Cromemco systems are the best serious computers available at the price. But because by doing so we can dedicate our time, energy and resources to giving you the highest standard of Cromemco support possible.

Cromemco Model Z-2H hard disc computer. 10 megabyte hard disc, 2 floppy discs, Z-80 computer and 64K memory. MicroCentre price £5,326.

Demonstrations

So when you visit MicroCentre expect to find Cromemco systems on permanent demonstration; expect the full range of Cromemco peripherals; single-user and multi-user systems; and interactive graphics.

Software

Expect a choice of operating systems and compilers to evaluate; expect complete documentation; and expect the largest collection of Cromemco systems software in the UK.

Expertise

Expect to find in-depth professional expertise at MicroCentre, the kind that is only acquired by installing Cromemco systems all over Britain. Expect a thorough appreciation of how Cromemco systems can be applied . . . in business, scientific research, industrial engineering, medicine and education.

Support

Expect to get frank, accurate answers to your questions at MicroCentre. Above all, once you've bought a Cromemco system from us, expect to get a very high standard of technical support with your hardware enhancements and continuing software needs.

At MicroCentre, simply expect the best.

MicroCentre's Cromemco demonstration room, with the full range of Cromemco computers, peripherals, operating systems and software products on permanent exhibition. Why not pay us a visit? We're only an hour's Shuttle flight from Heathrow!

For Cromemco... call the experts

NOW IN SPACIOUS NEW SHOWROOMS

Tel. 031-556 7354

MicroCentre

STILL IN CENTRAL EDINBURGH

Complete Micro Systems Ltd., 30 Dundas Street, Edinburgh EH3 6JN

● Circle No. 101

Book-makers may soon declare war on computers: read Rex Tingey's article on making micros forecast sports results.

Editor
Peter Laurie
Staff writer
Duncan Scot
Production editor
Toby Wolpe
Art
Margaret Smith
Editorial secretary
Susie Manning
Advertisement manager
Tom Moloney
Advertisement executive
David Lake
Advertisement secretary
Stephanie Hill
Consultants:
Technical **Nick Hampshire**
Software **Mike McDonald**
Videotex **Peter Sommer**
Publisher
Chris Hipwell
Editorial: 01-261 8752
Advertisements: 01-261 8000

Published by IPC Electrical Electronic Press Ltd, Dorset House, Stamford Street, London SE1 9LU, tel 01-261 8000, Telex/grams 25137 BISPRSG
Typesetting and artwork by Bow-Towning Ltd, London EC1
Printed by Eden Fisher Ltd, Southend-on-Sea
Distributed by IPC Sales and Distribution Ltd, 40 Bowling Green Lane, London EC1R 0NE
Subscriptions: UK, £6 per annum; Europe (ex UK), £12; rest of the world, £18 (including airmail postage). Enquire Subscription Manager, IPC Business Press (S & D) Ltd, Oakfield House, Perrymount Road, Haywards Heath, Sussex RH16 3DH, tel 0444 59188

ISSN 0141-5433

Would-be authors are welcome to send articles to the Editor but *PC* cannot undertake to return them. Payment is at £30 per published page. Programs intended for publication should ideally be justified to 22 or 44 or 66 characters per line.

Every effort is made to check articles and listings but *PC* cannot guarantee that programs will run and can accept no responsibility for any errors.

CONTENTS

45 **Editorial** / A whale of a task; Programmer of the Year Competition

46 **Printout** / Fire as security threat; Nascom System 80

51 **Feedback** / Adventure problems; TI-59 v. HP-41C; Microchess and user groups

55 **Z-Plus** / A review of the versatile business system by Vincent Tseng

58 **UK101** / Martin Collins tests a machine ideal for the hobby and educational fields

64 **Intelligent gambling** / An up-to-date account from Rex Tingey of how the micro might forecast sports results

72 **And there was light** / Fiction by John Abbatt

76 **Videotex and the micro** / Peter Sommer explores the possibilities in interfacing the micro to the viewdata and teletext systems

80 **Why good software is expensive** / A thorough investigation by Nick Horgan into the pitfalls of software development

86 **Cobol** / A business-orientated language examined by John McMillan

90 **Robotics** / How industrial robots are controlled by programs in part four of Mark Witkowski's series

96 **Parallel interfaces** / Advantages and headaches involved in attaching parallel printers by Tony Cassera

98 **ROM and firmware** / Julian Allason discusses the rapidly-growing software market based in ROM

102 **Conditional jumps and registers** / Part three of David Peckett's look at machine code

108 **6502 Special**

111 **Tandy Forum**

113 **Pet Corner**

117 **Apple Pie**

120 **COS for Apple II**/ Hugh Dobbs suggests a remedy for the Apple's lack of a cassette operating system

124 **Hardware Buyers' Guide**

160 **Diary and Programmer of the Year Competition details**

163 **Glossary** / Continuing the terminological gamut from S to T

Prestel page number 45631 / In another first, *Practical Computing* has launched its own pages on Prestel

A whale of a task

THE FIRST EDITORIAL I had the honour of writing for this magazine, in August of last year, had this striking passage:

> The Western world is like a ski jumper who has just pushed off at the top of the mountain. He's crouching, beginning to gain speed. The wind starts to whip at his goggles, he fidgets to get his balance. Before him, the great hill drops away in a sickening fall. People at the bottom are just dots.
>
> The thing is, he's never made the jump before. "He" is the state, the body politic, Hobbes' Leviathan. The minipeople riding on his shoulders are us — the hobbyists, the fanatics, the lovers of the emerging mind in the machine.

Nine months further on, in theory, we should be rushing helter-skelter downhill, the adrenalin pounding our veins.

This is not how it has turned out. Hobbes' Leviathan is still at the top of the hill fiddling with his ski bindings and asking friends what time the bar opens and the girls put on their *après-ski* pants. The manikins who were poised on his shoulders are beginning to clamber down and put on their own skis to do the jump alone, unencumbered by the giant with the boots of clay. Many of them may get hurt, but at least they'll start.

In plain terms, the national effort to foster microcomputing seems to be sinking, as usual, into a self-defeating slough of committees and further researches leading, possibly to reports and draft Pink Papers. The Government is doing so little, when it could achieve so much.

An infuriating example: it was recently made known £9 million was to be assigned to encourage microcomputing in schools. At first sight, laudable enough, but how will this money be spent? On 'curriculum changes, material for teacher training, the provision of advice on computer programmes (*sic*) and the development of programmes'.

The plain fact about teaching microcomputing to children is that you don't have to. Simply give the average child a machine and he will fall on it like a wolf on a well-fed Cabinet Minister. Before his teacher has time to say 'planned curriculum interdisciplinary re-vitalisation', the kid is already writing operating systems. Micros and children are made for each other. Therein, of course, lies the problem.

Children love computers because they are in charge of what the machine does. The thing has power, and they control that power. The teacher is relegated to the role of an advisory onlooker. To judge by the quality of much of the material sent to *Practical Computing* with the note 'aged 14' modestly attached, many teachers are an unnecessary encumbrance.

Children do not need to be taught computing in the formal sense, because they do not realise that it is difficult. Grown-ups only think it is difficult and has to be taught because they have persuaded themselves so — or rather, because the mainframe industry has spent millions in so persuading them. The micro revolution is shining a good deal of bright, illusion-dispelling light into that corner of intellectual mystification, and in the process, making many unhappy.

The mainframe business is unhappy because useful computing now costs two orders of magnitude less than it did two years ago. Teachers are unhappy because their dear little charges look as though they can quite happily teach themselves what they need to know, and go out and find well-paid jobs without any exams at all.

As a result, we have jokes of this £9 million kind — which is an expensive price to pay for a laugh, however wry.

For that much money you can buy 45,000 single-board machines with VDU, cassette and adequate Basic. Allow that an hour a day is enough computing for anyone, and that the school day — with some arriving early and staying on for a while after tea — is nine hours, that much money could introduce nearly half-a-million children to computing — and that's just this year. Allow 10 percent for maintenance per year, and for a mere £1 million, next year it could introduce another half-a-million kids, and so on.

I need hardly add that spending that much money would give a huge boost to our own microcomputer industry, which would in turn help to spread microcomputer use throughout society, which would provide jobs for the children when they leave.

Would our far-sighted masters do something so simple, so obviously useful and productive?

I labour the point. You know the answer already.

It is no use looking to Leviathan for common sense, even in his own self-interest. We have to look to ourselves; and happily the whole trend to the microelectronic revolution favours self-help. As an immediate example — pulled at random from the editorial in-tray and in striking contrast to the official position scathed above — Microsense and Apple were just about to give £30,000 worth of machines to MUSE, the association of computer teachers. No doubt these firms calculated their self-interest and found the gesture worth while. If it is worth two relatively small companies doing such a thing, how much more must it be worth to Leviathan?

In the traditional words of the crusading journalist — 'Wake up Britain, before it's too late'.

Programmer of the Year Competition

THAT DREAM of the science fiction writer, time travel, is a reality in the *Practical Computing* office. Here days take minutes, months days. We held the last Programmer of the Year Competition three weeks ago — or so it seems. It's May again and time to announce the next one.

We shall be holding our second Programmer of the Year Competition starting — wait for it — now. The rules are much the same as they were — see page 161 — and entries must be received by July 1, 1980.

Last year's entries produced some impressive programs. We hope this year will do even better.

Fires pose security problem for micros

THE NEWS that a consignment of Apples went up in flames at the hands of an arsonist in December has made the U.K. micro industry look again at its own problems of security. For the users, however, the main worry has been about how they would survive a system crash or a fire which destroyed all their business records. More than 50 percent of those companies suffering from a fire, which destroys their records, go out of business within six months.

In an attempt to benefit from the spread of micros, Chubb, the lock and safe manufacturer, has introduced a disc box which is designed to store floppy discs in a fire-resistant filing cabinet. Each unit will cost £610.

More of the Chubb fire-resistent safes are designed to protect paper records and rely on a concrete layer in the casing which contains chemically-bonded water. In a fire, this slowly boils away keeping the internal temperature at 100C — discs deteriorate at 65C. Despite the fact that the company will make no firm guarantees, it claims that discs, in the disc box, will be fully protected under the test conditions of one hour's heating in a furnace, taking the temperature from 21C to 927C, shuting down the burners and leaving the cabinet to cool for 20 hours.

Chubb admitted that the most difficult potential customers are those who claim that it is cheaper and easier to make back-up discs every night and take them home.

Hearing tested with the Pet

AN AUTOMATIC audiometer has been interfaced to the Pet and enables anyone employed at a noisy industrial plant to have his hearing tested regularly, the results analysed and compared to previous records held on floppy discs.

A noise induced hearing loss is irreversible and results in the loss not only of the ability to detect quiet sounds, but also the ability to discriminate speech. Industry is now being encouraged to use this kind of equipment because it provides a check on the efficiency of hearing conversation programs, can assist in the placement of individuals in which good hearing acuity is required or who may be susceptible to noise-induced deafness. The courts are beginning to recognise employers' responsibilities.

The courts were awarding claims 10 years ago of up to maximum of £1,000. Claims are now being settled at anything up to £30,000 and there are a number of cases, for example, some shipyards and steelworks, where the number of claims has run into thousands.

The audiometer is supplied by Alfred Peters and Son Ltd. Tel: (0246) 418861. The interface software and the analysis software was developed and is supplied by the Sheffield Computer Centre, which can be contacted on (0742) 53519.

A Pet attached to the audiometer.

New system

TRIUMPH-ADLER, the West German business systems manufacturer, backed by Volkswagen, will launch a £450 system aimed at the home and school markets soon. It will have 4K of RAM, cassette and TV interface. A 48K version with two minidiscs and a printer will cost £2,400. Basic will be provided.

PO approval for Apple

THE APPLE II has become the first popular microcomputer to be approved by the Post Office for connection to Post Office modems, and is now well on the way to being accepted as a suitable terminal for the Prestel viewdata system.

Post Office approval means that Apples can use standard telephone lines to communicate with remote terminals and other computers throughout the U.K. and internationally. It can also be used with the Datel 200 service, using the standard Apple communication Interface card, and with Datel 600.

The main concern of the Post Office is the safety and technological reliability of any equipment attached to its network. The approval means that the Post Office is unlikely to have any objection to Appletel, the Apple Prestel Terminal Package developed by Mike Gardner of Owl Computers. Apple users will be able to interface directly to the viewdata network without having to purchase expensive viewdata terminals. Viewdata already includes a number of pages of programs which could be loaded directly.

Challenge to schools

SCHOOLS were urged to send their pupils along to the Challenge of the Chip exhibition at the Science Museum, in London, when it was recently opened by Neil MacFarlane, Under-Secretary of State for Education and Science. The exhibition will run until the end of 1980 and has already attracted large crowds.

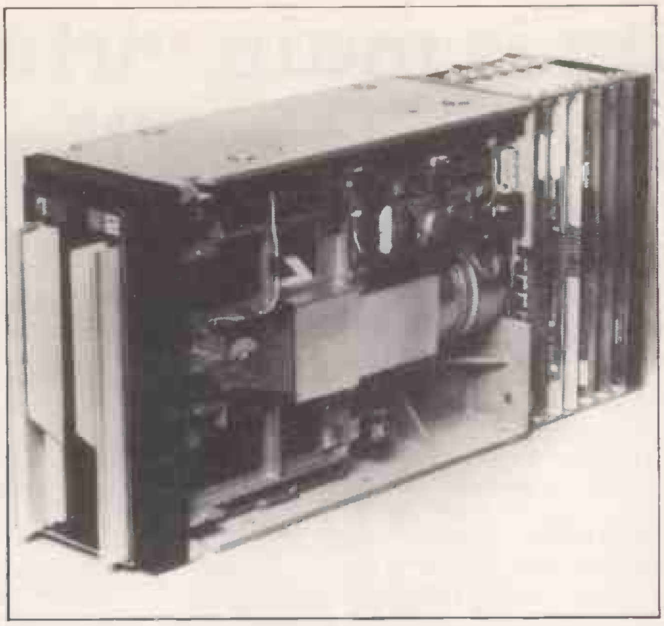
Products from about 60 companies are displayed in 10 sections covering shopping, offices, transport, communications, production and control, education, music, medicine, the home and toys. The exhibition also recalls the origins of the silicon chip and traces its development.

Some of the major exhibits include an industrial robot, word-processing systems, telecommunications equipment, the latest supermarket checkout from IBM, and a medical diagnostic package which can ask simple questions like "Now tell me, are you feeling well today"? and can then respond with the sympathetic but none too helpful "I am sorry to hear that you are feeling unwell". Research Machines also has two of its 380-Z microcomputers on display, one of which gives the public direct access to a full keyboard to control a number of demonstration packages.

Memorial lecture

A MEMORIAL lecture is to be held for the late Dr Christopher Evans, author of the book and television series, the Mighty Micro. The lecture will be in the Fyvie Hall, Polytechnic of Central London, 309 Regent Street, London W1 at 2pm on May 29. Professor Tom Stonier, of Bradford University will speak on Microprocessors and the Future. There will also be a speaker from the National Physical Laboratory where Dr Evans worked.

Applications for the free tickets should be made to the London Regional Management Centre at 311 Regent Street, London W1. Tel: 01-637 7583.



World's first 6MB floppy drive

THE OEM division of Burroughs Machines has launched the world's first 6MB floppy disc drive, the MD122. It is a compact unit, incorporating twin 8in. floppy discs and an integral controller in about the same amount of space as most conventional 1MB drives.

The MD122 drives have been developed at Burroughs U.K. plant, Glenrothes, Scotland.

Turing Trust discourse

THE TURING Trust is calling for a permanent memorial to Alan Turing, the mathematical founder of computing.

It would be run along the lines of the Turing Award presented every year by the Association for Computing Machinery for the most significant contribution of a technical nature to the computing community.

The trust proposes to endow an annual A M Turing Lecture to be delivered in a British centre of learning by a lecturer of international distinction of any scientific topic to which Turing contributed. The initial target is £10,000.

Further information about the appeal, the Trust, Turing's collected works, and a projected biography can be obtained from the A M Turing Trust, 1 Hope Park Square, Edinburgh EH8 9NW, Scotland.

Copyright protection proposals

GOOD NEWS for anyone still confused by the chaotic state of the law of copyright — a London barrister and programmer, Alistair Kelman, has drafted a Bill to give explicit protection to computer software and has started a campaign to have the Bill introduced before Parliament.

Some support for the Bill has already been drawn from Sir Keith Joseph, Secretary of State for Industry, and Phillip Vergo, Secretary of the Conservative Computer Forum, although the Civil Service still holds the view that the matter is less than urgent. A Green Paper will be published in May which will examine the whole area of copyright, without necessarily giving much attention to computing. On past record, it means that a new copyright law could be passed by Parliament in something like five to seven years.

Kelman's Bill has been drafted to ammend the Copyright Act of 1956 and is designed to tackle the following questions: Are Computer programs copyright works? If a copyright work is converted into digital impulses is this an infringing act under copyright law? Is output from a computer a copyright work? Can work written with the aid of a computer have a copyright?

The new Bill answers these questions by defining a new process "transmutation" — compilation already has a meaning in law — which will mean the "automatic conversion of a work into an object code by electronic or mechanical or similar techniques."

Let us take one example. A writes a medical diagnostic manual which is published and sold as a book. B takes A's manual and feeds the pages through an optical character reader to create a digital recording on disc. B then writes a computer program which uses A's manual as a source of information. C sits down at a terminal and responds to questions asked when running the program.

The Bill would ensure that the conversion of the manual was an infringing act. The responses of the user to the questions posed by the computer would not create a new copyright work. Nevertheless, if the terminal produced a permanent copy of the substantial part of the manual for the user to take away with him, it would infringe the copyright of the manual, if it were not to be used for research or private study. B would have copyright on his program.

One of the advantages of the Bill is that it could be implemented quickly and it would cover all likely technical developments. Kelman is now trying to persuade the Science and Technology Committee of the House of Lords to appoint a sub-committee which could introduce the Bill there. It is not a money Bill and it is not politically contentious so there should be no procedural objections.

SGS-ATES

MIDWICH Computer Co has asked us to correct an error in our review of the SGS-ATES — October, 1979. We said that the price of the machine without PSU was £350. In fact, the price should have been £260 plus £90 for the PSU. The price of the machine is now £240. We regret any difficulty this has caused Midwich.

Flexibility is main virtue of Nascom System 80

IF THERE is still any doubt that the market for microcomputers is moving away from the hobbyist and the kit, it will be dispelled by a look at the moves now being made by Nascom to house its Nascom 2, add some new cards and call it the System 80. The managing director of Nascom, John Marshall, hopes that it will enable the company to compete in the packaged desk-top microcomputer market and

by DUNCAN SCOT

retain its position as one of the pioneering British manufacturers.

A moulded glass-fibre case, a five-slot card frame with a Nasbus motherboard, the Nascom 2 processor in the first slot, the power supply and the keyboard, form the basis of the new system whose main virtue is flexibility. The choice of five new cards, each of which allows its own options, means that the System 80 can be configured for virtually any application suitable for microcomputers.

Colour board

The five new cards include a RAM board in 16K, 32K or 48K options, an I/O board, a programmable character generator board, a colour board and a floppy disc controller. Although all of the boards, except the disc controller, are now on display, production facilities are so limited that the boards will be released only one at a time during the next few months.

Disc controller

The floppy disc controller board should be ready later in the summer. The delay is the result of what Nascom claims were considerable difficulties in finding a reliable supplier of floppy drives. It has switched its preferred supplier twice and has now settled on Siemens mini floppy, double-sided, double-density discs. The board will be able to control up to four drives.

The floppy disc units will be based on the CP/M operating system, rapidly becoming the industry standard. A recent report which revealed that there are more than 29 CP/M word-processing packages demonstrates that Nascom, who regards itself as an engineering company and does not want to become entangled with supporting customers and software, should have little problem in finding dealers willing to supply suitable software.

Quite how successful the new system will be must depend on Nascom's ability to supply the products in volume, persuade the potential buyers that the price is right, and keep the dealer networks and users assured that it has sufficient control over its own financial affairs to survive the turbulance expected in the market during the next nine months.

The market

The produce must be a move in the right direction and yet again Nascom seems to have judged the state of the market and been able to produce the basic product just in time. It claims to have enough advance orders for chips to keep well in supply, avoiding some of the more damaging problems which emerged last year. John Marshall claims that sales of the Nascom 1 are now considerably more than 15,000 and that the Nascom 2 had sold more than 3,000 by the end of February 1980.

The company hopes that many of the new systems will be bought by its existing users in industry, which it thinks may well account for nearly half its sales, and its original kit hobbyists maturing into real home computing.

The System 80 will be sold on its flexibility, rather than its price. The system can work out expensive with the case at £85, a 3 amp power supply unit at £30, the Nascom 2 processor at £225 and the 32K RAM board at £165 totalling £505. The more complex configurations bring some economies with a 48K system with a programmable generator board and twin disc drives at £1,355.

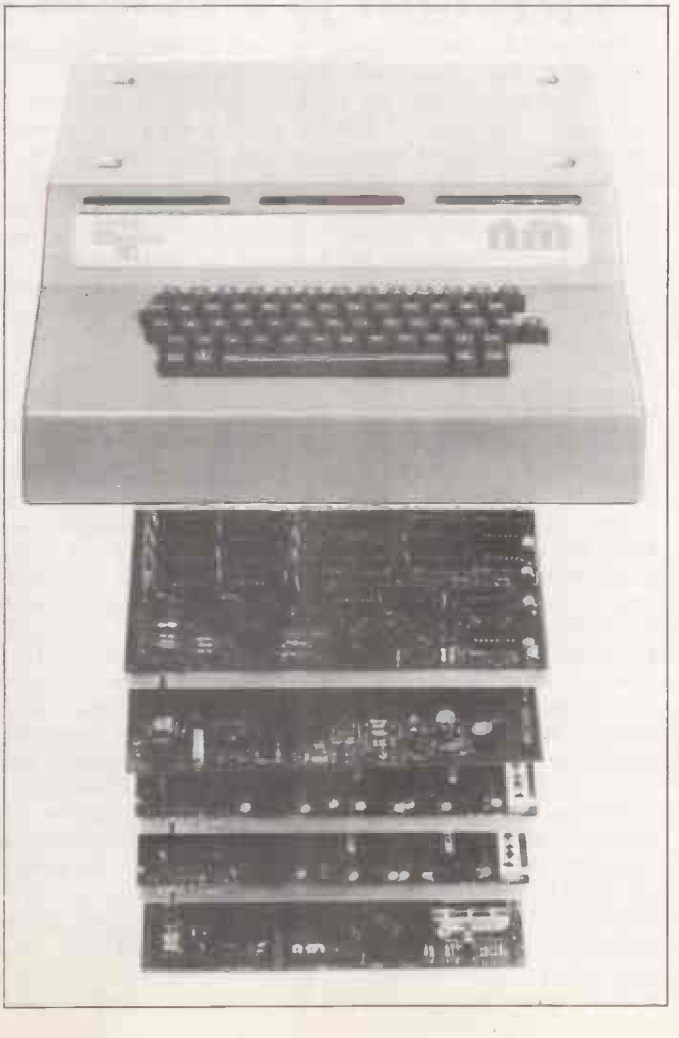
Peripherals

Peripherals for the System 80 will include the Nascom IMP matrix printer, announced last year and now becoming available at £325. It can be plugged straight into the System 80 and will accept cut sheets.

Breaking with the all-British tradition of the company, the boards will be assembled in the U.S., taking advantage of the economies of scale in the U.S. manufacturing plants and avoiding the dearth of appropriate testing facilities in the U.K.

Another important factor in the decision to assemble the boards on the other side of the great divide lies in the misguided patronage of the British tax laws. In an effort to protect our existing electronics manufacturing, chips, as electronic components, incur an import levy of 18 percent whereas assembled boards, as computer parts, only rate 6 percent. The cost of assembling boards in the U.S. can be more than offset by the savings in tax.

The System 80 case with the Nascom 2 CPU board and four expansion boards. The case has been designed with recesses to take a TV/monitor or an expansion box, still under development, which will hold a further five cards.



Our Feedback columns offer readers the opportunity of bringing their computing experience and problems to the attention of others, as well as to seek our advice or to make suggestions, which we are always happy to receive. Make sure you use Feedback—it is your chance to keep in touch.

Spirit of Adventure

I HAVE A small problem and was wondering if you or your readers could help me. On our micro — a Z-80 — in my office we have the Z-80 version of Adventure, and my colleagues and myself are, unfortunately, addicted to it. The problem is that we have discovered 14 of the treasures but cannot find any more. We have achieved a score of 235 out of the possible 350 but no higher. I was wondering if you can answer a few of my questions about the game and perhaps give me directions as how to find more treasure.

The treasure we have found so far is silver, gold, jewellery, diamonds, trident, golden eggs, pearl, spices, golden chain, pillow, vase, coins, emerald and the rug.

How do you explore beyond the plover — emerald — room? If you remember you cannot take your lamp into this room through the small crack so there is no source of light.

I have heard that there is a platinum pyramid in the cave somewhere, how do you arrive at this? Of what use is the mirror in Mirror Canyon? How do you get to the maze containing the pirates' chest? Are there any more magic words apart from XYZZY and PLUGH? Does rubbing or waving any of the treasure achieve anything? Finally, what happens in the Masters section?

It would be a great help if these questions were answered and save eight people from becoming demented with frustration.

Finally I would like to congratulate you on an excellent magazine and would like to see a few more articles about Cobol and assembly language for beginners.

Lawrence Davis,
North Cheam,
Surrey.

Difficult task

I CONFESS to feeling a certain amount of sympathy with Ian Crosswell — Feedback, March 1980. It is bad enough for an inexperienced programmer to be faced with the task of modifying an extremely-complicated and minimally-documented program. However, the suggestions made by the Feedback editor in reply to his plea for assistance make matters infinitely worse.

Not only does the internal organisation of the Basics concerned make the use of PEEK and POKE into array space extremely hazardous, in this particular program the elements of the array in question each require more than one byte of storage, a point explained at length in the original article. Finally, the suggested measures would not reclaim sufficient storage space even if the numbers would fit into one byte, chiefly because array elements in RML 9K Basic occupy four bytes, not the six claimed by your Feedback editor.

The most serious complaint is, however, that attempting to implement the program using the methods suggested would take several weeks of effort by someone unfamiliar with the Basic — it would take me a few days, and I wrote it — whereas

```
20 DIM A(500), C(3)
7010 A(500-Z) = A(N2-Z)
7030 G = 501-N2
```

takes but a few moments, and solves the problem by the simple expedient of reducing the size of the offending array.

The country is going to need many good programmers in the future — please don't put beginners off the whole subject by sending them on wild goose chases.

Steve Thomas,
Oxford.

QWERTY explanation

CONCERNING THE comments in the February issue on the QWERTY keyboard, surely a moment's thought would have made clear that the reason for its lay-out was not just an erroneous calculation of letter frequency.

Even the worst maths teachers might be expected to realise that the letter A is fairly common, yet it has been placed under the smallest finger on the left hand.

The more likely explanation, a factual source of which I am afraid I don't know but would be interested to find out from readers, was that technology did not allow fast use of the keyboard. QWERTY was, therefore, designed specifically to be inefficient and to slow down the typist, which it has been doing very effectively ever since.

R Fawdry,
Edinburgh.

Avoiding waste

THE PROGRAM Shooting Gallery, *Practical Computing*, March, page 102, demonstrates the ability with which bytes can be wasted on the Pet. Admittedly, the program is a short one, but in longer programs those extra few bytes can make all the difference between a faultless program and an ?OUT OF MEMORY ERROR IN.

The first waste is in line 10 — the statement 'X. % = 87. It is generally considered that integer variables — indicated by a % sign after the name — save memory in the variable table.

However, a look in the Pet manual shows the truth: an ordinary variable takes 7 bytes in the table — 2 for the variable name, and five for the actual value in binary floating point.

An integer variable such as A% has 2 bytes for the name, 2 for the actual valuable, a hi and a lo byte, and 3 bytes filled with zeros, so it still takes 7 bytes.

AJ Lea,
North Shields.

Grampian group

THE GRAMPIAN Amateur Computer Society, was started about a year ago and we formed a constitution in July, 1979. We meet every second Monday of the month, at the Holiday Inn, Bucksburn, Aberdeen. We have more than 30 paid-up members who each receive a newsletter every month. The newsletter is printed by a local Pet supplier, Pilgrim Business Machines, and contains details of meetings, short articles and snippets of useful information. Our meetings consist of demonstrations by local computer suppliers — Pet, Apple, Sharp — visits, club nights and an auction.

We recently purchased an Acorn computer for members to use and gain experience with a computer.

We have good attendances at our meetings and welcome anyone who is interested in computers. If anyone in Grampian region wishes to find out more they should ring me at Lumphanan (033 983) 284.

M Basil,
Lumphanan,
Kincardineshire,
Grampian.

Unfair comparison

VINCENT TSENG, in his article on the Hewlett-Packard HP-41C in *Practical Computing* January 1980 appears to conclude that it compares very closely to the Texas Instruments TI-59. How on earth can he come to that conclusion? I own a TI-59 and, comparing it to what Tseng says about the HP-41C, Texas win without even trying.

The HP-41C uses Reverse Polish Notation. On page 147 of the same issue of *Practical Computing* your glossary says of Polish Notation: "Now forget it". I doubt that anyone who has tried to use Reverse Polish ever will. The TI-59, by contrast, uses common or garden

(continued on page 53)

Now you can control your business for less than £2,500.

This could be your best investment opportunity yet. A complete computerised business system, including a Floppy Disk Unit, high-speed Printer and *Britain's best selling microcomputer* – the Commodore PET. All for under £2,500.

First Class Programs

A comprehensive range of first class programs is offered by *Commodore 'Business Software' Dealers*. These are available on disk from £50-£500. And they cover such applications as *Business Information, Stock Control, Word Processing, Payroll, Accounting* and *Mailing Systems*.

Service and Support

With over 10,000 PET computers installed in the UK, dealer support is growing fast. A nationwide network of 90 official *Commodore 'Business Software' Dealers* ensures that service and technical facilities are close to every PET user. Our dealers can even offer you a 24 hour on-site maintenance agreement.

Training and Instruction

The PET Business System is self-contained and simple to use. Should you require personalised programs or extensive installation training this can be arranged with your *Commodore 'Business Software' Dealer* who can also give details of official *Commodore Training Courses*. These include intensive 2 & 3 day workshops to train you to write your own programs.

For full details about the Commodore PET Business System, Training Courses, Programs, *and 'Business Software' Dealers*, simply fill in the coupon and post today.

To: Commodore Information Centre,
360 Euston Road, London NW1 3BL.

Please send me details of the PET Computer Business Systems.

Name ____________________

Position ____________________

Company ____________________

Address ____________________

____________________ Tel. No. ________

If you have a particular application in mind please specify:

____

____

____

P.C.B.3

commodore
We made small computers big business.

• Circle No. 170

(continued from page 51)

algebraic notation with good old brackets if you need them.

The TI-59 is supplied with built-in magnetic card reader; on the Hewlett-Packard machine it is an optional extra. While we're on the subject, the HP-41C retains its memory when switched-off, which the TI-59 does not, but then it does not need to with a built-in card reader.

Both calculators have built-in extra functions, but the HP-41C requires seven keystrokes to use them when the TI-59 needs only four. It is true that the HP-41C can display alphabetic characters and the TI-59 cannot.

It is apparently too easy to clear the program memory on the HP-41C. A similar mistake on the TI-59 requires 2 keystrokes, the second of which is labelled CP, i.e., clear program — not something easily done accidentally. Other than that, the program can be over-written step by step, but neither is this an easy mistake since the code already in the memory must be displayed in order that it be over-written. I agree with Tseng that the manual for the TI-59 is excellent.

The TI-59 can be bought for £156.50 from a well-known discount shop — the HP-41C costs £173.48 from the same supplier, and that is without a magnetic card reader.

In all the above points I have simply compared the TI-59 to the HP-41C as read from the article, and while I do not object to anything in the body of the article, I find his conclusion ridiculous. He says: "As a calculator the HP-41C is certainly one of the most sophisticated I have come across, although the TI-50 rivals it quite closely". Rivals it quite closely? The Texas is streets ahead.

C P Watters,
Aldershot, Hampshire.

Far ahead

I CANNOT agree with Vincent Tseng's evaluation of the Hewlett-Packard 41C calculator compared to the Texas Instruments TI-59. Indeed, on reading his article, I cannot imagine what he intends to use these instruments for. I use both — and previously the HP29C — for pioneering work on a multi-parametric problem to establish the conditions under which a Fortran-based investigation on a full-sized outfit is likely to prove rewarding.

For this field, which involves a great deal of thought, and plenty of trial and error, the Hewlett-Packards are far better. In terms of useful capacity, the TI-59 is little better, all in all, than the HP-29C, and the HP-41C is far ahead.

The Hewlett-Packard continuous memory means that the data for a problem can be held in the calculator, for months if need be, until a satisfactory program has been developed, whereas the TI-59 magnetic cards are fiddly even with the read facility. The Hewlett-Packard keys are easiest to use, and I have never had reason to suspect any Hewlett-Packard machine failure.

The batteries in the TI-59 last only an hour or two when the calculator is in continuous use, whereas the HP41C batteries have lasted from September 1979 and do not yet need replacement.

The fact that with the HP41C user-written programs can be assigned to keys means that very heavy calculations can be carried-out without over-complication.

There are omissions in both manuals, but the Hewlett-Packard manuals suggest to me that the liaison between designers and users has been better.

J M Craddock,
Norfolk.

West Midlands

THE WEST Midlands Amateur Computer Club meets on the second and fourth Tuesday of each month. The venue for the meetings, unless advised otherwise, is Elmfield School, Love Lane, Stourbridge, West Midlands. Each meeting usually commences at 7.30 pm with the first meeting being a lecture, talk/discussion or demonstration while the second meeting of each month is usually kept for systems to be brought along and compared, repaired and improved.

The annual subscription for 1980 is £3 — £2 if you are still receiving full-time education. Visitors are welcome to sample a meeting without obligation.

The club secretary is John Tracey of 100 Booth Close, Kingswinford, West Midlands, tel: Brierley Hill (STD 0384) 70097. Enquiries for further information should be addressed to him.

The club has a membership of 60 with several enquiries being received since our last meeting and systems in use in the club at present include eight Pets, 12 Nascom 1s, five Nascom 2s, three TRS-80s, Sharp MZ80, four Newbear 7768s, two Apples and 12 other assorted systems.

Malcolm Sparrow,
Wolverhampton.

Microchess

IN YOUR February edition you published a Microchess game on the Pet which was submitted to you as a quick revenge game. Having overcome my surprise at any one losing to the Pet, I was further surprised your contributor thought he won quickly in 22 moves.

He may like the following:

1. e2 e4	King's Pawn opening	e7 e5
2. dl h5	Novice attack	b8 c6
3. fl c4	Mate threat	g7 g6
4. h5 f3	Maintain threat	d8 f6
5. bl c3	Develop knight	f6 e7
6. c3 d5	Attack Queen & Bishop Pawn	e7 d6
7. d5 c7	Check	d6 c7
8. f3 f7	Check	e8 d8
9. f7 f8	You win	

Total time about 9 minutes — Pet takes 7 of them.

B E Robinson,
Coventry.

King's gambit

I FEAR YOU will now be inundated with games which beat Microchess in fewer than 22 moves. Here is a King's gambit — always worth a try against a greedy opponent:

1	e2e4		e7e5	
2	f2f4		e5f4	accepted
3	g1f3	against & attack	d8e7	
4	b1c3		g8f6	
5	f1c4		f6e4	greedy
6	c4f7	to stop castling	e8f7	
7	0-0	the attack	e7c5	a self-indulgent check
8	d2d4	tempts again	e4c3	
9	b2c3		c5c3	can't get back to defend, now
10	f3g5		f7g8	other moves are no better
11	f1f4		c3a1	the condemned man ate a hearty supper
12	d1f1		f8d6	
13	f1c4	mate		

John Race,
Brunel University,
Uxbridge.

Croydon micros

A MICRO group has been formed in Croydon, Surrey, by more than 20 local people.

To ensure a successful start, please send brief details of your interest beforehand.

Vernon Gifford,
111 Selhurst Road,
London SE25 6LH.

CPM society

CPM IS USED WIDELY in microcomputer systems throughout the country and it has become the operating system for 8080, 8085 and Z-80 systems. CPM is normally provided by the supplier of the microcomputer as the operating system needs tailoring to the particular hardware. This wide dispersion of supplies of CPM has handicapped the exchange of ideas between CPM users.

We have formed the CPM Users' Group as a non-profit-making society to provide a forum to encourage and foster contact between CPM users. We are currently planning our first quarterly magazine and we would welcome contributions from any CPM users.

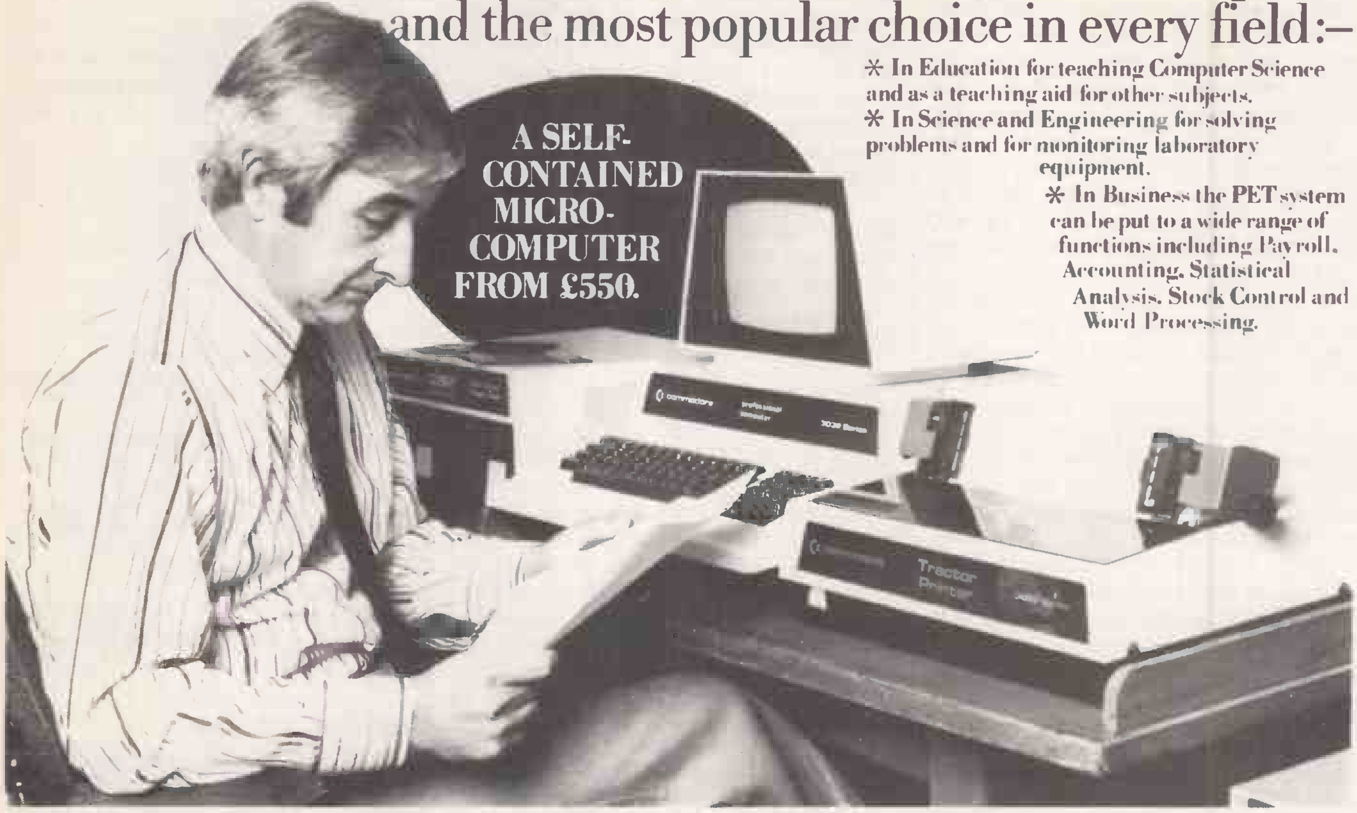
We are also compiling a list of all the microcomputer systems and software packages which are available in this country. The society intends to draw up a register of CPM specialists.

D Powys-Lybbe,
CPM Users' Group (U.K.),
C/O MML Ltd,
11 Sun Street, Finsbury Square,
London EC2.

Your Commodore PET System

The Commodore PET is Britain's best selling microcomputer and the most popular choice in every field:–

* In Education for teaching Computer Science and as a teaching aid for other subjects.
* In Science and Engineering for solving problems and for monitoring laboratory equipment.
* In Business the PET system can be put to a wide range of functions including Payroll, Accounting, Statistical Analysis, Stock Control and Word Processing.



Not least of its attractions is the price of a PET – from £550 for a self contained unit, to under £2,500 for the complete system including Floppy Disk Unit and high-speed Printer. Ask your nearest Commodore dealer below for details about Commodore hardware, software and training courses.

Our Dealer* Network

LONDON

Capital Computer Systems, W1. 637 5551
ACE (by Top TV Ltd), SW1. 730 1795
Micro Computer Centre, SW14. 876 6609
Logic Box Ltd, SW1. 222 1122
Sumlock Bondain Ltd, EC1. 250 0505
Da Vinci Computers Ltd, NW4. 202 9630
L & J Computers, NW9. 204 7525
Adda Computers, W5. 579 5845
CSS Business Equipment Ltd, E8. 254 9293
Advanced Management, EC2. 638 9319
Metyclean Ltd, SW1. 828 2511
Microcomputation, Southgate. 882 5104
T.L.C. World Trading Ltd, WC2. 839 3894

HOME COUNTIES

Orchard Electronics Ltd, OXON, 0491 35529
D. L. Chittenden Ltd, CHESHAM, 4441
J. R. Ward Computers Ltd, MILTON KEYNES, 562850
Dataview Ltd, COLCHESTER, 78811
South East Computers Ltd, HASTINGS, 426844
Symtec Systems Ltd, SOUTHAMPTON, 38868
Alphascan Ltd, BANBURY, 75606
Super-vision, SOUTHAMPTON, 774023
Millhouse Designs Ltd, ALTON, (042) 050374
Micro Facilities Ltd, MIDDX, 979 4546
DDM, BRENTWOOD, 230480
Stuart R. Dean Ltd, SOUTHEND, 62707
Alpha Business Systems, HERTFORD, 57423
HSV Microcomputers, BASINGSTOKE, 62444
HSV Microcomputers, SOUTHAMPTON, 22131
RUF Computers (UK), BURGESS HILL, 45211
Wego Computers Ltd, CATERHAM, 49235
T. & V. Johnson, CAMBERLEY, 62506
T. & V. Johnson, OXFORD, 721461
Petalect Electronic Services Ltd, WOKING, 23637/21776
Business Electronics, SOUTHAMPTON, 738248
Amplicon Micro Systems Ltd, BRIGHTON, 562163
Bromwall Data Services Ltd, HATFIELD, 60980/64840
MMS Computer Systems, BEDFORD, 40601
Isher-Woods, LUTON, 416202
Sumlock Bondain, NORWICH, 26259
CSE (Computers), READING, 61492
Oxford Computer Systems, WOODSTOCK, 811976

MIDLANDS & STH. HUMBERSIDE

Taylor Wilson Systems Ltd, KNOWLE, 6192
Betos (Systems) Ltd. NOTTINGHAM 48106
Holbrook Business Systems, DERBY, 368088
Lowe Electronics Limited, MATLOCK, 2817
Davidson-Richards Ltd, DERBY, 366803/4
Arden Data Processing, LEICESTER, 22255
Tekdata Ltd, STOKE-ON-TRENT, 813631
C.S.M. Computer Systems, BIRMINGHAM, 360 6264
Business & Leisure Microcomputers, KENILWORTH, 512127
Caddis Computer Systems Ltd, HINCKLEY, 613544
Allen Computers, GRIMSBY, 40568
CPS (Data Systems) Ltd, BIRMINGHAM, 707 3866
Camden Electronics, BIRMINGHAM, 773 8240
Cliffstock (Computer Systems) Ltd, WOLVERHAMPTON, 24221

YORKSHIRE & NTH. HUMBERSIDE

Microprocessor Services, HULL, 0482 23146
Microware Computers, HULL, 562107
Computer Workshop, LEEDS, 788466
Hallam Computer Systems Ltd, SHEFFIELD, 663125
Ackroyd Typewriters Ltd, BRADFORD, 31835
Datron Micro Centre, SHEFFIELD, 585490
Yorkshire Electronics Service Ltd, MORLEY, 522181
Sheffield Computer Centre, SHEFFIELD, 53519

NORTH EAST

Dyson Instruments, DURHAM, 66937
Currie & Maughan, GATESHEAD, 774540
Wards Office Supplies, GATESHEAD, 605915
Tripont Associated Systems, SUNDERLAND, 73310
Newcastle Computer Services, NEWCASTLE UPON TYNE, (0632) 615325

SOUTH WALES & WEST COUNTRY

Computer and Design, BROADSTONE, 0202 697341
A. C. Systems, EXETER, 71718
Computer Supplies (Swansea), SWANSEA, 290047
Sigma Systems Ltd, CARDIFF, 21515
Devon Computers, PAIGNTON, 526303
Bristol Computer Centre, BRISTOL, 23430
J. A. D. Integrated Services, PLYMOUTH, 62616
Sumlock Tabdown Ltd, BRISTOL, 26685
Radan Computational Ltd, BATH, 318483
T. & V. Johnson Ltd, BRISTOL. 422061

NORTH WEST & NORTH WALES

B. & B. Computers Ltd, BOLTON, 26644
Megapalm Ltd, CARNFORTH, 3801
Tharstern Ltd, BURNLEY, 38481
Fylde Business Machines Ltd, PRESTON, 731901
Preston Computer Centre, PRESTON, 57684
RPL Microsystems, DOUGLAS, 4247/8

LIVERPOOL

Microdigital, LIVERPOOL. 227 2535
Rockliff Brothers Ltd, LIVERPOOL, 521 5830

MANCHESTER

Cytek (UK) Ltd, MANCHESTER, 832 7604
Executive Reprographic Ltd, MANCHESTER, 228 1637
Sumlock Manchester Ltd, DEANSGATE, (0618) 834 4233
Computer Workshop, MANCHESTER, 832 2269
Professional Computer Services Ltd, OLDHAM, 061-624 4065
D. Kipping Ltd, SALFORD, 834 6367
Catlands Computers Ltd, 0625 527166

SCOTLAND

Microcentre, EDINBURGH, 225 2022
Thistle Computers, KIRKWALL, 3140
McAllister Business Equipment, EDINBURGH, 336 2402

IRELAND

Softech Ltd, DUBLIN, 784739
Medical and Scientific, LISBURN, 77533

*This is a list of dealers participating in associated advertising and not a full list.

commodore

We made small computers big business.

Commodore Information Centre, 360 Euston Road, NW1 3BL. 01-388 5702

• Circle No. 171

Strengths of Z-Plus reside in all-round business ability

THE Z-PLUS SYSTEM was delivered late one night by two amiable gentlemen called John and John, from Rostronics — it is reasonably heavy and requires two to carry it. The Z-Plus System is normally housed in a desk, but the computer box was delivered without desk for portability. The CPU box weighs, according to one of the Johns, 65lb. It looks like a large hi-fi speaker, but must not be stood on end — it should be laid on one of its sides.

The computer box is made by the U.S. firm Micromation and contains a Z-80A system with a claimed 64Kbytes of RAM working at a full 4MHz with no wait-states. There are dual 8in. — full-sized — floppy-disc drives capable of recording in both single- or double-density, soft-sectored format, which gives more than 1MB capacity per disc using double-density. The bus is S-100-compatible. Operating system is CP/M.

Preferred terminal

Now this may sound ordinary, but surprisingly few systems have all this, as John Barton of Rostronics points out. The preferred terminal for the Z-Plus, the Elbit DS 1920 VDU terminal, with 15in. screen and 96 ASCII character set capable of displaying lower-case with true descenders, accompanied my Z-Plus System, with a daisywheel printer, the Multiwriter.

After some muscular exertion, I managed to set the system up on a table; the VDU and printer alone occupied the whole 4ft. × 3ft. area. Because of its weight and size, the computer box was set on the floor by my feet — not an ideal arrangement. I am sure the fitted-desk version would not only look neater, but would probably be more convenient. Frequent moving of the equipment is not recommended.

Connecting-up was relatively simple. Cables were supplied to attach the VDU and the printer to the CPU box, then each

by Vincent Tseng

of the three items needed to be connected to the mains. Mains points were a minor inconvenience; unless one has three points in close proximity, the best way is to use an extension lead with a distribution panel.

Switch-on brought all the components of the Z-Plus to life. The indicator light on drive A of the two floppy-disc drives was on, showing that the drive was being accessed — but since it is not recommended to have a floppy disc inserted in the drives either on power-on or power-off, nothing much else happened. The cursor just blinked in the top-left corner of the VDU screen.

By inserting a system disc into drive A, pressing the re-set button on the top of the CPU box, sounds of movement came from the drive and CP/M announced itself on the screen. I noticed that for once there was no fuss in setting-up of the serial connection for the VDU, in terms of line speed — baud rate — parity, or number of bits matching. They have been pre-set and matched by Rostronics, on both the CPU box and the VDU, to 9600 baud, since it supplies both.

Clear display

Although the Elbit VDU supplied as standard is big, the larger-than-average 15in. screen did not look that much bigger than a standard 12in. VDU screen. The characters are displayed very clearly, though, to a standard rarely seen on lower-priced terminals.

The detachable keyboard was of the conventional typewriter QWERTY lay-out and handled satisfactorily. It has 96 keys but some were not used in the normal operation of this system, e.g., cursor control keys which were not used in the operating system CP/M, and some more obscure ones for screen editing.

Two versions

There were two small annoying points on the good VDU, my sample buzzed loudly, and there were two double-sized control keys on either side of the space bar, which were often used mistakenly instead of the shift keys.

(continued on next page)

(continued from previous page)

Two versions of CP/M were supplied for this review, 1.4 and 2.1 — it was marked 2.1 on the label of the disc, but when booted-up, it turned out to be version 2.0. Hereby hangs a rather sorry story. Version 2.0 is known to be unreliable and Rostronics does not recommend its use. Certainly I had some intermittent bad track errors from drive A only when using V2.0, which do not occur when using 1.4.

However, using CP/M 2.0 very sparingly, I found that the access time from the floppy discs when using double-density format is speeded by a factor of approximatley 3.3; i.e., the time taken to load the same version of 24K MBasic (V.5.03) was nine seconds for CP/M 2.0 and 30 seconds for CP/M 1.4.

The operating system for the Z-Plus is CP/M. This operating system has been reviewed numerous times with various other systems, in particular version 1.4 — see the Rair Black Box review *Practical Computing*, November, 1979.

Improvements

Version 2.0 is claimed to offer some improvements over 1.4, other than the disc access times mentioned. The improvements are, briefly, the ability to partition disc areas into user areas — that is not the same as multi-user. It merely allows user numbers 0 to 15, and different-numbered users cannot access other disc files. That is not very exciting and hardly essential. It might be useful for larger-capacity discs, but not really for only 1MB.

It is not that difficult for a new user to identify himself as any of the 0 to 15 numbers to access any file. The user number is not the same as logging-on or the use of a secret password for security.

File security

There are improvements to the STAT, PIP and ED commands, some to take care of the partitioning into user areas, but there are also more worth-while enhancements such as making individual files read only (R/O), i.e., write-protected. By declaring them to be system files ($SYS), their names will not be listed on a DIRectory enquiry — useful for security of files against casual enquiries.

CP/M in either version offers sensible and useful facilities for development work, which may not be everyone's ideal, but it is one of the better operating systems I have encountered for micro-computers. As with the Rair Black Box, this system is also very disc-dependent; without a system disc, the whole system is useless, as there is no other monitor or execute in ROM to be booted-up as an alternative.

Although this is not disastrous, it is not desirable. What happens, for example, if there is a fault with the disc drives and the discs will not boot-up? At least with a monitor in ROM, some work can be done and perhaps a diagnostic program entered and run to test the drives. So it is imperative to back-up — duplicate — the system discs in case of accidents.

CP/M supports several languages. Rostronics suggests CIS COBOL, but a review copy was never delivered — a pity, because the useful screen-formatting facilities offered by this version of Cobol by Micro Focus are intended specifically for interactive usage. What was supplied with the numerous discs were two inevitable versions of Basic.

MBasic 5.03 is an interpreter by Microsoft and there was CBasic by Software Systems. Running the standard benchmarks for MBasic showed that it was reasonably quick, in the same class as the Research Machines 380-Z and the Nascom 2 — 4MHz with one wait state. MBasic is well known so details will not be given. Version 5.03 is claimed to be free of some of the bugs of previous versions — none of which could have been that major. It does now have TRON and TROFF, line tracing facilities, for testing of programs.

CBasic 2.05 is totally different; it is, in fact, a compiler/interpreter version of Basic. That means code is not executed immediately, but needs preparation via a text editor, e.g., the CP/M ED, and submitted for compilation. The compilation checks for syntax errors, and creates intermediate code for the run-time interpreter to execute. That should in theory give more compact code than the usual Basic interpreters, and also run faster.

Not so in this case. Two of the benchmarks, BMK7 and 8, were compiled and run to see how they compared to the MBasic interpreter. The results for the two benchmarks were astonishingly slow: the timings were in the order of 2.8 and 12 times slower for BMK7 and 8 respectively. So much for theory.

Advantage

The main advantage of CBasic, I think, lies in its more versatile file-handling, which may make it better suited for business programming, but for other normal uses there does not seem to be any particular advantage in using CBasic. In fact, it involves a good deal more fuss than a straight interpreter Basic. Writers of business programs like to conceal their listings from competitors, hence the popularity of a slow compiled Basic.

The compiler part of CBasic seemed reasonably efficient, taking only nine seconds to load, under CP/M 1.4, and about six seconds to compile one of the benchmark programs of about 13 lines. A good feature is that a trace of the program can be taken during run-time, provided a toggle for this requirement is set during compile time.

Applications

Since this system is clearly aimed at the small business computer market, it is its applications which will interest the customer. Rostronics supplied two demonstration application business programs. One was an application, still under development, of stock control using compiled Basic. The other was a demonstration of sales ordering where CIS Cobol was used.

Both seemed workable by a user with intelligence, but what both lacked was any help facilities, so users would have to refer to manuals to use these programs — neither had manuals for this review. Both featured screen formatting and on-line entry of data, and the sales-ordering demonstration used secret security passwords.

Another application submitted for review was much more interesting — the word processing facility using the now well-known Wordstar package produced by the U.S. company Micropro International. Wordstar on the Z-Plus system does all that a layman would expect. The 15in. screen with the mentioned high-quality display also added to the pleasure of using Wordstar and, therefore, was a good choice by Rostronics.

Wordstar allows the creation of a document file or the editing of one — so far not much different from an ordinary text editor, but it is the useful screen editing features which make it a joy to use. The cursor can be moved anywhere in the text to manipulate, correct, insert or

Micromation

Micromation was founded three years ago by Ben Cooper, a physicist by training who established a company to design and manufacture video games, before graduating to computing. He launched Micromation with a design for a controller board and now has a staff of more than 30 and 15,000 sq.ft. of office and factory space. The company sells its micros and boards through 100 retailers and 70 OEM houses in the U.S., shipping $½ million's worth of equipment every month.

The main problem the company faces is its growth rate — a now all too familiar tale — but the company is entirely self-financed and most of the manufacturing is subcontracted and then tested in-house. At the West Coast Computer Faire, in March, Micromation launched a new multi-processing version of the Z-Plus based on five Z-80s with a 20MB hard disc and 300K of RAM giving each user his own CPU and 64K of RAM.

The U.S. price will be $16,000 for a four-terminal system. The vice-president, Loran Wiley, handles the company marketing. Micromation, 1620 Montgomery Street, San Francisco, California 94111, tel: 0101 415 398 0289.

move it. Blocks and whole files can be moved or appended or copied.

The page can be formatted to the required justification, i.e., the right-hand margin of the text can be made even — impossible on an ordinary typewriter. Characters or words can be highlighted by double-printing — or overprinting — or underlined.

Pages are numbered automatically, headings can be put on every page automatically. Top, bottom, left and right margins can be set. Line-lengths and formatting are automatically taken care of, so one types as if the text were on one continuous long line, using carriage return only when the line needs to end or a blank line inserted. These features make life so much easier. It is possible to obtain almost unlimited good clean top copies of documents.

Daisywheel printer

Some of those features depend on the printer used. Some of the described features, such as overprinting, micro-spacing to achieve the even right-hand margin and underlining can only be done on a word-processing printer, of which the Multiwriter is a good example. It has a good typeface and can do all the clever things that a word-processor like Wordstar requires of it. Its only disadvantage is speed. At approximately 45 cps it is typical of daisywheel printers, but at least it makes most of its type speed by buffering characters, bi-directional printing and printing only the length of line required.

There are a few things which other word-processors can do that Wordstar on Z-Plus cannot, such as inter-machine communications for the future in electronic mail, multi-user access and multi-tasking, but these do not affect it too much if used only as a stand-alone dedicated system. A criticism was the use of control keys for the cursor movement instead of the cursor keys.

I suppose that since there is no widely-accepted standard for cursor movement characters, Micropro chose control keys which were compatible with almost any

Summary specifications

CPU — Z-80 at MHz.

Memory — 64Kbytes of 4116 dynamic RAMs at full 4MHz with no wait states.

Mass storage — Twin 8in. floppy disc drives support both single- and double-density, soft-sectored formats — 1MB total on two drives recording double-density.

I/O — Two serial RS232 ports to support the VDU and printers and two parallel ports.

VDU — Elbit 1920 series with 15in. screen and detachable 96-key keyboard.

Operating System — CP/M 2.1 should be supplied, but 1.4 can also be supplied.

terminal. Nevertheless, it was frustrating to have the cursor keys but not be able to use them. A minor bug was that the "control-J" (or J) command did not work. It is intended to set and display the help levels at any time, so one had to set it at the beginning just after calling up Wordstar. It was still advantageous to be able to call a help menu on the screen as a reminder of the commands available.

Power supply

A quick look inside the CPU box showed it was neatly laid-out. The massive power supply constitutes most of the weight. The floppy-disc drives are laid horizontally — not a very good idea, as it gathers dust more easily than vertical drives. The logic is on only three standard S-100-sized cards, one for the floppy-disc controller, one for the I/Os and one containing the Z-80 and the full 64K of RAM. The RAMs were 4116 16K dynamic RAMs and there were 32 of them packed on to the single board.

I have some reservations about the wisdom of this, questions about the buffering of signals to and from the RAMs, and if the memory refresh is the unmodified RAM refresh signal from the Z-80, any DMA (direct memory access) operation could affect the refresh, and, therefore, the integrity, of the RAMs adversely. To be fair, I had no problems with the hardware, other than the occasional non-response when hitting the re-set button. A repeat press usually cured this, and it was a fairly rare occurrence; so this is a commendable performance.

Most of the documentation was standard from Digital Research on CP/M and Microsoft on MBasic. They have all the essentials but tend to be a little dry, but are fairly good for references. The manual for Wordstar was reasonably good but again very much in the style of the others mentioned, with perhaps a little more detail, but too few examples. The CBasic manual was very brief but workable, but again lacking in examples. Perhaps an overview of advantages in using CBasic and the concept behind it would not have gone amiss.

An operation manual for the Z-Plus system was supplied in draft form, on a document file for Wordstar on disc. In this situation, one has to know how to operate the system before being able to access the manual. Obviously Rostronics intends to print-out the manual on paper for their customers. This manual was brief but to the point.

Conclusions

- **The Z-Plus System is a well-known combination of components.**
- **The system is suitable for business purposes — more than can be said of some systems claiming to be for business applications.**
- **Wordstar, the word-processing package, is good, but one needs a good and expensive printer to obtain the full benefit from it.**
- **I liked the system very much, with its CPU box with twin floppies and the quality Elbit 15in. VDU.**
- **For £3,950 in a desk configuration, with CP/M and a Basic, it is good value for money.**

UK101 is ideal for hobbyist and kit-builder

THE COMPUKIT UK101 is a single-board computer supplied either as a kit or ready assembled. Included on the board is a memory-mapped VDU an upper- and lower-case QWERTY keyboard, up to 8K RAM — the kit is supplied with 4K RAM — a cassette interface using the Kansas City standard, and a 2K monitor and 8K Basic in EPROM.

The VDU controller has 1K of memory-mapped RAM and an ASTEC UHF modulator with co-axial socket into which your TV aerial may plug directly. The VDU will display up to 16 lines of 48 characters. When using Basic, the line width may be set to any value between 16 and 48. Compukit advises that you normally set the width to 46 as the last two characters will be lost from the edge of the screen if you attempt to display 48 characters to a line. I certainly found that to be the case on my television.

Character set

The display was excellent for this type of system. With the television tuned correctly, there was a slight wave motion on the screen, which could have been due to the television. Also there was some flickering, when a lot of rapidly-changing information was displayed. That was noticeable when running Hectic, one of the games supplied with our evaluation model.

The character set includes the full upper- and lower-case ASCII characters, and a range of special and graphic characters including Greek letters. Most of the characters are reasonably well defined and easy to read, although some of the Greek characters appear too complex to be displayed clearly.

The full depth of line is used for displaying some of the characters, so there are no gaps between lines of these characters. This is fine for certain types of graphic display but it means that lower-case text can run together as is the case

by Martin Collins

when a 'y' or 'g' on one line coincides with an 'l' or 'h' on the next.

The keyboard is in the standard QWERTY lay-out with the numerals along the top, and special keys for control, rub-out, return shaft and shift lock. There are two re-set keys on the right of the keyboard. The re-set keys must be pressed simultaneously to initialise the system which reduces the chances of accidentally re-setting the system. However, as the keys are both together, it is still possible to do that accidentally. To prevent it altogether, the keys should have been placed further apart.

The keys have a positive feel. If a key is held down for longer than a second, the character is repeated until the key is released. The key switches are fitted closely to each other, and as the manual warns you in the assembly instructions, you must be careful not to misalign the keys so that they bind against one another.

I had a problem when I first attempted to use our model; I followed the instructions on initial use of the system and was able to cold-start Basic — a cold start is employed when initially using the system, a warm start if you wish to return to Basic having used the monitor and already have a program loaded. When you cold-start Basic, the system requests the amount of memory to be allocated as Basic workspace, and the terminal width to be used.

When I first used the system, I was unable to respond to these prompts because the machine kept displaying rows of zeros. I eventually realised that there must be touching contacts on the zero key, and I was able to cure the fault by fiddling with the key until the zeros stopped.

Shift lock

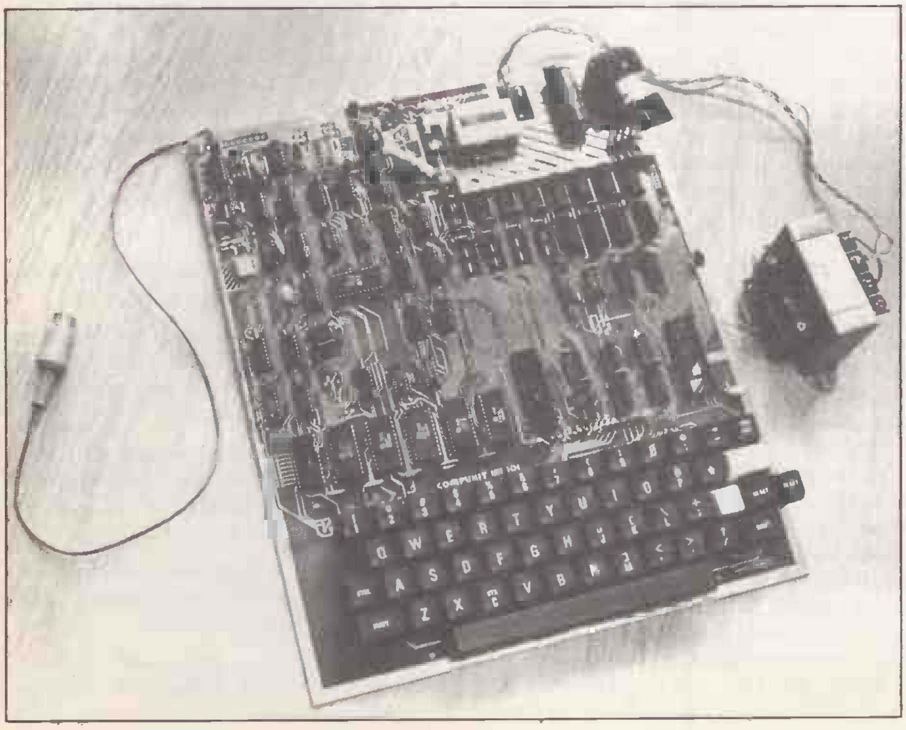
Another criticism I have concerns the use of the shift and shift lock keys. As well as both upper- and lower-case ASCII characters, the keyboard will generate the normal characters seen on typewriters and some of the graphics characters. On most keyboards, shift lock affects only the alphabetical characters, giving upper-case when the key is pressed down and lower-case when the key is in the up position. Numeric and special keys are only affected by the shift keys. On the UK101, however, shift lock affects all the keys which means it is possible to type only special characters like !, ", £, or by pressing shift with shift lock down. So typing lower-case text means continually pressing and releasing the shift lock key, hitting rub-out with shift lock up produces " ← " and hitting return gives the character "M" which is also rather inconvenient.

The keyboard is software-scanned which means the system continually checks each row of keys to see if any keys have been pressed, and it is possible to detect which key has been pressed within a program. Special keyboard functions can, therefore, be programmed which is useful for games programs.

I found the cassette interface straight-forward to use. Compukit supplied a tape with two Basic games; Taxi and Hectic, on one side, and an extended Monitor on the other. I had no problems loading any of these programs.

The manual supplied with the UK101 includes a very clear description of all the components and general hints on how to assemble the system. It doesn't include the

(continued on page 61)



(continued from page 58)

detailed step-by-step instructions you have with some kits, yet all but the complete beginner should find assembling the system easy. Beginners should, of course, practice on a simpler kit before attempting to assemble any microcomputer.

Having assembled the machine, I first switched-on to find an incoherent display and no response from the keyboard. Careful checking of the system revealed no obvious problems, so we telephoned Compshop who suggested that we took it in so that the board could be checked. We did so and within a few minutes of arriving, they had located and replaced a faulty IC.

The system is provided with a machine code monitor and 8K Microsoft Basic in EPROM. The monitor allows the user to examine areas of memory, input machine code routines, load machine code from tape, and start execution of a program from any point in memory. There is no facility within the monitor to save programs on tape although the user manual gives a routine to perform this function.

The Basic features two character-variable names — giving a possible 962 numeric variables. String variables are the same except they must be followed by an 8 sign. Numeric variables may be in the range 10 ± 38, but only the six most significant digits may be displayed as is standard for most implementations of Basic. Strings may be up to 255 characters long.

Both string and numeric arrays may have up to four dimensions. Although the maximum line width for the VDU is 48 characters, Basic lines may be up to 71 characters long. When you reach the edge of the screen, keep typing and the line will continue below. Line numbers may go from 1 to 63999, and several statements separated by ":" may be included in our program line.

The Basic commands include LIST, RUN, CLEAR — to set variables back to zero — NEW, CONT, LOAD and SAVE. These commands may be included in a program, if you wish, allowing the program to control the system, although I think this has a limited application. LOAD and SAVE are used to load and save programs using the cassette. I missed the presence of a DELETE command to allow blocks of lines to be deleted, and no edit command. Lines may only be deleted by re-typing the line number and altered by re-typing the complete line.

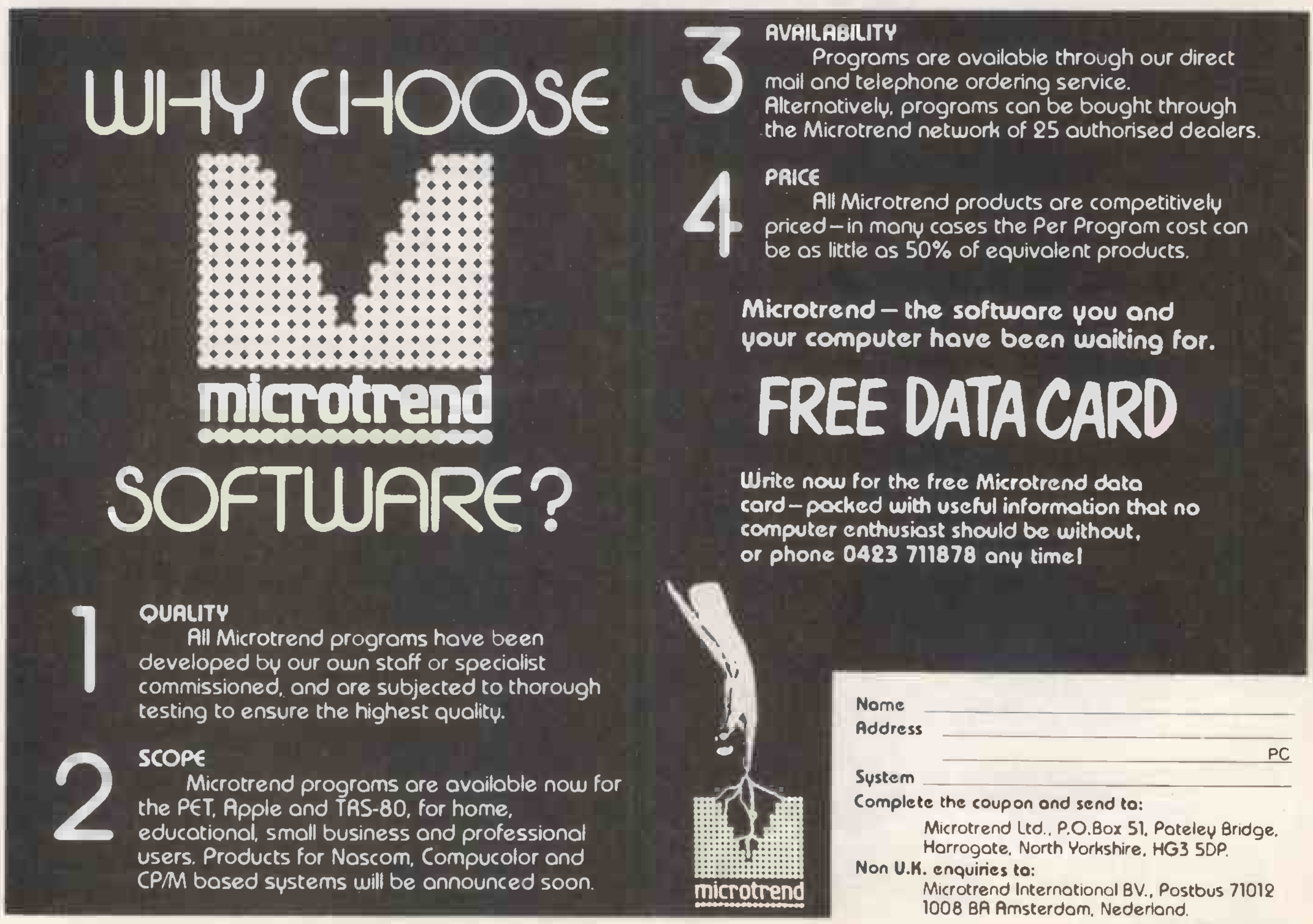
Basic lines and multiple statement lines may be executed in immediate mode. That means they will be executed immediately they have been typed-in, by typing the line with no line number. The multiple statement line feature is quite useful here, allowing FOR loops to be performed in immediate mode, for instance. Several versions of Basic only permit single statements to be executed in immediate mode.

All the usual Basic verbs and functions are implemented, including READ, DATA, RESTORE, INPUT, PRINT, DEF FN, DIM, FOR-NEXT-STEP, IF-THEN, GOTO, GOSUB, ON-GOTO, ON-GOSUB, and POKE amongst the

(continued on page 63)

UK101 background

The Compukit UK101 is an offshoot of the Ohio Superboard. In March 1979, Dr Anthony Berk was asked by Compshop to design a kit, similar to the Superboard, which could be produced in the U.K. He re-designed the hardware while Andy Fisher, of Compshop, adapted the software. The first prototype of the UK101 was prepared by June 1979 and the kits should have been available to the public by the middle of the summer. Compshop had been let down by their supplier of ROMs and the first kits were despatched incomplete — the ROMs were forwarded several weeks later. Dr Berk works as a hardware and software consultant with Modus Systems Ltd, in Letchworth, Hertfordshire. He also lectures on microcomputing at Brunel University, Uxbridge, and is designing a small educational kit, at the bottom end of the market, which he hopes to sell for less than £30.



• Circle No. 174

◀ • Circle No. 173

(continued from page 61)
verbs. The functions include FRE, PEEK, USR, ASC, CHR8, MID8, LEFT8, RIGHT8, LEN8, SIN, COS, and TAN. PEEK, POKE and USR enable the programmer to examine and alter bytes of data and to call machine code routines from a Basic program. When a cold start is executed the system asks for the amount of memory available to Basic. That enables one to protect the top part of RAM from being over-written by Basic programs or data. So a machine code routine may first be input to this part of memory using the monitor, then a Basic program may be loaded which uses the routine.

As can be seen, the Basic is reasonably powerful, the only other criticism I have, apart from the lack of DELETE and EDIT commands concerns the lack of any method within a Basic program for reading or writing data to a cassette.

Variation

The only way to do this is to write machine code routines to be called within the program. I would have thought it simple to include a variation of the INPUT and PRINT commands to access data on a cassette.

The documentation with the UK101 provides only the essentials. It consists of a single 48-page manual describing the circuitry, construction details, and giving an introduction to Basic and the monitor. As such, it is well written and easy to follow — anyone with a fundamental knowledge of computers and electronics should find it quite adequate.

At the back of the manual are some useful tables including the UK101 memory map and the 6502 machine code and architecture. The sections on Basic and the machine code are there only as quick reference guides. As it states in the manual, an introduction to Basic should be read if you do not already have any experience with the language. Similarly with the machine code, which is particularly difficult to understand unless you are experienced in machine code programming.

The advertised expansion capability of the system includes: an additional 4K RAM on board; an RS232 interface for a printer; a further 16K RAM; floppy discs, S-100 bus; colour display.

The first three are available, the additional 4K RAM on board costs £49 — sockets for it are included in the kit. The RS232 interface requires a few extra components, and instructions for adding it are included in the manual. The 16K RAM can be from an Ohio Scientific board or from a board specifically designed for the UK101. In either case the cost is around £150.

Further expansion seems to be rather restricted. The introduction to the manual states that floppy discs plug-in directly to the board, but the Compshop says that it would not recommend the use of floppy discs with the UK101. The original intention was to use the Ohio Scientific discs. They were made to work with the first version of the UK101 but apparently do not work with the current version. The Compshop suggests that the Ohio scientific controller has been changed since the UK101 was designed. The Compshop does not recommend the UK101 to anyone who is likely to want to expand their system to use floppy discs.

The colour board, which was due to be available in November 1979, has also run into problems. It has now been developed but not released.

Conclusions

- **The UK101 seems to be an excellent machine for the hobbyist and kit-builder.**
- **It is good for games programs and educational use.**
- **Unfortunately, it is limited to these applications by the lack of facilities for mass storage of data.**
- **The Basic is good but has some restrictions, chiefly the absence of DELETE and EDIT commands, and I/O commands for storing data on the cassette.**
- **It appears very difficult to expand the system.**
- **Some of the expansion capabilities mentioned are not yet available or no longer available.**

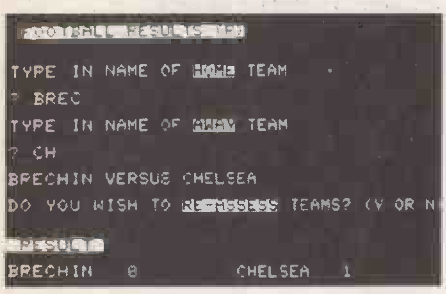
Nascom 2.

The review of the Nascom 2, April 1980, included photographs of the assembled kit. These were taken at Henry's, Edgware Road, London W2.

Search for winners is backed by micro

Gambling has exercised a powerful fascination throughout history and our attempt to devise a system for winning using the micro is only the latest in a long series. Rex Tingey presents logically correct programs for football, greyhound and horse racing forecasts and illustrates the principle of multiple data banking.

MULTIPLE data banking is the method of using groups of stored data which each contain as a single item, two or more items which are in some way inter-related, each item being as complex as required within a total maximum possibility of approximately 72 symbols in Pet Basic. Approximately because the 80 possibilities on line 10 consists of Space-10-Space-DATA 72



symbols maximum. Step numbers larger than 99 reduce the possibilities. A unit of multiple data can consist, for example, of the name of a chemical element, its symbol, and its atomic number together with no separation. In the programs, the multiple data banking contains the English and Scottish Football League teams, each with a figure for its last season's performance, plus a figure for its last season's away performance. For example, IPSWICH 628 means that Ipswich is allocated a scale score of 62 for overall performance, and 8 out of 9 for away performance — which was good. The numbers are extracted, in this case, by taking out a complete unit of multiple data as a string assigned a definite string variable, and then using the usual string functions to remove and use the values required. The length of the team names can, therefore, vary without affecting anything else.

The major advantages of the multiple data bank unit are that it is economical, it is easy to program as all the parts are one, and that it is easy to find and extract, for the same reason. In the example, the data is particularly easy to extract by the user since all the names are different and at the left. It means that input to extract data need only be enough to identify the left end, but with priority for being first. Thus an input of B will always extract BRISTOL C, BI will always find BIRMINGHAM, but to find Bristol Rovers the full BRISTOL R must be input. BREC will extract Brechin right from the end of the bank as BREC can only mean Brechin.

The method of initial extraction of data is to measure the LENgth of the user's input and to loop through all the data, breaking-off the first same-size length from each unit and comparing it to the string of input. If it is the same, the loop is forsaken and the next part of the program entered. In Pet Basic, spaces and some punctuation marks are allowed in data without the necessity of using double quotes around each unit — regardless of what the handbook states — some other Basics do not allow this.

Two versions of the program are given. The first version uses the standard data bank wherein those lines are indicated by DATA, and the units are separated by commas — if a unit contains a comma it must be enclosed in double quotes. In the second version, the data is assigned a string variable AS(A), and A varying from 0 to 10 in groups. That is a convenient way of assembling data which is very complex, or which requires a definite address within a program. When each unit of data has to be checked against input it has the disadvantage of taking longer to operate; up to 9 seconds for data in the Scottish League as opposed to under 1 second for the other version.

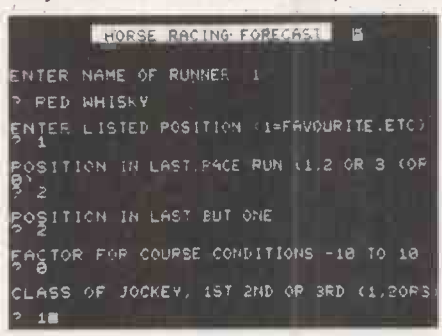
It is interesting to note, however, that if the strings are numbered sequentially by re-DIMensioning AS, the find and extract time increases by a factor of about × 12.

When the multiple data is short and limited, as in this case, with the pro-

gram occupying little more than half of 8K, it is convenient to use a series of assignments for the string variables when manipulating the data, in either version. In programs handling lengthy and complex data, this convenience will result in "Out of memory" errors, due to the great amount of memory space used in running and building string after string. To avoid this, keep re-using the same assignments, making sure that the manipulations have first extracted all the required information before re-using.

The data was formulated from the League tables for the season 1978-79, and a figure was found for each team, assessed from table position, division and performance, with a possible range of 01 to 99. Then the away performance was assessed with a range of 1 to 9, and this figure is used to modify the normal performance figure whenever that team is playing away.

When the home and away teams have been selected by input, their assessed figures are compared, and if the values are close enough a drawn game results, a random score is selected, and the result screened. If there is a sufficient discrepancy between the two values, a home



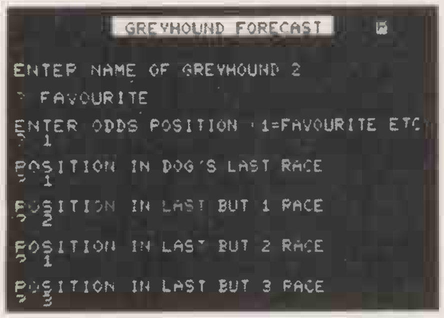
or away result is screened to correspond with the discrepancy, also with a random selected score.

To introduce a small element of chance, the program allows a random feature to be selected, if required, which may modify the result by altering one of the performance figures unpredictably. Further, the user can input updated information about each team's recent performance record as above or below average. Following this, the user is requested for an input regarding the team's strength, to take into account injuries, suspensions or substitutions.

So that an error condition is avoided should a team's name be incorrectly spelt, and, therefore, not be found in the listed data, the program goes to 9000 whenever the condition occurs and offers a choice of two dummy teams "AA" and "BB". These can also be used as substitutes for non-league teams which may play from time to time.

Selection of a team is by means of the Input ZS, then entering the loop at 400 which reads through the data as AS, assigned CS for manipulation, and whenever ZS is matched, the pointer falls out of the looping to 4000. If the ZS is not matched, line 2000 directs to 9000. Line 4000 checks if the information is "home" or "away", and if it is "home" then CS become DS, and the pointer returns to pick up the "away" input, which remains as CS as B = 1. Notice that data is Restored before both of these pickups. If desired, CREWE at home can be played against CREWE away; without the Restores the program would otherwise Goto 9000.

The values are extracted from the data, and the home versus away teams displayed. The data in use is no longer



multiple but consists of the home team name ES and value D, and away team FS with its corrected value H.

If random results have been initially selected then the value of one team, the away team, is affected, as there is no point in altering both figures. The random number generator produces a figure between 0 and 22, corrected to be between —4 and 18, which, when added to the away score, produces a tendency towards more draws and away wins. The tendency is small because the home-win bias is already large, as in real results. The random altered result still has the core produced in the random fashion, to conform to the calculated result. Whether random or not, the results produced in the calculating sections are subject to one of two small adjustments, to conform to the pattern of normal results.

A section of the string data version of the program is given to illustrate this method of moving multiple data banked in this way. The subroutine 2000 is the check to find a match with ZS. Notice the ON Z count through, directing the pointer to the next set of addresses of the data stored as strings, in sets of 11. It can be seen that if complex data has to be readily accessed, time after time, but without identity other than an address, then this method is the best way to store such complexities. Data statements have limits in Restore, where several areas of data exist with no real boundaries. Running through data, and then having a need for a particular item of back data to be collected before returning to the run through can cause serious programming problems. The string multiple data statements simplify this area of programming, taking up more memory in the process.

Both programs employ the same principles of calculation and assessments. The 1979-80 League Results are soon to be published and new assessments will need to be made and programmed for the teams. You may care to devise your own methods of assessment and results calculations.

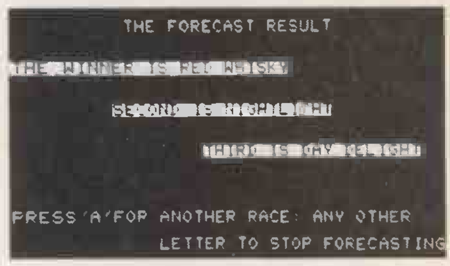
Two programs, Racing Forecast and Greyhound Forecast, give complete computed results of races, dependent upon accurate assessments being completed by the user. Both programs give a particular insight into multiple data banking; this time in the form of assigned string data held only in memory.

Data which is formed during the run of a program can, of course, be placed against DATA statements allocated step numbers by the program during the Run, and then used as data, but this is really only suitable for progressively-updated programs to be Saved, after each completed updating. If data formed during the run of the program is to be used later, for just that run, and then Cleared — so that further use can be made of the program held without re-Running or re-Loading — then all complex data must be held in memory against assigned string variables. Single values of data can be assigned to letters or letter/numbers, such as J2 or JJ8, giving vast possibilities of storage within memory. String variables can be re-Dimensioned, with new ROMs in the Pet, so that AS(B), for example, can now be expanded to store quantities of multiple data above that of 256, the old limit, to the extremes of available memory.

AS(B) can be used in a program with increasing values of B to form chained strings, so that if a string is formed when B = 0, that string is duplicated at that one time, in memory, as AS(B) and AS(0). On the count-on of B = 1 and the formation of a new string, AS(0) remains but AS(B) equals AS(1), and so on, forming a chain of associated strings with its ex-B values available for a count-through of data, which is what is required. The count-through of data, however, does not now have to be AS(B) but can be, say, AS(Y), with a changing value of Y and still access the data.

If your particular brand of Basic cannot handle this complexity, tending to lose parts of mixed data, with the method, a more positive method is available where AS(0), AS(1) etc. are programmed as stops with an address, and when a multiple data string is required to be formed, the data is sent to one of the string-forming step-set, before re-direction. This technique is shown in the Greyhound Result program.

The Racing Forecast program is designed to give the possible result of a horse race, calculated from various known factors entered for each runner. The minimum number of runners to be entered is three. If a race with only two runners is to be forecast, then a "dummy" horse with a low assessed score must be entered as well as the two runners, or an error will result. The



maximum number of runners which can be programmed to run in a race is 25, as the program stands, but it can be re-Dimensioned for another maximum number, if required.

To introduce the element of chance, a random number can be selected to be used to vary the value of the first assessment entry, which is the odds position. It can be altered to affect a different assessment simply by causing it to be added to an input other than B. The random element

(continued on next page)

An alternative method of storing data.

```
350 B=1:PRINT"TYPE IN NAME OF AWAY TEAM":INPUTZ$:L=LEN(Z$)
400 A=-1
410 Z=Z+1:ONZGOTO480,580,680,780,880,980,1080,1180,1280,1380,1480,1580,9000
480 A=A+1:IFA>10GOTO400
500 A$(0)="LIVERPOOL917":A$(1)="NOTT'M F799":A$(2)="WEST BROM748"
510 A$(3)="EVERTON664":A$(4)="LEEDS636":A$(5)="IPSWICH628":A$(6)="ARSENAL615"
520 A$(7)="ASTON VILLA599":A$(8)="MAN UTD577":A$(9)="COVENTRY563"
522 A$(10)="TOTTENHAM539":GOSUB20000:GOTO480
580 A=A+1:IFA>10GOTO400
600 A$(0)="MIDDLESBRO525":A$(1)="BRISTOL C524":A$(2)="SOUTH'PTON523"
610 A$(3)="MAN CITY494":A$(4)="NORWICH471":A$(5)="BOLTON462"
620 A$(6)="WOLVES453":A$(7)="DERBY423":A$(8)="Q P R365":A$(9)="CHELSEA317"
622 A$(10)="BIRMINGHAM332":GOSUB20000:GOTO580
680 A=A+1:IFA>10GOTO400
700 A$(0)="CRYSTAL P636":A$(1)="BRIGHTON604":A$(2)="STOKE618"
710 A$(3)="SUNDERLAND597":A$(4)="WEST HAM545":A$(5)="NOTTS C468"
720 A$(6)="PRESTON447":A$(7)="NEWCASTLE443":A$(8)="CARDIFF443"
722 A$(9)="FULHAM433":A$(10)="ORIENT424":GOSUB20000:GOTO680
780 A=A+1:IFA>10GOTO400
800 A$(0)="CAMB'GE U427":A$(1)="BURNLEY423":A$(2)="OLDHAM413":A$(3)="LUTON382"
810 A$(4)="WREXHAM402":A$(5)="BRISTOL R404":A$(6)="LEICESTER394"
820 A$(7)="CHARLTON378":A$(8)="SHEFF UTD352":A$(9)="MILLWALL336"
822 A$(10)="BLACKBURN319":GOSUB20000:GOTO780
880 A=A+1:IFA>10GOTO400
900 A$(0)="SHREWSBURY585":A$(1)="WATFORD566":A$(2)="SWANSEA565"
910 A$(3)="GILLINGHAM554":A$(4)="SWINDON525":A$(5)="CARLISLE474"
920 A$(6)="COLCHESTER463":A$(7)="HULL446":A$(8)="EXETER432"
922 A$(9)="BRENTFORD414":A$(10)="OXFORD UTD404":GOSUB20000:GOTO880
980 A=A+1:IFA>10GOTO400
1000 A$(0)="BLACKPOOL395":A$(1)="SOUTHEND394":A$(2)="SHEFF WED394"
1010 A$(3)="PLYMOUTH384":A$(4)="CHESTER383":A$(5)="ROTHERHAM383"
```

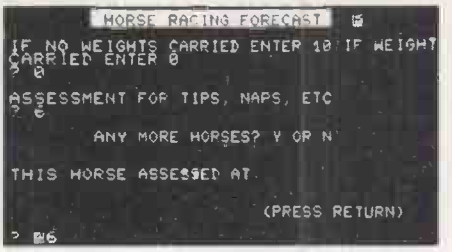
```
FOOTBALL RESULTS PROGRAMME

30 REM:COPYRIGHT (C) REX L TINGEY - JAN 1980
100 GOSUB10000:PRINT"THIS PROGRAMME GIVES COMPUTED FOOTBALL
104 PRINT"RESULTS IN TWO INTERWORKING MODES.
108 PRINT"PRESS Q TO EXCLUDE SMALL RANDOM FEATURE
112 PRINT"PRESS ANY OTHER LETTER TO INCLUDE IT (R)
116 PRINT"FORM IS BASED ON LAST SEASON: YOU MAY
118 PRINT"RE-ASSESS A TEAM'S PERFORMANCE/STRENGTH
120 PRINT"OR LEAVE THE ASSESSMENT AS IT STANDS.
122 PRINT"________________________________________
128 PRINT"ENTER TEAMS AS ON LIST, NO FULL STOPS.
140 PRINT"PRESS ANY LETTER OR Q.
142 GETZ$:IFZ$=""GOTO142
150 IFZ$="Q"THENQ=1
200 RESTORE:GOSUB10000:B=0:IFQ=0THENPRINTTAB(18)"(R)"
250 PRINT"TYPE IN NAME OF HOME TEAM":INPUTZ$:L=LEN(Z$):GOTO400
350 RESTORE:B=1:PRINT"TYPE IN NAME OF AWAY TEAM":INPUTZ$:L=LEN(Z$)
400 FORA=1TO132:READA$:C$=A$:IFZ$=LEFT$(C$,L)GOTO4000
410 NEXT
500 DATALIVERPOOL917,NOTT'M F799,WEST BROM748,EVERTON664,LEEDS636,IPSWICH628
520 DATAASTON VILLA599,MAN UTD577,COVENTRY563,ARSENAL615,TOTTENHAM539
600 DATAMIDDLESBRO525,BRISTOL C524,SOUTH'PTON523,BIRMINGHAM332,WOLVES453
610 DATAMAN CITY494,NORWICH471,BOLTON462,DERBY423,Q P R365,CHELSEA317
700 DATACRYSTAL P636,BRIGHTON604,STOKE618,PRESTON447,NEWCASTLE443,CARDIFF443
710 DATASUNDERLAND597,WEST HAM545,NOTTS C468,FULHAM433,ORIENT424
800 DATACAMB'GE U427,BURNLEY423,OLDHAM413,LUTON382,BLACKBURN319,SHREWSBURY585
810 DATAWREXHAM402,BRISTOL R404,LEICESTER394,WATFORD566,SWANSEA565
820 DATACHARLTON378,SHEFF UTD352,MILLWALL336,CHESTER463,HULL446,EXETER432
910 DATAGILLINGHAM554,SWINDON525,CARLISLE474,BRENTFORD414,OXFORD UTD404
1000 DATABLACKPOOL395,SOUTHEND394,SHEFF WED394,PLYMOUTH384,CHESTER383
1020 DATAMANSFIELD377,BURY368,CHEST'FIELD343,ROTHERHAM383,PETERBORO294
1100 DATATRANMERE215,LINCOLN184,READING574,WALSALL254,NEWPORT408,HUDD'FIELD344
1110 DATAGRIMSBY507,WIMBLEDON504,BARNSLEY506,YORK346,TORQUAY334,SCUNTHORPE324
1120 DATAALDERSHOT463,WIGAN A435,PORTSMOUTH405,NORTH'PTON263,ROCHDALE254
1210 DATAHARTLEPOOL319,HEREFORD303,BRADFORD C305,DARLINGTON234,HALIFAX123
1220 DATAPORT VALE298,STOCKPORT273,BOURN'M'TH263,CREWE129,DONCASTER236
1320 DATADUNDEE558,ABERDEEN534,HIBERNIAN497,CELTIC626,RANGERS595,ST MIRREN489
1400 DATAPARTICK463,HEARTS346,MOTHERWELL289,MORTON483,DUMBARTON366,CLYDE316
1410 DATADUNDEE U585,KILMARNOCK547,STIRLING A329,CLYDEBANK536,MEADOWB'K106
1420 DATAAYR UTD458,HAMILTON403,AIRDRIE378,ARBROATH304,RAITH295,BERWICK449
1510 DATAS JOHNST'NE287,MONTROSE219,Q OF SOUTH201,DUNF'MLINE425,FALKIRK395
1600 DATAALLOA286,AA304,ALBION R277,STRANRAER256,EAST FIFE327,COWDENB'TH306
1610 DATABB607,FORFAR254,BRECHIN181,STENH'MUIR199,E STIRLING182,QUEENS P149
2000 IFZ$<>LEFT$(C$,L)GOTO9000
4000 IFB=1GOTO4020
4010 D$=C$:GOTO350
4020 PRINT"
4022 L=LEN(D$):E$=LEFT$(D$,L-3):L=LEN(C$):F$=LEFT$(C$,L-3)
4030 PRINTE$SPC(1)"VERSUS"SPC(1)F$
5000 G$=RIGHT$(D$,3):D$=LEFT$(G$,2):G$=RIGHT$(C$,3):H$=LEFT$(G$,2)
5010 C$=RIGHT$(G$,1):D=VAL(D$):H=VAL(H$):C=VAL(C$):C=C/10:H=H*C
5018 PRINT"DO YOU WISH TO RE-ASSESS TEAMS? (Y OR N)
5030 GETY$:IFY$=""GOTO5030
5032 IFY$="Y"THENGOSUB8500
5040 IFQ=1GOTO6000
5500 Y=INT(RND(.5)*23):Y=Y-4:H=H+Y
6000 H=INT(H/10):D=INT(D/10):H=H+1:IFH=DGOTO6200
6020 IFH>DGOTO7000
6030 IFH<DGOTO6500
6200 H=INT(RND(.5)*12):IFH=11GOTO6220
6202 IFH>2GOTO6200
6204 GOTO6230
6220 H=3
6230 D=H:GOTO8000
6500 D=INT(RND(.6)*4)+1:H=INT(RND(.5)*2):IFH>=DGOTO6500
6520 GOTO8000
7000 H=INT(RND(.6)*4)+1:D=INT(RND(.5)*2):IFD>=HGOTO7000
8000 TI$="000000":PRINT" RESULT ":PRINTE$SPC(2)D,F$SPC(2)H
8200 IFTI$<"000006"GOTO8200
8300 GOTO200
8500 PRINT""E$" REASSESSMENT ":GOSUB30000:INPUTK:K=K-10:D=D+K
8760 GOSUB30100:INPUTK:K=K-10:D=D+K
8800 PRINT""F$" REASSESSMENT ":GOSUB30000:INPUTK:K=K-10:H=H+K
8960 GOSUB30100:INPUTK:K=K-10:H=H+K:RETURN
9000 PRINT"YOUR TEAM IS NOT ON MY LIST, CHECK YOUR
9010 PRINT"SPELLING.IF THE TEAM IS NOT IN THE SCOTS
9016 PRINT"OR ENGLISH LEAGUE THEN CALL THE TEAM:
9020 PRINT,"AA OR BB
9022 PRINT,"________
9030 PRINT,"AA = FAIR
9050 PRINT,"BB = GOOD
9070 PRINT," PRESS ANY LETTER "
9080 GETZ$:IFZ$=""GOTO9080
9090 GOTO200
10000 PRINTTAB(9)" FOOTBALL RESULTS ":RETURN
20000 C$=A$(A):IFZ$=LEFT$(C$,L)GOTO4000
20010 RETURN
30000 PRINT"TEAM RECORD THIS SEASON (0 TO 9 BELOW   AVERAGE,11 TO 20 ABOVE)
30010 RETURN
30100 PRINT"PRESENT TEAM STRENGTH DUE TO INJURY ETC.
30110 PRINT"(10=NORMAL STRENGTH)":RETURN
READY.
```

(continued from previous page)

can be avoided by selecting "Q" initially from the opening instructions, to enter the practical run. If the random element has been selected, a reverse field "R" appears after the main title block; this element of chance is introduced to all runners in that race.

Assessments are broad-based and can be extended to cover finer points such as, for example, how far a horse has been brought to run in the race since an owner would be unlikely to bring a horse from America unless it has a good chance of winning. You may thus wish to introduce your own set of assessments and calculations — the program is easily altered. A small random element works on the last assessment of whether the horse is tipped or fancied otherwise, which is itself a form



of chance; this can be avoided by entering 0 for all the runners as this assessment.

When assessment is complete, answer (Y or N) is required to the question "Any more horses?", but if 25 runners have been entered, the program requires N to be pressed, instructing the user to do so. The assessments for each runner are manipulated and added to form a single assessment, at line 760, of Y. Each Y value in the program has to be two symbols long, thus the value is checked for greater than 100 — a marvellous horse — and reduced to 99 if it is. The Y is checked for less than 1, and made up to 1. Now, if Y is less than 10 it is divided by 10. Thus 9 becomes .9, 2 becomes .2; two symbols. This is brought back × 10 later, after the two symbols have been removed from the end of the data string for the display of the name of the runner. The two-symbol requirement is to prevent the cut-off of the initial letter of the name with a figure less than 10, or having a figure on the head of the name with a Y in the hundreds.

The assessment value is displayed with the flashing cursor which enables the user to change the assessment, should he so desire, before pressing Return to convert the figure to YS input. Alternatively, this part of the program could be dropped and lines 830-850 changed to give YS = STRS(Y), a statement with the assessment not revealed.

A crucial program step at line 850 forms the multiple data bank of AS(J)s in memory, looping back each time to pick up the next runner's data, but if A1 = 9, either all runners or 25 runners have been entered and the pointer is directed to 900 where selection commences of the first, second and third place. Selection consists

(continued on page 69)

RACING FORECAST PROGRAMME

```
50 REM COPYRIGHT (C) REX L TINGEY JANUARY 1980
90 DIMA$(24):Z=0:GOSUB10000
100 PRINT"THIS PROGRAM FORECASTS THE OUTCOME OF
110 PRINT"A RACE WITH UP TO 25 RUNNERS.
130 PRINT"PRESS Q TO EXCLUDE THE RANDOM FEATURE
132 PRINT"FROM THE 1ST ASSESSMENT,ANY OTHER LETTER
134 PRINT"INCLUDES IT. IF RANDOM,SYMBOL R APPEARS
136 PRINT"AT THE END OF THE TITLE BLOCK.
150 GETZ$:IFZ$=""GOTO150
155 IFZ$="Q"THENA2=1
200 IFA1=9GOTO900
206 Z=Z+1:GOSUB10000:IFA2=1GOTO220
210 PRINTSPC(32)"R"
220 PRINT"ENTER NAME OF RUNNER "Z"":INPUTA$
240 PRINT"ENTER LISTED POSITION (1=FAVOURITE,ETC)":INPUTB
260 PRINT"POSITION IN LAST RACE RUN (1,2 OR 3 (OR 0)":INPUTC
280 PRINT"POSITION IN LAST BUT ONE":INPUTD
300 PRINT"FACTOR FOR COURSE CONDITIONS -10 TO 10":INPUTE
320 PRINT"CLASS OF JOCKEY, 1ST 2ND OR 3RD (1,2OR3)":INPUTF
342 GOSUB10000:IFA2=1GOTO350
344 PRINTSPC(32)"R"
350 PRINT"IF NO WEIGHTS CARRIED ENTER 10:IF WEIGHTCARRIED ENTER 0":INPUTG
370 PRINT"ASSESSMENT FOR TIPS, NAPS, ETC":INPUTH
380 IFZ=25THENPRINT"NO MORE HORSES:PRESS N":A1=9:GOTO400
390 PRINTSPC(7)"ANY MORE HORSES? Y OR N
400 GETI$:IFI$=""GOTO400
410 IFI$="N"THENA1=9
700 B=B+2:B=INT(67/B):IFA2=1GOTO720
710 A3=INT(RND(.6)*19):A4=A3-7:B=B+A4
720 IFC=0GOTO740
730 C=INT(10/C)
740 IFD=0GOTO760
750 D=INT(10/D)
760 F=12/F:A4=INT(RND(.4)*3):H=H*A4:Y=B+C+D+E+F+G+H:Y=INT(Y):IFY>100THENY=99
800 IFY<1THENY=1
810 IFY<10THENY=Y/10
830 PRINT"THIS HORSE ASSESSED AT:(PRESS RETURN)":PRINT""Y
850 PRINT"":INPUTY$:A$(J)=Y$+A$:J=J+1:GOTO200
900 Z=Z-1:D=1:X=0:PRINT"":C=Z
920 FORB=0TOC:E=VAL(A$(B)):IFE<10THENE=E*10
940 E=100-E:IFE=DGOTO1000
950 NEXTB:D=D+1:B=0:GOTO920
1000 X=X+1:ONXGOTO1010,1110,1210
1010 PRINT""SPC(8)" THE FORECAST RESULT "
1030 GOSUB4000:PRINT"THE WINNER IS "K$:GOTO920
1110 GOSUB4000:PRINTSPC(8)"SECOND IS "K$:GOTO920
1210 GOSUB4000:PRINTSPC(16)"THIRD IS "K$:CLR
1250 PRINT"PRESS'A'FOR ANOTHER RACE: ANY OTHER
1260 PRINT""TAB(12)"LETTER TO STOP FORECASTING.
1300 GETZ$:IFZ$=""GOTO1300
1320 IFZ$="A"GOTO90
2000 END
4000 K$=A$(B):A$(B)="0":L=LEN(K$):K$=RIGHT$(K$,L-2):RETURN
10000 PRINT""SPC(7)"
10010 PRINTSPC(7)" HORSE RACING FORECAST ":RETURN
READY.
```

Thinking Computers?

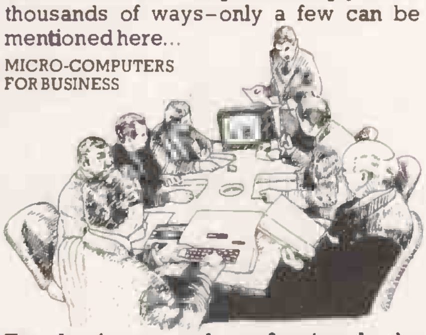
Then come to the number one micro-computer centre

If you're wondering if a micro-computer can help you, we are here to advise you. At Lion House – London's leading centre for micro-computers – you'll find:

* Experts who'll explain the equipment in a way you can easily understand, showing how and where it applies to *your* work.
* Demonstration areas where you can get immediate experience of using micro-computers yourself.
* Probably the biggest range of software in the UK.
* Programmes can be tailored for your particular commercial needs by our In-House Analysts and Programmers.
* Total service – including the availability of full maintenance after you've bought an installation.
* Leasing and H.P. facilities immediately available.
* A computer book section with publications that give you new insight into the world of micro-computers.

How will micro-computers help you? In thousands of ways – only a few can be mentioned here...

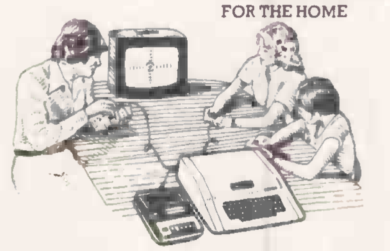
MICRO-COMPUTERS FOR BUSINESS



For business and professional, the versatility of compact micro-computers means that all the benefits of big computers are made available to all at low cost. The businessman can now computerise his accountancy, his stock control, his records and much more – cutting his overheads and improving his efficiency.

For the home, micro-computers have innumerable uses and considerable value too – sometimes in unexpected ways.

MICRO-COMPUTERS FOR THE HOME



Budgeting . . . investments . . . controlling heating or security . . . storing information on things like recipes . . . designing complex and fascinating games . . . education . . .

Come and see. We invite you to visit us and investigate the possibilities and the potential. If you're too far away, phone or write and we'll send you more information.

You need a micro-computer. We can supply it.

LION MICRO-COMPUTERS

SMALL COMPUTERS – TO MAKE YOUR BUSINESS BIGGER

Lion Computer Shops Ltd, Lion House, 227 Tottenham Court Road, London W1 (First Floor). Telephone: 01-637 1601.

Telex: 28394 Lion G.

Open 9 to 6, Monday to Saturday (Thursday to 7).

• **Circle No. 177**

GREYHOUND FORECAST PROGRAMME

```
1 GOTO100
2 A$(0)=Y$+A$:GOTO200
3 A$(1)=Y$+A$:GOTO200
4 A$(2)=Y$+A$:GOTO200
5 A$(3)=Y$+A$:GOTO200
6 A$(4)=Y$+A$:GOTO200
7 A$(5)=Y$+A$:GOTO200
50 REM COPYRIGHT (C) REX L TINGEY FEBRUARY 1980
100 Z=0:GOSUB10000:PRINT"THIS PROGRAMME FORECASTS THE OUTCOME OF
110 PRINT"A RACE BASED UPON INFORMATION ENTERED.
116 PRINT"IF THERE ARE ONLY 2 RUNNERS THEN A DUMMY
120 PRINT"MUST BE ENTERED WITH A LOW ASSESSMENT.
130 PRINT"PRESS'Q'TO ELIMINATE THE RANDOM FEATURE
132 PRINT"FROM THE 1ST ASSESSMENT,ANY OTHER LETTER
134 PRINT"INCLUDES IT. WHEN INCLUDED R APPEARS AT
136 PRINTSPC(6)"THE END OF THE TITLE BLOCK.
150 GETZ$:IFZ$=""GOTO150
160 IFZ$="Q"THENA2=1
200 IFA1=9GOTO900
201 Z=Z+1:GOSUB10000:IFA2=1GOTO220
210 PRINTSPC(32)"R"
220 PRINT"ENTER NAME, OF GREYHOUND"Z"":INPUTA$
240 PRINT"ENTER ODDS POSITION (1=FAVOURITE ETC)":INPUTB
260 PRINT"POSITION IN DOG'S LAST RACE:":INPUTC
280 PRINT"POSITION IN LAST BUT 1 RACE:":INPUTD
300 PRINT"POSITION IN LAST BUT 2 RACE:":INPUTE
320 PRINT"POSITION IN LAST BUT 3 RACE:":INPUTF
340 GOSUB10000:IFA2=1GOTO350
344 PRINTSPC(32)"R"
350 PRINT"TIME OVER LENGTH (ENTER 0 IF NOT KNOWN)":INPUTG
370 PRINT"IF RACE TIME ENTER 0; TRIAL TIME 1":INPUTM
382 PRINT"PERSONAL ASSESSMENT- TIPS, NAPS, ETC":INPUTH
388 IFZ=6THENPRINT"NO MORE RUNNERS:PRESS N":A1=9:GOTO400
390 PRINTSPC(7)"ANY MORE RUNNERS? Y OR N
400 GETI$:IFI$=""GOTO400
410 IFI$="N"THENA1=9
700 B=B+3:B=INT(67/B):IFA2=1GOTO720
710 A3=INT(RND(.6)*19):A4=A3-7:B=B+A4
720 IFC=0GOTO740
730 C=INT(14/C)
740 IFD=0GOTO752
750 D=INT(10/D)
752 IFE=0GOTO756
754 E=INT(7/E)
756 IFF=0GOTO760
758 F=INT(5/F):IFG=0GOTO790
760 IFM=1THENG=G-.4
770 IFZ=1THENS=G
780 IFG=STHENG=4
782 IFG<STHENG=8
784 IFG>STHENG=0
790 A4=INT(RND(.4)*3):H=H*A4:Y=B+C+D+E+F+G+H:Y=INT(Y):IFY=100THENY=99
800 IFY<1THENY=1
810 IFY<10THENY=Y/10
830 PRINT"THIS GREYHOUND ASSESSED AT:PRESS RETURN":PRINT""Y
850 PRINT"":INPUTY$:ONZGOTO2,3,4,5,6,7
900 Z=Z-1:D=1:X=0:PRINT"":C=Z
920 FORB=0TOC:E=VAL(A$(B)):IFE<10THENE=E*10
940 E=100-E:IFE=DGOTO1000
950 NEXTB:D=D+1:B=0:GOTO920
1000 X=X+1:ONXGOTO1010,1110,1210
1010 GOSUB4000:PRINT""SPC(8)" THE FORECAST RESULT "
1030 PRINT"THE WINNER IS "K$:GOTO920
1110 GOSUB4000:PRINTSPC(8)"SECOND IS "K$:GOTO920
1210 GOSUB4000:PRINTSPC(16)"THIRD IS "K$:CLR
1250 PRINT"PRESS'A'FOR ANOTHER RACE: ANY OTHER
1260 PRINT""TAB(12)"LETTER TO STOP FORECASTING.
1300 GETZ$:IFZ$=""GOTO1300
1400 IFZ$="A"GOTO100
2000 END
4000 K$=A$(B):A$(B)="0":L=LEN(K$):K$=RIGHT$(K$,L-2):RETURN
10000 PRINT""SPC(8)"
10010 PRINTSPC(8)" GREYHOUND FORECAST ":RETURN
READY.
```

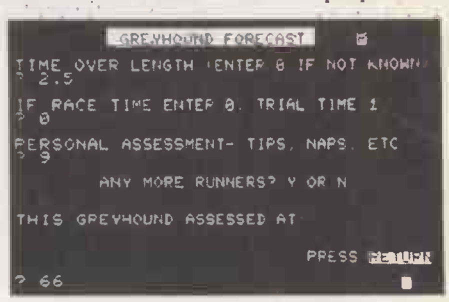
(continued from page 66)

of a minor loop through the values of AS(B), which is the chained string bank formed as AS(J), for the highest value starting at 99. If a value of 99 is found, the runners name with this value is extracted and displayed as the winner. If two runners have the same value then the one which was entered first — with the lowest assigned number — will be the winner and the other one second because return is made direct into the loop, but with the winning data cleared from memory. To keep selection logical, runners should always be entered favourite first, and then in order down the odds list.

The program does not guarantee to find the winner of a race every time. With correct assessment of all runners, It might give approximately 80 percent of either the first or second placed in a complete days racing within the three screened, during an average day. On the day races all run true to form, the forecasts might approach 99 percent, excluding the random feature, on the day they do not, the random feature could be a blessing.

A dog race involves no human element; there are only six dogs in a race; full details of previous races are given in the sports pages; times over the length, or trial times are usually given. All these aids enable the forecast assessments of a greyhound race to be reasonably more accurate than that of a horse race. This program might forecast the winner or second place, taken as one of the three forecast, about 90 percent of the time.

Only the first- and second-placed dogs are taken as winners for the purpose of



course and tote betting. The first three are given by the program to increase the chance of the winner being included. Any serious punter might be able to produce a system to make a bet profitable, using the computed winners from several races. The three also facilitate the amalgamation of the horse and the dog forecast programs into one.

The program uses the device of direct formation of the multiple data bank by directing the elements of data to program-steps of definite assigned string variables — in this case at lines 2 to 7 for the six runners, directed there by line 850. This process compiles a multiple data bank in memory similar to that of the Racing Forecast program.

A major assessment difference in this program, compared to the other, is the

(continued on page 71)

(continued from page 69)

"time over distance" entered for each dog, and whether the time was in a race or in a trial run; these details, when available, are always published in the newspapers. By using the assessment of the favourite — the first dog entered — and using it as a standard time, then the dogs entered afterwards are assessed about this mean, accordingly.

It is fair to generalise that during some periods, a set of races will run true to form when the programs will prove their worth, but be disappointing during times when the opposite is true. It should be realised that all the values and manipulations of assessment must be compromises, particularly as the value of one assessment cannot be put in hard terms against another. A great deal of thought went into the methods used, the final thought always being that each runner is given the same (mis)chance, even if some gross error has been inadvertently included.

It's all in the game

OTHERS HAVE attempted to rationalise and sharpen up their own systems of prediction by using the computer — but essentially as a secondary process. In the first place, they would argue, it is hunch that counts; the means by which those hunches are stored for future reference, to accumulate a rolling total of canny predictions, may be stored in any form. It is just that the computer happens to represent the most efficient means of doing so.

Phil Bull, race enthusiast and former maths teacher runs a company called Timeform based in Halifax, Yorkshire, which specialises in issuing form notes for runners throughout the country. Only one aspect of his business is computerised, though dp manager John Witley rather wishes that others, such as the mailing lists, were as well.

Yardstick

The computerised aspect of the Timeform business revolves around the idea of standard time-figures. The standard time-figure is a yardstick by which to compare any one performance of a particular horse and is expressed as the time difference per furlong against the standard time.

This method has certain implications which Bull is at pains to point out. The first is that it is only a particular performance which is assessed. A very good time-figure for any one horse — that is, where the horse is substantially faster over the five furlongs than the standard — implies that that particular performance is a good one, and that the horse is capable of very good performance.

Yet the converse does not hold true. A good horse may turn in a poor performance — even if it wins — for a multitude of reasons; poor going, too short or too long a distance, insufficiently strong competition.

Where the computer really scores is in the mass of detail which it can process to arrive at a true standard figure for each course. Obviously, for example, there will be wide variations, all other factors being equal, between the time taken to run the five furlongs at Epsom, which is predominantly downhill, and Ascot, which is uphill — the difference is about 15 percent.

Add to this variables such as the state of the going, the age of the horse — horses are supposed to mature at four years old — the weight carried and the wind strength — arrived at for each track by a little cunning computer vector analysis dependent on the configuration of the course — and it can be seen that the corrected time-figure on the day is a very sophisticated assessment of the time a horse clocks-up.

It is also one which it would be extremely tedious to calculate without a computer. Indeed, it can be said that the use of the computer has permitted time-figures to be far more accurately assessed than before.

by Martin Hayman

Timeform prints-out the results of a day's racing with the time rating expressed as the difference between each horse's performance and the standard time rating. A negative figure implies that the horse is quicker than standard, and a quick bit of mental arithmetic converts the time figure into a time-rating, which is on the same scale as Bull's other operations which deal with more conventional form predictions.

This technique forms the basis for an extremely efficient form of betting for the punter who is more interested in maths than in horses. Bull claims that the time-figure can reveal a horse's potential long before the formbook and that it can spot early winners at long odds — indeed, one is urged not to be dissuaded by long odds when one thinks one has located a potential early-season winner in the time-ratings. Bull is presumably speaking from experience when he writes: "Success in systematic betting of this kind depends on the occasional long-priced winner, and there is nothing more galling or exasperating than to find you have missed one because you didn't stick to the rules".

Information

The cost to subscribe to the weekly Timeform print-outs during the season is £200 and for the more detailed information, the whole season's racecards, the cost mounts to £472 — individual racecards cost £3.50.

For this money you are paying for some serious computing power. Witley programs a £23,000 Wang 2200 mini with 5MB disc drive and reckons that with the number of variables involved and using Timeform's own highly-developed statistical method, the operation is "reasonably labour-intensive". A reader of *Practical Computing,* Witley thinks that it would be "inconvenient" to use a micro like a Z-80 with multiple Winchester drives, and more or less impossible with floppy discs. "The discs must be large and fast. You could try to do it on a Tandy or a Pet, but it would be a bit of a Micky Mouse job".

It is fortunate for Witley that, like his employer Phil Bull, he is a race enthusiast himself. He says he still goes to the races himself, but only occasionally; the TV has supplanted his jaunts to the courses. Reassuringly, Bull, who set up the business on the basis of writing (non-computerised) form reports after the second war, still occasionally bets himself, though the majority of his income is now supposed to derive from Timeform.

Imponderables

If capable micro users are rubbing their hands in glee as they dream of doubled earnings from the horses and the pools, the book-makers at least are not quaking in their boots. Doubtless they have seen aids and cunning ruses before — generally their attitude towards the electronic punter is that he is like all the others in the proverbial no-win situation.

Generally, variable weather conditions suit the book-makers. Consistent going usually means that horses are running, and football teams playing, more to form. The fewer the imponderables the more amenable are results to systematic analysis. During the 1978 flat racing season, for example, Bull was able to claim that using his top computer time-figures, "a straightforward bet on the top adjusted time-rating in every race throughout the season showed a 3.91 percent profit on level stakes".

That is no mean result for a punter who is not required to know anything about horses — only to understand a particular mathematical system. Yet the fact remains that the British punter collectively loses £300 million every flat season.

The Inland Revenue is, needless to say, equally insouciant about the possibility of electronic aids improving the punter's lot. The Tote has declined slightly in popularity of recent years — particularly on horses — but off-course betting turnover in Britain was a staggering £2,267 million (all races) during 1978/9, reports the Central Statistical Office. On-course bets totalled £236 million.

And there was light

by John Abbatt

SHE LOOKS intelligent. She sits alone for hours, motionless, seemingly deep in thought. Her eyes focus beyond the confines of the room. The pupils move and the expression changes as if in response to some scene beyond our perception. Sometimes a thoughtful smile lingers on her lips. If you try to attract her attention, she reacts angrily.

But you said nothing about her beauty, did you Marjorie? The oval, innocent face with the sallow complexion; the soft, shoulder-length, dark hair and the wide, clear eyes.

"Your sister has treated Selena for some years, Mr White, so I expect you know all about her autism".

All that I knew about autism had been gleaned from a lay medical publication and from Marjorie's busy chatter over the last week. Already I felt like a charlatan. Nor did Goldsmith make me feel any less uneasy. Homely, craggy features such as his could be seen all over the country, behind the wheels of trucks, at factory benches and in public bars. So the contrast with his cultured tones and the obvious opulence of his home and surroundings was unsettling. He was plainly used to giving instructions. I found myself apologising.

"I expect my sister, Marjorie, told you that I have no sociological or medical qualifications. I am an electrical engineer. But as Selena now responds so violently to her presence, she thought that I might be able to help in a sort of last-ditch effort to communicate with her", I said.

"Yes, yes". He was in a hurry. "And this is the box of tricks, is it: uses the TV I believe"?

I swallowed my irritation at his patronising remark. "It's just that Marjorie said that she did sometimes respond to colour television. In particular she reacted to anything involving patterns or shapes — cartoons for instance. I've brought with me some programs that I wrote for a computer exhibition. They produce a variety of kaleidoscopic effects on the screen".

He frowned: "Well, I've always had the greatest respect for your sister but all this sounds a bit — well, passive to me".

Goldsmith may seem brusque and cynical to you at first, but you must remember that he has had Selena for sixteen years now. He has spent a small fortune on specialist treatment but all to no avail. They have even paid faith healers to work on her. Most other people would have given up by now and institutionalised her, but still he persists. Just think: 16 years of one disappointment after another, and trying to run a thriving business at the same time.

"There's a joystick as well so she can participate if she wants to". I opened the case and showed him.

"You control the formation of the patterns with this . There are four pattern-producing programs driven by a microprocessor in this case. One does spirals, one mosaics and so on. You can push the joystick in four directions to bring in each pattern and merge it with the others. You can twist the know as well. In one direction it changes the colour mix and intensity. The other one controls the direction of movement of the pattern.

"Marjorie thought that if she could be encouraged to use this we could tell a little more about her mental processes by observing the patterns that she creates. I've got a video tape-recorder with me so that I can make a permanent record of the session for later study".

Goldsmith shrugged. "Sorry, but it's all mumbo-jumbo to me. As long as it doesn't do her any actual harm then go ahead. Look, I'll leave you to it. I'll be in the study not far away. Give us a shout if she misbehaves. My wife Helen is upstairs in her studio and she'll be down later".

"Helen's paintings look grotesque to me but people buy them. Sometimes I wonder if she just stands back and chucks the colour at the canvas. I don't know what a psychiatrist would make of them".

I had not expected to be left alone with Selena just like that — I had heard one or two harrowing stories about her 'misbehaviour'. But she continued to ignore me as I took the micro out of its case, plugged in the keyboard and the video-recorder, and took the output lead to the aerial socket of the large colour television in the corner; it seemed incongruous amidst the antique furniture.

I switched everything on and logged-in Then I slipped my program cassette into the holder and loaded the four programs into core. Finally I manhandled Selena's chair so that she faced the screen, sat on an armchair beside her, and placed the joystick on the arm between us.

I started with a rainbow of coloured bands across the screen. Then I formed them up into a rotating spiral. I called in the mosaic routine and carefully transformed the spiral. The complex lattice evolved slowly, growing and fragmenting like an amoeba.

So subtle was the touch on the back of my hand, that I failed to notice it at first until, glancing down, I was surprised to see a slim, pale hand covering mine. I moved the knob slowly in each of the four directions and saw that she was staring intently at the effects created. I twisted it this way and that and the pressure on the back of my hand became more positive.

Once, years ago, four of us had mocked up a seance with a tumbler, upside-down on a polished round table. We cut out pastboard squares with the letters of the alphabet inscribed and the words YES and NO, and placed them in a circle with the tumbler at the centre. Then we sat in the darkened room with the centre finger of our left hands on the base of the upturned tumbler.

Very little happened at first but, just as we were beginning to get bored, the glass took off into motion, sliding rapidly from one letter to another. We hissed accusations at one another but each vehemently denied being the guiding force.

The message which I wrote down with my free hand turned out to be garbled nonsense, but a few days later, out of curiosity, I counted up the frequency of occurrence of each of the letters and discovered that they were the same as for normal English. That is, the Es occurred most frequently, then T, O, A, N and so on. I decided that I had got an anagram and laboured for weeks to solve it, but to no avail.

So it was with Selena. I could not tell which of us was really manipulating the joystick, but the pattern changes became more positive and intricate.

We stared in unison at the screen. Some sort of rising frequency of change was building up, faster than I could comprehend. Here was some visible insight into the torment that possessed the mind of the fragile creature beside me.

The whirling shapes suddenly meshed and co-ordinated in an entirely original way to produce a multi-dimensional effect. The spiral program was in the ascendancy in the form of a complex spinning vortex. It gripped the mind and drew it into the unmeasured depths. Distant inchoate shapes became a beckoning magnet for our perceptions. In some manner our senses fused and we entered that funnel together, revolving, giddy and then levelling-off and gliding down into a world of colour.

We hovered in a cathedral of light, with opaline naves and shimmering apses. Glittering globes and iridescent cones wove involved patterns within the translucent floor, while spitting fire-drakes

(continued from page 73)
phone that I thought I had better bring it along straight away".

I made no move to get it out of my handbag and, after a moment's hesitation, he said, "Would you like to come in"?

The furnishings and decor of the flat were tasteful, if a little unimaginative and overdue for renovation. I noted with approval the reproduction Picassos on the walls of his sitting room. White's belongings were strewn around in casual bacherlordom. Odd pieces of electrical apparatus decorated the shelves and sideboard. I knew that I would have to sit there suppressing the urge to twitch the tablecloth straight or snip the frayed end off of the curtain.

"I wondered how you were", I opened. "Marjorie phoned and she said that you had been off work".

He still looked strained and drawn. Did he have those streaks of grey hair three days ago? He nodded briefly towards a bottle of tablets on the mantelpiece. "The doctor's put me on some sort of tranquiliser. Tea"?

He jumped up and went into a tea-making ritual without waiting for an answer. "How is she"? he asked.

There was no point in prevaricating. "It's a complete regression I'm afraid. It's almost as though Marjorie had never treated her. We are having to feed her and wash her again, and of course she is incontinent. Bill should never have left you both along together. I felt that I should come along and make some sort of apology. Oh, and you wanted this."

He took the cassette eagerly, slipped it straight away into a machine, pressed a button and watched the wheels go round with satisfaction.

"I'm erasing it", he said. "It's the only copy, and I've torn up all the program listings and my notes. No-one can reproduce my work easily, if at all, so we should be safe".

"I thought that you put quite a lot of effort into designing those," I countered.

A look of regret crossed his face.

"Yes. I was doing a lot of work in my spare time on pattern theory. I was working my way towards a single, unifying program covering all mobile topology. But I can't take the risk of what happened the other evening ever happening again, either to me or to anybody".

"I wondered if you could tell me something about it — if it's not too painful for you. You see you are the first person ever to gain any real rapport with my daughter. She took you into her private world, from which I have been excluded all these years".

He thought for a moment. "Yes, I suppose you are entitled to some explanation. Well then: think about the way our brains must work. We have the five senses: touch, smell and so on. Each one is associated with an explicit part of our minds and produces a small analogue of some aspect of the external world.

"Now, can you conceive of a type of mind where every single input is converted into visual terms? You can see the smell of a rose and the sound of an echo. Now add to that a child's capacity to imagine and to fantasise, and you have a being which can produce a form of controlled hallucinations at will. A sort of in-built television set.

"This could be an explanation for all of the visionaries down the ages you know: Joan of Arc, Bernadette Soubirous. But they only spent part of their time in that state. With Selena it's total".

He paused at the disbelief that must have shown on my face. "It's a plausible theory", I responded, "but I was there, remember. I watched you and I had the impression that it was more than just a fantasy to you".

The cup he was holding gave a little rattle in its saucer, and his reply was a little strident.

> **"I have destroyed the programs. It has all gone for ever, and I am left with a set of withdrawal symptoms as though I had come off heroin."**

"You're right, damn it. It was real. It really happened. That other place must exist somewhere and there is something there . . . some presence that I can't explain, but I was drawn to it. The colours there were more varied, more vivid and intense. Ever since I came back to what we call reality I have been consumed with the feeling that it is just a shallow imitation of her world. It all seems — well — drab, and I feel isolated somehow".

I was tracing patterns on the tablecloth with a teaspoon handle. "Isolated, yes I know what you mean. I suppose that what you have given me is your technician's explanation. It was inevitable that you would see it as some sort of wiring problem in the brain circuitry. Well I too have a theory, which may seem fanciful to you but the truth lies probably somewhere between the two of us".

"I believe that for her Evolution has taken a different turn. Her personality is still frozen at day one of Genesis, '. . . and there was light'. If you think about it, during our lives we mimic the whole evolutionary model. We start as a single-celled creature and then evolve rapidly into a more and more complex organism. At birth, the bursting of the amniotic fluid parallels the move from aquatic to land forms of life. We crawl at first, then we become quadrupeds and then bipeds.

"Now just suppose that there is a parallel mental development. Then in our 'aquatic' state we are all inhabitants of some all-pervading sea of consciousness. It is a function of our society to raise each of us above the surface of this shared state to the status of a lonely island.

"Isolated, you said. But she resists us. She lies in the shallows looking at us but when we try to catch her attention she gives a flick of her mind and swims off into that other realm where we cannot follow. We can only peer into the depths and marvel.

"You know Marjorie nearly netted her. She was pulling her across the threshhold of our consciousness hand over hand, and she was struggling to escape. But you somehow got in there with her and she severed the connection and all but took you with her.

"I was going to ask you to repeat the experiment, but with me. It would be that I could at last communicate with her, even if it were for only a few minutes".

The cups of tea lay cooling between us and he was shaking his head vigorously. "No, Never"!

"Twenty-four hours ago I might have helped you. Look, perhaps you don't realise that I took a video-tape of that session. When I got back here, I played back the tape and exactly the same thing happened. I was back there again, in Selena's world with her. It was irresistible. I would play the tape and then try to get some sleep or occupy myself with something else. Each time I found that I would lose concentration and I would be back, re-loading it and sitting down for another session.

"This went on for three days, but in the early hours of this morning I made a supreme mental effort and broke free. You see I had a small, sane vision of my own. I could picture tapes of this sort being marketed around the world. Can you guess the likely impact of something as easily copied and instantly addictive as this on civilisation? I could picture everyone crouched helplessly in front of television screens, unwilling and unable to break free from that other, infinitely exciting world.

"So now I have succeeded in erasing the tapes and destroying the programs. It has all gone for ever and I am left with a set of withdrawal symptoms as though I had come off heroin."

There seemed little more to say. I took my leave and left him slumped, hollow-eyed against the doorpost, watching me drive away. I felt sorry for him.

I could not, of course, tell him that right now Bill was assembling a duplicate of his apparatus at home, and that when I returned we should load it with the copy of the programs that we had made from White's cassette.

Interfacing the individual to information sources

Peter Sommer describes how to interface a microcomputer to the viewdata and teletext systems and how the sure way to enhance viewdata and teletext information is to treat them as direct inputs to local micros which then interpret the results for their users.

SINCE THE CONCEPT of Videotex is still relatively new, I should explain that Videotex is the umbrella term used to cover both broadcast teletext and telephone transmission viewdata systems. It is important not for its sophistication, but in its availability and generality.

It offers an agreed set of standards on protocols and display; centralised support via — in the case of viewdata — the public switched telephone network and — in the case of teletext — redundant TV lines; and the general availability of terminals in the form of enhanced TV receivers. For the U.K. and many other countries, it is the form in which much information will reach homes and businesses.

What we are beginning to see now is a series of attempts to link the processing power of the micro to this information source. The local micro can take desirable data input from the great information banks available to viewdata — Prestel — and teletext — BBC Ceefax and IBA Oracle. Videotex services are the more useful because they supply in homes and businesses not only 'raw' files of information but also data which can be interpreted extensively in numerous ways beyond the abilities — and sometimes imaginations — of the original information providers.

The local micro can also give the local user a flexibility of handling Videotex, both as to speed of access to the required information and, in the case of viewdata, the cost of obtaining, which is available to the ordinary user who owns a simple 'dumb' terminal. It is early days yet, but already there are some quite remarkable results. I believe that potentially they will create great changes in the way in which use of the small micro will develop, first in the U.K. then, perhaps, in other countries.

Viewdata technology

First we need to know something about viewdata/Videotex technology. Perhaps the most difficult thing to appreciate for an individual familiar with conventional information retrieval computers is that ordinary direct interrogation of the viewdata computer — typically a GEC8040 — is not possible. It is, of course, even less possible with the broadcast teletext transmissions.

Videotex is stored as a series of frames or file cards; in viewdata, the only thing the main computer understands is the top row of each frame, which contains page numbering and charging, and the bottom row, row 24, which contains the routeing instructions. Actually, this is not entirely true, but nevertheless the content of each frame is 'invisible' to the main computer — it could be anything — lines of text, columns of figures, graphics, or even data instructions.

Viewdata's aim is to be user-friendly: what the ordinary user has in the home or office is a dumb terminal — which in most cases is also a receiver of broadcast television and a resolver of teletext signals. To remind those who have yet to use viewdata, the only commands available are those from the calculator-like pad — simple numeric demands for specific pages. The individual databases route the user through a series of choices, routeing pages, to those which contain the information desired — end pages.

This apparent limitation is a considered trade-off in the interest of making the system available to the totally untrained. If viewdata is to work for the mass audience, then you have to remember that a good number of the users will be those same people who today can't quite get the hang of re-setting the time on their multi-function digital watches.

To persuade a user that a database is easy to use, the information providers — viewdata's name for the independent publishers (IP) who put information on the system — have to structure their databases in a variety of ingenious ways. They have to ensure that any conceivable whim of the browsing user is anticipated so that he can be routed effortlessly in all manner of directions. Although some people have described a typical viewdatabase as having a tree structure, in fact such is the amount of looping back and forth and branch-hopping that all sorts of other metaphors have to be employed.

These two factors inherent in the viewdata system suggested quite early on the need for 'intelligent' terminals: the 'invisibility' of the useful data to the main computer; and the complexity of the structure of the database created by the IP to make it easy for the untrained user.

Almost as soon as the viewdata show was on the road, critics were suggesting ways in which it might be improved and enhanced. Many of them forgot that it is the availability of viewdata and imposed standards that are among its most significant features. The intelligent terminal provides enhancement without interfering with the simplicity of the basic protocols. More recently, other uses requiring local intelligence have made themselves apparent. Intelligent terminals are of use to both the Information Provider and the more sophisticated user.

The ordinary IP editing terminal in viewdata provides the facilities necessary for frame generation — alpha-numerics, a limited graphics set, colours, double-height characters, flashing, conceal/reveal and so on. The Post Office computers provide facilities for inputting, editing, routeing, numbering of pages, and so on, but these features inherent in the system are often not enough even for quite small IPs. These are some of the things that an intelligent editing terminal can supply:

- Text and word processing — this may be quite simple or, in the case of very large IP organisations, amount to re-formatting data already available in another form.
- Storage of pages or sections of pages for transfer — if you generate in graphics an elaborate map you don't want to have to hand-key it every time.
- Bulk up-dating, transferring whole sections of a database from one location to another, while preserving the integrity of the overall tree structure.
- Database management — because of the variety of ways in which the user can be routed through a database in viewdata, the IP often needs more information than the central computer provides about how his frames are organised and used. Further, a prudent IP needs to know how his pages are accessed so that he can perform suitable analyses — he needs a breakdown of sales figures like everyone else.

Great strengths

Not all IPs are large conglomerates or public utilities, although that's what a lot of people seem to think. One of the great strengths of viewdata is the provision of a form of classified listing or advertising — the sort of thing that a local estate agent, department store, or weekly newspaper may wish to let potential customers in a specific locality know about. We're talking of information that isn't sophisticated in content but derives its value from being up-dated regularly.

IPs in this category could seek services from a consultancy or umbrella IP but they may prefer to do their own inputting.

It is precisely this class of business that has bought micro-based computer systems. In fact, to let the thought hang — such a business would also be a user of viewdata as well.

Turning now to the user, these are some of the possibilities of the intelligent receiving terminal:

Indexing and keyword searching of the database. The ordinary way to use viewdata is to follow a series of routes developed for you by the IP, but suppose you use one particular database — or maybe two or three — rather frequently. Wouldn't it be more convenient to be able to go straight to the page of interest without involving the time and expense of conventional routeing?

Disc storage

Your local micro can store all those page numbers for you on a disc and you can make your selection off-line using a menu that is convenient for your particular purposes — and addressing it with full alpha-numerics as well — and then have the micro call up the viewdata computer and demand the page instantly.

A travel agent would find such a system useful now in terms of what is available — there is a choice of railways and airline timetables and also several tour operators offering a variety of holiday packages and availabilities. At present, a travel agent has to tour Prestel manually. The same can apply to businessmen seeking financial data and to lawyers and accountants wanting legal information. Indexing of this sort would be particularly useful to the Closed User Group which, though on the public switched telephone network, denies access to its pages to non-subscribers.

Infolex, which gives a legal update service, for example, uses a minimum of routeing pages and a maximum of end pages with useful data on them. At the moment, it sends subscribers photocopied indexes through the post.

However, such indexing works only if the IP can guarantee a hardened database structure. In other words, no matter how frequently he updates specific items of information on his pages, the frame numbers corresponding to the class of information have to remain constant. To use a well-known example, British Rail can alter the timings of its early morning services from London to Edinburgh several times a month, if need be; but the information must always appear on the same Prestel page — that is if the index system is to work. A number of IPs already use such a hardened structure, notably *The Economist*.

Analysis of frame content. So far the local intelligence has merely been asked to look at the top row of the viewdata page — the one with the page number and charging instructions.

You can make a micro take a page off the viewdata base and then input it into its own memory — as opposed to displaying it dumbly on a screen. Providing the IP uses a reasonably consistent means of arranging his material, you can then make the micro embark on a string search of the contents of the page, and, having located the class of information, input it into a local program.

Or, if the class of information required is always located in a particular field on the VDU, you can shorten the process of search. Incidentally, you can do the same with teletext — Ceefax and Oracle — and if you think how frequently some of the data on those services is changed and updated, you can see the value.

Suppose you want to record and analyse the movement of a share or commodity or currency over a period. You can instruct the micro to call up a series Prestel, Ceefax, or Oracle pages at selected times, conduct a search of those pages for pre-determined data, and then input that data into a locally resident analytical program.

The movement of gold prices over the last few months may not be available in any detail from an IP, but with a suitable micro and software — actually a very simple program — you can have this on tap whenever you want it — with graphs and histograms of your choice.

Regulation of viewdata use. Suppose you are a large user, with Prestel terminals strung throughout the premises. Not all the individual stations can be supervised all the time and you wish to ensure that your employees don't use the facilities for private enquiries — is it possible to limit access to certain approved pages?

Suppose you are a retailer or a hotelier offering public access, possibly through a coin-op terminal. Each IP specifies the charge per frame, but perhaps you want to 'give' away certain priced frames, because they relate to the services you feel you are supplying your customers — can you extract those particular frames so that they are free to your customers while you pay the bill to the IP?

Local decisions

The answer is that you can — it's another example of using the micro to scrutinise the top row and make local decisions.

Private viewdata. Not all viewdata is on the public network — the technology can — and is — being used as an in-house information dispersal and collection device. Remember that in a few years' time the terminals — TV sets — will cost far less than an office typewriter. You can limit the access of various classes of user.

Already Philips and GEC are offering large-scale private viewdata systems, but smaller-scale varieties could be run from a fastish micro with reasonable memory. The private viewdata systems and the public version could interface, of course.

Telesoftware. Earlier I said that an IP could put anything he liked on his frames — test, graphics — and data. A viewdata frame need not be instantly readable — why not send machine-code data instructions through the air or down the telephone lines?

This is already being done, and the name is Telesoftware. You can see it on some of the Oracle pages and the software house of CAP is providing experimental versions on the Prestel pages.

A typical example for domestic application would be a mortgage calculator. Or you could send encrypted data on lost cheque and credit cards. Or games — instead of buying a cassette or cartridge for your dedicated games computer, you can have one piped down the line.

Public service

I would like to see the development of public service Telesoftware designed to take people through the ins and outs of welfare benefits and facilities. However, such a service isn't likely until viewdata is already a well-established mass medium — and until the Government of the day feels it can afford such things.

Loading up time is roughly comparable to cassette — a modest program may need seven - 10 frames. You are almost compelled to use a machine code because loading-up in a high-level language like Basic would be impossibly long. You also need some error correction facility to guard against line noise.

There is no reason, though, why a resident ROM in the intelligent terminal shouldn't be able to make the program resolve into Basic, if that is what a particular application demands. Typically, though, once the Telesoftware is loaded, the user commands it through numerics or alpha-numerics.

In the broadcast teletext version, loading may be quicker than on viewdata. In the current scheme, the teletext frames can be loaded into the local micro in any order — you don't have to depend on having each frame cycle round correctly in sequence. Program steps can be put in order in the terminal. All this is achieved by the appropriate heading code to each frame.

Telesoftware, however, needs agreement as to protocols and either a standard resident processor, preferably a very popular one like a Z-80 or 6502, or a portable language — CAP hopes to have one in Microcobol. Even then, because of the slowness in loading and anxieties about copyright protection, at the moment it looks as though Telesoftware will only be used for short, limited applications — at least for the next few years.

These then are the possibilities offered by linking micros and viewdata — how actually are the connections being achieved?

A variety of approaches are being demonstrated and others suggest themselves, but whatever you do has to take as its starting point the problem of linking

(continued on page 79)

Keen Computers Micromarket

Keen Computers are one of this country's leading microcomputer consultants, with a reputation for quality after sales service. Recent extensions to our product range mean that we now offer the North Star Horizon and D.E.C. computer systems along with the Apple, and Corvus and Mountain Hardware accessories. Add this to our ability to provide software from a comprehensive range of packages or on a special consultancy basis (we write the software to your specifications) and you can see why we can provide the solution to your problem . . . whatever it may be!

Apple

Apple II + 16k	750.00
Additional 16k RAM	69.00
Disk drive + controller	398.00
Pascal language system	296.00
Printer card	110.00
RS232 Interface	110.00
Integer Basic ROM	110.00
Supercolour	90.00
Serial/Parallel Interface	65.00
Floating Point Board	205.00

North Star Horizon

16k Horizon computer system with floppy disk drive + controller	1085.00
32k Horizon computer system with floppy disk drive + controller	1415.00
Additional floppy disk	310.00
16k RAM	235.00

Peripherals

PRINTERS	
Anadex DP8000	715.00
Lear Siegler 200	1925.00
Diablo 1610	1950.00
CORVUS 10Mb HARD DISK . . . FOR Apple (DOS+Pascal), Tandy, S-100 and LSI-11 from	3500.00
100Mb tape back-up from	500.00
MOUNTAIN HARDWARE	
Supertalker	190.00
Apple Clock	140.00
S-100 Clock	140.00
Romplus+	120.00

Software

Packaged titles include Word Processor, Direct Mailing, General Ledger, Sales Ledger, Purchase Ledger, Payroll, Stock Control, Information Retrieval, Estate Agents Management, and many more. Full details sent on request.

We can also design and produce software systems to your specifications . . . for more information contact Bob Ellis on 0602 583254.

To find out more . . . phone or write to:

Keen Computers

5b the Poultry, Nottingham.
tele: 0602 583254
telex: 37297 (keenco)

CRA COMPUTER RETAILERS ASSOCIATION

• Circle No. 180

(continued from page 77)

anything to Post Office lines. The familiar arguments about Post Office policy towards the modem are the more pointed because in the case of Prestel, you are dealing with a facility that links, not one private computer to another, but a private device to a public one.

In fact, Prestel has been, so far, the only area in which Post Office rigidity has softened — you can use your own modem, provided it is type-approved. The fact that you are allowed to do this is a tribute to the extent to which the Prestel staff at the Post Office have been able to prevail over the anxieties of their telecommunications colleagues. Nevertheless we are a long way away from instant, easy hook-up; and every Prestel terminal, dumb or intelligent, has to follow rigorous testing procedures.

Assuming that you can solve these problems — how do you develop an intelligent terminal cost-effectively?

Dedicated approach

The first method of approach is to take a basic TV set with a teletext generator, a display board and an approved modem — a dumb receiver in other words — and put a microcomputer into it. We could call this the dedicated approach. Some TV manufacturers are demonstrating models of this sort, but with the functions of the CPU severely curtailed so that the end result is a terminal responsive only to a very limited range of inputs — particular sets of Telesoftware, for example.

Here you liberate to the user the full functions of the micro — you have a stand-alone that will function independently off-line, though the main idea is that it functions with Videotex. So you have a built-in modem and, in some cases, a TV tuner, for the capture of teletext.

This is the approach of the TECS developed by the Liverpool company of Technologics. The processor is a 6800 and they offer a package which includes the reception of teletext, Prestel, and a 3K-mini-Basic available either in ROM, cassette, or disc. A fuller Basic, a machine code monitor and other high-level languages are being developed.

You can buy it in either table-top or rack format. This could be a system with a lot of potential, but at the moment there is very little software and because of the relative unpopularity of the 6800 chip, there are fewer ready-made sources to be tapped. Cost, in the simplest form is around £2,000.

Using the same approach is the Mighty Sparrow, from ISE, the firm which also created the coin-op terminal — which incidentally, I believe, could be an extremely important vehicle in developing viewdata as a mass medium in the period while viewdata TV sets are too pricey for the residential market. Whereas Technologics seems to be thinking also of the user and hobbyist markets, ISE is more immediately concerned with the IP.

The Mighty Sparrow is the most complete working unit available, though it is under constant development as ISE becomes aware of what the market wants. Full Post Office approval is just about to be secured. Its choice of chip is the well-known Z-80 and it means that it has been able to create operating sytems for editing and disc by adopting the CP/M suite from Digital Research and then extending and adapting it simply for its needs.

As a result, it is able to offer many of the desirable facilities mentioned earlier: you can edit off-line, lift frames from one location to another, store frame numbers, carry out limited word processing, and so on.

On the hardware side, ISE will offer a choice of keyboards — the one you can buy now is designed more for the user and they are currently seeking advice from the industry about the design of an intelligent editing keyboard, although you can edit from the present one. Everything appears to have been designed for maximum flexibility, so that its units can be used overseas on systems with slightly differing standards, so that baud rates can be altered, and so on.

There is also a V24 interface to permit direct communication with mainframes and international viewdata systems. The price is around £2,000, including RGB monitor, but discs are extra. You don't have teletext. A reasonably full software suite will be just under £5,000.

It's possible to buy a special VDU add-on to your existing micro or mini configuration. The VDU works to teletext specification. You then need a modem and interface board, and you're away. A Southampton company, Hi-Tech, is offering a VDU board configured to a S-100 bus. In association with another Southampton company, Xitan, this is being offered either with a North Star Horizon or Cromemco as a source of ingelligence. At the moment, this system can be seen as a private viewdata system, which mirrors many of the features of the public service. The software seems to be in a fairly advanced stage.

High standards

Similar facilities are offered by yet another Southampton company, Leenshire, whose main business until now has been the provision of dedicated display units for such places as airports and who provided the VDU facilities for the current editing suite at Ceefax. Leenshire also has a S-100 VDU and associated software and are promising the imminent appearance of a S-100 TV tuner and teletext capture board.

Neither of these configurations has Post Office approval yet: the Post Office prefers to grant modem approval for complete installations. One of the difficulties of the add-on approach is that they can never be certain of the environment in which a modem may have to operate. As many people are now becoming aware, modems are subject to interference from RF (radio frequency), interference from VDU time-bases, PSUs, and a number of other electronic sub-assemblies. If the add-on modem approach is to work, therefore, very high standards of RF screening are likely to be demanded. There's also the thought that this particular path drives still deeper the wedge into the Post Office monopoly on modem supply.

Formal acceptance

In some cases, however, you don't need an external VDU. One system, already in demonstration and awaiting formal acceptance for connection, is based on the Apple. The display is derived by using the Apple high-resolution graphics. The arrangement was created by Owl Computers and will be marketed by Microsense. The add-on cost will be about £600, which unfortunately does not include colour, just monochrome, but this might be acceptable for business.

ITT has a similar development for the ITT2020, though since the high-resolution graphics are significantly different — it's the main distinguishing mark between the Apple and ITT2020 — the Owl Appletel solution will not work directly. ITT has been heard to say that it hopes to be able to offer three or four colours, but nothing has been mentioned recently. Neither the Apple nor ITT offer capture of teletext signals.

Stand-alone

Other systems are just about to appear. The Swedish Luxor AB680 uses the teletext character generator as its VDU. Through its Swedish parent, which has a special arrangement with Prestel International, it has been linked into the Prestel system. However, at the moment the ABC80 is sold as a stand-alone.

Similarly, the Acorn uses the same approach for its VDU. Again, a specially-configured Acorn has been linked to the Prestel system — the man who did it is the same one who performed the Appletel miracle — Mike Gardner. Both these systems are some way from being approved by the Post Office.

There's one last approach that suggests itself, one which has yet to be tried, because it awaits the TV industry. In the Prestel Terminal specification issued in November, the Post Office indicated a series of approved in-outs to terminals, whether dumb or intelligent. One interface is for an audio tape recorder, so that you can freeze a few frames in viewdata audio tones for later off-line playback — you can already see this on the latest models to be shown.

With the introduction of message store-and-forward facilities on Prestel, message and data transmission in the viewdata protocols will become very significant, and facilities available similar to those offered by Source in the U.S.

Why good software is expensive

Nick Horgan looks at the efforts required to make good computer software and why, consequently, it can cost so much. His step-by-step approach reveals the hidden traps which await the unwary.

IT IS A SORRY fact of life that software development is very costly and time-consuming. The arrival of microcomputers has done little to change this. Since they lack the software aids found on large computers, the cost in certain cases may be even higher.

For a long time the mainframe computer user has been aware of, and worried by, the ever-increasing cost of getting projects up and running on his machine. If the company spending millions of pounds on hardware is worried about software costs, then how much greater a problem will these costs be to the company spending only a few thousand on a micro.

There have been two main areas of effort in trying to reduce the impact of this cost escalation. Firstly, higher and higher level languages have been devised in order that the writing line of a new system can be kept as low as possible. Secondly, companies have tried to write systems that can be used as packages, thereby spreading the initial costs.

Some of the high-level languages running on large computers are as far removed from procedural languages such as Basic and Pascal as Basic is from machine code. With these high-level languages, the computer can be made to perform a large amount of work with very few instructions from the programmer.

Packages have tended to concentrate on areas of business which are well defined by accounting/legal requirements: payroll, bookkeeping and auditing packages, for example. They have steered clear of the less well-defined but most important parts of a business such as invoicing and stock control. Most concerns tend to have their own particular method which would require expensive modification to software packages.

These problems with the micro tend to be glossed over by many dealers and manufacturers intent on selling hardware to the small business computer user, and who are unaware of his real requirements.

1. Identification of the problem.
2. Exploration of the problem.
3. Identification of the solution.
4. Implementation of the solution.
5. Monitoring of the results.

All these stages are equally important in the design of the software system. If you are quoted a cheap price for a one-off piece of software, you can be reasonably sure that the person giving the quote has their eye set upon stage 4 as the biggest part of the solution.

The reason for this is simple: stage 4 can always be made to work — it consists mainly of programming, and for the supplier this is an easy thing to cost and be correct according to the specification. Stages 1 to 3 are much more difficult to justify and cost and, although they are absolutely essential to the project as a whole, they tend to be pushed to the back of the quoter's mind.

To be fair, the prospective user often regards these stages as trivial and to a large extent common sense; he is therefore reluctant to pay as much for this part of the project as he is for the mystique of programming. Hence most quotes tend to underestimate the work required prior to the actual programming and the user ends up with a bad system, and the system designer with an unprofitable contract.

Before we go into more detail, I must point out that the stages I have defined are rather artificial; of course, in real life they will overlap and interrelate. When monitoring results, you will often find that small problems appear which were not anticipated in the original design and which have now to be included in the system; this should only happen in a small way and not too often.

Identification of the problem

There are commonly two types of problem:

- Problems of which we are aware and which are easy to identify.
- Problems of which we are not aware and which are, therefore, not easy, or even impossible, to identify.

We are all aware of the first type of problem. Examples are:

- General accounting and in particular accounts receivable.
- Payroll.
- Order processing.
- Stock control.
- Mailing lists.
- Word processing.

These problems may become acute if a company wishes to expand without taking on more staff, to get a tighter grip on finances or, less often, to integrate the company's activities. Whatever the reason, the problem can usually be identified easily and will in most cases make itself obvious by causing trouble in day-to-day running.

Not so easy to understand are the hidden problems. Broad categories are:

- Management information.
- Customer services.
- Internal company facilities.

Management information can be defined as something you thought you knew by intuition, didn't want, bought, and now can't see how you ever ran your company without.

It should answer such questions as:

- When will the running cost of a piece of equipment exceed its replacement cost?
- Does the customer who receives a 20 percent discount on all his purchases really earn the company any money?
- How much does that kilo of mince produced from a carcass bought at the market have to sell at to preserve the required gross margin?
- How much extra stock do I need to carry me over a peak seasonal period?

With profit margins reducing all round, this sort of information is vital to keep your company profitable.

Customer service is even more difficult to quantify in straight money terms; but unless you look after your customers, they will fall easy prey to your competitors. Some of the things that you can do for your customer are:

- Keep him up to date on product information and prices.
- Contact him regularly, especially if his order has dropped.
- Make him aware of any discount he could receive by increasing the amount or number of products he can buy from you.

What usually hinders a company from supplying this kind of customer service is lack of information on how the company is running.

Lastly, we have internal company facilities. For the small company with few customers and products, these can largely be ignored, but for the larger, geographically spread-out company they can be of great benefit. Some examples are:

- Job cost allocation.
- Personnel records.
- Telephone lists.
- Internal registers of any kind.

Whatever the perceived problem, the next stage is to look at it in detail.

Exploration of the problem

We must first determine whether the problem we have identified is the real one or just a symptom of a deeper malaise within the basic system. For example, the failure of an accounts receivable system may be caused by incorrect invoicing.

We then need to define the bounds of the problem. Using this example, the problem with the invoicing could be incorrect extension of price and quantity; to double-check this may involve more work at the outset, but as the number of credits fall, the time saved on the accounts may more than balance the initial investments in checking the invoices.

When running such a check, watch for opportunities to improve other parts of the system which, although not a problem, could benefit from an overall improvement in the general operation.

At this stage, keep an open mind. If you think to yourself, 'Here is the problem, how can I get a computer to solve it?' it may bias the investigation in what could turn out to be the wrong direction. You should concentrate on the problem and document it in such a way that when you have finished this stage, you can apply a number of possible solutions and see which will give you the best results.

You may decide to call in outside help to analyse the situation for you, or you may decide to do it yourself, but before you make up your mind, you should consider some of the pros and cons in Table 1.

Reading Table 1, you may consider that it has a bias towards outside help, but there are three major points you should bear in mind in addition to those shown. Firstly, some points may be different for your company. For example, your staff may well prefer to speak to you rather than an outsider. Secondly, you have to find a good reliable consultant; lastly, it will cost money. Only you are in a position to make the decision.

However you decide to carry out the problem investigation, you must, at the end of the day, have a detailed, well-thought-out written report in a format which is going to enable you to try the various solutions.

Always be aware of other non-crucial problems which can be easily included in the main scheme. Try to think ahead to see how things will change in the next few years, and estimate what effect these changes will have on your existing procedures.

Identification of the solution

At this point you should have a clear idea of the problem, include such things as the amount of money which can be spent on the solution, the time-scale in which to fix it and the resources you are prepared to commit to it.

Table 1. Do-it-yourself v. outside help.

Using your own staff	Using a consultant
Lack of experience of how others run their business.	Should have a wide experience of different methods.
They know how your own business works.	May take some time to familiarise himself with the business. You will have to pay for this.
Bound to be biased towards your existing systems which they may have set-up.	Will have an open mind.
Staff may be reluctant to talk about defects in the existing system if they feel it will get back to the boss.	If presented in the correct way, staff may feel happier criticising the existing system with an outsider.
Difficult to discover the true cost of using your own people.	The cost may be high but at least you know what it is.
To do a good job requires full-time work, not usually possible with own staff.	Will be able to dedicate himself to the project.

Before you start to examine the available solutions, you must write down any extraneous factors which will affect the situation. A particular solution may be justified on grounds of cost but impossible to implement. For example, hiring an extra member of staff may make sense but if there is no more space in your office, you can't do it.

We must also look again at the question of outside help. In this stage of the project, you are going to be looking at methods and equipment of which you have no experience so you are not able to compare satisfactorily the various possibilities. Salesmen and dealers can help but, of course, they have an axe to grind.

Some of the solutions you should consider are:

- Re-organise your existing system, manual or computerised.
- Distribute some of the work to external agencies.
- Install a manual/electro-mechanical filing system.
- Install a simple accounting machine.
- Install a visable record computer.
- Use an outside computer bureau.
- Use an in-house computer.

The re-organisation of your existing system is obviously the best and cheapest solution and is one which will almost inevitably have to be done before any other solution is adopted. Some simple thing like using window envelopes or batching certain types of clerical job can give worth-while savings with very little effort.

The distribution of work to a agency: it may be cheaper to let specialist outside services handle certain parts of your business. There is no limit to the types of work that can be sub-contracted; they vary from simple secretarial work to virtually all your accounting and management functions, including things like mailing lists, advertising etc.

Manual/electro-mechanical systems: nowadays there are so many good filing systems available that it is a very involved subject in itself. To obtain an idea of what is available, you should read some of the office equipment magazines. If data retrieval is your problem rather than data updating, then some filing systems can, in some cases, outperform computers not just in price, but also in speed.

Simple accounting machines have been available for a long time and usually employ amazing examples of mechanical ingenuity. Most are now being replaced by their micro-driven counterparts, but they do work and an inexpensive second-hand one might be the answer.

Visible record computers are, to all intents and purposes, microcomputers which have been specifically designed to run a small business. They are worth considering if you are prepared to modify company procedures to fit in with the packages which are supplied and you want a solution that is very much "plug-in-and-go". As their name implies, they provide a visible record of what is going on, as well as storing information magnetically. Many users find this very reassuring and don't feel that their company is being taken over by the computer.

Computer bureaux: if you feel you need a computer to solve your problem but are not too happy that you or your staff possess the necessary talents to control one of your own, then a computer bureau may be the answer.

There are two basic methods of using computer bureaux. The first, and at one time most common, method is called batch processing. Communications between you and the bureau will normally be by post or courier. It is ideally suited to jobs such as mailing lists where time is not a crucial factor and the number of changes to your data kept at the bureau is small.

The second, and now more usual, method is called on-line, where you communicate directly with the bureau computer. The user is in contact via Post Office lines and a terminal, often a visual display unit (VDU), and can update any of his information directly on the computer, receiving information on any errors or changes made. Some bureaux have combined batch/on-line systems and others allow you to have a small computer in your office which, when it wants some help, can communicate with the much more powerful machine at the bureau.

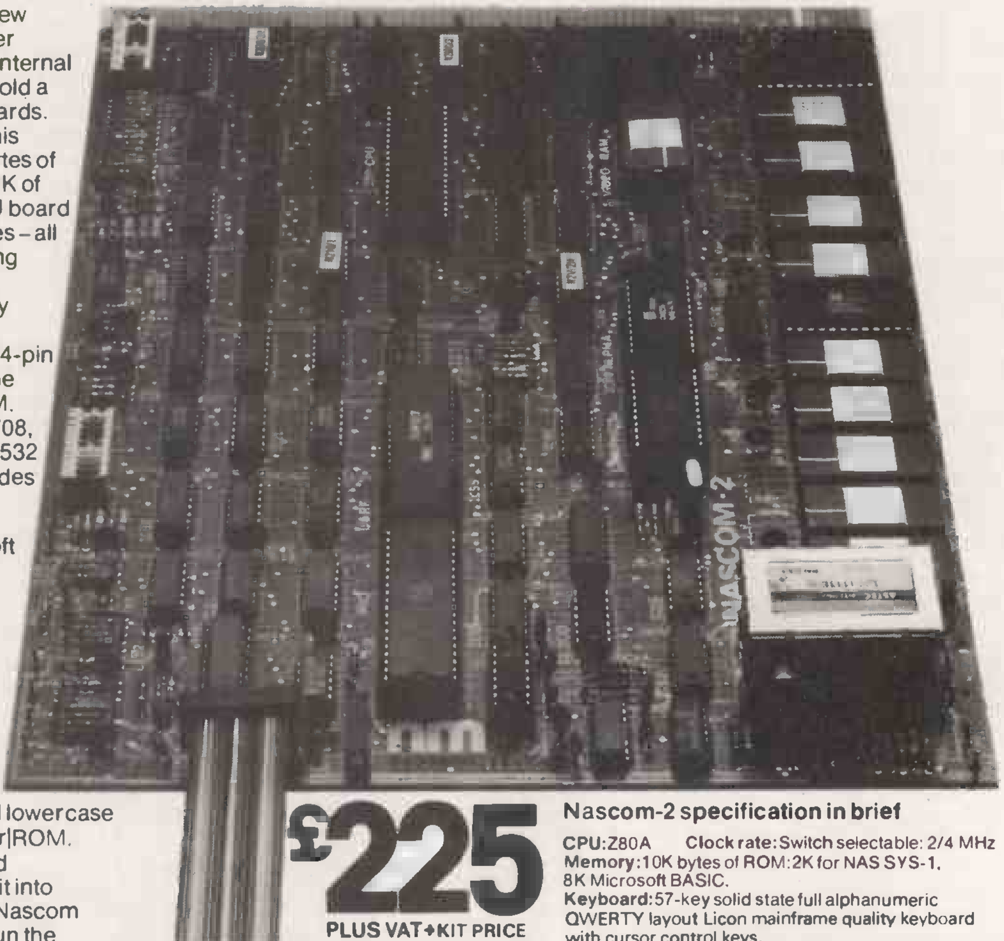
The advantages of a bureau are large data storage, experienced staff to handle any problem, no worries for the user about hard/software problems and very fast printers for mailing lists.

However, there are disadvantages. You are locked into a system and could find it hard to avoid paying higher charges because the cost of transferring to another bureau or to your own machine, would be too high. As the bureau finds more and more customers, the response time — that is, the time it takes the

(continued on page 83)

The first step towards 208K bytes of internal addressable memory...

nascom-2



The Nascom-2 is the heart of our new System 80 desk top microcomputer system. It fits into the bottom of an internal frame racking that is designed to hold a further four 8in x 8in expansion boards.

With our new 48K RAM board this means you can have up to 192K bytes of internal memory. Add to this the 16K of user memory available on the CPU board and you have a potential 208K bytes – all addressable by the Nascom-2 using page mode software.

To give maximum user flexibility Nascom-2 is now being supplied without user RAM. The eight, free, 24-pin sockets have link option to allow the use of either RAM, ROM or EPROM. With EPROM the choice is wide: 2708, 2716, 2508, 2516, TMS2516, 2758, 2532 and 2732. Using 2716s alone provides over 16K.

In ROM there is a 2K NAS SYS operating monitor and 8K Microsoft BASIC. NAS SYS is a powerful monitor with full cursor control allowing you to edit any part of the screen without having to re-enter whole lines.

A character generator ROM is provided to enable Nascom-2 to have 128 graphics characters which can be displayed in three different modes. This is in addition to the 128 character full upper and lowercase alphanumeric character generator ROM.

The 57-key solid state keyboard supplied with Nascom-2 will also fit into the System 80 housing as will the Nascom 3 amp power supply required to run the Nascom-2. (A 5 amp supply will be needed to run the CPU board plus four expansion boards.) So you can see the Nascom-2 is totally compatible with System 80. Buying a Nascom-2 gives you a head start to a very powerful infinitely flexible and expandable system designed by one of the leading microcomputer design teams in Europe and offering unrivalled value for money. Now isn't that worth all the waiting?

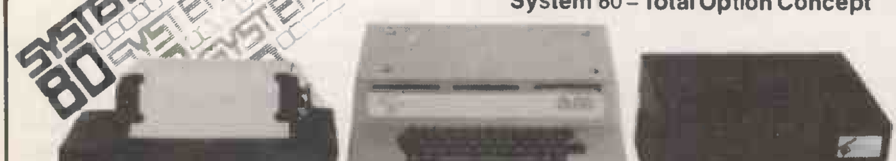
£225

PLUS VAT+KIT PRICE

Nascom-2 specification in brief

CPU: Z80A **Clock rate:** Switch selectable: 2/4 MHz
Memory: 10K bytes of ROM: 2K for NAS SYS-1, 8K Microsoft BASIC.
Keyboard: 57-key solid state full alphanumeric QWERTY layout Licon mainframe quality keyboard with cursor control keys.
On board interfaces: Domestic TV at 50Hz 625 lines (adaptable to 60Hz/525 lines) displaying 16 lines of 48 characters.
Kansas City type audio cassette (300/1200 baud) or Teletype. RS232 for printer. 16 free, programmable PIO lines.
Graphics: 2K bytes of ROM: 128 graphics characters, 128 character ASCII with full upper and lower case Alphanumerics.

System 80 – Total Option Concept



For further details and stockists please contact:
Nascom Microcomputers Limited,
92 Broad Street, Chesham, Bucks.
Telephone: (02405) 75155

nm
Nascom Microcomputers

• Circle No. 181

(continued from page 81)

computer to reply to you — may slow down. If the bureau encounters any reliability problems and you are a small user, you could find that the bureau will allocate most of its remaining resources to their larger revenue earners.

If you are thinking of using a bureau you must, in your own interest, find an outside consultant both to vet the initial start-up and to review your charges year by year.

An in-house computer: if you think that installing a micro in your company is a matter of signing a cheque to your local dealer, putting it under your arm and starting to run your business on it, then I am afraid that you are in for a nasty shock. If you look at table 3, you will see some of the not-so-obvious problems that you will have to face at one time or another. Yet with thought and planning, all these problems can be overcome.

A glance through this magazine will give you some idea of the vast choice that is available to the microcomputer user; even to scratch the surface of the factors required to separate one system from another is beyond the scope of a short article.

Table 2. Some not-so-obvious reasons for buying your own computer.

- Tends to enforce a common method of working amongst the staff concerned.
- A well-thought-out system should give you confidence that you can cope with an increase in business, without an increase in office overheads.
- Again, with a good system it will give you greater security both against frauds and disasters. All accounting information can be password-protected for restricted access and you can lodge copies of all your files with your bank to guard against fire risk.
- If you have more than one branch, then sharing information via magnetic media is fast, cheap and easy.
- You should find that staff have increased job satisfaction and involvement.
- Early detection and prevention of errors. Validation procedures should trap most of the common clerical errors.
- Better presentation of your company to the outside world.

Table 2 shows some of the not-so-obvious advantages of the in-house computer. Add the obvious advantages of speed, accuracy and flexibility, plus the unbeatable cost/performance ratio that gives you computing power that would have been unthinkable even a few years ago, and you can see why there is so much excitement among potential users of micro business systems.

There are a few quite basic considerations you need to bear in mind before you start to look at the micros that are available:

- The amount of disc storage you will need will be a big factor in your choice of hard or floppy discs.
- Do you want more than one number of your staff to be using the computer simultaneously? If you require to adjust the accounts at the same time as doing some routine order-processing, then you will need a multi-user system.
- Make sure that the supplier has some software which is suited to your needs.
- Check that delivery, maintenance and financing are all acceptable.

These four simple criteria will help limit the number of computers which you need to consider in depth.

For the rest of this article, I will ssume that you have decided that your needs are best served by buying an in-house microcomputer.

Implementing the solution

Once you have decided on the hardware, you now have to consider the software. There are three possibilities:

- Use a pre-written package.
- Do-it-yourself.
- Commission purpose-built software.

There is the fourth possibility of putting a computer expert on your payroll. I feel that this is beyond the budget of most companies using a micro, so I shall deal only with the other three solutions in detail.

A pre-written package: firstly, if you can possibly accept a pre-written package as it stands, then you should do so. Some packages are built to accommodate certain changes — these are usually restricted to the reports lay-outs — and your special needs in that direction should not present problems to the software supplier.

However, if you need some fundamental changes to the package, then you must be prepared for considerable extra charges which will arise from the extra work that the software supplier will need to carry out on the package. These charges will arise from the following:

- His staff will have to discuss the changes with you and produce a specification for you to agree.
- The supplier will have to consider the effect of the change on the rest of the system.
- Programming and testing of the whole package.
- Changes to documentation and provision to support your non-standard version of the package.

So if you can possibly be flexible and fit your requirements to the package, then you will be making life much easier and cheaper for yourself.

Choosing a software package can be more confusing than choosing a computer, particularly as, unlike, hardware, there appear to be very few good software reviews published. You have one big advantage in that you are now dealing with a problem that you, as a businessman, can understand. If the software supplier starts talking about bits and bytes in answer to a simple question such as 'Can I invoice by quantity as well as weight?', then there is a good change that he does not really have the business side of the package clear in his mind.

There are a number of simple points you should consider in selecting a package which initially appears to do what you require.

- Has the software writer — this may not be the same person as the one who sells it — a commitment to this product? If it's written by an amateur, then you can expect amateur back-up.
- Are there any other users that you could see or talk to?
- Is there any documentation for the system?
- Make sure that the package has actually been written.
- Will you receive any updates of the package without charge?

One thing you must remember is that once you have installed a package and transferred a large amount of data to the computer, you will find it very costly and difficult to change to another package. I suggest that you should start with a small section of your business, both to prove the package and to become familiar with its operation and impact on the rest of your organisation.

Do-it-yourself unless you have considerable experience in computers — not just writing noughts and crosses on a cassette-based system — this is extremely risky. Do not believe the people who maintain that anyone can write an invoice system in Basic in two weeks. Like all things, you may do the first 80 percent in this time, but it's the other 20 percent which will take the major effort and it is this 20 percent which, if not done correctly, is going to plague you for the rest of the life of the system.

Before you go ahead remember two things:

- Who is going to run your company while you learn to program, write and test the system.
- Playing with a computer is great fun, but having to use it and then live with the results for the next few years is another thing altogether.

There are people who have done it and the results have been very good, but they are few and far between.

Of these three steps, the last one is by far the most important. You will need to define the starting date and period of the test with great care; for example, with an accounting system either month- or year-end would be a good starting point. A month is the minimum running time to obtain comparative figures.

It is probable that at the end of the parallel run there will be inconsistencies between the two systems. This may be as a result of a deliberate change in working in the new system but, whatever the reason, this difference must be reconciled. Not finding an error until the auditor points it out tends to make life rather fraught.

(continued on next page)

(continued from previous page)

One last warning: the parallel run will involve both you and your staff in a good deal of extra work, so choose a slack period rather than in the middle of holidays or Christmas.

However good your testing, there will still be problems, so keep an eye on the system for the first few months. Most of the problems you encounter will be small and you will be able to solve them, but you must keep a written log as at some time you may decide it is worth spending some money on reviewing the system.

Take into account the views of your staff who are using the computer and remember you may find you should change some of the routine manual procedures on the periphery of the computer to speed up the system.

Summary

Microcomputers are here to stay and, for the small businessman, will provide a powerful and sophisticated method of avoiding much of the tedium of running the business and free him and his staff to concentrate on the purpose of the business and its profitability.

It is also obvious that packages are the only answer for the small computer user. It is quicker and faster to change people and routines to fit into a package than it is to have the software changed to your own requirements. This fact just cannot be overstated.

Four final points:

- If you can't see it work, then it's a good bet it doesn't, so have all promises in writing.
- If you are using any form of financial programs, then let your accountant and/or auditor vet them — essential in the case of limited companies.
- If you think you and your problems are exceptional, then you may well be mistaken. Try looking at them in a different light — it could save you a good deal of money.
- Do you really need a comuterised system now or would a minor re-organisation of the present system employed satisfy the requirements — consider obtaining outside professional advice. It could save money and time.

Purpose-built software

There are many companies and individuals who will supply you with a purpose-built system. This may be hardware and software combined, or just software. For anything other than a trivial system, this is going to cost you a great deal of money. The software cost of a decent system will nearly always be more than you paid for your hardware, particularly if you have reliable advice and system development.

Qualified analysists and programmers cost a good deal of money — in excess of £400 a week in the London area. Add this to the cost of the systems investigation, testing, documentation and support, and you can see how the costs mount. In addition, both you and your staff will need to devote some time to the project, to ensure that the software writers are doing what is required.

Unforeseen problems

If you still decide to go ahead, you must prepare a good detailed specification of your requirements and submit it to a number of software companies for a fixed-price quote on time and cost.

There is, of course, no such thing as a fixed-price contract. As both you and the software people start looking at the nuts and bolts of the system, there will be unforeseen problems and new requirements, so unless your specification was excellent or you are prepared to accept the odd system problems, you should assume that there will be an increase in the cost. To limit this increase you should insist that the software house supplies you, at regular intervals and in writing, at least the following information:

- How much the project has cost so far.
- The cost and time needed to complete system if the system design is frozen and all programming completed.
- The cost and time needed to finish the system including all the changes envisaged.

There are several ways to reduce some of these costs associated with one-off software:

- If you think that there are other companies who could use your proposed system, then you could suggest to the software company that you enter into some form of financial arrangement with them and market the system, either yourself or in configuration with them.
- The converse of this is to enquire within your trade/professional association to see if anyone is already using a system which they might be prepared to sell to you.
- Certain computers have very high-level languages that allow a user, with a small amount of learning, to write his own systems. It is unlikely that you will be able to design the system to fulfil your exact requirements, or that it will run very quickly, but this method will become more and more popular as these type of programming languages become more available.

Table 3. Unpleasant truths to think about in the real world of computing.

● Don't expect the micro to work on the same power circuit as a power drill or compressor because it probably won't. You will either have to intall a 'clean' line or install £250's worth of isolation transformer, or both.
● If your office is a shed that reaches the same temperature as the outside air, then expect trouble. The adverts say that micros don't need air conditioning; this is true, but you should find out just what temperature and humidity range your machine, and in particular the disc drive, will stand.
● There is also the problem of temperature gradient — there is no point coming in to a freezing office on a Monday morning and warming up the case of the computer with a fan heater and expecting it to work. The machine and the discs should be warm all the time; if there is any risk of condensation, then on no account use the disc drive.
● The price of continuous stationery that is prepared to your specification can be very high, so see if your dealer can help you out in any way. If you want to use your existing — i.e., non-continuous — stationery, then be sure that your printer is pressure-fed or has a form-feed option.
● Don't expect the office dumbo to be able to use the system without training. Even the best-documented system requires that the person using it be aware of what the computer is trying to accomplish in your business.
● You will need to spend a considerable amount of time growing used to the computer's impact on your company and in transferring all the initial data which will be required to start the system running.
● It will go wrong, one way or another and you must assume that you could be without the machine for at least 24 hours and probably more. Secure a good maintenance contract and ensure that you have a manual back-up system which can both take over from the computer and provide an easy method to re-start when the machine is repaired.

Implementation of your solution

Professional computer users have for a long time used a well-tried method of getting a new system up and running. The method breaks down into the following steps:

- **Individual program testing:** each program is tested with test data to ensure that it functions in accordance with its specification.
- **System testing:** the flow of data through the programs is checked to ensure that the data from one program is suitable for input to the next program and that all file lay-outs are compatible. The system test should, if possible, use real data from the existing procedures to test the validity of the computer system as a whole.
- **The parallel run:** this, as the name implies, means running the old system and the new system side-by-side using the same data as input. Doing this ensures that both systems agree, that your staff can run the new system and that it operates with success in the real world.

Cobol means business

ONE REASON for the popularity of Cobol is that there are a great many Cobol programmers but the main reason is that Cobol is designed specifically for business.

So far most users of microcomputers have used scientific languages, notably Basic. To understand the form of a Cobol program, it is useful to consider the difference between scientific and business computing. The main difference is subtle but important — scientific computing is mainly concerned with calculations and regards data as something on which to perform calculations, while business computing is mostly concerned with data.

Most business systems can be considered as having three parts — data input, file updating and reporting. There will usually be large volumes of input which consist of mixed records of varying length. Cobol has a structure which reflects these requirements exactly. It uses long mnemonics which make it easy to understand and it is designed to allow easily-followed structured programming.

Comparisons

For example, while a Basic program would read

```
LET  X3 = X3 — C2
LET  Y4 = C2 * X5
```

a Cobol program would read:

```
SUBTRACT TRANS-QTY FROM MFILE-
STOCK-QTY.
MULTIPLY TRANS-QTY BY MFILE-
UNIT-PRICE GIVING COST.
```

One of the most popular Cobol compilers available on microcomputer is the Micro Focus CIS Cobol. CIS Cobol is available for several machines, most notably it will run under the CP/M operating system. CIS Cobol is designed specifically for interactive working, which must be one reason for its success. Micro Focus tell me that 70 percent of their customers use CIS Cobol to run mainframe programs on microcomputers.

FOR MANY YEARS Cobol has been the most popular programming language for business computing. PL/1 and, more recently, Pascal have challenged Cobol but so far without success. John McMillan explains the language and describes how to understand a Cobol program.

The first thing a Cobol programmer learns is that a Cobol program is split into four divisions. Each division performs separate functions. The divisions are known as identification, environment, data and procedure — the last two are the most important.

The identification division does little other than identify the program. I will not bore the reader with details of the identification division, but there is an example of one in figure 1, lines 10-30. The environment division provides information relevant to the host computer and reflects the fact that there is a defined standard — American National Standards Institute (ANSI) — for Cobol, so Cobol programs are reasonably portable. In theory, if a Cobol program has to be transferred to a different computer, only the environment division has to be re-written.

The environment division is split into three sections. The configuration section describes the host computer. The special names section gives information on special uses of standard words. It also provides program switches which can be useful and allows the programmer to specify a currency symbol and decimal point, if the default values are to be overridden.

The third section in the environment division is input-output. It consists mainly of file control. File control defines every file to be used by the system. Cobol files are called by names, rather than channel numbers, and the line printer is considered to be a file. For a disc file, file control specifies the file organisation, and if the file is indexed, identifies the key.

Most important

As mentioned earlier, the data and procedure divisions are the most important. The data division defines all the data to be used in the program, and the procedure division contains the actual procedure or calculations. Before I describe these divisions, I need to explain Cobol syntax.

Cobol is designed to be self-documenting — the program explains what it is doing. For this reason, all Cobol statements are constructed from words. A word is a string of characters separated by spaces. Words form sentences, each sentence being terminated by a full stop.

Statements

In the procedure division, each command is called a statement, as in Basic. Each statement is started by a verb. A procedure division sentence can contain several statements. As an example, a sentence could consist of the single statement:

MOVE A TO B.

Figure 1.

```
** CIS COBOL V4.2                 STOCK1.DEM                        PAGE: 0001
**
000010 IDENTIFICATION DIVISION.                                          0118
000020 PROGRAM-ID.                         STOCK-FILE-SET-UP.            0118
000030 AUTHOR.                             MICRO FOCUS LTD.              0118
000040 ENVIRONMENT DIVISION.                                             0118
000050 CONFIGURATION SECTION.                                            0118
000060 SOURCE-COMPUTER.                    MDS-800.                      0118
000070 OBJECT-COMPUTER.                    MDS-800.                      0118
000080 SPECIAL-NAMES.                      CONSOLE IS CRT.               0118
000090 INPUT-OUTPUT SECTION.                                             0118
000100 FILE-CONTROL.                                                     0118
000110     SELECT STOCK-FILE ASSIGN "STOCK.IT"                           0176
000120            ORGANIZATION INDEXED, ACCESS DYNAMIC                   0176
000130            RECORD KEY STOCK-CODE.                                 0176
000140 DATA DIVISION.                                                    01A5
000150 FILE SECTION.                                                     01A5
000160 FD  STOCK-FILE  RECORD CONTAINS 32 CHARACTERS.                    01A5
000170 01  STOCK-ITEM.                                                   01A5
000180     02  STOCK-CODE                  PIC X(4).                     01A5
000190     02  PRODUCT-DESC                PIC X(24).                    01A9
```

(continued on next page)

where MOVE is the verb. The statement is the equivalent of the Basic:

LET B = A

So for simple uses, Cobol is no more complicated than Basic.

A Cobol statement, however, contains several clauses which extend the processing of the statement. For instance, the multiply statement has the simple form

MULTIPLY A BY B.

which corresponds to LET B = A * B. However, up to three extra clauses may be added to the multiply statement so we could have

```
MULTIPLY A BY B GIVING C.
MULTIPLY A BY B GIVING C ROUNDED.
MULTIPLY A BY B ON SIZE ERROR.
   GO TO STEP-7.
```

Cobol operands can, like Basic operands be either field names, literal strings or numbers. Field names can contain up to 30 characters which is one of Cobol's most useful features. A good Cobol programmer will always select field names which describe the field, such as EMPLOYEE-NUMBER or HOURS-WORKED.

Large programs

Many installations use standards to make maintenance easier. Typically, each field name will be preceded by an identifier showing where the field originated, for example, L53-ITEM-NO could be read from an input transaction while LIO-TERM-NO is from a stock file, L21-ITEM-NO from a transaction file and so on. The usefulness of this is immense during program testing.

The long data names explain why Cobol is preferred for large programs. It is difficult to keep track of much more than 12 two-character mnemonics. The largest program on which I have worked used considerably more than 200 fields. Imagine trying to debug a standard Basic program with that many fields.

Having thus discussed syntax, we can now study the data division. It is divided into three sections, file, working-storage, and linkage. The file section provides information about the files and describes every record in the files. A file is, of course, divided into records, each record containing a number of fields. A business file may contain a mixture of records.

Let us look at any example of a record on a stock file containing an eight-character item number, a six-digit stock quantity, a four-character unit field, a selling price up to £999.99 and the date the record was last updated. A picture of the record would look like this:

STOCK-RECORD			
ITEM-NO			
STOCK-QTY			
UNIT			
ITEM-PRICE			
LAST-DATE	DAY	MONTH	YEAR

This shows exactly how the Cobol data division holds data. The programmer assigns a level to each field, the highest level being 01. To program the example, the record STOCK-RECORD would have level 01. The next level, ITEM-NO, STOCK-QTY and so on would have a lower level, say, 03, while the fields below LAST-DATE could have level 05.

Before a program can read this record, it needs to know the length and format of each field. The PICTURE clause is used for this purpose. To describe the field ITEM-NO, the clause PICTURE IS X(8) is required indicating the field is eight-characters long. Usually this will be abbreviated to PIC X(8).

To describe the record to the program, the following data division entry is required:

```
01 STOCK-RECORD.
03 ITEM-NO          PIC X(8)
03 STOCK-QTY        PIC 9(6).
03  UNIT            PIC X(4).
03 ITEM-PRICE       PIC 999V99.
03 LAST-DATE
   05 DAY           PIC 99.
   05 MONTH         PIC 99.
   05 YEAR          PIC 99.
```

This is difficult to describe but is easy to use. Coding the data division is difficult only if you try to design the program after you start writing it.

The advantage of the multi-level approach to data is the ease with which part-fields may be addressed. This approach is neater than the Basic substring facility and has the advantage that groups of fields may be moved around with single statements.

If a number is stored in a packed BCD format, the clause USAGE iS COMPUTATIONAL is added to the field description. This is usually abbreviated to COMP. An example is:

```
02 QUANTITY      PIC 9(6) COMP.
```

Useful feature

As the printer is also considered to be a file, it is in the data division that report lines are defined. A useful feature for reporting is the data name FILLER. FILLER defines part of a record which is not used in processing. FILLER is often used with the VALUE clause, so a heading could be built-up:

```
01-HEADING.
02 FILLER          PIC X(8) VALUE
                   SPACES.
02 FILLER          PIC X(20) VALUE
                   "STOCK SUMMARY
                   REPORT".
```

The PICTURE clause can be used to edit fields. Editing is very powerful in Cobol and includes possibilities such as zero suppression, insertion of decimal points and commas in numbers, insertion of asterisks for cheque protection and

(continued on next page)

(continued from previous page)

```
         000200     02  UNIT-SIZE                   PIC 9(4).                 01C1
         000210 WORKING-STORAGE SECTION.                                      01C7
         000220 01  SCREEN-HEADINGS.                                          01C7 00
         000230     02  ASK-CODE PIC X(21) VALUE "STOCK CODE      <    >".    01C7 00
         000240     02  FILLER   PIC X(59).                                   01DC 15
         000250     02  ASK-DESC PIC X(41) VALUE "DESCRIPTION     <           0217 50
         000260-    "         >".                                             0217 50
         000270     02  FILLER   PIC X(39).                                   0240 79
         000280     02  ASK-SIZE PIC X(21) VALUE "UNIT SIZE       <0000>".    0267 A0
   (2) ▶000290  01  ENTER-IT REDEFINES SCREEN-HEADINGS.                       01C7 00
         000300     02  FILLER                  PIC X(16).                    01C7 00
         000310     02  CRT-STOCK-CODE          PIC X(4).                     01D7 10
         000320     02  FILLER                  PIC X(76).                    01DB 14
         000330     02  CRT-PROD-DESC           PIC X(24).                    0227 60
         000340     02  FILLER                  PIC X(56).                    023F 78
         000350     02  CRT-UNIT-SIZE           PIC 9(4).                     0277 B0
         000360     02  FILLER                  PIC X.                        027B B4
         000370 PROCEDURE DIVISION.                                           0000
         000380 START-UP.                                                     001A
   (3) ▶000390      DISPLAY SPACE.                                            001B
         000400      OPEN I-O STOCK-FILE.                                     001E
```

(continued on next page)

(continued from previous page)

floating-currency symbols. I have so far described the file section of the data division.

The other two sections in the data division are working storage and linkage. Working storage is used to define each working field in the program and linkage is used to transfer data between modules — this is an advanced feature for very large programs. Both sections have the same format as the data division.

One feature likely to arise in the working storage section is the OCCURS clause. In its simplest form, it has the effect of creating an array. However, by using OCCURS, a whole group of fields can be repeated. For example, a statement could be specified:

```
01 STATEMENT-PAGE.
02 STATEMENT-LINE OCCURS 15 TIMES.
03 DATE          PIC 9(6).
03 REF           PIC X(4).
03 AMOUNT        PIC 9(6) COMP.
```

Each field can be referred using a subscript, e.g.,

```
DATE (I), AMOUNT (N).
```

The main commands in any programming language perform arithmetic, control and file handling and Cobol is no exception. Cobol is more restricted than Basic in its arithmetic because it lacks the mathematical functions provided with Basic. However, as I mentioned earlier, a Cobol statement is more powerful than its Basic equivalent because extension clauses may be added to a statement. Thus we could write

```
ADD A TO B GIVING C
ON SIZE ERROR GO TO PARA-3.
```

These extensions enable, among other things, the programmer to keep control of error conditions.

The arithmetic of Cobol reflects its business use. Floating point numbers are not used, precision being specified instead by data division entries. This makes sense when you consider how many Cobol programs perform financial calculations. A banking program, for instance, will want all balances to be accurate to a penny even though it may cater for accounts holding a £100 million.

On the other hand, it will probably calculate interest to a tenth of a penny and use another field that can only contain numbers from 0 to 1,000.

There are six arithmetic verbs in Cobol, ADD, SUBTRACT, MULTIPLY, DIVIDE, MOVE and COMPUTE. COMPUTE allows arithmetic expressions, almost identical to Basic ones, to be written.

The control statements in Cobol are similar to those in Basic but increased power is obtained by extensions to the Cobol statements. One difference in control is that a Cobol procedure division is divided into paragraphs. A paragraph is given a name which, like a data name, consists of up to 30 characters. A paragraph is introduced by a single sentence containing only the paragraph name. Examples can be seen in lines 380, 420 and 540 of figure 1. The use of paragraphs aids modular programming methods.

The Cobol GO TO verb is the same as Basic's GO TO except that in Cobol, control may only be transferred to a paragraph. GO TO ..... DEPENDING ON... corresponds to ON ..... GO TO.

One of the most useful statements in Cobol is PERFORM. In its simple form, this causes a paragraph to be performed and has the same effect as the Basic GOSUB statement. PERFORM may be extended, however, by the THRU clause to perform a number of consecutive paragraphs. The UNTIL, VARYING, FROM AND BY clauses provide the same facilities as Basic's FOR..... NEXT loop, for example:

```
PERFORM A THRU B
VARYING J FROM 1 BY 1
UNTIL J = N
```

The Cobol IF statement is more powerful than the Basic IF. After the IF test, a number of statements may be performed, and an ELSE clause may be added. It is also possible to use the IF statement to validate data, thus enabling the programmer to keep control after errors in input data. For example:

```
IF TRANS-QTY NOT NUMERIC
MOVE 3 TO ERROR-CODE
PERFORM REJECT-TRANS
GO TO NEXT-TRANS
ELSE MOVE TRANS-QTY TO
STOCK-QTY.
```

Note the full stop — this terminates the IF condition.

Cobol's file handling is similar to that of Basic. The OPEN and CLOSE verbs have the same function as their Basic equivalents. READ causes a record to be read from a file and WRITE causes a named record to be output. When printing, the WRITE verb can be extended to throw pages, skip lines or perform specified actions when page overflow is detected.

Of the commoner verbs, two I will mention are ACCEPT and DISPLAY. They normally enable communication with the console, but in CIS Cobol they are used for all-purpose screen handling.

ACCEPT and DISPLAY are used like READ and WRITE in CIS Cobol. Whole screen lay-outs are specified in working storage, as can be seen in lines 220 to 360 of figure 1. The DISPLAY verb causes a screenful of information to be displayed and the ACCEPT verb causes the program to wait until the Return key is struck and then input specified data.

In figure 1, the record SCREEN-HEADINGS is used to send the following headings to the VDU screen:

```
STOCK CODE < >
DESCRIPTION <      >
UNIT SIZE <0000>
```

Then the record ENTER-IT overlays the input fields on the headings. The REDEFINES clause is used to achieve this. The input record — for the ACCEPT verb — defines fields into which data may be input. Other fields are called FILLER and are protected which means that data may not be input into them.

(continued from previous page)

```
    (4) ▶000410      DISPLAY SCREEN-HEADINGS.                                    0022
         000420 READ-INPUT.                                                      0033
    (5) ▶000430      ACCEPT ENTER-IT.                                            0034
         000440      IF CRT-STOCK-CODE = SPACE GO TO END-IT.                     004B
    (6) ▶000450      IF CRT-UNIT-SIZE > 999 GO TO READ-INPUT                     0055
         000460         ELSE IF CRT-UNIT-SIZE < 1 GO TO READ-INPUT.              0062
         000470      MOVE CRT-PROD-DESC TO PRODUCT-DESC                          0076
         000480         MOVE CRT-UNIT-SIZE TO UNIT-SIZE                          0076
         000490         MOVE CRT-STOCK-CODE TO STOCK-CODE.                       007C
         000500      WRITE STOCK-ITEM INVALID KEY GO TO READ-INPUT.              008A
    (7) ▶000510      MOVE SPACES TO ENTER-IT MOVE ZERO TO CRT-UNIT-SIZE          0096
         000520         DISPLAY ENTER-IT.                                        009C
         000530      GO TO READ-INPUT.                                           00BA
         000540 END-IT.                                                          00BD
         000550      CLOSE STOCK-FILE.                                           00BE
         000560      DISPLAY SPACE.                                              00C2
         000570      DISPLAY "END OF PROGRAM" UPON CONSOLE.                      00C5
         000580      STOP RUN.                                                   00DC
         ** CIS COBOL V4.2 COMPILER COPYRIGHT (C) 1979 MICRO FOCUS LTD URN AA/0000/AD
         **
         **ERRORS=00000 DATA=00636 CODE=00255 DICT=00392:19791         END OF LIST
```

MEET THE FAMILY

Exidy inc.

. . . THERE MUST BE ONE HERE FOR YOU

The Exidy family is a sophisticated range of products designed to meet the ever increasing variety of computing needs including home and business uses, educational and laboratory applications, industrial process control, etc.
The Exidy range has been designed with built-in growth capacity to take the risk factor out of computer investment. The Sorcerer Computer now has a memory expansion capability from 8K up to 48K within its own cabinet, enabling the system to grow with your needs plus the ability to add many additional plug-in facilities such as disk drives, printers and pre-programmed ROM PAC's. All Exidy products are backed up by a 12 month warranty covering both parts and labour.
The Exidy Sorcerer provides a standard typewriter keyboard plus a 16 key numeric pad. Connections are provided for future expansion within the keyboard unit.
If you want an accounting machine, a word processor, or a program development system, the SORCERER offers all of these within the same unit.
The Word Processing ROM PAC allows you to create, edit, re-arrange and format text. The operator's work is displayed clearly on a screen—you can see the whole of an average business letter—correct mistakes—re-arrange it. Once perfected—press a key and the printer will type a perfect letter in record time—as many times as you like—no wasted letterheadings. Features include auto wraparound, dynamic curser control, variable line length, global search and replace, holding buffer for re-arrangement of text, right justification, line width and line to line spacing, underlining or boldfacing, text merging and a macro-facility permitting tasks such as formletter typing, multiple column printing or automatic forms entry.

Now contact your nearest dealer:

The NORTH
BASIC COMPUTING, Oakville, Oakworth Road, Keighley, W. Yorkshire. 0535 65094
E.S. MICROCOMPUTERS, 7 Berkeley Prec., Ecclesall Road, Sheffield S11 8PN. 0742 668767
GLYSTHYDON LTD., Nook Farm, Rake Lane, Warton, Preston, Lancs. 0772 633138
MAGNUM CONSULTANTS, 57 Fairburn Drive, Garforth, Leeds. 0532 867892
MICROPUTE, 7 Westbourne Grove, Manchester M20 8JA. 0625 612818
MICRODIGITAL, 25 Bruswick Street, Liverpool L2 0BJ. 051 227 2535
TW COMPUTERS LTD., 293 London Road, Hazel Grove, Stockport. 061 456 8187

MIDLANDS
M&R SYSTEMS, 20 Norwich Road, Wisbech, Cambs. 0945 5900
MIDLAND MICROS, 1 Cherry Wood Drive, Aspley, Nottingham. 0602 298 281
GRIFFIN & GEORGE, Birmingham, Manchester, London, Eastkilbride. 01 997 3344

LONDON and SOUTH EAST
E.M.G. MICROCOMPUTERS, 30 Heathfield Road, Croydon, Surrey CR0 1EU. 01 688 0088
HOME & BUSINESS COMPUTERS, 445 High Street North, Manor Park, London E12. 01 472 5107
MICROBITS, 34b London Road, Blackwater, Camberley, Surrey. 0276 34044
SLOUGH MICROSHOP, 120 High Street, Slough, Berks. 0753 22855
TVJ MICROCOMPUTERS, 165 London Road, Camberley, Surrey GU15 3JS. 0276 62506
C.C.S. MICROSALES, 7 The Arcase, Letchworth, Herts. N2 9EN. 046 26 73301
G.P.W. ELECTRONICS, 146a London Road, North End, Portsmouth, Hants. PO2 9DJ. 0705 693341
INFORMEX-LONDON LTD., 8-12 Lee High Road, London SE13. 01 318 4213
NIC, 61 Broad Lane, Tottenham, London N15. 01-808 0377

WALES and the WEST
TRYFAN COMPUTERS, 3 Switfts Blds, High Street, Bangor, Gwynedd LL57 1UW. 0248 52042
ELECTROPRINT, 5 Kingsdown Parade, Bristol BS6 5UD. 0272 292375
LIVEPORT DATA PRODUCTS, The Ivory Works, St. Ives, Cornwall. 0736 798157
TVJ MICROCOMPUTERS, 48 Gloucester Road, Bristol BS7 8BH. 0272 422061

Or send coupon for further information to:

SOLE DISTRIBUTORS
Geoff Wilkinson
LIVEPORT DATA PRODUCTS
The Ivory Works, St. Ives, Cornwall.

PLEASE SEND DETAILS OF THE EXIDY RANGE

NAME ____________________

ADDRESS ____________________

____

____

____

____

• **Circle No. 183**

Communication is problem in programmed control

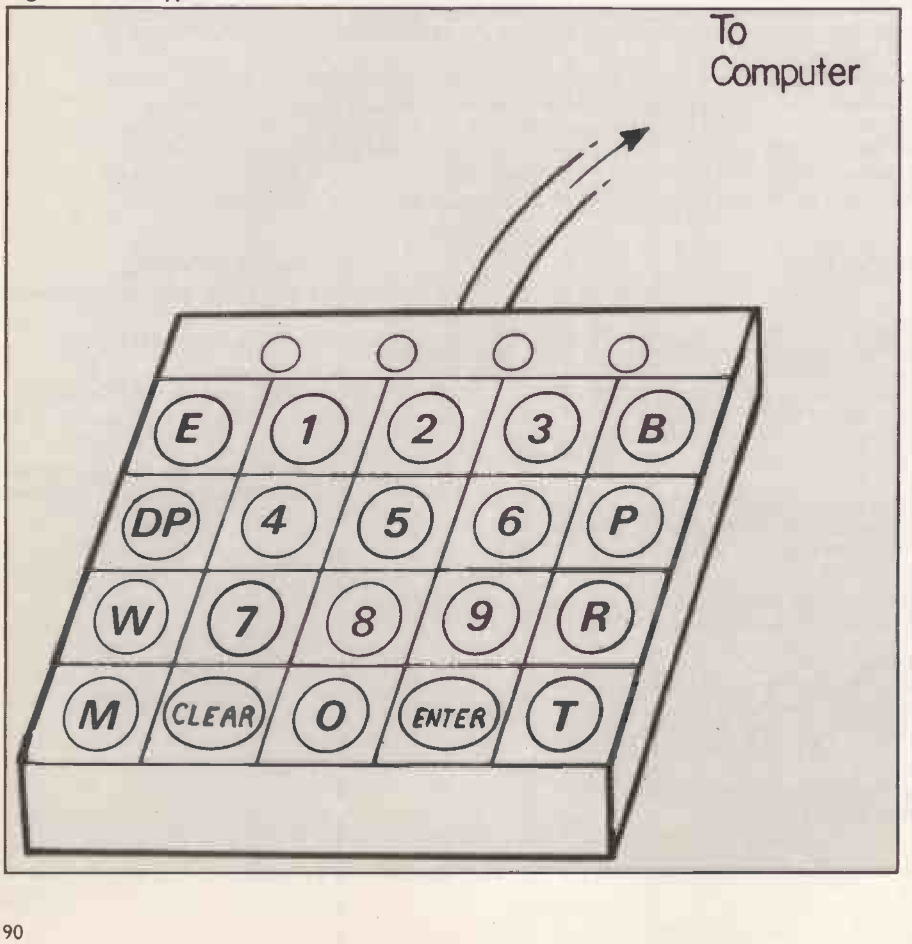
Part four of Mark Witkowski's series concerns itself with the programmed control of industrial-style robots and problems of the communication of ideas from the user to the machine.

HOWEVER BRILLIANTLY designed and constructed, and whatever sensory provision is made on a robot device, its usefulness, usability and performance is ultimately dependent on the control algorithms used. Not that one would want to spend a year writing code — assume that robots are all computer-controlled which is, of course, untrue — to compensate for an unstable mechanism, or trying to guess that the world is doing because the robot is insufficiently instrumented. Generally a good robot is made usable by virtue of its programming, a lesser one may be saved.

Deterministic approach

In some cases, the software is so central to an idea that the robot is not built at all, just simulated on a computer graphics terminal. Programming of robots tends to fall into two categories, first the deterministic approach, in which the robot is programmed with specific actions known to perform a given task. New ways are always sought to program robots with the minimum of effort for the maximum effect and some have met with success.

Figure 1c. V/I keypad.



When direct teach mode instruction becomes too cumbersome, programming languages are developed to describe the problem and its solution. As there is an incredible mass of detail in the most trivial of everyday tasks, these languages are being developed continually and re-structured to cope. This first category is the province of the industrial robot and industrially-orientated robot research.

The second software category falls within the bounds of artificial intelligence research, where the emphasis is on robot problem solving, and where instead of being instructed in minute detail, only a broad outline or the final goal need be specified, possibly in a natural language. Artificial intelligence techniques are being incorporated slowly into robot programming languages as the tasks the robots are required to perform become more complex and the current intuitive methods are found to be inadequate.

Currently, unfortunately, each robot and manipulator tends to have its own teach technique or programming language. There has been little conformity and standardisation, with no universally-accepted language. Unlike different types of computer which, even though they have distinct order-codes, have standard-

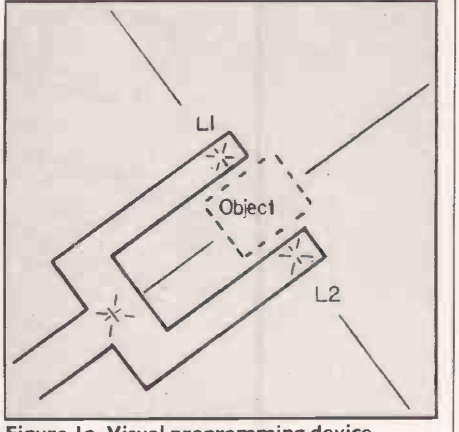


Figure 1a. Visual programming device.

ised user languages, manipulators are still sufficiently diverse in design to defeat the compiler writer.

However, numerically-controlled machine tools have been programmed in APT for years — ITT research institute, 1967 — different numerically-controlled machines are catered for by post-processing a universally applicable intermediate codeform from a single compiler into the specific control signals required.

Simplest method

By far the most straightforward method of programming an industrial robot is to manhandle it through the desired sequence of actions. Continuous path robots, as used in paint spraying or welding, are effectively programmed by a skilled human operator leading the arm's spray/welding head through a complete spray or welding job with the actuator power turned-off. The joint position sensor values are recorded at frequent intervals, either in computer memory, or on tape. When the job is completed to the operator's satisfaction the power can be re-applied and the robot will repeat the operator's actions exactly when the stored data is re-played.

Assuming that one has taken the precaution of placing a new workpiece in precisely the position and orientation of the original, the robot will do as good a job as the man did.

With pick-and-place-type robots, the arm is moved to a series of significant positions in the sequence of actions with a joystick control. Although there are many possible designs for this type of control unit, in principle there will be a switch for each of the degrees of freedom, and a teach button. In a typical application, the arm may be required to move to a component feeder, grasp an item, move to a press and deposit it in place, move out of the way, pick-up the piece after stamping, deposit it in an outgoing hopper and finally return to the feeder to collect the next blank.

This process will involve many discrete steps and the manipulator is moved to each using the multi-switch control. When it is aligned perfectly at each point in the cycle the teach button is pressed, and the joint positions recorded. This is repeated for each significant point, and there may be many before the cycle is re-played to check the sequence.

Specialisations

As the main control unit will compute a straight-line trajectory between the points during playback, it is essential for the user to define sufficient intermediate points to avoid obstacles — none of the actions made between teach points is stored.

There are a number of possible specialisations to this mode of robot instruction by teaching. Figure 1a shows a Visual Programming Device (VPD) used to program the University of Rhode Island (URI) five-degree-of-freedom arm which is shown in figure 1b — Birk and Kelly, 1976, and Kelly and Silvestro, 1977. A computer-compatible TV camera views the base upon which the objects the robot will manipulate are placed. When the VPD is placed on the base, it is possible to calculate the co-ordinates and orientation of the two lights, L1 and L2, from the TV image. The VPD is placed round the object to be grasped and the 'P' button pressed on the keypad — figure 1c. During the playback phase, this will have the combined effect of moving to, orientating with, grasping and lifting the selected object. 'R' has the effect moving to, lowering and releasing the object at that specified VPD position. Other commands include 'B', Begin, and define the 'home' position. 'T', move Through a point specified by the VPD, 'E', End, and move back to the home position. 'Wnnn', Wait for NN.N seconds, allowing operator intervention. Recorded or memorised points are available, TMn, RMn and PMn for Through, Release at the Pick up at the co-ordinates stored in Memory location n. This is a particularly useful feature as it is difficult to re-position the VPD repeatably by hand.

Well-suited

This form of programming is well-suited to the overhead gantry, five-degree-of-freedom manipulator used. A more general six-degree-of-freedom arm would need programming in three dimensions. Perhaps this could be done with stereo television cameras or a navigation-style position sensor.

The visual instruction scheme (V/I) is used in four stages. First the calibration phase. Two fiducial lights, to the left on the base-plate in figure 1b, are used to calibrate the camera co-ordinate-generating program. Recording the sequence of actions using the VPD and keypad is the next stage. Then the Edit/Verify mode is entered. With the aid of a single-step facility, all the points can be checked and changed if they are incorrect.

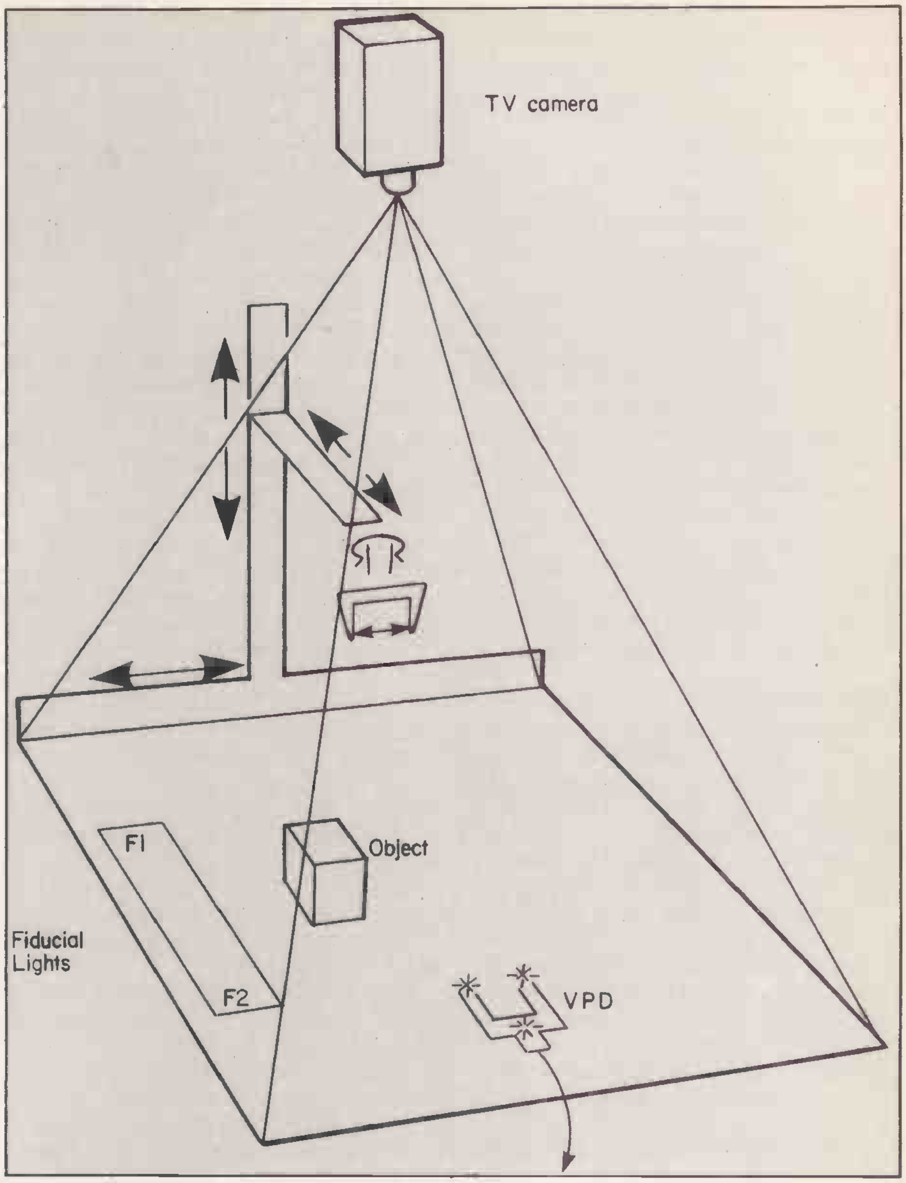


Figure 1b. URI V/I lay-out.

Misalignment can be corrected by altering one or more of the individual co-ordinate components. Points may be added, if, for instance, insufficient clearance was allowed around some obstacle or when some new sub-sequence has to be added. Points may be deleted if a path-length proves to be excessively long. When all is as it should be, the arm is put into playback mode and used.

When a manipulator is used as a disabled persons' aid, particular care has to be taken with the design and lay-out of the input mechanism. Todd (1979) describes a multi-mode input cluster with which tetraplegics may operate a manipulator in a number of different ways by head movements alone. He used a ring of 12 photocells suspended in front of a video monitor, which would operate the manipulator when a light beam projector attached to a pair of glasses frame shon on to one of the cells.

Tree structure

Information displayed on the monitor close to each of the photocells labelled their function. Certain cells would have the effect of changing the labels, and hence the effects the Z-80-based controlling microprocessor had on the manipulator. These changes were organised into a tree structure of different modes, including direct control, pre-programmed automatic picking-up and changing to a different input device.

In addition to the photocells, there was also a ring of 12 momentary push switches, arranged in the same manner as the photocells — so that the monitor labels were still useful — operated by a stick held in the mouth. There was also a joystick, operated by placing the lever in

(continued on next page)

(continued from previous page)

the mouth, which offered a multi-dimensional input mode, up/down, left/right, in/out and two levels of breath pressure. Furthermore, three switches could be operated with the side of the user's head. Some of these modes could be used together, some separately. While the patient was using the manipulator to feed himself, it would be unreasonable to expect mouth-operated control.

With any taught-sequence robot, the ability to edit, modify, add and delete actions is particularly important. In the simplest case one would just record the values of the manipulator joints in sequential computer store. The URI team has suggested that the use of a linked list is a more effective approach. Data for each point, or node, which will include a label — so it is named — a mnemonic — Move, Wait — the arm joint values — X, Y, Z, angles, grip, speed — and finally a 'pointer', the place in a memory where the next data unit is to be found — figure 2a.

Data nodes

The position in memory has no significance to the order in which the data nodes will be used, the edit/verify and playback modes will follow a line of linked pointers which the record mode set-up. Playback is a matter of taking the data from the current instruction and interpolating a path until the robot assumes the attitude specified in the successor node, which then becomes the current instruction. This is then repeated until a node with a special end-of-queue pointer "*" is found.

Deleting a node is a matter of changing the current predecessor pointer so that it points to the current successor node. The removed node's pointer is changed to point to the first node in the free list, which initially consists of all nodes, the action list is built-up during record mode by changing pointers from the free list to the action list. The special pointer that indicates the start of the free list FREE is then changed to point to the recently removed action node, figure 2b.

Adding a node to the action list is a similar matter of altering the current instruction's pointer to the top of the free list, which is contained in FREE, altering the FREE pointer to the next free node and pointing this new successor node the the old successor node as in figure 2c. Linked list storage allocation is a standard computing technique, about which more can be found in the majority of books on data structures — Knuth, 1968.

A doubly-linked list, in which a second set of pointers point from the successor to current and current to predecessors would allow the actions to be re-played or searched in reverse order. There are doubtless some instances where this would be helpful.

A majority of robots will be programmed by teaching them. It offers a number of important advantages over other methods. There is no need for the operator, who is presumably already skilled in the work the robot is to perform, to understand the intricate detail of robot operation. There is also no need for the operator to learn a specialised programming language, and the machine is ready for use as soon as it is commissioned. Program development and debugging are, therefore, accomplished in the minimum time. Furthermore, there is a minimum of sophisticated equipment in the work-area, at most keypad or joystick control, improving the potential reliability of the whole system.

There are also many disadvantages and limitations to this form of programming, while the robot's action may be performed *ad infinitum* with no variation all is well, however there are many situations in which a robot should be programmed by telling rather than showing it — Hohn, 1979, Holt, 1979.

Consider the task of picking eggs from a feeder, i.e., a fixed location, passing through some inspection processes, and finally transferring them to a carton — figure 3 — or that of picking the next item of a neatly-stacked pile, each of which is to be found at a position lower than the last.

One fairly bad solution would be to train the whole sequence explicitly. By training the fixed sequence, supply, P1, P2, P3, P4, branch, it could be saved as a macro. Then it would only be necessary to train each of the different branch paths and, after each, press the macro-expand control which substitutes automatically the stored path into the linked-list sequence action queue.

Real power

The real power a programming language gives a robot user is in relation to acting on sensory data. We didn't decide on how to describe the tests on those eggs, or how to dispose of bad ones. As soon as anything more than a few binary interlocks are considered the possible combinations of sensor tests explodes and finding some orderly way of handling the ensuing branch points, feedback loops and error recovery becomes essential.

Subroutine call or macro-expansion can, as with all computing, reduce programming effort considerably, as well as impart a much more modular, top down, control structure to the task solution, particularly where small modifications are required to a basic action sequence. Language makes the description of transformations to the already-programmed actions more powerful. It becomes possible to describe actions relative to some object or previous action, rather than absolute position, or to superimpose the motion of a conveyor belt on which the work-piece is moving.

In some circumstances, it may be desirable to rotate, expand or contract the sequence, or reflect it to give a mirror image. Consider the left- and right- hand sides of car assembly. Where absolute positioning is required, manual control of the action sequence may be insufficiently accurate or repeatable. Print statements in the language are used to provide a written log of robot activity, display messages and sound alarms when operator assistance is

Figure 2a. A linked list.

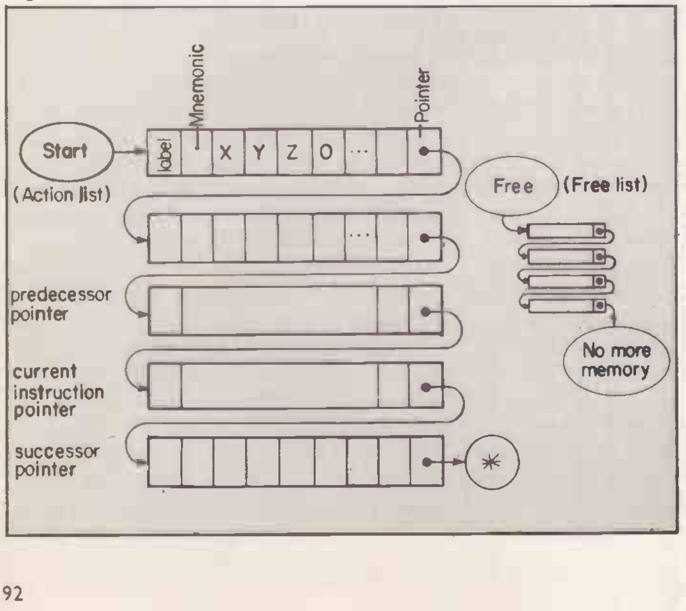
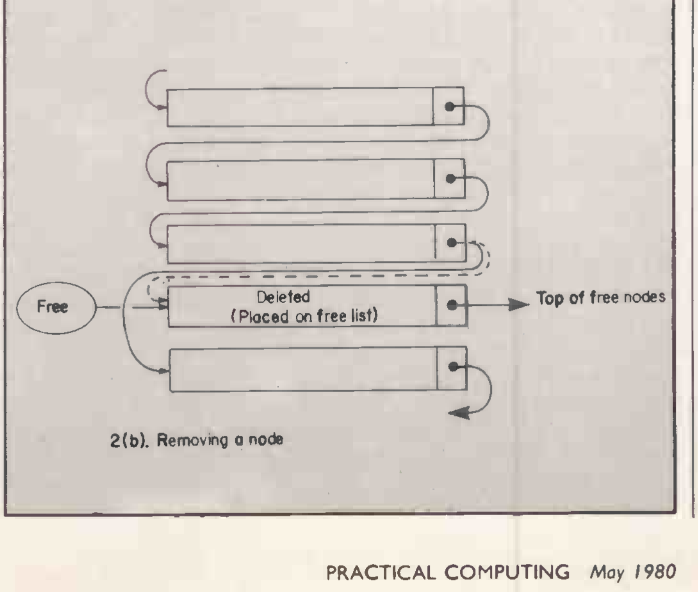


Figure 2b. Removing a node.



called for. There are many instances where the design criteria for a good robot control language are similar to those of any other type of computer language. They must allow the user to specify every aspect of the task to be performed, without being too cumbersome. Robot languages for manipulators may either describe the task in terms of robot motions or the position and transformations of the work pieces.

WAVE from the Stanford artificial intelligence laboratories — Paul, 1977 — is an example of an industrially-orientated manipulator control language. It is written as a sequence of one-line instructions, and is worth closer examination. In WAVE, an object is described by the position the manipulator must be in to grasp it. There are six items required to specify the position and they are assigned to a variable name thus:

TRANS variablename 30,20,10,0,90,0

assigns a particular position (X = 30, Y = 20, Z = 10) in co-ordinate space to the gripper with a unique orientation. The Scheinman arm at Stanford has six degrees of freedom and the latter three parameters to TRANS specify the angle of attack of the gripper completely in relation to fixed reference orientations.

Instructions

MOVE variablename

would then cause the manipulator to move from its current position and, assume the co-ordinates and orientation specified in a previous TRANS instruction, which in itself caused no action. MOVE is an absolute instruction, motions relative to the current position can be made with:

CHANGE vector1,scalar,vector2,angle,time

which moves the arm a distance specified by scalar in the direction given in vector 1, also rotating it by angle about vector 2, at an optional speed.

VECT variablename x,y,z

is used to specify a vector with x,y and z components, and can equally be used to give a direction or a force heading and value. The gripper is opened and closed with:

OPEN 5

open the gripper to five inches and:

CLOSE 1

close the gripper, the jaws will close until either physical resistance or a specified force is met by appropriate sensors. If they close more than the parameter allows, less than one inch, a well-defined error condition is generated, usually meaning that the object to be grasped was not in the expected position.

CENTER 1

centres the hand about an object using touch sensors on the insides of the fingers, without moving it. CLOSE and CENTER use sensory data inherently, whereas MOVE and CHANGE do not.

The STOP instruction may be used to abort a movement when a certain, expected, pre-condition is met. So the code:

VECT DOWN 0,0, —1

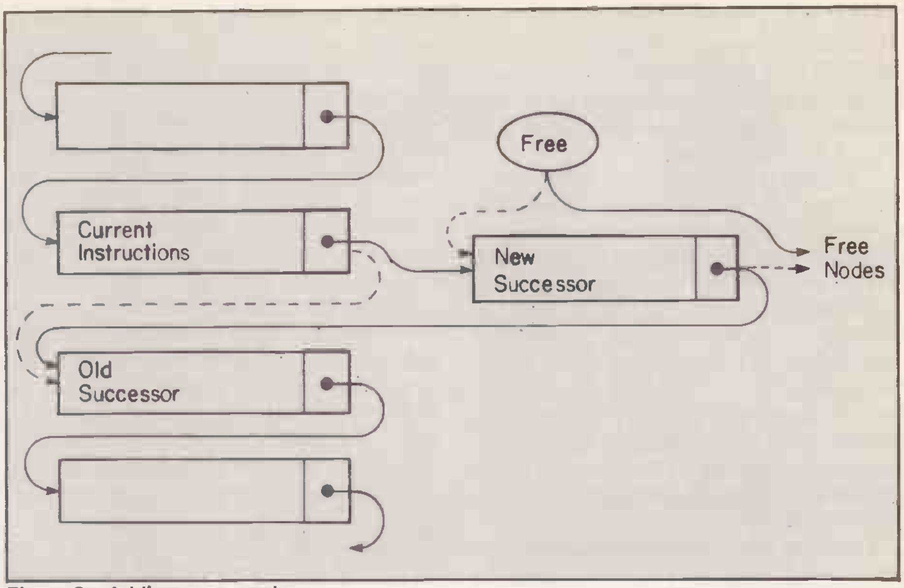


Figure 2c. Adding a new node.

VECT HIT 0,0, —30
STOP HIT,NIL
CHANGE DOWN, 10,NIL,0,0

moves the arm down a maximum of 10 inches in the Z direction only, vector DOWN times a scale factor of 10, but this stops if a force vector of 30 oz. is encountered in the minus Z direction, upwards. The co-ordinates of the obstacle that caused the arm to halt are saved in FLOOR, they may be used later by:

RESTORE FLOOR

this save and restore feature is particularly important in remembering where objects already manipulated are — rather than where they ought to be.

There are several forms of program control:

JUMP label

clearly transfers control to code at "label:". Iterative actions are controlled by the loop instruction:

SOJG loopvalue, label

which decrements the contents of the variable loopvalue and, if it is still positive, jumps to label. Loopvalue is initialised with the instruction:

ASSIGN loopvalue, n

for n times round the loop, these loops can either be used to repeat a total sequence of actions or make final position adjustment iterations using sensory data.

Error handling is by the SKIPE n instruction, skip over the next instruction if error condition 'n' is encountered. There are many possible predictable error conditions — gripper closes beyond expected amount, failed to STOP, object encountered where none should be. Alternatively, SKIPN skips the next instruction if an error is not trapped. This form of error handler is all very well, as long as the programmer is aware of what is likely to happen so provision can be made.

WAVE allows a complete program to be built-up in 'macro' modules, each of which may be tested in isolation, starting with some clearly-defined condition and exiting in an equally clear state for the next module, or on giving an error message. The 'WAIT error message' command halts the system, prints the "error message" and waits for operator intervention. Because of the uncertainties inherent in all real-world manipulations, WAVE offers a number of facilities.

SEARCH X,Y,0.1

sets-up a box search in the x and y planes, starting in the x direction, with increments of 0.1 in figure 4. This pattern of initial guess followed by a sensor driven search, or some variation, is a standard technique in robot assembly programming.

Assembly operations

In a number of assembly operations, close fitting parts can be better mated with some of the degrees of freedom released, such that they are only balanced against gravity and acceleration forces. They will then comply to external imposed forces to prevent jamming:

FREE 2,X,Y

gives translational compliance in the x and y directions.

SPIN 1,Z

gives rotational compliance.

FORCE vector

maintain the given force in the direction of the vector. To further ease the problems associated with close assembly:

WOBBLE 0.1

superimposes a 0.1in. sinusoidal perturbation on the hands' movement. These compliance and oscillatory modifiers are designed to reduce the incidence of close-fitting parts seizing together if force is applied at some angle not exactly perpendicular to the line of best fit.

The amount of processing required to convert these instructions into a form suitable to drive the arms motions is not trivial.

In this case the actual drive parameters are planned, using a model of the arms physical dimensions, possible motions and dynamic considerations in a time-shared

(continued on next page)

(continued from previous page)

PDP 10. The assembly-language-like form of WAVE is translated into arm control object program. Planning also check that the requested action is not impossible with the configuration used.

The plan is executed in a PDP 6 which interprets the object code, evaluating trajectories and acting as a six-degree-of-freedom servo, re-computing as needed where pre-planned actions have been modified by CENTER or STOP. Efficient dynamic control, particularly where a manipulator is operated close to its design tolerances, as well as other kinetic and static considerations are covered by Raibert and Horn, 1978, and Horn, 1979. Drazan and Jeffery, 1976, control a three-degree-of-freedom pneumatic arm with on/off valves.

The more recent trend in languages for manipulator control has been to bring the syntax more in line with current Algol-like programming languages. BEGIN .... END block structure, IF ... THEN ... conditionals, WHILE ... DO ... and REPEAT ... UNTIL ... control structures — Paul and Nof, 1979.

Stanford's later language AL, a successor to WAVE, also has an Algol-like structure and introduces new ideas into this problem area — Finkel et al., 1974. Al, and the MIT language LAMA — Lorano-Perez 1979 — and others — Ambler and Popplestone, 1974, are all concerned, to varying extents, with describing assembly tasks by the objects which are to be manipulated, rather than the actions actually required to perform the task.

It would be by far preferable to state the problem in English, or some subset, than describe actions. In all cases this will dramatically increase the analysis and planning stages to produce the executable plan. Consider the effects of inserting the primative instruction:

INSERT (OBJECT,HOLE)

on both computational and knowledge database requirements.

With all the advantages of programming languages, it would be elegant to also incorporate the directness of the teach mode. Gini and Gine, 1978, report on the POINTY system in which the manipulator is used to point at objects and generate data structures automatically about that item in AL. Eventually the best of these ideas will find their way on to the work-shop floor. One may even, one day, be able to program a robot in natural language — Bernorio, 1977.

Programming of mobile robots does not need the same degree of transformational arithmetic as manipulators, as they are, in effect, only two-degree-of-freedom devices. Because of this, there is almost no need for highly-specialised languages to control them — any computer language will suffice so long as the robot hardware is interfaced to the software in some logical manner. Furthermore, the robot is seldom instructed in terms of absolute co-ordinates, MOVETO X,Y; but rather in terms of relative motions, MOVEFORWARD 10 or GOLEFT UNTIL SENSOR3 > X.

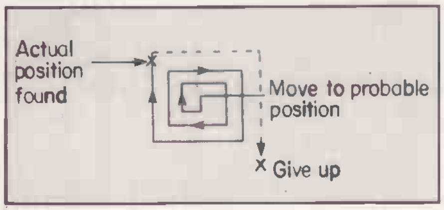


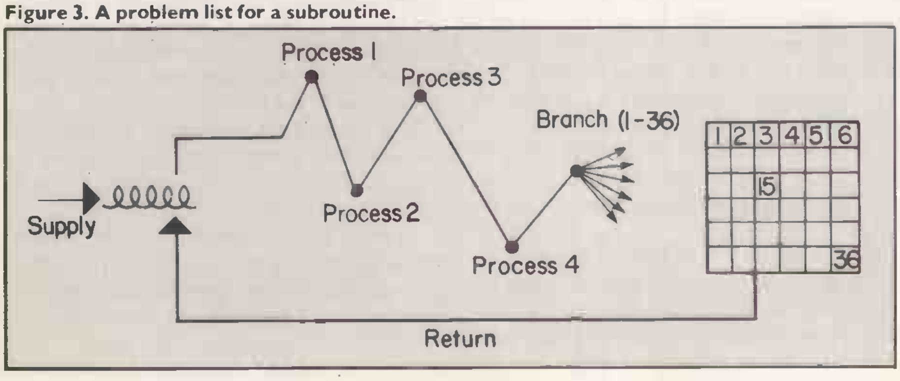
Figure 4. A box search.

Even when the algorithm functions in absolute co-ordinate space the transformation to relative motion, even if must be planned, is straightforward. The programming language LOGO has been used to teach children about various concepts in mathematics and computing using small, two-wheeled, turtles which, with a pen attached to their undersides can be programmed to draw pictures on the floor, according to programs the children write — Papert, 1971a and 1971b, and Papert and Solomon, 1971.

Less computation

As there is far less computation involved in determining the vehicle's actual path, these languages can be interpreted. Input text is scanned directly to perform the actions, whereas WAVE had to pass through a planning stage. The advantages of easy testing, editing, rapid turnaround and good diagnostics usually more than outweigh the time overheads imposed by interpretation. While the school children will see only the simpler aspects of LOGO, a full implementation of the language can be used for complex A.I. programming — bundy et al., 1978.

Figure 3. A problem list for a subroutine.



References

Ambler AP and Popplestone R J (1974) Inferring the position of bodies from specified spatial relationships in AISB Summer Conference, July 1974 held at University of Sussex. pp. 1-13.

Bernorio M, Bertoni M, Dabbene A and Somaluico M. (1977). Programming a robot in quasi-natural language in The Industrial Robot. 4-3 (September 1977) pp. 132-140.

Birk J R and Kelly R B (1976). New robot programming devices for teaching assembly, inspection, materials handling, and pallitising tasks in 3CIRT/6ISIR paper B4, pp. B4-33 to 42. International Fluidics Services Ltd, Kempson, Bedford.

Bundy A (ed.) (1978) Artificial Intelligence: An introductory course. Edinburgh University Press. ISBN 085224-340-5.

Drazan P J and Jeffery M F (1976) Microprocessor control and pneumatic drive of a manipulator arm in 3CIRT/6ISIR paper D2, pp. D2-9 to 20. International Fluidics Services Ltd, Kempson, Bedford.

Finkel R, Taylor R, Bolles R, Paul R, and Fieldman J (1974). AL, a programming system for automation. Stanford A I Lab. Memo AIM-243.

Gini G and Gini M (1978) Object description with a manipulator in The Industrial Robot 5-1 (March 1978) pp. 32-35.

Hohn R E (1979) Application flexibility of a computer-controlled industrial robot in Industrial robotics, Vol 1/Fundamentals. Mitchigan: Society of manufacturing engineers. ISBN 0-87263-045-5 pp. 177-195.

Holt H R (1979) Robot decision making in Industrial robotics, Vol 1/Fundamentals. Mitchigan: Society of manufacturing engineers. ISBN 0-87263-045-5. pp. 197-205.

Horn B K P (1979) Kinematics, statics, and dynamics of two-dimensional manipulators. In: Artificial Intelligence: An MIT perspective Vol 2 (Winston P H and Brown R H edgs.) pp. 273-308. The MIT Press, ISBN 0-262-23097-6.

IIT Research Institute (The APT long-range program staff) (1967) APT Part Programming. McGraw-Hill Book Company.

Kelly R R and Silvestro K C (1977) V/I A Visual instruction software system for programming industrial robots in: The Industrial Robot 4-2 (June 1977) pp. 59-75.

Knuth D E (1968) The art of computer programming, Vol 1/Fundamental algorithms. Addison-Wesley Publishing Co.

Lorano-Perez T (1979). A language for automatic mechanical assembly in Artificial Intelligence. An MIT perspective, Vol. 2 (Winston P H and Brown R H eds.) pp. 244-271. The MIT Press, ISBN 0-262-23097-6.

Papert S (1971a). A computer laboratory for elementary schools. M.I.T. A.I. Laboratory. Artificial Intelligence Memo no. 246.

Papert S (1971b). Teaching children to be mathematicians v. teaching about mathematics. MIT AI Laboratory. Artificial Intelligence Memo no. 249.

Papert S and Solomon C (1971). Twenty things to ao with a computer. MIT AI Laboratory. Artificial Intelligence Memo no. 248.

Paul R (1977) WAVE — A model-based language for manipulator control in The Industrial Robot 4-1 (March 1977) pp. 10-17.

Paul R L and Nof S Y (1979) Human and Robot task performance in Computer vision-and sensor-based robots. (Dodd G G and Rossol L etd.) pp. 23-50. New York: Plenum Press. ISBN 0-306-40305-6.

Raibert M H and Horn B K P (1978). Manipulator control using the configuration space method in The Industrial Robot 5-2 (June 1978) pp. 69-73.

Todd D J (1979). An investigation into the uses of a microcomputer in the control of a manipulator for tetraplegics in Engineering and Medicine 8-4. pp. 193-200.

Introducing the PET COLLECTION

... A suite of powerful business programs at a budget price – from ACT Petsoft, the professional software specialists!

1 ACT SALES LEDGER £120

Commodore Disk **£95 Cassette version**

Full facilities for the maintenance of the Sales Ledger, the preparation of a list of outstanding balances and printing of statements. All data including new customer details, invoices, credits, cash and transfers are entered under step by step guidance on the display screen. Printed results include Audit List, Aged Debtors List, Control Account and Statement.

For 32K PETs

Both Sales Ledger and Purchase Ledger are . . . OFFICIALLY APPROVED BY COMMODORE

2 ACT PURCHASE LEDGER £120

Commodore Disk **£95 Cassette version**

Full facilities for maintenance of the Purchase Ledger, the preparation of a list of outstanding balances and printing of remittance advices. The system produces the following printed results: Audit List, Aged Creditors List, Control Accounts, Purchase Ledger Record, Remittance Advice, Cheques and Payment List. For 32K PETs

PLUS

3 INVOICING WITH STOCK £75

A powerful, easy-to-use system for the CompuThink Disk, handling 1200 or 2400 stock items per diskette.

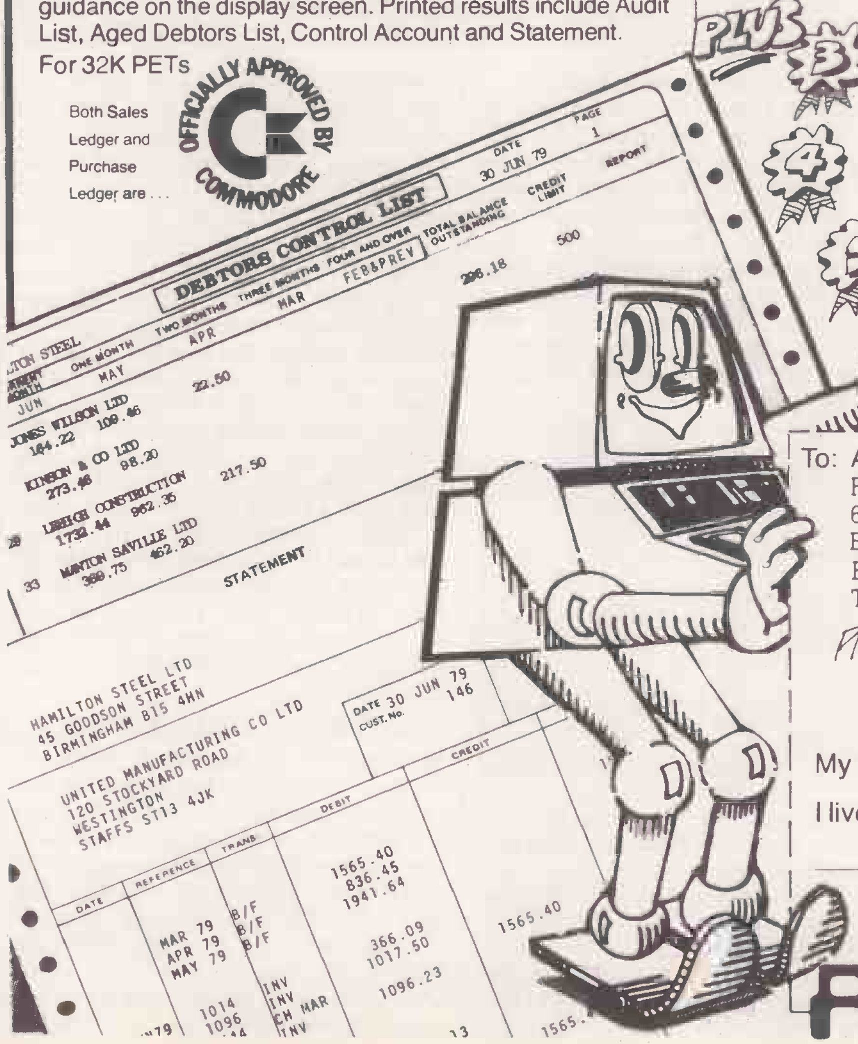
4 PAYROLL 200 £50

For up to 200 employees, on disk or cassette.

5 WORDCRAFT £325

The ultimate PET Word Processor, now on CompuThink Disk.

. . . and over 200 more business programs, games and programming aids in the NEW PETSOFT CATALOGUE.



To: ACT PETSOFT
Radclyffe House,
66-68 Hagley Road,
Edgbaston, Birmingham
B16 8PF. Tel: 021-455 8585
Telex: 339396

Please rush me information on ... 1☐ 2☐ 3☐ 4☐ 5☐ also the NEW PETSOFT CATALOGUE ☐

My Name is ____________

I live at ____________

____

Tel. No. ____________

ACT Petsoft

• Circle No. 184

Parallel ports mean faster data transfer

Attaching parallel printers to micros can produce problems unless one understands what is involved. Tony Cassera anticipates some of those problems.

MOST PRINTERS connected to microcomputer systems use serial data transfer in a RS232 configuration, but on many minicomputer systems the printer is driven from a parallel interface. This type of connection can allow faster transfer of data and if the device is run with interrupt control, less processor time is used in servicing the printer.

Undoubtedly as the powers of microcomputers grow, parallel ports for printers will become common. In a parallel interface, data is transferred eight bits — one byte — at a time. This type of data transfer is sometimes called byte serial to contrast it with the bit-serial nature of the serial interface.

In most present microsystems, the data is being moved round in eight-bit words so that it makes sense to output all these at one time to the printer. The data is taken from the data bus into an eight-bit latch and the interface logic started by the I/O instruction. The transfer of data from port to printer has to be done in a controlled way otherwise data may be lost. The control of data transfer is described as "handshaking" because processor and printer interlock as in the grasp of a handshake. Many more wires must run between printer and host with a parallel interface than with a serial one. There are eight data wires and their returns and the handshaking signals add more wires. More signals can be added to cover such eventualities as the printer running out of paper, the power supply of the printer failing, or the interface plug being pulled by a careless operator. All these signals can make the parallel interface look fearsomely complicated — have you tried to understand the IEEE 488 interface? We can come to grips with these complexities by considering one of the many parallel printer interfaces, the Centronics interface.

This type of interface is something of an unofficial industry standard and is not confined to printers made by Centronics. Many other manufacturers offer Centronics-type interfaces in their product line. They sometimes are called basic or standard parallel interfaces but beware, some manufacturers call Data-Products-type interfaces by these names. The pin allocation of the interface is given in table 1 and this acts as a check list of the signal names. The socket on the back of the printer will be a 36-way Amphenol one so that your external cable must end in a 36-way Amphenol plug. Let's look at some of the signal in much greater detail.

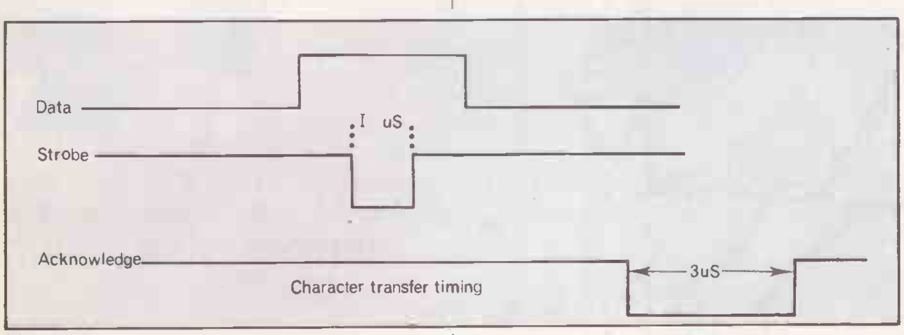


Figure 1.

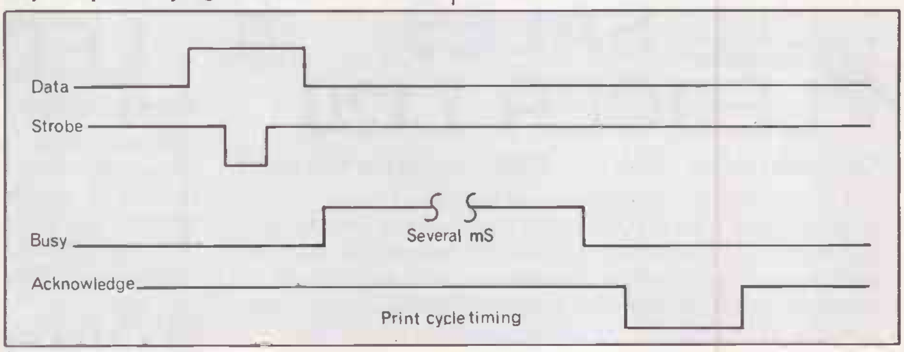


Figure 2.

Data bits 1 to 8 are carried on pins 2 through 9. The standard ASCII code is a seven-bit code so that the eighth bit is not really needed. It is sometimes used to do things like switching character sets or as a parity bit so don't ignore it or you may obtain some strange results. Discover what it can do in the particular machine you are using. If it is not required, it may be safest to ground it at some point in the system. The most important signal is $\overline{\text{STROBE}}$ on pin 1.

After data is taken into a latch some time is needed for the data lines to settle. To appreciate this, remember that the data is being put on to wires several metres in length and possibly not terminated very well at the far end. Under these conditions the line tends to oscillate for a while. Only when the data has settled, is the pulse called $\overline{\text{STROBE}}$ issued. The $\overline{\text{STROBE}}$ signal in the Centronics interface is negative-going and lasts about one microsecond. Active low signals are denoted by having a bar over their name. You can think of $\overline{\text{STROBE}}$ as a message from the processor to the printer saying "here's a character for you". It is the first half of the handshake.

You can't go on streaming characters at the printer indefinitely because at some time you will set it printing and the printer can't print characters on to the paper as fast as you can stream them from the processor. This is the second part of the handshake. Each time the printer receives a character it responds by sending back a signal called $\overline{\text{ACKNOWLEDGE}}$ to the processor. You can think of this as the printer saying "got it", in response to the "here's a character" of the $\overline{\text{STROBE}}$.

The processor in its turn will not transmit another character and $\overline{\text{STROBE}}$ it until it has had the ACKNOWLEDGE back. When the printer buffer is full or when some printing is taking place, the printer delays sending back $\overline{\text{ACKNOWLEDGE}}$ until it is ready again to take data. In the Centronics interface, the $\overline{\text{ACKNOWLEDGE}}$ signal is like $\overline{\text{STROBE}}$, a negative-going signal. When data is transferred without printer action, it is issued typically some 5 microseconds after $\overline{\text{STROBE}}$ and is some 3 microseconds long. The relationship between DATA, $\overline{\text{STROBE}}$ and $\overline{\text{ACKNOWLEDGE}}$ is shown in figure 1.

The $\overline{\text{STROBE}}$ / $\overline{\text{ACKNOWLEDGE}}$ signals constitute the handshake. It is quite possible to run the printer using these two signals alone, and some installations do, but there are other control signals available. Suppose the printer runs out of paper or the print switch is raised. The $\overline{\text{ACKNOWLEDGE}}$ signal may never be sent back and the processor can hang waiting for a signal which will never arrive. Further, in the long time which the printer spends printing and emptying its buffer, valuable processor time can be lost. For these reasons other supervisory signals are provided.

The most important of these is BUSY. When the printer is not ready to take data, the BUSY line is raised. It can be used to tell the processor that the printer will be unavailable for at least a few milliseconds and that it can do something else during that time. When the printer is ready, that is not busy, it drops the BUSY signal. This change is either noted as a flag that the processor polls from time to time or causes an interrupt that demands services. Of course, when BUSY is dropped, the ACKNOWLEDGE is sent. The relationship between STROBE, ACKNOWLEDGE and BUSY are shown in figure 2.

There are other signals which deserve a mention although you don't have to use them — in fact the majority of installations don't.

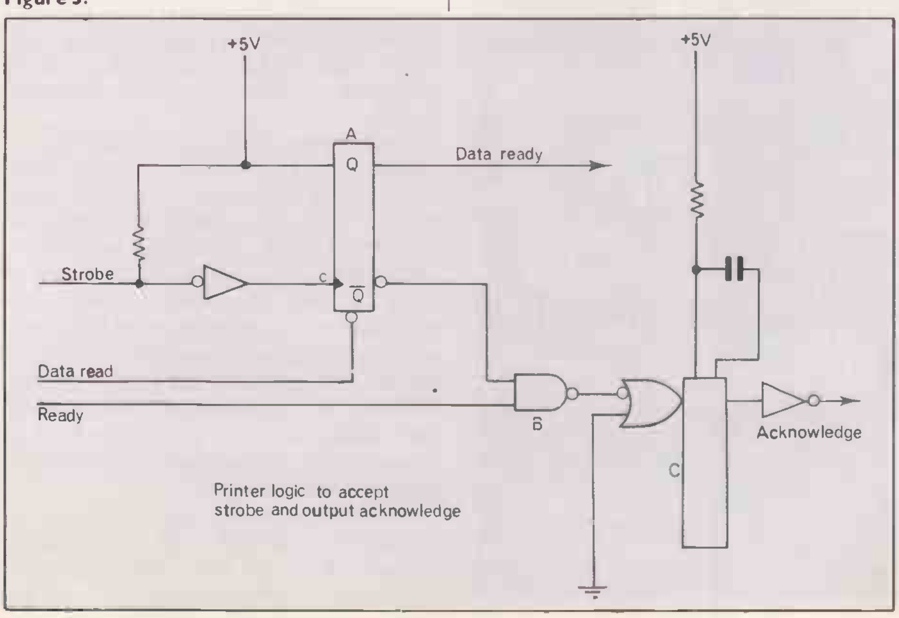
PAPER EMPTY is self-explanatory. You don't need to worry because the printer going out of paper goes BUSY anyway. In some systems it causes an error message to appear at the user consol.

FAULT is again obvious. Often it just senses the fall of the 5-volt logic supply.

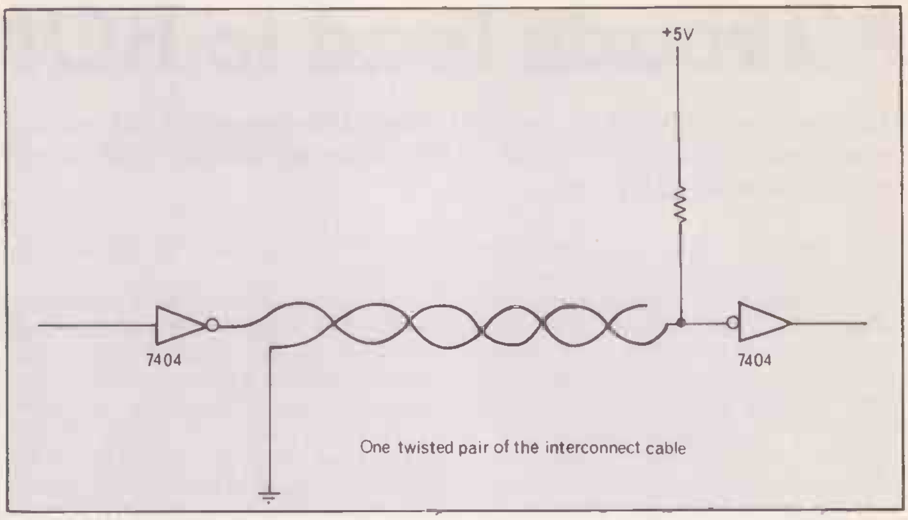
SELECTED needs a little explanation. In early printers an electric motor was used which ran all the time the printer was on. To save wear and noise, circuitry was added which could start the motor before print data was sent. This was done by sending a code over the interface. The code was often XON to start and XOFF to stop the motor. SELECTED reported back that the motor had started.

Nowadays few printers need this facility but be careful you don't send the appropriate code, as some printers respond by switching-off their interface to data except the code word to re-select the printer. Often the print light is used to show the selected/deselected status.

Figure 3.



Printer logic to accept strobe and output acknowledge



One twisted pair of the interconnect cable

Figure 4.

INITIALIZE is sometimes called PRIME. It clears-out all unprinted data and re-sets all logic. Watch out what you do with this signal or you can lose data. Its best grounded-out if you don't want to use it.

OSCXT is a square wave output from the printer. It is useful for a test-message generator as is the + 5-volt output.

Figure 3 shows the kind of logic used in the printer to handle the STROBE and ACKNOWLEDGE signals. The input, STROBE after inversion, clocks flip-flop A, the D input of which is wired high. The Q output goes high, creating a signal DATA READY that strobes data into the data latches. Satisfactory entry of data sends back a signal, DATA ENTERED. This goes to re-set A, putting Q low and $\overline{Q}$ high. Provided the printer is still READY, gate B output goes low and one-shot C fires to generate, after inversion, the signal ACKNOWLEDGE. At the processor interface the circuitry is much the same. The incoming ACKNOWLEDGE is used to clock the flip-flop and the one-shot generates STROBE. If you want to trim the strobe pulse length, as you may have to, do this by altering the value of the capacitor on the one-shot.

Improper construction of the interconnecting cable can cause installation problems. An examination of the pin allocations for the connector shows that a number of pins are marked as RETURNS. All these return wires must be present as second wires in twisted pairs, ordinary twisted twin-flex will do. These return wires act as electrostatic shields to the signal lines. Without these, cross-talk between signals may occur.

Table I
INPUT/OUTPUT SIGNALS FOR THE CENTRONICS INTERFACE

INPUT SIGNALS

Signal Name	Signal Pin No.	Return Pin No.
STROBE	1	19
DATA 1	2	20
DATA 2	3	21
DATA 3	4	22
DATA 4	5	23
DATA 5	6	24
DATA 6	7	25
DATA 7	8	26
DATA 8	9	27
INITIALISE	31	30

OUTPUT SIGNALS

Signal Name	Signal Pin No.	Return Pin No.
ACKNOWLEDGE	10	28
BUSY	11	29
PAPER OUT	12	—
SELECTED	13	—
FAULT	32	—
OSCXT	15	—
SIGNAL GROUND		16
+ 5 VOLTS	18	—
CHASSIS GROUND	—	17

Driving pulses down long wires with TTL logic is not good practice but it seems to work in most cases. However, keep the cable as short as possible and not longer than five metres. At the receiver end the wires are often pulled-up with resistors. If the data transfer is error-prone, try changing the value of these. The set-up is shown in figure 4. If necessary, look at the shape of the signals with an oscilloscope, the pulses of STROBE and ACKNOWLEDGE should look fairly square.

All roads lead to ROM

There are two schools of thought about the storage of software: ROM and disc. Julian Allason of Petsoft puts the case for ROM as the emerging medium and investigates the rapidly-growing market for ROM-based software.

WHEN THE PET made its appearance some two and a half years ago, much was made of the fact that it had Basic resident in Read Only Memory (ROM). Reviewers, used to the chore of loading the language from tape, marvelled at the new-found ability to start computing immediately after power-on. In the last few months, dealers have become half-buried in an avalanche of announcements about plug-in ROMs, ROM cartridges, ROM PACs and EPROMs. To determine whether this is the beginning of a ROM revolution or something rather different, it is worth looking at what is, or about to become, available.

New systems

Several interesting new microcomputer systems are on the point of arriving in the U.K. One, the Texas Instruments TI-99/4, has already been launched while the arrival of the Atari and Mattel systems is imminent. Their importance lies principally in the huge volume of sales which the manufacturers anticipate, but they have something else in common; the use of ROM as a medium for applications software.

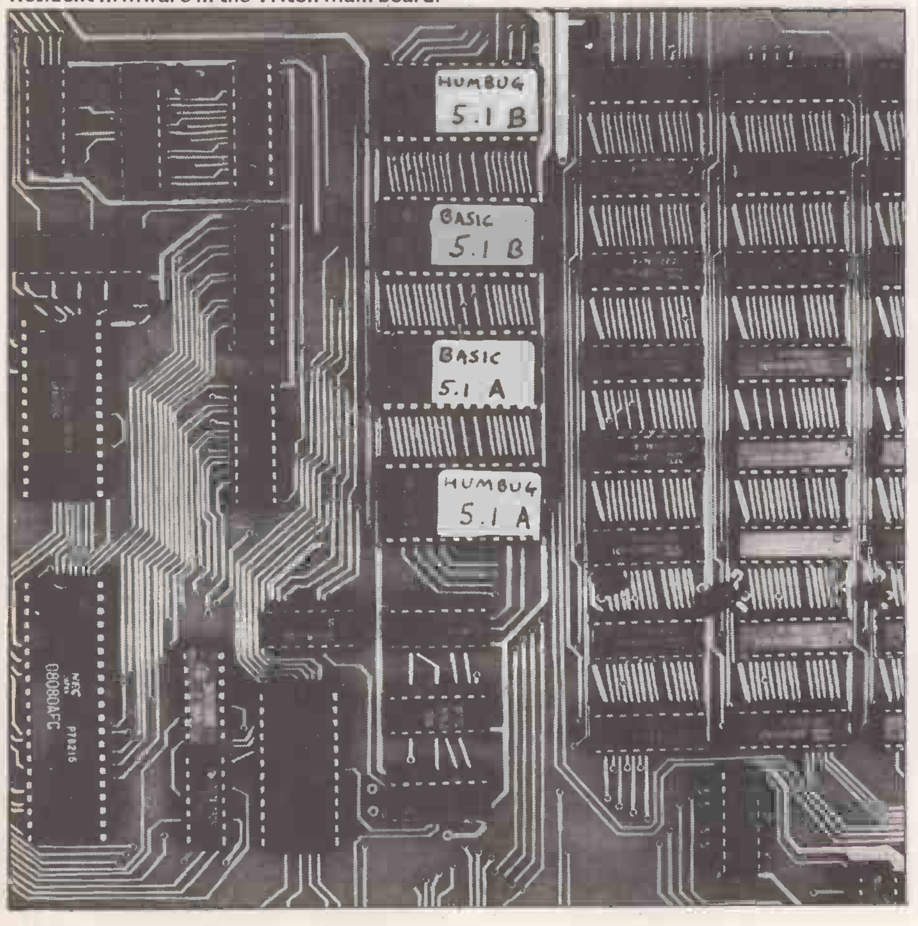
The reasoning is not hard to follow. The TI-99/4, which, incidentally, is based on a 16-bit central processor and is sold with 16Kbytes of RAM as standard. The impressive Video Chess game developed for Texas Instruments by British Master, David Levy, requires 30Kbytes. ROM memory is cheaper than RAM. Hence Video Chess is to be supplied as a solid-state command module containing no less than five 6K-masked ROMs. At present, the range of command modules is rather limited, perhaps because they are programmed in a graphics-orientated language, GPL, which is said to be known only to the few.

Atari, which has yet to dream-up a really fancy name for its ROM cartridge, has an impressive list of games available, as one might expect. A Basic interpreter cartridge is already supplied with most machines sold in the U.S., while an assembler/editor package and a music composer are due for release soon. The latter allows the entry of musical notes through the keyboard; you hear the melody as it is displayed simultaneously on screen in the form of musical notation. The tempo, and any note, can be changed at will.

Even the established systems have started to appear on the scene with plug-in ROM packages. Many Pet owners have bought the Programmers' Toolkit chip, which contains 2Kbytes of machine code routines. Once plugged into an empty ROM expansion socket and initialised with a SYS command, the Toolkit adds 10 extra commands to the standard Microsoft Basic. These include RENUMBER, FIND, APPEND, TRACE and HELP.

Meanwhile, Skyles Electric Works has just released MacroTea, a powerful 6502 assembler on a chip, complete with relocatable assembler, conditional assembler, macro assembler, text editor and enhanced monitor. Other Pet utilities are said to be on the way from Personal Software and possibly Commodore.

Resident firmware in the Triton main board.



Facilities

For Apple users there is Programmers' Aid, which includes RAM test, RENUMBER, Tape Verify and a number of high-resolution graphics facilities among its features. There is also available a peripheral board called ROMPLUS+ which provides six sockets to accept individually addressable 2K ROMs or EPROMs. The board is supplied with the keyboard-filter ROM already fitted. It includes routines to enable upper- and lower-case letters, multiple user-defined character sets, coloured or inverse letters, mixed text and graphics, and a number of other features.

Surprisingly, ROM firmware for the Tandy TRS-80 has yet to appear, although AJ Harding (Molimerx) has said it is considering the idea. Physical access will almost certainly be via the keyboard module.

The Sorcerer is another popular microcomputer with provision to accept what Exidy calls ROM-PACs. At present there are four; a standard Basic PAC, a development PAC which contains an assembler/editor, most interestingly a Word Processing PAC, and an EPROM PAC. The latter allows one to create ROM-resident software using your own PROMs. It is supplied jumpered for use with type 2716 EPROMs for a total of 8Kbytes storage. By changing jumpers, the EPROM PAC can be altered to accept other common bipolar and MOS PROMs.

Anyone who has ever had to fiddle with the volume control of a cassette deck or watched glumly as the screen displayed some cryptic disc-error message, will be aware of the benefits of having software in ROM. In addition to speed and reliability of access, must be included the advantages of cost. Producing a masked ROM in volume is relatively inexpensive. 10,000 units would be the minimum quantity for a low-priced firmware product according to one manufacturer, although other sources suggest that a smaller figure could be economic. In fact, there are a number of EPROM burners available where low volumes are required. Both Personal Computers Ltd of London and G R Electronics of Newport, Gwent, offer inexpensive burners to handle the ubiquitous 2716 EPROM. Unfortunately 2716s have been in chronically short supply, although delivery has improved lately.

Interim solution

Simple plug-in ROMs are probably only an interim solution since the chip is both vulnerable to static electricity and tricky to insert without causing damage to the pins. Various types of carriers have been devised which make insertion and extraction easier. The cartridge idea appears more attractive, however, where regular firmware changes are required. Even there, excessive wear has been noted on the connector strips of some frequently-used cartridges.

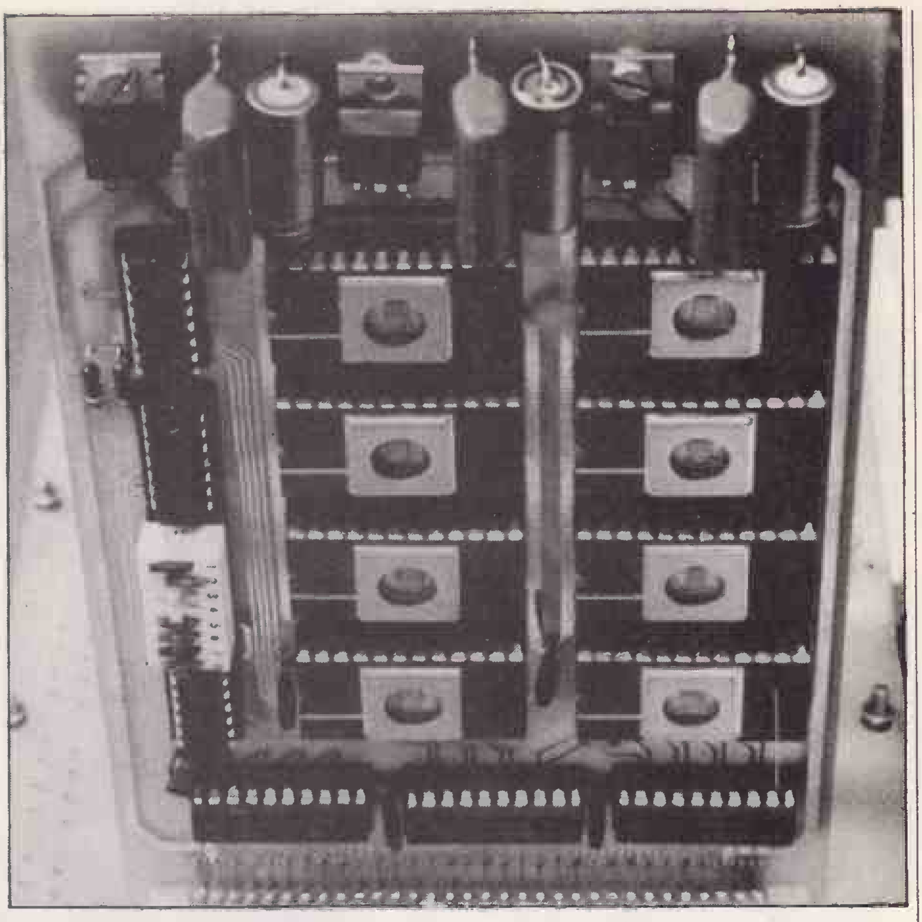
Limited addressable space is a problem with systems like the Pet and TRS-80. Since the 8-bit architecture of the 6502 only allows up to 64K of memory to be addressed directly, one is left with no more than 12K of expansion ROM space, once system and RAM requirements are accounted for. On current models of Pet there are two empty ROM expansion sockets. The result has been a clash between various competing address ROMs such as the Programmers' Toolkit and Commodore applications software protection ROM. To complicate matters further, the plug-compatible Compu-Think mini-floppy disc system uses the same slot.

Empty sockets

A short-term solution has appeared in the form of the SpaceSaver — a small card which plugs into a ROM expansion socket. On it are mounted three empty sockets into which address competitive ROMs may be inserted. A simple mechanical switch enables the user to select whichever ROM he wishes to use. An additional device called a ROM Driver will allow selection to be made under software control.

This game of musical slots is likely to recommence when Commodore releases its new SuperPets which will have an extended Basic implemented. Designated Basic 4.0, it will contain an improved Disc Operating System in ROM.

The extention of systems software in ROM is a logical development which most



The Triton 8K extension EPROM card. A further 8K of resident firmware can be housed in 2708 EPROM and as many as eight of these can be plugged into the Triton extension motherboard.

manufacturers are likely to follow. What is a good deal more controversial is the question of putting applications packages into firmware.

The proposition offers software houses something of a mixed bag of advantages and disadvantages. On the one hand copyright is easier to protect, since rather more trouble and equipment are required to duplicate a program for a friend if it is ROM. The programs are also protected against unauthorised meddling. The other side of this coin is the need for software to be totally bug-free. With sophisticated business packages, this is asking a good deal and the prospect of replacing several thousand ROMs is enough to make the most flamboyant software supplier uneasy. Unless distribution is on a large scale, or prices high, software maintenance could be ruled-out altogether.

It is possible, although inefficient, to produce ROMs programmed in Basic. But even with programs written in machine code there are limits to what can be achieved in the amount of ROM space available on most systems. Several solutions including intelligent bank-switching devices, have been proposed to resolve this limitation.

Some 8-bit micros already employ a form of bank switching to extend the normal 64K addressing limit. For example, the ACT800 also makes use of some of the space assigned to the Basic interpreter for Disc Input/Output RAM. One way of making more space available for ROM is to discard Basic altogether.

To activate another language such as Pascal, the entire Basic interpreter and most of the operating system might be switched-out. Using this approach, one OEM manufacturer has already developed an 'executive Pet' capable of handling five languages.

Big business

Many programmers want to be able to use high-level languages other than Basic, so it is a fair assumption that ROM-based languages are likely to become very big business indeed. One of the first to arrive was the Western Digital Pascal Micro-engine, which is available as a 16-bit chip set which executes Pascal programs five-times faster than is possible with conventional systems software. Not surprisingly, their order book is said to be full for months ahead.

Bank switching is, nonetheless, a clumsy, and potentially inefficient, means of handling extensive ROM. For this reason, future generations of micro-computers will be based on new CPUs, the architecture of which is better suited to the task.

Texas Instruments has been using 128K ROMs for more than two years, and a

(continued on next page)

(continued from previous page)

256K ROM was demonstrated recently for the Sorcerer. At the International Solid State Circuits Conference in San Francisco, Japanese manufacturers gave their U.S. counterparts a nasty shock when they announced a 4MB ROM on a wafer. Such large capacity devices will be capable of supporting software of considerable sophistication.

Talking to software originators, one encounters a measure of disagreement on the applications best suited to firmware implementation. Several word-processing packages written in machine code have reached a sufficiently stable state to be ROMed and there are a number of information retrieval packages which are also agreed to be suitable. Another much-discussed possibility is firmware capable of converting a microcomputer into an intelligent terminal.

Pros and cons

Several of the major software houses have looked quite carefully at the pros and cons of making their applications software available in ROM. Chris Hawkins of CAP comments: "The nature of our business is to wholesale to a rapidly growing number of licensees worldwide. To do this, we already supply our software in a variety of forms. Our view is that it is early days yet and the software is changing too fast, but if a manufacturer asks us to make our software available in ROM, that is a different matter".

Microcomputer manufacturers have reservations about tying their machines to particular market areas. "If we stuff the computer full of business firmware, what would it do to our educational sales"? asks one manufacturer.

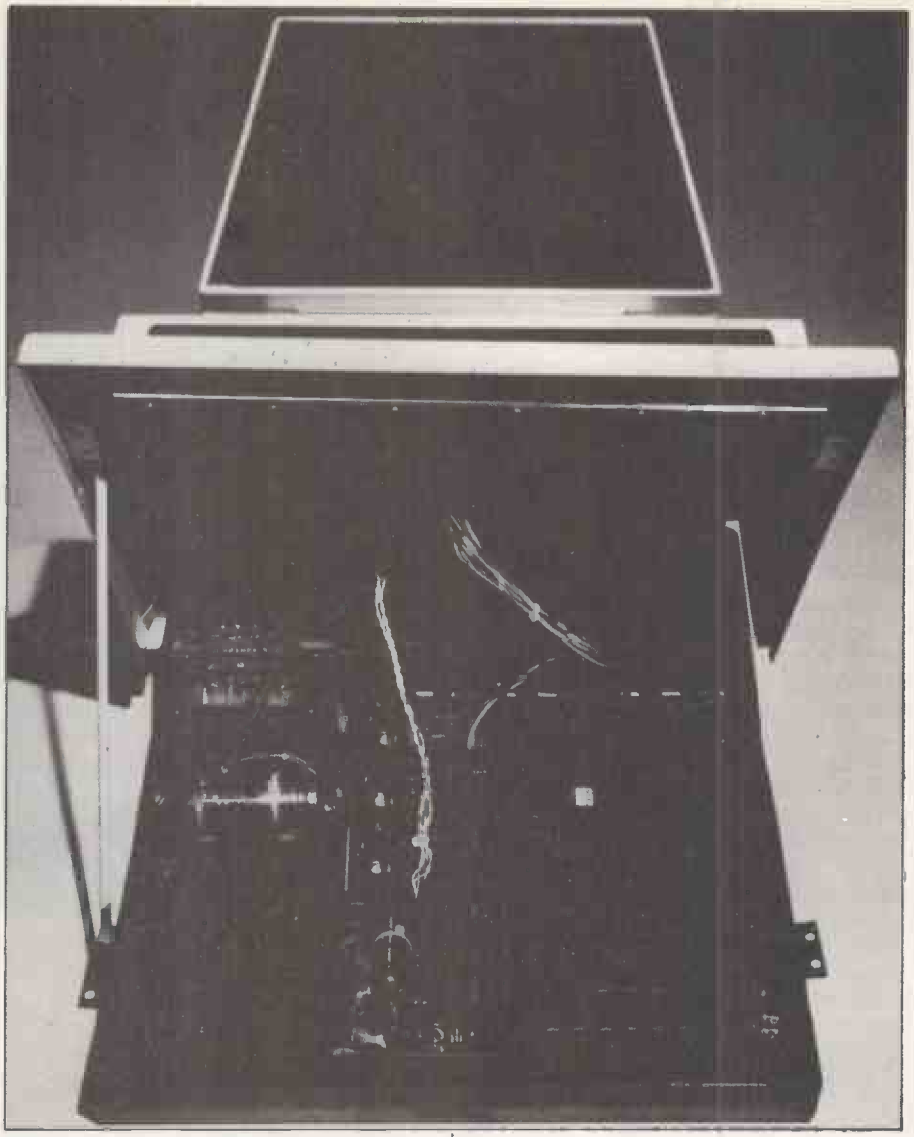
Transam is one of the few suppliers offering a range of off-the-shelf firmware. It sells the single-board Triton computer for £294 including 4K of firmware. One can subsequently up-grade by adding a motherboard at a cost of £50. It provides eight slots into which can be plugged extra RAM cards or an 8K EPROM board which cost £30 and will hold up to eight EPROMs.

Most customers up-grade from the original 4K of firmware to an 8K extended Basic and a more powerful monitor. Because erasable firmware has been used, customers can bring their original chips back for re-programming. Transam charge £1 each for erasure and a modest £2 for re-programming each EPROM.

Packages

Further up-grades includes a resident assembler package and a resident Pascal compiler, of which there is also a CP/M version for disc users.

"Because all the firmware can be re-programmed, the system is both inexpensive and very flexible", says Nigel Stride



of Transam. "However, we find it is not really practical to have more than 30K of firmware. At that price you are competing with the cost of a disc system".

One requirement of outside software houses considering the development of plug-in firmware, is stable systems software. "It is no good if the manufacturers keep mucking about with the operating system", comments the managing director of one large software house. "We would have been in firmware by now if it were not for that".

Derek Rowe of Abacus Computers is another who has held back until now: "We have already developed a package to handle matrix arithmetic on the Pet, which lacks a MAT statement at present. We are also looking at several alternative languages for the Pet and other micros".

Industry observer, Robin Bradbeer thinks software houses will have to learn to live with continuing system changes. Nevertheless, he predicts that "ROM-based applications software will do for microcomputers that the golf-ball did for the typewriter".

Most people in the software industry are agreed that the types of package best suited to ROMing are those where the application is broad and sufficiently standard to be used without customisation.

Unfortunately, highly-standardised packages of the stock control variety tend to be those which have a substantial random access requirement. Since disc storage is going to be needed anyway, the program might just as well be loaded from disc in the first place, or so it is argued.

For this reason it appears that the types of firmware most likely to become available for use on general purpose microcomputers will fall into the quasi-systems category, such as database management. The general level of knowledge among computer purchasers is said to be declining.

The future

In future generations, microcomputers may evolve into highly-specialised devices. The authoritative Infotech state-of-the-art report forecast recently the availability of complete off-the-shelf applications software systems based on firmware modules. It is not unreasonable to expect to see relatively inexpensive microcomputers dedicated to a wide range of business and scientific applications.

Hotel MicroSystems Ltd

WEST LONDON MICROCOMPUTER SPECIALISTS

Demonstrations and advice gladly given.

NORTH STAR HORIZON

Dual Density Twin Drives 32K £1585
Dual Density Twin Drives 48K £1780
Quad Density Twin Drives 32K £1795
Quad Density Twin Drives 48K £1990
Many other configurations available.

VDUs ADM-3A

Teletype replacement VDU with addressable cursor £595

ADM-31

Sophisticated terminal with many features including 2-page memory, protected fields, addressable cursor. £795

PERICOM 6801

High-quality robust terminal with many features £1080

PRINTER LS300

Very high speed dot-matrix printer. Bi-directional, logic-seeking, high speed tabbing. 180cps. Serial or parallel interface £1345

CLOCK/CALENDAR BOARDS

These S100 boards store the time in intervals from 100us to 100000days. Supplied with software to turn days into date, day of week, month, year. £185.

WINCHESTER DRIVE

Complete packaged drive and controller with S100 interface and Multi-User Operating System and BASIC (7 users)
6.2MB £3000 31.2MB £4000 18.6MB £3500

Full after-sales service.

NORTH STAR SOFTWARE WORDSTAR

A sophistcated and versatile screen-oriented word processing package. Runs under CP/M and works with most printers and VDUs £255

CP/M

For 48K Horizon £75

SPOOLER

A modified DOS which operates a 7K5 interrupt-driven RAM printer buffer. Gives increased system throughout £55 *

HMS BASIC

A modified North Star BASIC interpreter incorporating several extra functions especially useful to the commercial programmer. Has MOD, DAY, DATES, CHRS (C,n), COPY (disk copy in BASIC) and SEEK (fast string/table search). £40

Auto RAM

test. A version of DOS which tests your RAM during terminal wait loops. Displays sufficient information on detection of a fault to enable you to pinpoint the faulty IC £15 *

INSTAR

An information storage/retrieval program with many applications. Very fast searching facilities £185

* these three items FREE with every Horizon purchased from us.

All prices exclusive of VAT and delivery   Callers (by appointment) welcome   473, Stains Rd, Bedfont, Feltham, Middx. Tel. 01-890 9696

• Circle No. 185

PETAID

PET SOFTWARE WITH A FUTURE. EVERY PET SHOULD HAVE ONE

OFFICIALLY APPROVED BY COMMODORE

LOOK NO PROGRAMMERS!
PETAID — THE ANSWER TO A USER'S PRAYER

STAGE ONE SOFTWARE DEALERS

Name	*Address*	*Telephone No.*
Adda Computers Ltd	17-19 The Broadway Ealing LONDON W5 2NH	01 579 5845
Amplicon Micro Systems Ltd	143e Ditchling Road BRIGHTON, Sussex BN1 6JA	0273 562163
Computer Services Midlands Ltd.	Refuge Assurance House Sutton New Road, Erdington, BIRMINGHAM B23 6QX	021 382 4171
D.A.M.s Office Equipment Ltd.	30/36 Dale Street, LIVERPOOL L2 5SF Merseyside	051 227 3301
Jeffrey Martin Computer Services Ltd	78 Fore Street, NEWQUAY, Cornwall	063 73 2863
M.M.S.	26 Mill Street, BEDFORD	0234 40602
R.P.L. Microsystems	Goldie House, Upper Church Street DOUGLAS, I.O.M.	0624 4247
Sheffield Computer Centre	225 Abbeydale Road SHEFFIELD, S7 1FJ	0742 53519
Slough Microshop	120 High Street, SLOUGH, Berks	0753 72470
T & V Johnson (Microcomputers)	165 London Road Camberley, Surrey GU15 3JS	0276 62506
Thistle Computers	29 Albert Street KIRKWALL, Orkney KW15 1HO	0856 3140
Walters Computers Systems Ltd	Field House 107 Worcester Lane STOURBRIDGE, W. Midlands DY9 0SJ	0562 885937
Alpha Business Systems	89 Railway Street, HERTFORD	0992 57423
Catlands Computers Ltd Ltd	Green Lane WILMSLOW, Cheshire	0625 527255
G M Marketing	Old Barn House Amport, ANDOVER, Hants	026 471 410

All your installed programs could have an identical structure — think how easy subsequent support will be — programs in Days not Weeks. Weeks not Months, Months not years —

PETAID FEATURES: Create Files, Insert, Amend, Delete, Display, Print, Search, Copy, Examine, Extract and Sort — Good Documentation.

NOW AVAILABLE FOR:

Sequential	Commodore Disks
Random Access	Commodore and Compu/Think Disks
Printers	Commodore, Anadex, Qume

Tape based Version only £51.75
Sequential Disk based version £149.50
Random Access Disk based version £207.00
Extract & Sort £86.25

Soon to be available and PETAID compatible!

1. Various Print generators
2. Search and Extract new file
3. Transaction Handlers

ALSO
Word Processor £115.00
Word Processor with mailing £138.00
Quote Processor £115.00
Mailing List £115.00
Estate Agents Pack:
Applicants/ Properties £287.50
Stock £115.00
Incomplete Records £862.50
Pension Calculations £57.50
Bond 10 + £57.50
All prices shown are VAT) P&P UK inclusive

MAIL ORDER

Written orders with cheque or Access/Visa No. to:
STAGE ONE COMPUTERS 6 Criterion Arcade, Old Christchurch Road, Bournemouth. 23570

BARCLAYCARD VISA   Buy it with Access

Please supply: ITEM .... QUANTITY ....

AMOUNT £ .... CHEQUE NO. ....
Name ....
Address .... ACCESS/VISA NO. ....

• Circle No. 186

Conditional jumps and registers under review

The previous two articles in David Peckett's series looked at basic arithmetical operators. This month he deals with the instructions for effective practical programs — conditional jumps. He explains more about the hardware of the 6502 and the 8080A and, in particular, the registers.

THE 6502 architecture with all the user-accessible registers is shown in figure 1a. There are three registers which are new to us. These are "X" and "Y", both 8-bit registers, and the stack pointer (SP). The SP also has eight bits, and we must understand its operation if we want to use the micro effectively.

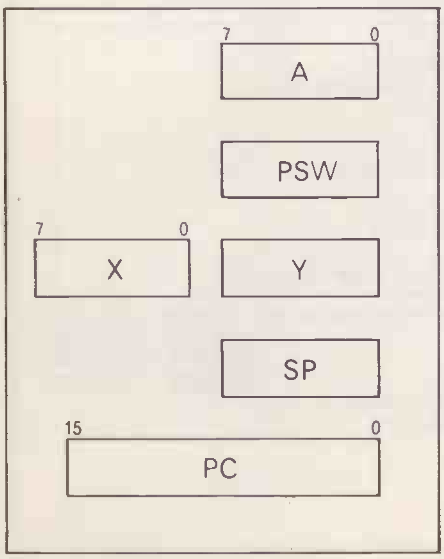
The stack represents a quick way of preserving data, and of passing it from one program segment to another; the SP is used to move data into and from the stack.

In the first article, I introduced the idea of indexed addressing — the 8080A has to use it during arithmetical operations. Indexing is a powerful technique and the 6502 X and Y registers are used as index pointers. Their other main use is as counters during loops, although they can also be used as temporary data stores.

What extra instructions do we have to manipulate these registers? Table 1 gives this month's operations. Data can be loaded into X and Y, either from memory or directly, using the "LDX" (LoaD X) and "LDY" instructions. Equally, their contents can be placed in memory by "STX" (STore X) and "STY". These four instructions are used just like "LDA" and "STA", and have the same effect(s) on the flags. We can also pass data between the accumulator and either X or Y, using "TAX" (Transfer A to X) and "TAY", and "TXA" (Transfer X to A) and "TYA". We can't, however, move data directly between X and Y.

In general terms, the only way to manipulate data in the 6502 is via the accumulator. Memory, and X and Y, are essentially used for data storage. We can, however, increment and decrement the data in X and Y via the "INX" and "INY", and "DEX" and "DEY" instructions. This ability is vital when we use the registers as either pointers or counters.

Figure 1a. Architecture of 6502.



The architecture of an 8080A is shown in Figure 1b, and is rather more complex than that of the 6502. The micro has six 8-bit auxiliary registers: "B", "C", "D", "E", "H" and "L". For certain operations, these can be paired to form three 16-bit registers: (B,C), (D,E) and (H,L). We've already met a user for (H,L). The 8080A also has a 16-bit Stack Pointer.

The six auxiliary registers give us, not surprisingly, a more complicated set of programming options than we had with the 6502. The registers can be used for the same three basic purposes; indexing, counting and temporary storage, but with a wide range of sub-options.

- **Indexing.** As we have seen, (H,L) is the 8080A primary index register, and gives access to the implied memory location "M". In fact, this is the only way we can move data from memory to manipulate the contents of the accumulator. The other two register pairs (RPs) can, however, be used as pointers to load the accumulator or to store its contents. The appropriate instructions are "LDAX B" (LoaD Accumulator indeXed from B) and "LDAS D", and "STAX B" and "STAX D". In each case, the address is that implied by the contents of the RP, with "B" meaning (B,C) and "D" meaning (D,E). The 8080A does not have "LDAX H" or "STAX H" instructions. These would be the same as "MOV A,M" and "MOV M,A" respectively.

  Because indexing normally means that we have to manipulate the data in a pointer, the 8080A has "INX r" (INcrement indeX r) and "DCX r" instructions. In these, "r" can be "B", "D" or "H". The instructions increment or decrement the appropriate RP as a single 16-bit word. Beware, however — they don't affect any flags.
- **Counting.** As we shall see later, counting the number of iterations of a loop is a very common procedure. Any of the 8080A 8-bit registers (A-E, H, L) can be used as counters via the "INR r" (INcrement Register r) and "DCR r" instructions. When you use these instructions, you must insert the appropriate register letter in place of "r"; it's best to use B-D as counters. An important point is that the 8080A effectively uses M as an extra auxiliary register. However, it takes longer to manipulate M than any internal register.
- **Temporary storage.** The 8080A design allows great freedom when moving data between 8-bit registers. Data can be moved from any register to any other, including M. We have already met the MOV A,M instructions. In fact, this is just a special case of "MOV r_1,r_2" — MOVe the contents of register 2 to register 1. Thus, MOV B,E makes B equal to E, and MOV M,D loads M with whatever is in D. The micro will even solemnly perform operations such as MOV A,A!

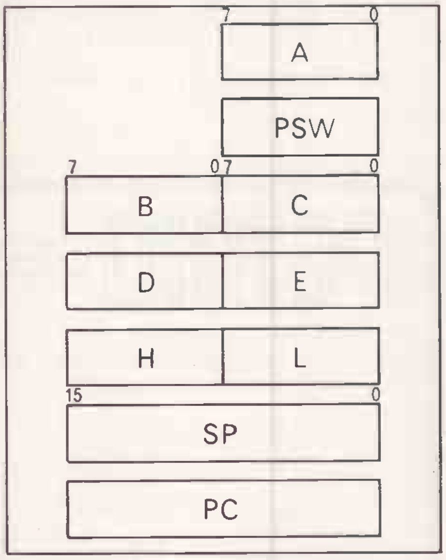


Figure 1b. Architecture of 8080A.

You will also remember that we used "MVI A,data"; again, this is a particular form of "MVI r,data" — MOVe immediate data to register. This instruction gives an immediate load of any of the 8080A 8-bit registers. Finally, when doing arithmetic we used mnemonics such as "ADD M". It'll be no surprise to find that this is a special case of "ADD r". Thus data can be stored easily in any register, and then used in arithmetic.

This facility is very useful, and can be used to speed programs considerably by reducing the number of memory accesses. All arithmetic has to be performed in the accumulator, however. As a final note, "SUB A" is one way of setting the Accumulator to zero.

Finally with this rich choice of register manipulation instructions, how do we actually move data into them? We've already seen that "MOV r_1,r_2" and "MVI r,data" can move single bytes to any register — can we also move 16 bits? We've already met "LXI H,data", and this is a special case of "LXI r,data". In addition, (H,L), and only (H,L), can be loaded directly from memory by "LHLD address" — Load H,L Direct from address. The 16-bit address points to an 8-bit byte; this goes into L, and the data at (address + 1) goes to H. The opposite operation, "SHLD address" — Store H,L Direct at address — is also available. Figures 2a and 2b show the effect of "LHLD pqrs" an "SHLD pqrs" respectively.

There are many times that a program must do something only if something else is true. For instance, we may need to jump to an error-handling routine if incoming data is outside a specified range. Alternatively, we may want to perform a loop a given number of times and then leave it. In Basic, this sort of function is given by the instructions:

```
150 FOR I = 1 TO 15
300 NEXT I
```

which carries out the instructions between line 150 and 300 a total of 15 times. In a micro we can obtain the same effect by loading a counter with the desired number of iterations. At the end of each cycle, the counter is decremented; if the result is equal to zero, the program goes on to the next segment, otherwise it jumps back to the start of the loop.

All computers have flags which record the results of various operations, and we've met the flags provided by the 6502 and the 8080A. The two micros also each provide a set of instructions which test individual flags and cause a jump if the

(continued on next page)

	6502			8080A		
Operation	**Mnem**	**Flags**	**Effect**	**Mnem**	**Flags**	**Effect**
Load X/Y	LDX/Y o	N,Z	X/Y = d/(a)	—		
Store X/Y	STX/Y a	None	(a) = X/Y	—		
Transfer A to X/Y	TAX/Y	N,Z	X/Y = A	—		
Transfer X/Y to A	TX/YA	N,Z	A = X/Y	—		
Increment X/Y	INX/Y	N,Z	X/Y = X/Y + 1	—		
Decrement X/Y	DEX/Y	N,Z	X/Y = X/Y — 1	—		
Use Register r In Arith Oper'n	—			Opnr	All	A = A opn r
Load Reg 1 with Conts of Reg 2	—			MOV r_1,r_2	None	$r_1 = r_2$
Load Acc, Indexed by Reg Pair	—			LDAX rp	None	A = (rp)
Store Acc, Indexed by Reg Pair	—			STAX rp	None	(rp) = A
Load (H,L) from Memory	—			LHLD a	None	L = (a), H = (a + 1)
Store (H,L) in Memory	—			SHLD a	None	(a) = L, (a + 1) = H
Load Reg Pair Imm.	—			LXI rp,d	None	rp = d
Increment Reg Pair	—			INX rp	None	rp = rp + 1
Decrement Reg Pair	—			DEX rp	None	rp = rp — 1
Jump if Zero	BEQ d	None	PC = PC + d if Z = 1	JZ a	None	PC = a if Z = 1
Jump if not Zero	BNE d	None	PC = PC + d if Z = 0	JNZ a	None	PC = a if Z = 0
Jump if Carry	BCS d	None	PC = PC + d if C = 1	JC a	None	PC = a if C = 1
Jump if not Carry	BCC d	None	PC = PC + d if C = 0	JNC a	None	PC = a if C = 0
Jump if Positive	BPL d	None	PC = PC + d if N = 0	JP a	None	PC = a if S = 0
Jump if Negative	BMI d	None	PC = PC + d if N = 1	JM a	None	PC = a if S = 1
Jump if Overflow	BVS d	None	PC = PC + d if V = 1	—		
Jump if not Overflow	BVC d	None	PC = PC + d if V = 0	—		
Jump If Odd Parity	—			JPO a	None	PC = a if P = 0
Jump if Even Parity	—			JPE a	None	if P = 1

Notes on Table 1
"a" = Address (defined by the program)
"d" = Data (defined by the program)
"o" = Operand — can be an address or data
"r" = Any 8080A register, including M
"rp" = Any 8080A register pair
A = Accumulator
X/Y = Register X or Register Y (in the 6502)
Opn = Any 8080A arithmetical operation (e.g., ADD, SBB)
The flags are as defined in the second article
Brackets mean: "Contents of the address, or address defined by the contents of the register pair, contained between the brackets."
A "/" is to be read as "or".

Table 1. This month's instructions.

Figure 2a. 'LHLD pqrs'.

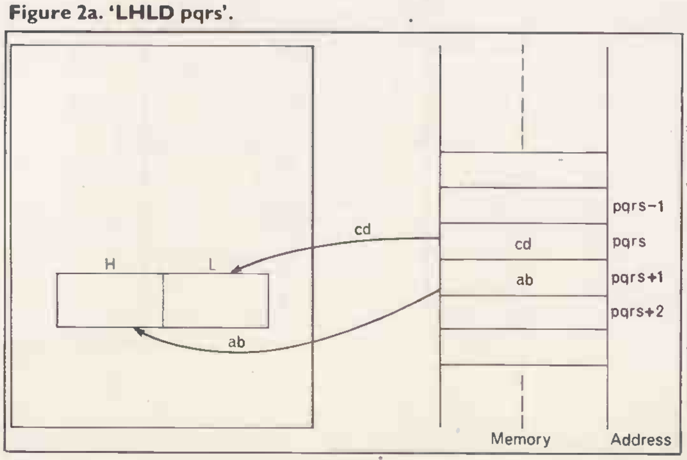
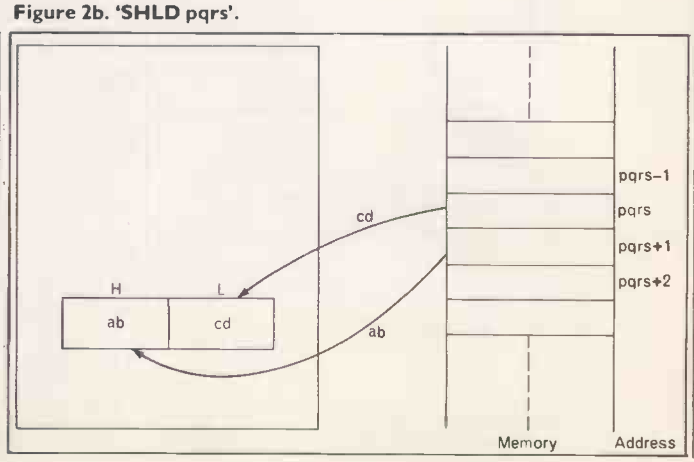


Figure 2b. 'SHLD pqrs'.



```
;6502 MULTIPLICATION ROUTINE
          CLD               ;INITIALIZE FLAGS
          SEC               ;NO-BORROW CONDITION
          LDA DATA1         ;"A" — MULTIPLICAND
          LDX DATA2         ;"B" — MULTIPLIER
          STA MULT          ;SAVE FOR LATER USE
          SBC DATA2         ;TEST FOR MAGNITUDE
; IF NO BORROW, "B" IS NOT GREATER THAN "A"
          BCS NOSWAP        ;CARRY = 1 IS NO BORROW
;EXCHANGE "A" AND "B"
          STX MULT          ;MULT NOW CONTAINS "B"
          LDX DATA1         ;"A" BECOMES MULTIPLIER
;SET UP SUMMING REGISTERS
NOSWAP    LDA#0
          TAY               ;Y WILL HOLD MSBS
;DO MULTIPLICATION
LOOP      CLC               ;ENSURE NO CARRY
          ADC MULT
          BCC NOCRRY        ;INCREMENT MSBS?
          INY
NOCRRY    DEX               ;DECREMENT MULTIPLIER
          BNE LOOP          ;FINISHED?
;MULTIPLICATION FINISHED HERE — SET DATA1 ND DATA2
          STA DATA1         ;LSBS
          STY DATA2         ;MSBS
;END OF PROGRAM SEGMENT
```

Figure 6a. 6502 multiplication.

```
;8080A MULTIPLICATION ROUTINE
          LXI H,DATA1       ;SET UP POINTER
          MOV A,M           ;"A" — MULTIPLIER
          MOV B,A           ;SAVE "A" IN B
          INX H             ;POINT TO DATA2
          MOV C,M           ;C CONTAINS "B" — MULTIPLIER
          SUB C             ;TEST FOR MAGNITUDE
;IF NO BORROW, "B" IS NOT GREATER THAN "A"
          JNC NOSWAP        ;CARRY = 0 IS NO BORROW
;EXCHANGE "A" AND "B"
          MOV C,B           ;C NOW CONTAINS "A" — MULTIPLIER
          MOV B,M           ;B NOW CONTAINS "B" — MULTIPLICAND
;SET UP SUMMING REGISTERS
NOSWAP    MVI A,O           ;LSBS
          MOV D,A           ;D = 0 = MSBS
;DO MULTIPLICATION
LOOP      ADD B
          JNC NOCRRY        ;INCREMENT MSBS?
          INR D
NOCRRY    DCR C             ;DECREMENT MULTIPLIER
          JNZ LOOP          ;FINISHED?
;MULTIPLICATION FINISHED HERE — SET DATA1 AND DATA2
          MOV M,D           ;STORE MSBS
          DCX H             ;POINT TO DATA1
          MOV M,A           ;STORE LSBS
;END OF PROGRAM SEGMENT
```

Figure 6b. 8080A multiplication.

(continued from previous page)
appropriate flag is set or not set, depending on the test.

What kind of tests do we make? By far the most common is to see whether a result is zero or not, and therefore uses the zero flag. The carry flag is also important, particularly during multi-byte arithmetic. During input operations, the MSB is often used to indicate the presence, or absence, of fresh data; tests of the sign flag can be used to monitor the MSB. Occasionally, the other flags may need testing, such as the overflow flag during 2s-complement arithmetic.

The 6502 and the 8080A both allow tests of the carry, zero and sign flags. In addition, the 6502 can test its overflow flag and the 8080A its parity flag. In each case, tests can be made for the presence, or the absence, of the flag. Each micro thus provides eight conditional jumps.

There is a major different, however, between the way in which conditional jumps are mechanised in the two devices. The 8080A conditional jumps are exactly analogous to its unconditional jumps. The instructions use three bytes:

	address	
opcode	low byte	high byte

If the condition — e.g., zero, odd parity — implied by the opcode is true, the PC is loaded with the new address, and control is passed to the new part of the program. Otherwise, the PC points to the instruction after the conditional jump. The key point is that the 8080A conditional jump uses a 16-bit address field, which explicitly defines the destination of the jump.

In the case of the 6502, the mechanism is rather different. The instructions are called branches, and each uses only two bytes:

opcode	address

Obviously, we need two bytes to define an address, so what does the 6502 do? The address field of the instruction is used as a displacement which modifies the PC in a technique called relative addressing. The address field of the instruction is treated as a 2s-complement number, and is added to the contents of the PC if the branch is to take place. It is thus only possible to branch between (PC-128) and (PC + 127).

Remember, once the micro has read an instruction in order to execute it, the PC points to the next instruction; this is always two bytes ahead of the branch. The range of the displacement is thus —126 to + 129 from the test itself.

Figures 3a and 3b may make this clearer. In the first case, the PC contains 1234_{16}, and the next instruction BEQ 20 is loaded. The PC then contains 1236_{16}. Suppose the test succeeds; PC is set to (1236 + 20), or 1256_{16}, to where the program immediately jumps. In the second case, the branch displacement is

Figure 3a. Forward branch.

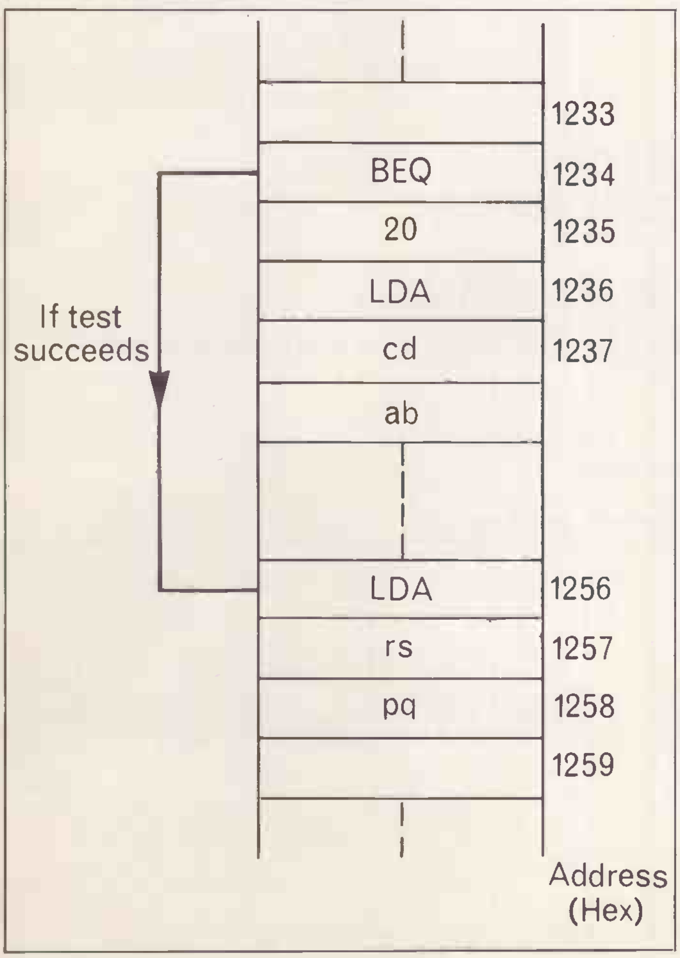
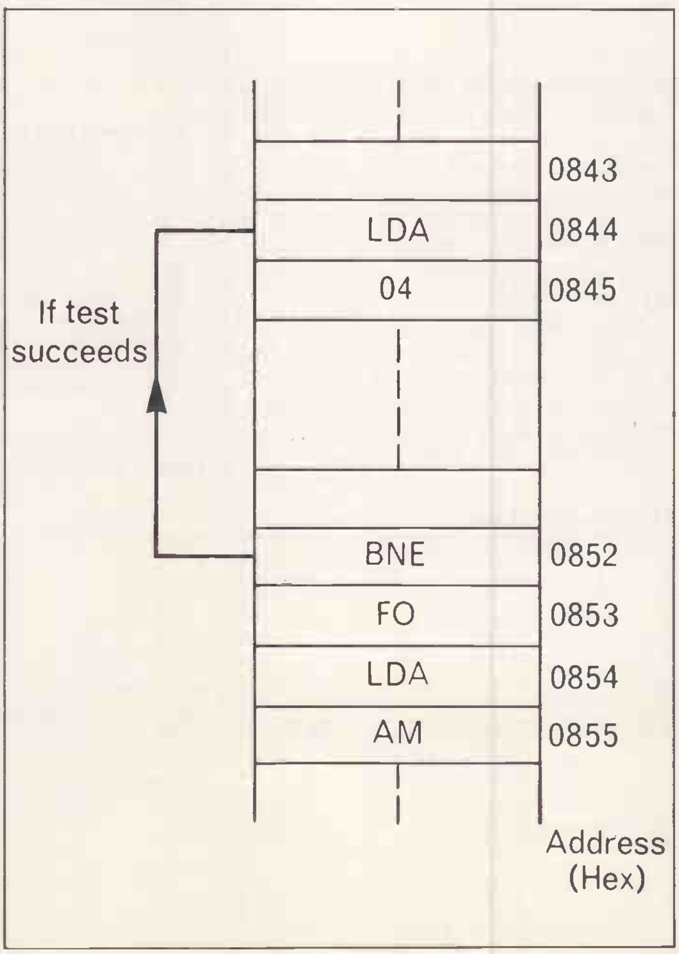


Figure 3b. Backward branch.



negative (FO_{16} = -10_{16}), and the test is at address 0852_{16}. Thus, if the test succeeds, the PC is set to (0854 — 10), and the program leaps backwards to 0844_{16}.

By now, you may be wondering what happens if you want to (or have to) branch beyond the range given by a single byte. The answer is easy — you use a branch and a jump:

```
        BNE FLAG
   FLAG JMP TARGET ;DUMMY
                    STEP
 TARGET LDA DATA ;WE WANT TO
                  GET HERE
```

This is obviously clumsy, but it is the only way.

What are the advantages and disadvantages of the two approaches? With an 8080A, there is no need for stepping stones like the one above, and it is not too difficult to calculate jump addresses manually. On the other hand, having to use three bytes every time leads to a somewhat longer, and possibly slower-running, program. With the 6502s branching technique, it can be difficult to calculate the displacement manually, and an assembler becomes even more desirable.

Branching has the major advantage of being intrinsically re-locatable — it doesn't use absolute addresses, and a routine can be placed, unchanged, anywhere in memory. This has considerable value when it comes to building a library of program segments which can be used in different programs in different places. There are ways of making programs which are absolute addressing re-locatable, but they are unsuited to manual assembly.

The 6502 and 8080A conditional jumps are shown in Table 1. For the sake of convenience, I've called the 6502 branches jumps, but don't be confused. The mnemonics should be fairly clear — "B" for branch and "J" for jump — but the following notes may help:

- **6502 — BEQ,BNE.** These are the 6502 instructions which test the "Z" flag. It may help to remember them as "Branch if EQual to zero" (Z = 1) and "Branch if Not Equal to zero" (Z = 0). The equivalent 8080A mnemomics are, I think, more understandable.
- **Carry** tests. Remember that these tests monitor the carry flag, and the two micros set this flag in different ways. During substraction, the 6502 sets C to "0" to show a borrow, while the 8080A represents this conditions with "1". Thus, if you're testing for a borrow, you must use "BCC" for a 6502, and the opposite test, "JC", with an 8080A. There are no differences during addition.

Let's now test the effect of jumps with two programs. The first is a multiplication routine, and the second is a time-delay generator.

There are many ways of performing multiplication in a computer. The simplest and also probably the least efficient, way of computing:

$$c = a * b$$

is initially to set "c" to zero, and then add "a" to it "b" times. Figure 4 is a flowchart for multiplying two unsigned numbers together using this approach. You can see how it uses "b" as a iteration counter, and makes a decision i.e., a conditional jump — depending on whether or not "b" is zero. I'm assuming that neither "a" nor "b" is initially zero.

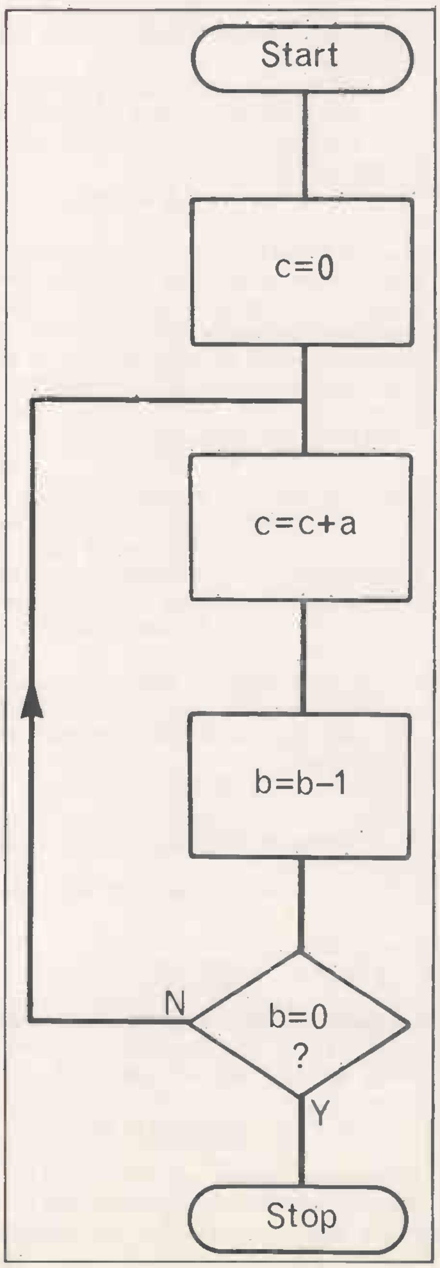
If "a" and "b" are both 8-bit numbers, then the final answer will be 16 bits long. Using the 8-bit arithmetic of our two micros, we can get this effect by forming the low byte of "c" in the accumulator, and incrementing a register containing the high byte of "c" every time a carry occurs. There will never be a carry of more than "1" from a single addition, of course.

The multiplication routine may go through the addition loop up to 255 times — i.e., if b = FF_{16}. We can speed things by initially testing "a" and "b", and making the larger one the multiplicand "a" and the smaller one the multiplier "b". The number of passes through the addition loop will then be "b'".

One way of finding the smaller of the two numbers is to subtract one from the other, and look for a borrow. If there is one, then the subtrahend is larger than the minuend. Don't forget to save the original values of "a" and "b" while making the test. With these refinements, the flowchart of figure 4 becomes that of figure 5. address — "DATA1" — will contain the least significant half of the answer. The resulting programs for the 6502 and the 8080A form figure 6a and 6b repsectively. We can draw a number of points:

(continued on next page)

Figure 4. Basic multiplication flowchart.



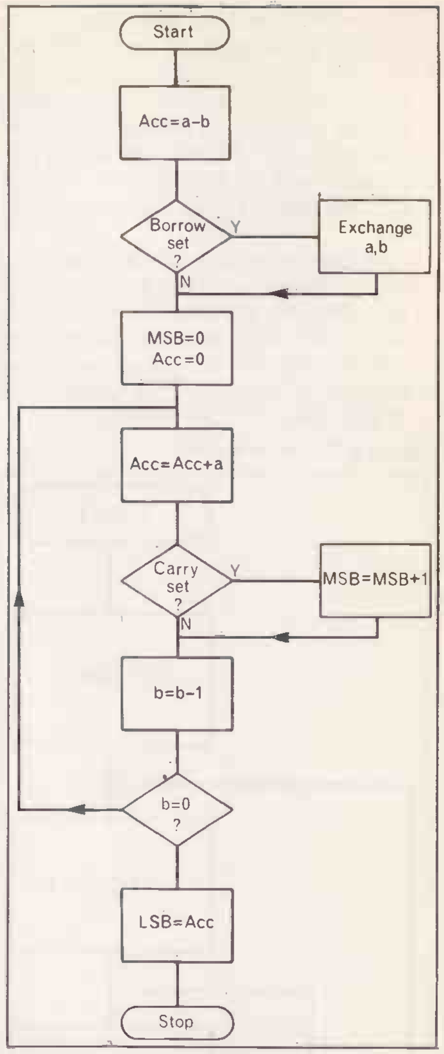


Figure 5. Final multiplication flowchart.

Number of clock cycles per instruction.

Table 2a — 6502

Instruction	Cycles	Remarks
LDX	2	Immediate Load
DEX	2	
BNE	3	2 cycles if no branch occurs
SEC	2	
SBC	2	Immediate subtract

Table 2b — 8080A

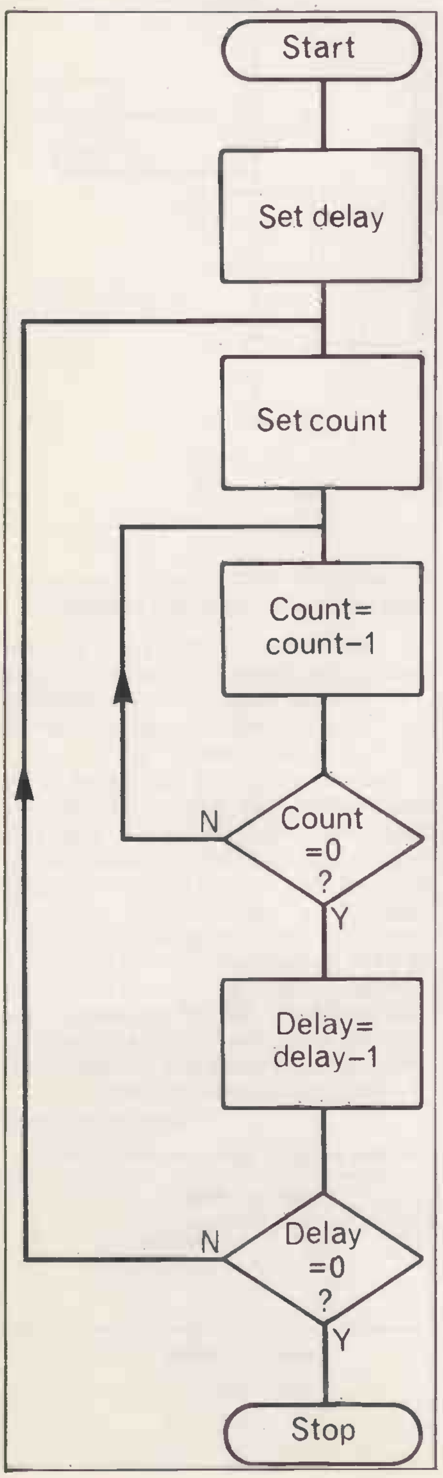
Instruction	Cycles	Remarks
MVI r,	7	
DCR r	5	
JNZ	10	Whether or not jump occurs

(continued from previous page)

- **6502 Program.** Initially, we need to put the "D" and "C" flags into the right state — i.e., "0" and "1" respectively. Having done this, X is loaded with the multiplier "b" to be used as a loop counter, and the accumulator is set to multiplicand "a" for the magnitude test.

Before we modify the accumulator, "a" is stored in "MULT" for future use. We then perform the subtraction which tests for magnitude — note the test for "carry = 1", which means no borrow. If necessary, the contents of X and "MULT" are swopped.

Figure 7. Timer flowchart.



- **8080A Program.** The 8080A program is logically identical to the 6502, but makes use of the extra internal registers, and of the indexed addressing provided by (H,L). This means that the program would be shorter and faster-running than that of the 6502.

The B register is loaded with the multiplicand via "MOV B,A" rather than "MOV B,M", since the former instruction executes more quickly; having loaded the data, B corresponds to the 6502 "MULT", and C to the 6502 X — i.e., the loop counter. The magnitude test and the exchange are similar to those in the 6502, apart from the test being for "no carry".

It is often useful to be able to generate a variable time delay in a program — a typical requirement is to delay a defined number of mSec. The program to do this is very simple and is shown in figure 7.

The inner loop, and the outer loop instructions exercised on each 1 microsecond iteration, are adjusted via the value of "COUNT" to run for exactly 1 microsecond. Then, the time taken to exercise the full routine "DELAY" times is "DELAY" microsecond.

The first step is to write the program. Then, from a knowledge of how long each instruction takes to be executed, we can calculate a suitable value for "COUNT". First the two programs; figure 8a and 8b show them for the 6502 and 8080A respectively. They are both very simple, and use the accumulator to count the number of microseconds to be delayed; a second register is used to control the internal timing loop.

The key, however, is finding the right value to set "COUNT" to in each case. To do this, we need to know two things: how many clock cycles each instruction takes to be completed, and the duration of each clock cycle. It is normal for a 6502 to use a 1 MHz clock, and an 8080A a 2 MHz clock. The clock cycles of the two micros thus last 1 microsecond and 0.5 microsecond respectively — table 2a and 2b.

Figure 8a. Basic 6502 timer.

```
;START OF BASIC 6502 TIMING LOOP. ASSUME THAT THE
;DELAY IN MSEC IS PASSED IN THE ACC
;
;SET UP "COUNT"
DELAY    LDX#COUNT    ;HERE AT START OF EACH MSEC
DELAY1   DEX
         BNE  DELAY1  ;INNER LOOP OVER?
         SEC          ;CARRY = 1 FOR NO-BORROW
         SBC # !      ;DECREMENT ACC
         BNE  DELAY   ;TIME DELAY OVER?
;
END OF TIMING ROUTINE
```

Figure 8b. Basic 8080A timer.

```
;START OF BASIC 8080A TIMING LOOP. ASSUME THAT THE
;DELAY IN MSEC IS PASSED IN THE ACC
;
;SET UP "COUNT"
DELAY    MVI  B,COUNT ;HERE AT START OF EACH MSEC
DELAY1   DCR B
         JNZ  DELAY1  ;INNER LOOP OVER?
         DCR A        ;DECREMENT MSEC
         JNZ  DELAY   ;TIME DELAY OVER?
;
;END OF TIMING ROUTINE
```

Taking now the 6502, making an N microsecond delay, the following number of instructions occur:

N * (LDX + "COUNT" * (DEX + BNE) + SEC + SBC + BNE)

Putting in the time of each instruction, this gives:

N*1000 = N*(2 + "COUNT"*(2 + 3) —1 + 2 + 2 + 3)—1

The two "—1"s represent the last pass through each "BNE", when no branch occurs, and correct the normal 3 microseconds duration to 2 microseconds for those occasions.

Figure 9a. Accurate 6502 timer.

```
;ACCURATE 6502 TIMING LOOP
DELAY    LDX#198
DELAY1   DEX
         BNE  DELAY1
         NOP          ;FINE TUNING
         SEC
         SBC# 1
         BNE  DELAY
;END OF TIMER
```

Figure 9b. Accurate 8080A timer.

```
;ACCURATE 8080A TIMING LOOP
DELAY    MVI  B,86
DELAY1   DCR B
         NOP          ;FINE
         NOP          ;TUNING
         JNZ  DELAY1
         DCR  A
         JNZ  DELAY
;END OF TIMER
```

Cancelling out, and ignoring the final "—1 microsecond" — thus accepting a fixed 1 microsecond error in any delay period — we have:

$$\text{"COUNT"} = \frac{1000-8}{5} = 198.4$$

If we set "COUNT" to 198, we will have an 0.2 percent error in any time delay; this is probably acceptable. If, however, the delay must be exactly N microseconds — we must tune the program slightly. If we could introduce an extra 2 microseconds' delay into the outer loop, then:

$$\text{"COUNT"} = \frac{1000-10}{5} = 198.0$$

Can we do this? Yes we can — the trick is to use an "NOP". You'll recall that this instruction does nothing, but it takes 2 microseconds to do so. Thus, it can be used to make fine timing adjustments of this type. The final, accurate, program is in figure 9a.

Let's go through the same process for the 8080A program. An N microsecond delay generates the following number of instructions:

N * (MVI + "COUNT" * (DCR + JNZ) + DCR + JNZ)

or, in terms of 0.5 microsecond cycle times:

N*2000 = N*(7 + "COUNT"*(5 + 10) + 5 + 10)

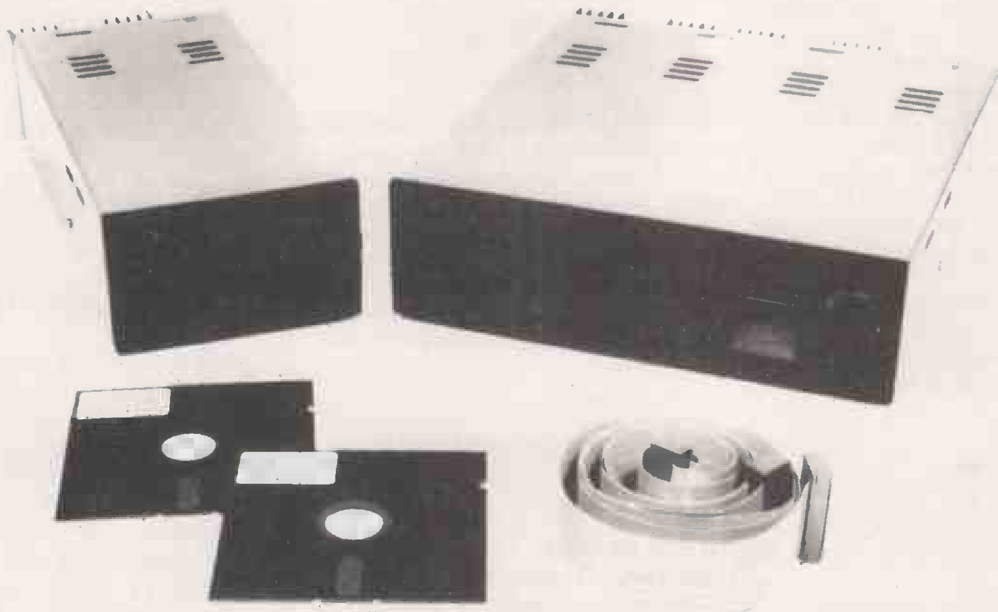
from which:

$$\text{"COUNT"} = \frac{2000-22}{15} = 131.9$$

Normally, it would be good enough to set "COUNT" to 132. If we must be completely accurate, add an eight cycle — 4 microseconds delay to the inner loop:

$$\text{"COUNT"} = \frac{2000-22}{23} = 86.0$$

Remarkably enough, an 8080A "NOP" requires four cycles. We can thus add two to the inner loop to give the accurate timer of figure 9b.



Useful tips

WALTER WALLENBORN, secretary of the 6502 Users' Club, has sent some tips for those who have machines using one of the versions of Microsoft Basic Pet, AIM, UK101. One thing which is handy especially in business applications is to store some constant, he writes. These constants — such as day and date — need to be passed from one program to the next without the loading of successive programs corrupting it. The easiest solution is to lower the end of memory pointers. Once this is done the space above the new top of memory will not be interfered with by the interpreter.

This program will put away string variables and recall them. The area created in this way is the ideal place for your machine language programs such as printer drivers.

The 6502 Users' Club also reports that there is now a club in Denmark. Those interested should write to E Skovgaard, Nordlundsvej 10, dk-2650 Hvidovre, Denmark.

```
5 REM TO STORE YOUR DATA
10 INPUT "YOUR DATA"; A$
20 A=LEN(A$)-1:POKE MEMEND+1,A
30 FOR I= MEMEND+2 TO MEMEND+2+A
40 POKE I,ASC(MID$(A$, I- MEMEND,1))
50 NEXT
60 END
105 REM TO RECALL YOUR DATA LATER IN THE DAY
110 A= PEEK(MEMEND+1)
120 FOR I= MEMEND +2 TO MEMEND +2+A
130 A$=A$+CHR$(PEEK(I))
140 NEXT
150 PRINT A$
```

Data saving

TWO SUBROUTINES which could help UK101 owners with their data saving problems have been submitted by L Ritchie, of Baberton, Edinburgh, who believes that R J Newman's letter in February 1980 highlights a weakness in the UK101 interpreter — and that of any other computer using a similar version of Microsoft Basic — namely the absence of specific instructions to save and load data from the serial interface and hence the cassette.

The method of using a machine code subroutine to perform such operations is perfectly acceptable, he writes, and the most efficient approach, but may be inconvenient in some cases. An alternative which requires no departure from Basic is very simple.

The key to the solution lies in the control of the LOAD and SAVE flag bytes at memory locations 515 and 517 respectively. When bit 0 of the SAVE flag byte is set to "1" a PRINT statement will send the variable to be printed to the cassette as well as to the VDU. Similarly, when bit 7 of the LOAD flag byte is set to "1", an INPUT statement will take its input from the cassette rather than from the keyboard. The following subroutine can therefore be used to save the value of a variable A on cassette:

THE 6502 SPECIAL re-appears this month after its successful debut in the January edition which prompted more than 50 letters to the 6502 Users' Club, but none to *Practical Computing*. It is up to you, the reader, to help establish this page with your ideas, problems and guidance for other 6502 users. Please mark your letters 6502 Special.

```
110 POKE 517,1
120 PRINT "D";A
130 POKE 517,0
140 RETURN
```

The "D" in line 120 may be any non-numeric character. This character is required when re-loading the data from cassette. The input data will appear as a string of ASCII characters and if the first of these characters is numeric this will be interpreted as a line number with disastrous consequences. The following subroutine will input the variable A from the cassette.

```
210 POKE 515,128
220 INPUT A$
230 A = VAL (RIGHT$(A$,LEN(A$)—1))
240 POKE 515,0
250 RETURN
```

String to tape

KEVIN FORD of Lincoln has also offered a solution to R J Newman's problem. He suggests that one writes a string to tape:

```
1Ø POKE 517,1
2Ø PRINT"#";X ;"%"
2Ø PRINT"#";A;"%"
3Ø POKE 517,Ø
```

One disadvantage with this method could arise with long strings into which Basic would insert a CR during output. The following routine avoids this problem by directly POKEing the cassette port:

```
9999 REM STORE X$ TO
TAPE
1ØØØØ CA=6144Ø:X$="#"+
"%"
1ØØØ5 REM# AND % ARE
DLIMITERS
1ØØ1Ø FOR I = 1 TO LEN
(X$)
1ØØ2Ø P=ASC(MID$(X$,I,
1))
1ØØ25 REM GET CHAR FROM
X$
1ØØ3Ø WAIT CA,2
1ØØ35 REM WAIT UNTIL
PORT IS READY
1ØØ4Ø POKE CA+1,P
1ØØ45 SEND BYTE TO PORT
1ØØ5Ø NEXT:RETURN
```

The # and % characters are used by the input routine and if strings of 253 characters or more are to be stored, would have to be output separately. The # is not essential but the % sign tells the input routine when to stop looking.

The next routine will input a string from cassette whichever of the above methods was used to store it:

```
19999 REM LOADS X$X
FROM TAPE
2ØØØØ X$="":IND=Ø:CA=
6144Ø:REM SET UP STRING
ETC.
2ØØ1Ø WAIT CA,1:REM
WAIT FOR PORT TO FIND
BYTE
2ØØ2Ø P=PEEK(CA+1):REM
GET BYTE FROM PORT
2ØØ3Ø IF IND=Ø AND P<>
35 GOTO 2ØØ1Ø:REM IGNO
RE UNTIL IS FOUND
2ØØ4Ø IF P=35 THEN IND
=1:GOTO 2ØØ1Ø:REM FOUND
NOW FIND STRING
2ØØ5Ø IF P=37 THEN RET
URN: REM FOUND % SO
FINISHED
2ØØ6Ø X$=X$+CHR$(P):
GOTO2ØØ1Ø:REM ADD CHAR
TO STRING
```

To save a numeric variable using the second routine, the program must:

```
1Ø POKE 517,1:REM POKE
SAVE FLAG TO ACTIVATE
PORT
2Ø X$=STR$(A)  or 2Ø X$
=A$for a string
3Ø GOSUB 1ØØØØ
4Ø POKE 517,Ø:REM POKE
SAVE FLAG TO DEACTIVATE
PORT
```

To load, a variable program must:

```
1Ø POKE 515,1:REM POKE
LOAD FLAG TO ACTIVATE
PORT
2Ø GOSUB 2ØØØØ
3Ø A=VAL(X$) for numeric
or 3Ø A$=X$ for a string
4Ø POKE 515,Ø:REM POKE
TO DEACTIVATE PORT
```

The above routines are expanded for clarity and no provision has been made for tape economy. Preferably a block of values would be collected and saved together though on tape machines with a pause facility, the "#" delimiter helps to prevent spurious loading if it is used.

TANDY FORUM is devoted to the Tandy TRS-80. Sometimes we will use it to pass on news about the TRS-80 but, above all, it is for users, and would-be users, of the well-established model I and now the new model II. With your tips, queries, moans and comments, this page can become a market-place for TRS-80 information.

Name display

TANDY USERS are still supplying solutions to the problem of displaying the name of a SYSTEM program. Stephen Zara, who lives in Sussex, claims that this Level II Basic program will work, although we have not been able to test it. The name is to be found as the second and seventh characters after the header bytes. The program should run without a MEMORY SIZE? allocation.

```
0  CLS
1  POKE 16553,255:CLS:PRINT"INSERT
   TAPE & PRESS PLAY";
2  PRINT"HIT ANY KEY WHEN READY"
3  R$=INKEY$:IF R$="" THEN 3
   'PAUSE UNTIL KEY PRESSED
4  FOR N=20224 TO 20248 'SET UP
   MACHINE CODE ROUTINE
5  READ X:POKE N,X
6  NEXT N
7  PRINT"PROGRAM NAME : " 'THIS
   ON 2ND LINE OF VDU FOLLOWED
   BY NAME
8  POKE 16526,0:POKE 16527,79
   'MACHINE CODE AT 20224 HEX 4F00
9  A=USR(0) 'CALL ROUTINE— READ
   FROM 'DATA' BELOW
10 DATA 205,18,2,205,150,2,205,53,2,33,
   79,60,6,6,205,53,2,119,35,16,249,205,
   248,1,201
```

The program calls the following ROM routines (HEX addresses)

```
0212  start tape
0296  read past header
0235  read a byte from tape into register A
01F8  stop tape
```

POKE 16553,255 ensures that READ works after use of cassette.

Briefer answer

AN EVEN SHORTER solution to the same problem has been submitted by Clive Davidson of Derby. He claims that this small Basic program will extract the same said file name.

```
5  CLS
10 INPUT #-1,A$
20 REM EXTRACT FILE NAME
30 B$ = MID$(A$,2, (LEN(A$)-3))
40 PRINT "FILE NAME IS";B$
```

Note the file name is always preceded by U and <. This information is removed by line 30 to give the file name.

Invisible plan

CHESS FERRIER of Homerton, London, has written in with this method of making a program listing invisible: Write your program as normal, and once finished, edit each line in the following manner:

- Type 'EDIT' followed by the line number ENTER.
- When in edit mode, type 'X', which will put the cursor at the end of the line.
- Press the 'SHIFT' key down and keep it pressed down throughout the rest of the operation.
- With the shift key still down, type '←' until the cursor is at the extreme left of the line — at the beginning of the line number. Keeping the shift key down, the space bar is pressed until you have moved the cursor well past the end of the line to be made invisible.
- If you now list the program, that line will not appear, though the program will still work well.

To change the invisible lists back to visible, again edit the line. This time, type 'L'. You will now find that as you space over the line, the characters will re-appear.

Chess Ferrier adds that he hopes that Tandy Forum readers will write in with plenty of short-cuts and neat tricks for using the Tandy Quick Printer. The main use for this line printer, he writes, must be for the hobbyist, as a hard-copy listing printer. As the paper is only 6cm — 2.5in. — wide, its use is restricted.

You have two rolls of aluminium paper about 20ft. each, the bus-cable to plug directly to the TRS-80 keyboard, a very clear, spacious and thin user manual and the Quick Printer thrown in to help stop the bits rattling in the box.

The printer can be used with or without the expansion interface — one of the main reasons he bought one — though the bus-cable for the expansion interface is supplied separately for about £12.

There are four inputs to the printer: the TRS-80 bus 'TRS BUS' which is direct to the keyboard; the 'PARALLEL' expansion interface (26-1140); the 'SERIAL' (RS-232C) devices and 600 baud.

Mode settings are ON LINE, OFF LINE and RESET. The paper loading is easy; it takes about 30 seconds for a beginner or 10 minutes blindfolded with one arm tied behind your back. It can print upper- or lower-case in 32 or 16 characters per line. Note that when in normal use, upper-case 32 characters per line will be used in a 7 by 5 matrix.

Decoding puzzle

ALAN PEARMAN'S information on decoding the TRS-80 keyboard in January 1980 is incorrect in several important respects, writes SJ Baker from Worthing, Sussex.

The SHIFT keys do map on to the memory area and can be tested for in a manner identical to the remainder of the keys — PEEK (14528) = 1 if depressed. Both SHIFTS appear at the same address and with the same value so you cannot tell which of the two shifts it was.

The BREAK key does appear in the position Pearman suggests — but you have to use an assembler routine to find this out reliably.

The PEEK addresses suggested are not the best — try the table given below — thus one would have to peek 14336 + 8 to detect either X, Y or Z, the value returned telling you which of them are currently depressed.

	(ADDRESS - 14336)							
Value	1	2	4	8	16	32	64	128
1	@	H	P	X	0	8	ENTER	SHIFT
2	A	I	Q	Y	1	9	CLEAR	
4	B	J	R	Z	2	:	BREAK	
8	C	K	S		3	;	↑	
16	D	L	T		4	,	↓	
32	E	M	U		5	.	←	
64	F	N	V		6	-	→	
128	G	O	W		7	/	SPACE	

Pearman implies that it is only necessary to test the location for equality with the desired key value. Try this routine:

```
10 IF PEEK(14400)=1 THEN PRINT
   "STOP PRESSING MY ENTER KEY !!!"
20 GOTO 10
```

The message should appear whenever the enter key is being depressed. If you run the program, hold down the space bar and simultaneously depress the ENTER key, the message does not appear. The solution is to mask the effects of other keys using the AND function.

```
IF (PEEK(14400) AND 1) 0 THEN . . . . .
```

L&J COMPUTERS

3 CRUNDALE AVENUE, KINGSBURY NW9 9PJ 01-204 7525

THE "PET" SPECIALISTS

HARDWARE

COMPARE OUR PRICES!

NEW PET 8K	£499
New PET 16K	£599
New PET 32K	£725
EXTENSION CASSETTE DECKS (WITH COUNTER)	£55

DUAL DRIVE DISK UNITS

PET 3040	£740
COMPUTHINK 400K	£949
COMPUTHINK 800K	£1099

PRINTERS

PET 3022	£620
CENTRONIC 779 *(Tractor feed)*	£869
SWT PR 40 (Inc Interface) *(becomes an on-line printer)*	£295
TELETYPE 43	£375

SUNDRIES

TOOL KITS	from £55
VARIOUS INTERFACES	from £40
FLOPPY DISCS	from £2

PAPER: ROLL & TRACTOR FEED
FULL RANGE OF PETSOFT & COMMODORE SOFTWARE
If you can't see what you want — ask.
We pride ourselves on getting what the Customer wants — quickly!

PRICES DO NOT INCLUDE VAT

SPECIAL OFFER

BASF 5¼ FLOPPY DISCS — 30 for £72

SYSTEMS

WE OFFER SMALL BUSINESS SYSTEMS FOR UNDER £2099 + WHICH CAN NOT ONLY CARRY OUT MOST OF THE DAY TO DAY CHORES, BUT ALSO HAVE ROOM TO SPARE TO COPE WITH THOSE "ODD JOBS". TAKE YOUR CHOICE!

EACH PACKAGE COMPRISES A 'PET' 32K, CABLES, ANY INTERFACES REQUIRED, BUT DOES NOT INCLUDE PAPER, DISKS ETC.

	TRACTOR FEED PRINTER		
DUAL DRIVE DISK UNIT	*PET 2022*	*TELETYPE43*	*CENTRONIC779*
PET 3040	*£2099*	*£2399*	*£2399*
COMPUTHINK 200K × 2	*£2129*	*£2429*	*£2379*
200K × 4	*£2289*	*£2589*	*£2539*

PRICES DO NOT INCLUDE VAT

SOFTWARE

As well as a full range of Petsoft and Commodore Software, we have some highly reliable "Home-Brewed" programs available.

STOCK CONTROL & INVOICING £60
(Handles up to 500 items — 32K) (180 on 16K). Stock depleted on invoicing, search etc. Cassette, disk (& print option).

STOCK CONTROL & INVOICE (RANDOM ACCESS) £120
On-line handling of 3400 items (50 char.) per disk. Disk (+ printer option) (16 or 32K) search etc.

ADDRESS/PHONE BOOK £35
Create, amend, enlarge, search (+ print option) (16K or 32K).

OUTSIDE SERVICES £220
Weekly or monthly invoices — cheque writing facility — optional deductions. (16 or 32K + disk + printer).

YOUR MONTHLY PLANNER £7
Print your own planner on your own printer! Any year or month.

RANDOM ENTRY & ANALYSIS £40
Makes adding up all those different invoices childs' play! Cash, cheques etc., balances & VAT.

CASH BOOK £90
Enter daily/weekly amounts — printout and totals, weekly/monthly analysis, totals and balances.

SEND SAE. FOR FREE BROCHURE
PRICES DO NOT INCLUDE VAT.
We can write specialist programs for you. Ask us for a quotation.

BREAKTHROUGH!

IF YOU ARE A PET 2040/3040 DISC USER WE CAN DRAMATICALLY INCREASE YOUR DISC STORAGE CAPACITY. FULL OPERATING INSTRUCTIONS ONLY £10 — GUARANTEED!!

ACQUISITION

You are welcome to try, browse & purchase at our premises.
You can phone your order.
You can fill in & send off the form below.
Whichever way you choose, you cannot help but be delighted with our service.

ORDER FORM

TO L & J COMPUTERS:
PLEASE SEND ME BY RETURN:—

____________________ £ __________
____________________ £ __________
____________________ £ __________

ADD £12 for large items **
ADD £1 for small items

VAT £ __________

TOTAL £ __________

CHEQUE/P.O ENCLOSED*
CHARGE TO MY ACCESS/BARCLAY/VISA CARD No.*
*(*delete as req'd)*

____

ALL GOODS SENT SAME DAY WHEREVER POSSIBLE
*Recorded delivery by post: or Securicor.***

• Circle No. 190

Header damage

IN FEBRUARY we discussed how to recover data from a program tape on which the header had been lost. J Bloore, from the West Midlands, has written to describe a case where the header section of a tape was damaged and the data was recovered successfully.

The procedure I used, he writes, was to place a small radio near the cassette tuned so that I could hear the cassette output to the Pet. The damaged tape was first set to the start of readable data — in any event, after the end of the header tape section — then switch tapes in the following way.

Type Load, press Play and listen for end of header which can be recognised easily by a pause. When that point is reached, stop tape — don't touch Pet — switch cassettes and press Play button again. The program will now load until an end of file is found.

The next part is critical — do not attempt to run. If you then SAVE the program using normal techniques, I found that I had the complete program on tape on re-loading back into the Pet as a new program.

Going back to the end of the faulty tape load, if an attempt is made to run or if machine-code monitor is called, I found that the Pet crashed. I am not sure why but I think the reason is that the flags and pointers for program length, start or end of text, variables are fouled. By saving and re-loading, these flags are set correctly.

Educating with ERIC

AS A TEACHER responsible for computer studies in a secondary school, writes Clive Bulmer of Sunderland, Tyne and Wear, I have found that, although micros are a great help, there appears to be a terrific lack of educational programs on the market to back them up.

It is with this view in mind that I enclose 'ERIC', a program written by myself

which I hope will be of interest to other teachers and readers.

ERIC, written for the Pet is a teaching aid, tried and tested with lower-ability maths classes in mind. The program, running for approximately 30 minutes, shows a face — full size of screen — with large eyes, nose and mouth. Within ERIC's mouth, each multiplication table is printed, a line at a time, so that a small group of children — I've used up to 4 — can say their tables with ERIC.

At the end of each set of tables ERIC asks six questions on the last table. If the correct answer is given, ERIC's eyes flash and he responds with a "Correct". If the wrong answer is given, then the children are asked the question again until they give the correct answer.

At the end of the 12× table, ERIC lets the children know how many questions they had correct and how many wrong, thus allowing the teacher to keep some form of record of the children's progress.

The program can be adapted easily to ask any type of question within the various subjects taught within our schools. The important thing is that children can identify with a face rather than numbers on a screen. They are no longer reluctant to try.

Persistent problems

A L MINTER, of the Alphabet Company, has had some trouble with the Pet and Computhink disc drive and Expandamem and has had no joy from official sources.

Any program line that is moved across the join between the Pet memory and the start of the Expandamem memory becomes corrupted, he writes. This joint is at 8192 — top of Pet 8K memory — and 8193 — bottom of Expandamem memory. The problem has persisted through two exchange Pet main circuit boards and one exchange Expandamem board. It happens whether the Computhink disc system is installed or not.

The only cure I have been able to find, and it is not really a cure at all, is to have eight or 10 REM lines in the program at this position, so that as the program is edited below them, they take the corruption instead of proper program lines.

RND function

MIKE TODD, from Welwyn Garden City, Hertfordshire, has produced a lengthy explanation of the RND function on the Pet, and hopes that, after all the publicity given to the problems, it will be the final word. He claims that the clumsy PEEK/POKE solutions to the RND problem will no longer be necessary.

A great deal of publicity and comment has arisen about the RND function on the Commodore Pet, he writes. However, it seems to me that no-one really knows how to use the random number facility to its best advantage; thus the following ought to be the final — and definitive — word on the matter.

The RND function on the Pet makes use of a simple algorithm to generate a pseudo-random number. It does this by taking a number — the seed — and multiplying it by 0.0000000392767778, adding 11879546.4, swapping the bytes of the result, forcing the result into the range 0-1 and then putting the result in memory. Thus, if the seed remains the same, the

(continued on page 115)

```
READY.

1 PRINT"J"
2 REM ERIC BY CLIVE BULMER NOV.'79 (C)
5 GOSUB100
10 GOSUB300
12 GOSUB499
15 PRINT"                    HI THERE ! I'M ERIC "
40 FORG=1TO12 N=0
43 FORS=1TO1500:NEXTS:GOSUB300
44 PRINT"                    LET'S SAY OUR"
45 PRINT"             ";G;" X TABLE"
48 FORS=1TO1500 NEXTS GOSUB300
50 N=N+1:F=N*G IFN=13THEN60
51 PRINT"                    ";N;"X";G;"IS ";F
52 FORY=1TO2000:NEXTY
57 GOSUB300:GOTO50
60 FORS=1TO1500:NEXTS:GOSUB300
62 PRINT"                    I'LL TEST YOU ON"
64 PRINT"           THE";G;" X TABLE"
66 FORS=1TO2000:NEXTS:GOSUB300
68 PRINT"                    PLEASE PUSH MY BIG"
70 PRINT"          RETURN BUTTON AFTER"
72 PRINT"               YOUR ANSWER"
74 FORS=1TO3000:NEXTS:GOSUB300
75 H=INT(100*RND(1)+1):IFH>12THEN75
76 PRINT"                    WHAT IS";H;"X";G;
78 INPUTR
80 P=H*G
82 IFR=PTHENPRINT"                CORRECT !!!":L=L+1:GOSUB499
83 IFR<>PTHENPRINT"                  WRONG !!!":M=M+1:GOSUB280:GOTO76
84 J=J+1:IFJ=6THEN88
86 FORS=1TO1000:NEXTS:GOSUB300:GOTO75
88 J=0:NEXTG
90 FORS=1TO1500:NEXTS:GOSUB300
92 PRINT"                    I HOPE YOU ENJOYED"
94 PRINT"         THAT !!  YOU GOT"
96 PRINT"        ";L;"QUESTIONS RIGHT"
97 PRINT"         AND";M;"WRONG"
98 FORS=1TO6000:NEXTS:GOSUB300:GOSUB499
99 PRINT"                    BYE BYE !!":FORS=1TO3500:NEXTS:PRINT"J":END
100 FORX=32768TO33567
110 POKEX,160
120 NEXTX
130 PRINT"                                 "
140 PRINT"                                 "
150 PRINT"                                 "
160 PRINT"                       "
170 PRINT"                   "
180 PRINT"                   "
190 PRINT"                              "
200 PRINT"                            "
210 PRINT"                            "
215 PRINT"  "
220 PRINT"              E               ";
221 PRINT"             |R|              ";
222 PRINT"             |I|              ";
223 PRINT"             \C/              "
240 FORZ=1TO2000:NEXTZ:RETURN
280 FORL=1TO1000:NEXTL
300 PRINT"                                       "
310 PRINT"                            "
320 PRINT"                            "
330 PRINT"                            "
340 RETURN
499 FORT=1TO6
500 PRINT"   "
501 PRINT"          X            X  "
502 FORS=1TO75:NEXTS
503 PRINT"   "
504 PRINT"                          "
505 FORS=1TO75:NEXTS
506 NEXTT
507 RETURN
READY.
```

(continued from page 113)

random number will always be the same.

If one uses RND(1), however, the seed is always the last random number generated — the initial seed being set on switch-on. Therefore, RND(1) will always generate the same series of numbers from switch-on.

Most games programs really need something a little more random than this, so the RND routine allows RND(0) to set a truly random number on the new ROM Pets — on the old ROM Pets the seed is set from non-existent RAM locations which returns a noise-based number, while on the new ROMs, the 1MHz timers are used which produce a number which is unlikely to be repeated once in four million times. Unfortunately, continuous calling of RND(0) is not recommended since the timers change fairly little and the numbers returned are very similar. But if you use A=RND(0) at the start of your program, before you start using RND(1) throughout, the seed will be set at random, and the consequent series produced will be as good a random sequence as you could want. RND(1) should therefore always be used in the main program.

Additionally, using A=RND(—1) at the start would allow easy debugging, since this makes the seed = 1, and the same series of numbers will be called every time the program is run. When debugged, just change the —1 to 0 — or even —TI which has a similar effect, and is recommended for old ROM users — and a truly random series will be generated. So A=RND(0) or A=RND(—TI) is equivalent to RANDOMIZE, and could, of course, be used now and again through a program to give the sequence of numbers a slight shake-up.

Lisp club

IF ANYONE is still ambitious with their Pets, Cliff Stanford, of Impetus Computer Systems, who has taught himself Basic, Assembler and now Lisp has something of interest. He would now like to hear from any readers who would like to join him in starting a Lisp programmers' club. Impetus Computer Systems is at Classic Offices, Hendon Central, London NW4. Tel: 01-202 2726.

I mastered Basic, he writes, I struggled with Assembler, then, at last, my Lisp disc arrived. I read the manual, and decided to ignore the warnings that the manual is not designed to teach programming in the language.

The manual is well written, the software is excellent, although I have found two or three ways in which to crash it. For example, printing to a non-existent file causes a total crash, requiring powering the Pet off.

I am now able to program relatively simple functions in Lisp. A Lisp function is, in fact, a short program and a Lisp program is simply a function which calls a series of other functions. The first function I produced was called DISKERROR which returns the error string from the disc unit. Subsequently, I produced functions to mimic the Basic functions LEFT$, RIGHT$, MID$ and LEN. I also prepared a function to reverse a string — e.g., ABCDE becomes EDCBA.

Structured programming is something about which I had read but did not totally understand. I expect that many readers will feel likewise. Having been forced to structure functions for Lisp, however, a whole new world has been opened for me and GOTO may never be the same. Still, although I would not like to have to prepare a complete serious business package in Lisp, programming in it is an experience which I would recommend to anyone who was interested in looking beyond Basic.

```
(DE DISKERROR
    ((X) (Q))
    (CLOSE 15)
    (OPEN 15 8 15)
    (LOOP
        (UNTIL (EQ Q CR))
        (SETQ Q (GETCHAR 15))
        (SETQ X (CONS Q X)))
    (CLOSE 15)
    (IMPLODE (REVERSE X)))
```

Super Pet

THE NEW Commodore super Pet will include changes in the firmware Basic interpreter and the disc-operating system which could, according to some reports, mean trouble for the plug-compatible manufacturers such as Compu/Think, marketed in the U.K. by ACT. Yet ACT does not seem too worried. Here Julian Allason makes his own comment about the super Pet and the changes it could bring.

Returning from California, he writes, I found myself faced with persistent questioning about the new Commodore range of super Pets. On the subject of delivery dates, I should be surprised if Commodore does not make some announcement soon.

Major improvement

The super Pet, now officially-designated the 8000 series, made a brief appearance at the Las Vegas Show. It closely resembles present models, but has a larger 12in. screen. The upgrade from 40 to 80 columns is a major improvement, which will be welcomed by business users, particularly those interested in word processing applications.

More than 100 pre-production models have been built and despatched to Commodore offices worldwide. Internal reaction has been favourable not least from software managers relieved at the high degree of compatibility with current models. This means that existing Pet users should be able to up-grade to the super Pet without having to ditch all their existing software.

The revised Basic 4.0 implemented on the new machines is somewhat faster in operation. An attractive feature is the resident DOS which will be welcomed by those who find the present disc handling syntax unwieldy. There was some talk in the States about offering Basic 4.0 as a ROM retrofit for users of the current range at $250. This may happen here in due course but I am not sure how many takers there would be at such a price.

Compatibility

Harry Saal, who manufactures the Programmers' Toolkit, has been canvassing the idea that the super Pets could use a 6809 CPU instead of the 6502. The 6809 contains a series of 16-bit extentions to the current 8-bit architecture. Yet to the outside world it looks like an 8-bit processor, so all standard peripherals chips remain compatible.

Some 24 software houses and dealers participated in a remarkable exhibition of Pet-compatible products in February. The show for the benefit of Pet retailers, provided a fascinating overview of the support which the system now enjoys. Considerable interest was expressed in the MTU high-density graphics board imported by IJJ Design of Marlborough. The system appears relatively easy to use and is supplied with some software. Small Systems Engineering demonstrated the NEC Spinwriter complete with Pet interface which is widely tipped to become the standard letter quality printer for word-processing applications. Commodore has accepted it into their range of officially-approved products and both Petsoft and Dataview recommend it for the Wordcraft word processor, which was also on show.

New add-on

For dealers the show provided an opportunity to compare the many business packages now available. Pet users will have the same opportunity at the Pet Show which Commodore is staging at London's Cafe Royal on June 13-14.

A new Pet add-on which I have enjoyed using is the Presto-Digitizer tablet developed by Dr David Thornburg. Once plugged into the user port, and with a control program loaded, the table can recognise alpha-numeric characters written on the tablet. These are displayed on the screen or can be sent direct to a printer. One of the progams provided allows Pet ot learn to recognise your own handwriting. Recognition is extremely fast. One application which could be developed for the tablet is for the program to recognise shorthand characters. Referenced to a disc-based, look-up table, this could allow direct conversion of shorthand into hard copy.

Anagram program

HAS ANYONE any ideas about how to design a simple anagram program which convincingly covers all the permutations for words up to, say, 20 characters?

Sine table program

FREQUENTLY WHEN producing programs relating to, or actually making music, it is necessary to have some form of sine table somewhere in memory, which when accessed at different rates will produce different notes when suitably processed, writes BW Lawrence of Harlow, Essex. It is obviously very tedious to make such tables up by hand.

This simple program does the table construction for you, after asking for the various parameters necessary for the jobs. Its use is not restricted to the Apple and ITT 2020, but will probably run on any machine which has the INT and SIN functions.

```
1Ø   =PRINT "ENTER THE VALUES OF
      THE FOLLOWING PARAMETERS:"
      : PRINT
2Ø   INPUT "ADDRESS (IN DECIMAL)
      OF LOCATION OF START OF
      TABLE";ADDR
3Ø   INPUT "LOWER LIMIT OF VALUES
      IN TABLE (USUALLY Ø)";OFFSET
4Ø   INPUT "AMPLITUDE OF SINE
      FUNCTION";AMPL
5Ø   INPUT "NUMBER OF VALUES IN
      TABLES";NUM
6Ø   PI = 3.14159
7Ø   FOR I = Ø TO NUM
8Ø   X = INT(OFFSET + 0.5 + AMPL +
      AMPL * SIN(2 * PI * I / NUM))
9Ø   POKE (ADDR + I), X
1ØØ  PRINT X; " ";: REM: LETS YOU SEE
      HOW TABLE CONSTRUCTION IS
      PROGRESSING
11Ø  NEXT
```

Analogue input

MICROSENSE has an analogue input card for the Apple II — the A1-02 from Interactive Structures Inc. Up to 16 analogue channels may be monitored by the system with 8-bit resolution. Channels are addressable individually and conversion time is 70 microseconds. The system can be operated from Basic and also provides interrupt capability for more efficient software implementation.

Faster printing

KEITH PARKES, of Armstrong Computer Services, Telford, Salop, found the program from WH Skipton in Apple Pie, January, a trifle slow, especially when using a fast printer. The subroutine was used for printing whatever was currently displayed on the screen.

He has written two machine language routines to increase the speed.

The first routine is called from a Basic subroutine added to the body of the program from which you wish to obtain screen prints and is suitable for use both by parallel and serial printers, he writes.

The second routine is restricted to Apples with the Autostart ROM and a high-speed serial interface card. It is much more flexible in its use, however, since all you have to do is hit RESET any time you want the VDU screen captured in print. After the screen is printed, you are left in the Basic of the former program, but you have to re-run the program.

The Poke into $1A (dec 26) sets-up the left-hand margin. Memory locations Hex $308, $31F & $329 all contain $C2. This is for a card in slot 2. For any other slot, use $Cn, where n is the slot number.

I am sure that these can be improved upon; perhaps someone can make it continue the program as though the RESET had not been hit.

Parallel printer.

Call the subroutine by a GOSUB to:

```
63ØØØ D$ = CHR$(4):PRINT D$;"PR ≠
1":PRINT CHR$(9);"80N": CALL 768:
PRINT D$;"PR ≠Ø":RETURN
```

Serial printer.

Call the subroutine by a GOSUB to:

```
63ØØØ D$ = CHR$(4):PRINT D$;"PR ≠
1":CALL 768:PRINT D$; "PR ≠Ø":
RETURN
```

```
0300—   A9 00      LDA   #$00
0302—   85 00      STA   $00
0304—   85 18      STA   $18
0306—   A9 04      LDA   #$04
0308—   85 01      STA   $01
030A—   85 19      STA   $19
030C—   A0 00      LDY   #$00
030E—   B1 00      LDA   ($00),Y
0310—   20 ED FD   JSR   $FDED
0313—   C8         INY
0314—   C0 28      CPY   #$28
0316—   D0 F6      BNE   $030E
0318—   A9 8D      LDA   #$8D
031A—   20 ED FD   JSR   $FDED
031D—   A5 00      LDA   $00
031F—   18         CLC
0320—   69 80      ADC   #$80
0322—   85 00      STA   $00
0324—   C5 18      CMP   $18
0326—   D0 E4      BNE   $030C
0328—   E6 01      INC   $01
032A—   A5 01      LDA   $01
032C—   C9 08      CMP   #$08
032E—   F0 07      BEQ   $0337
0330—   A5 18      LDA   $18
0332—   85 00      STA   $00
0334—   4C 0C 03   JMP   $030C
0337—   A5 00      LDA   $00
0339—   18         CLC
033A—   69 28      ADC   #$28
033C—   85 00      STA   $00
033E—   85 18      STA   $18
0340—   C9 78      CMP   #$78
0342—   F0 07      BEQ   $034B
0344—   A5 19      LDA   $19
0346—   85 01      STA   $01
0348—   4C 0C 03   JMP   $030C
034B—   60         RTS
034C—   00         BRK
034D—   00         BRK
*
```

Insert this line in your program and hit RESET to print the screen:

```
2 POKE 1Ø1Ø,Ø:POKE 1Ø11,3:CALL
—1169:POKE 26,Ø:REM LEFT-MARGIN
```

```
0300—   A5 1A      LDA   $1A
0302—   85 24      STA   $24
0304—   A9 08      LDA   #$08
0306—   20 00 C2   JSR   $C200
0309—   A9 00      LDA   #$00
030B—   85 00      STA   $00
030D—   85 18      STA   $18
030F—   A9 04      LDA   #$04
0311—   85 01      STA   $01
0313—   85 19      STA   $19
0315—   A0 00      LDY   #$00
0317    A5 1A      LDA   $1A
0319—   85 24      STA   $24
031B—   B1 00      LDA   ($00),Y
031D—   20 00 C2   JSR   $C200
0320—   C8         INY
0321—   C0 28      CPY   #$28
0323—   D0 F6      BNE   $031B
0325—   A9 8D      LDA   #$8D
0327—   20 00 C2   JSR   $C200
032A—   A5 00      LDA   $00
032C—   18         CLC
032D—   69 80      ADC   #$80
032F—   85 00      STA   $00
0331—   C5 18      CMP   $18
0333—   D0 E0      BNE   $0315
0335—   E6 01      INC   $01
0337—   A5 01      LDA   $01
0339—   C9 08      CMP   #$08
033B—   F0 07      BEQ   $0344
033D—   A5 18      LDA   $18
033F—   85 00      STA   $00
0341—   18         CLC
0342—   90 D1      BCC   $0315
0344—   A5 00      LDA   $00
0346—   18         CLC
0347—   69 28      ADC   #$28
0349—   85 00      STA   $00
034B—   85 18      STA   $18
034D—   C9 78      CMP   #$78
034F—   F0 07      BEQ   $0358
0351—   A5 19      LDA   $19
0353—   85 01      STA   $01
0355—   18         CLC
0356—   90 BD      BCC   $0315
0358—   4C D0 03   JMP   $03D0
035B—   00         BRK
035C—   00         BRK
035D—   00         BRK
035E—   00         BRK
035F—   00         BRK
0360—   00         BRK
0361—   00         BRK
0362—   00         BRK
0363—   00         BRK
0364—   00         BRK
0365—   00         BRK
0366—   00         BRK
0367—   00         BRK
0368—   00         BRK
```

This section is open to the Apple user. In every issue we hope to print ideas, hints and comments about the Apple and its suppliers. They must come from you, so write and tell us what you know.

THE RESPONSE to our appeal for hints, tips and ideas from Apple users in January was enormous and the letters poured in for weeks, but it seems that many decided that one letter was enough. Don't think that if you are using your Apple for something really sophisticated, nobody else will be interested. There must be many subroutines you have written and tricks you have discovered which could help other Apple users. Apple Pie is your page and your chance to show the world you are still alive.

apple® computer

Sales and Service

16K Apple £695
+ VAT and carriage

We also supply • Lear Siegler ADM3A — **£515** for one.
£500 each for two or more.

• Verbatim
Single-sided minidiskette — **£23** per box of 10.
— **£21** for two boxes or more.
Double-sided 8m. diskettes — **£40** per box of 10
— **£38** for two boxes or more.
Carriage inclusive, VAT extra.

We are also a Cromemco dealer supplying turnkey packages for Estate Agents, Solicitors and Wholesalers.

Total Concept Systems Ltd

270 Wanstead Park Rd., Ilford, Essex. Tel: 01-554 1152

• Circle No. 198

TRS-80 OWNERS!

MODEL I

LEV II CASSETTE

Adventures:		Astronomy II	£7.50
The Count*	£8.50	Basic 1P	£11.00
Land*	£8.50	Basic Toolkit	£11.00
Mission Impossible*	£8.50	Biorythms	£4.00
Odyssey*	£8.50	Calendar Functions	£7.50
Voodoo Castle*	£8.50	Complex Maths	£8.50
Air Raid*	£8.50	Electric Pencil*	£50.00
Amaz' In Mazes	£5.00	Electronics Asst.	£6.00
Balloon Race	£5.00	ESP Tester	£4.00
Barricade*	£8.50	File Handling	£7.50
Baseball	£5.00	Finance I	£7.50
Bingo	£4.00	Finance II	£7.50
Bowling (Ten Pin)	£5.00	Forth (Incl. Primer)	£40.00
Breakaway	£4.00	Fourier Transforms	£7.50
Bridge Challenger	£8.50	Graph Builder	£6.50
Backgammon	£5.00	Ham Radio	£6.00
Concentration	£5.00	Home Finance	£6.50
Cribbage	£5.00	General Accounting	£8.50
Dogstar	£6.00	Inst. Calculator	£7.50
Fastgammon*	£10.00	Inventory Mod.	£11.00
Galactic Blockade	£5.00	Inventory FP.	£13.00
Game of Life*	£6.00	Inventory Control	£11.00
Hangman	£4.00	IQ Builder	£7.50
Kamikaze	£5.00	Keyboard 80*	£6.00
Mastermind*	£5.00	Level-III Basic	£26.00
Mean Checkers*	£11.00	Linear Programming	£7.50
Othello III	£4.00	Line Printer Plot	£7.50
Pentominoes	£6.00	Math Drill	£4.00
Pork Barrel	£6.00	Math Library I	£18.50
Pre School Games	£6.00	Microtext Editor	£6.00
Remainder	£5.00	Mortgage Calculator	£5.00
Robots	£4.00	Personal Finance	£6.00
Round The Horn	£6.00	PASCAL (Incl. Manual)	£26.00
Safari	£5.00	Pilot 2.0	£9.00
Santa Paravia	£5.00	Pre-Flight	£11.00
Sargon I	£11.00	Renumber*	£8.50
Sargon II	£16.00	RPN Calculator	£6.00
Ship Air Battles	£5.00	RSM 2 Monitor	£14.50
Slalom	£5.00	Simplify It	£13.50
Snake Eggs	£8.50	Spelling Narration	£7.50
Space Battles	£8.50	Spelling Programs	£7.50
Startrek III 3.4	£8.50	Statistics	£11.00
Taipan	£6.00	Star Finder	£7.50
Time Bomb	£4.00	Story Builder	£7.50
Treasure Hunt	£4.50	System Copy*	£8.50
Trek 80	£8.00	Tarot Cards	£6.00
Trolls Gold	£4.00	Teachers Assistant	£6.00
Warfare I	£5.00	76 Basic Programs	£20.00
X-Wing Fighter II	£5.00	Manual for Above	£7.00

DISK

Accounts Receivable II	£40.00
Advanced Personal Finance	£13.50
Amateur Radio System	£13.50
Data Base II	£25.00
Electric Pencil*	£75.00
Forth (Incl. Primer)	£45.00
General Ledger	£40.00
Inventory 2.2	£30.00
Inventory 2.3	£40.00
Inventory II	£50.00
KVP Extender	£16.00
Level I In Level II	£13.50
Mailist IV	£40.00
Newdos Plus	£47.50
Pilot 3.0	£15.00
RSM 2D Monitor	£15.00
Text-80	£30.00
ST 80-D	£40.00

Taranto & Associates Conversions of Osbourne & Associates Business Programmes

Accounts Payable	£90.00
Cash Journal	£40.00
Invoicing	£90.00
Accounts Receivable	£90.00
General Ledger	£90.00
Manuals for Above	£12.00
Complete Co-ordinated System With Manuals	£350.00

MODEL II

CP/M 2.0	£70.00
CBASIC-2	£70.00
Accounts Payable	£255.00
Accounts Receivable	£255.00
General Ledger	£255.00
Inventory System	£255.00
PASCAL/Z Ver 3.	£150.00
Super Sort III	£70.00
Word Star	£240.00
Manuals for Above	£13.50

NOTE. User Licence Agreement Require for Above Businessd Programmes.

Osbourne & Associates Business Programs In CBASIC Each £140.00

MANY OTHER PROGRAMMES AVAILABLE CALL FOR DETAILS.
ALL PRICES INCLUDE FIRST CLASS POST AND PACKING (UK ONLY).
SEND 50P FOR FURTHER DETAILS OF PROGRAMMES.
* Denotes Machine Language

MICROCOMPUTER APPLICATIONS
11 RIVERSIDE COURT, CAVERSHAM,
READING RG4 8AL, ENGLAND

Buy it with Access

TELEPHONE:
(0734) 470425

BARCLAYCARD VISA

Towards a COS for Apple II

The lack of a cassette operating system for the Apple II is highly restrictive in many situations including the schools and college environment. Hugh Dobbs examines possible solutions to the problem.

ONE OF THE few weaknesses of Apple II is the absence of a reasonable cassette operating system which can support named files. Cassettes are ideal for school use where students can keep copies of their own programs without going to the expense of buying their own discs. The system as it is does not allow for data files, nor is there any easy way of loading an unidentified file — and cassette storage becomes expensive if you can have only one program per tape, which is the recommended solution.

The trouble is that there are three different types of program file on some of my tapes; Applesoft (FP) programs, Integer Basic (INT) programs, and machine code programs or pure memory dump. It is not possible to LOAD an INT program from FP, nor the other way around — either nothing happens at all, or the Basic crashes altogether and has to be re-started.

Diagnosis

You then have to decide whether the problem is due to a tape fault — rare — to a wrong volume setting on the recorder — common if you listen to your programs — or to your being in the wrong version of Basic. Having decided on the last possibility, you then have to re-set, start the other version, re-wind the tape, find the start of the track, re-connect the input lead, re-set the volume control, and try LOADing again which is hard work.

The mcp/dumps are at least recognisable as such by ear — only one header, while the other tracks have two — but there is no indication of how long they are, where they are supposed to go, or what their function is, though there is nothing to stop you adding comments if you know how and have some spare memory space.

So, what should a cassette operating system do? And how?

At least, it should allow one to SAVE, LOAD and RUN named files in any language, from any language — assuming that one has FP in ROM, for the time being — and be extendable. Easy extensions would be automatic switching from one language to another — FP, INT, machine code and assembler, probably — and warmstarting whichever Basic one was in already — equivalent of 3DØ G in the DOS — without losing any existing program.

Harder extensions would include handling text files and a method, of some sort, of loading unidentified files and relocating them, when their type has been recognised, to the normal areas for INT or FP programs, or to some plausible area for machine code — the last would require human intervention.

The system would have to coexist with, and be callable from, the Basics, monitor, and assembler — at present the only way of leaving the assembler is to hit Re-set, but not DOS — in fact, it could duplicate DOS in many ways, and perhaps the easiest thing to do would be to modify DOS.

It would be loaded from tape using the normal monitor Read, and probably, like DOS, it would re-locate itself to the top of memory and set HIMEM: below itself when booted, leaving a JMP instruction in page 3 — either at 3DØ for compatibility with DOS, or at 3F8 so that it would re-start from monitor with CTRL-L, or somewhere else memorable. Temporarily, it will start at CØØ because I don't know how long it will be, and also because I have some ideas about using the second page of video RAM for advice and information, and that ends at about BFF.

Shared control

It will have to share control with whichever language is being used; this is only possible at two points in the system: at FD18 in the monitor input routine, where there is an indirect JMP to — contents of 38 and 39 — and at FDED in the monitor output routine, where there is an indirect JMP to — contents of 36 and 37.

All the languages use these routines, so it is possible to censor all input and output by changing the vectors at 36, 37, 38, and 39 H to point to the appropriate place in the COS program. COS then has to return control to the ordinary routines if it is not being addressed directly. The vectors are re-set to point into the monitor when one hits the re-set key, so it is then necessary to re-start COS.

Normally the vectors point to FD1B and FDFØ; to have a program which doesn't crash when one tries to Trace it, I have defined the entry points to COS as ØC1B and ØCF0 respectively, but have written the first versions of COS in such a way that (a) when debugged it can be fairly easily re-located, and (b) it can coexist with a peripheral such as a printer which would operate normally by changing the printout vector to point into its own control routines — PR 3, or whatever, directs output to C3ØØ, the first address in the printer interface ROM, and this is presumably re-initialised by the first few instructions there to point to the real entry point — or eventually with the text-in-HIRES-graphics routines — part of the APPLEVISION demonstration program — which also operate as a pseudo-peripheral 'attached' to the output vector. To do this, the initialisation routine for COS picks-up the existing output vector, stores it as its own output vector, and substitutes its own output entry point.

Routines

I shall assume that your system, like ours, is an Apple II — not Europlus — with 32K RAM — not critical — and Applesoft II ROM — not tape — and for the moment nothing else apart from a cassette recorder. Switch-on with hardware select switch set for on-board ROM so that you can use the mini-assembler conveniently. Type as follows — the first character is the prompt, so you won't type that —:

*CØØ:Ø (retn)

That sets the contents of RAM location CØØ to Ø.

*CØ1 < CØØ.CFE M

That is block Move instruction which stores the contents of CØØ in CØ1, the contents of CØ1 in CØ2, and the contents of CFE in CFF. Thus it sets the whole of page C to Ø.

* F666G

Note the change of prompt here, as that is a Go instruction which starts the min'as'm'. There should also be a 'bell'. If it didn't work, your hardware switch on the FP ROM board is in the wrong position — while the monitor is identical for FP and INT, FP does not include the utility programs minasm, floating point routines, and Sweet Sixteen.

```
!CØØ:LDX  4        four locations to be changed, 36, to 39
! LDA 35,X         pick up contents of 39 (...38.....)
! STA C12,X        store it in C16 (..C15 .....)
! LDA C16,X        pick up COS I/O vectors from C1A (..C19 .....)
! STA 35,X         and store them as appropriate
! DEX              next location
! BNE CØ2          do it again until all four done
! STX C2B          put a Ø in C2B; we'll use that as a flag
! RTS              end of initialisation
!C1B: PHA          re-start minasm at entry point for input
! LDA # 20         bit mask
! BIT C2B          bit 7 of ØC2B to N, 6 to V; Ø in bit 5 sets Z
! BMI C27          stub...branch on bit 7 of ØC2B
! BVS C27          stub...branch on bit 6
! BEQ C27          stub...branch if bit 5 not set
! PLA              A only contains original contents of square which has been set to flash
! JMP (C15)        exit to normal keyboard input
!CFØ: JMP (C13)    stub...exit to normal print routine
! (Re-set)         back to monitor
*C17: FØ ØC 1B ØC  put COS vectors CFØ and C1B where the initialisation routine will find them
```

The miniassembler is particular about the first character on each line. A re-start address must start immediately, while if you want it to go straight on to the next available location you must start with a space. It has only one error message, a^ under the first character which it can't understand.

*CØØL at this stage will give a disassembled version of the initialisation bit, four BRKs for C13 to C16, and junk for C17 to C1A, and then part of the input but C1B. Note those BRKs, as they will change when we test it.

*36.39 will display the contents of locations 36 to 39, which now should be FØ FD 1B FD, as usual.

*CØØG initialises the embryonic COS, but all you should see is * because so far it is almost totally 'transparent', and you can use the ordinary monitor routines, or the assembler, or Basic and not notice any change at all.

*36.39, however, will now show the jump vectors as FØ Ø 1B 0C, showing that COS is attached, and

*CØØL will show more junk at C13 to C16 where the old vectors have been stored. Nasty things will happen if you use either monitor or minasm to change COS, and the thing becomes autistic literally, totally introverted if you type

*CØØG as that re-sets the exist vectors to point to the entry points. The only cure for this is the re-set key, since the rest of the keyboard is disabled; re-set initialises the monitor, putting in the normal values for the jump vectors, and

*CØØG can then be used to restore COS. This behaviour is clearly not ideal, but that is part of the educational process; version 2 does not have this problem.

Now let us try doing something to the printout; but first save the first attempt at COS on tape (CØØ.CFF W) in case anything goes wrong. Then start the minasm.

```
!CFØ:CMP#C1    crash if COS was still connected; re-set and try again.
!CFØ: CMP#C1   compare letter to be printed with 'A'
```

(continued on next page)

(continued from previous page)

```
!BCS CF6        skip next line if carry set
! LDA#87        replace letter by 'bell'
! JMP (C13)     exit to monitor print routine
! (Re-set)      back into monitor from minasm
```

This will now be far from transparent — try connecting it by
C00G and then try every key in turn. Then re-set, change to INT, write a program of 10 lines or so, then CALL 3072, then LIST it. Re-set, back into minasm and change to BCS to BCC:

```
!CF2: BCC CF6   then re-set, connect it and repeat the tests.
```

Or how about a replacement code? Re-set, minasm, then

```
!CF0: CMP#8D    is it a 'return'?
! BEQ CF6       if so, leave it alone
! EOR# 00       no change at the moment
! JMP (C13)     exit
```

This is transparent, but try doing
*CF5:FF and see what happens — the ASCII series is reversed.
*CF5:00 restores normal communications.
Now change to INT:

```
>10 FOR A = 1 TO 10
>20 POKE 3317, A          decimal equiv-
>30 PRINT "HELLO"         alent of CF3 or
>40 NEXT A                something
 RUN
```

and if 'COS' is connected you obtain 10 lines or five-letter 'words' starting with 'IDMMN' and then a mysterious message. Now here's a funny thing: use the cursor controls, escD and escB, to return to the start of the message, and use the forward arrow to run over it. Notice anything? Use escB to return to the start the back arrow is disabled, and try again. Can you explain what happens?

We are going to need a command table for COS, to see if you can write a routine starting at CF0 which will allow you to type text directly into memory starting perhaps at D40. Test it by typing in a few suitable commands, then re-set, esc @ to clear the screen, and display the results 'if any' by

```
(retn)          down to third line or lower
```

*400 D40.D67M which should Move the first forty characters of your table into the top line of video RAM where you can see it. There are other ways of putting text into memory, but this will allow us to store control characters as well, without trouble; any strange things such as inverted black on white script, or 00s, will have to be added later.

So to conclude the first section: we have established the beginnings of both an input and an output routine for COS. The input routine will be used for READing from files and for EXECuting stored instructions — and probably for other purposes. The output routine will be used for WRITEing to files; for suppressing video and printer under NOMON conditions; but most of all, at first, for identifying COS commands — initially from keyboard, later from PRINT statements inside programs.

The second page of video RAM is used rarely as such, and probably most Apple users are only vaguely aware of its existence. It occupies the region 800 to BFF H 2048 to 3071 decimal and has the same slightly odd structure as the first page. In it the first 40 bytes are the top line on the screen, say, line 0, the next 40 are line 8, the next 40 are line 16; then there are eight unused bytes to round-off the count to 80 H, and the pattern is repeated with lines 1, 9, 17, gap, 2, 10, 18, ...7, 15, 23, and a final gap.

It can be displayed, from monitor, by typing
*C055 or, from either Basic by either PEEKing or POKEing at location —16299. C055 and C054, when addressed in any way, i.e., the operations LDA C055 and STA C055 and BIT C055 all have the same external effect, only differing in their effects on the accumulator and the processor status flags will set and re-set the switch or latch which selects page 2.

If you have not been using Basic, you should now see a page of junk which looks just like page 1 immediately after you switched-on, except that there is no monitor prompt and no cursor. The cursor lives on page 1; if you want to type directly on to page 2, the easiest way would be to re-locate the monitor printout routines into RAM — changing the JSR and JMP addresses, of course. Change the BASCALC section to read ORA # $ 08 instead of ...04, and ... it may be enough to attach it as an output bug, or it might be easier to copy a bit more of the monitor and simply Go into it — the printout vector will need to be changed anyway. Since I only want to use page 2 for storage, this does not concern me at present because there is an easier way.

What does matter is that page 2 is used for other purposes by Basic. INT uses it for variable storage, and FP uses it for program test — in the ROM version at any rate. That means that COS will have to set the LOMEM pointer — or the start-of-program pointer for FP — to some location beyond itself, leaving room perhaps for a few files in the final version. I will return to this problem later, under initialisation and changing between languages.

To become accustomed to switching between the two pages, try typing blindly. First CTRL-X in case you hit accidentally a few keys while admiring page 2, then C054 returns. Page 1 should now reappear, complete with everything that you have typed. Now switch back to page 2 and see how carefully you can type.

```
800:A0 retn         should give you a
```

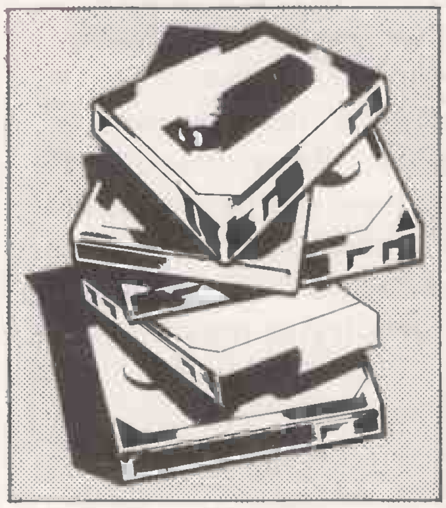
space at the top left-hand corner of the screen, and

```
801 800.BFE M retn  should clear the
```

whole screen to black, just to prove we

can do it. Now repeat the process but start with
800:00 retn instead, and you should end up with a screenful of black-on-white @ signs. Switch back to page 1 and type esc @CTRL-X and then
*400 800.BFF M which should give a copy of page 2 except for one line which starts with * followed by a flashing @ which is your cursor.

Now to store some useful information, type esc @ without hitting return, and then proceed as follows: type
HELP.....DISPLAYS THIS PAGE; HIT ANY KEY by which time the cursor has reached the start of the next line. Now use the left-arrow, and repeat, to run back to the H of HELP, then escC on to the next line. Then
FP........COLDSTARTS APPLESOFT
INT..COLDSTARTS INTEGER BASIC



ASM .....STARTS MINI-ASSEMBLER
ETC
and then use CTRL-J cautiously to reach the bottom line. If you go one step too far, you will have to start again; so when you are about five lines from the bottom type CTRL-X and then
*7D0:0 which will give a b-o-w @ in the bottom left-hand corner. Use CTRL-J and escB to reach that point and type THIS IS PAGE 2 OF TEXT starting with a space to remove the

Now escD up about five lines, and then CTRL-X, escB, escE, escD. This should remove the * and leave you on a line with a backslash. If there is any other junk on the screen, remove it similarly, then type 800 <400.7FF M but don't bit return. Instead, use escB repeatedly to move the cursor back to the '8'. Unlike the left-arrow, escB does not delete characters from the input buffer — 200 to 2FF H — so that your Move command is still there even when you clear that line with escE and finally hit return.

For those who are worried about this, I should explain that on Apple, esc is an ordinary key but is treated specially in the keyboard input routine which is shared by all the languages. Neither esc nor whatever key follows it is stored in the input buffer; nor are they echoed on the screen. Esc @ clears the screen and leaves the cursor in the top left-hand corner; escA, escB, escC, and escD move the cursor one step right, left, down, and up respectively; escE clears to the end of the current line, and escF clears to the end of the page; nothing else, including escesc, has any effect at all, and none of these has any effect on the input buffer which can be very confusing.

The end result of the last bit, anyway, is to copy the desired text onto page 2, with no extra rubbish; in this way, with modifications, it is possible to fill the whole page apart from the final character of one line — not the last line. To check the effect, switch to page 2; then switch back to page 1 and type
*800.83F which will give a Hex display of the first 64 bytes of page 2: ASCII values for the text displayed, but with the high bit set so that the last 24, for instance, are A0 — space = 20 H.
*840.87F will give the next 64 bytes, and note that it finishes with eight 00s which are not involved in the screen display.

The map so far is 800 to BFF, HELP page and storage and non-ROMable routines; C00 to C1A, initialisation; C1B to CEF, input bug; CF0 to D7F or so, output bug; D80 to DFF or so, keyword table: E00 to EFF will be either keyword-handling routines, or, if they are too long for that, at least jumps to those. We can write the HELP routine now, though the beginning of it will need to be modified later so that HELP can only be called from command mode.

```
*E00:EA EA EA EA EA EA
six NOPs to leave space for change
*F666G   ASM
!E06:LDA C051
display to text, not graphics
!     LDA C055
switch to page 2 display
!     LDA C000        ;LOOP
check for keyboard input
!     BPL E0C
LOOP until high bit is set
!     LDA C010
then clear keyboard
!     LDA C054
switch back to page 1 display
!     JMP FD67
new line in command mode, then RTS
!     (re-set)
```

*E00G test the routine: it should display page 2 until you hit a key, then return to page 1 and give you what looks like a normal monitor prompt and cursor, but, in fact, the return address on the stack is FF84 and not FF6F, so don't try any commands just hit return. Try CALLing 3584 from either Basic, similarly.

Actually the start address will be E01, for reasons to be stated later, FD67 is the monitor routine GETLNZ, which gives double-spacing for consecutive input lines; we could also use FD6A which gives single-spacing.

BUYERS' GUIDE

The Buyers' Guide is a summary of low-cost computers available in this country. It appears each month; we add new computers and amend existing information, as required, to keep it up-to-date. Systems are listed by manufacturer.

As the Buyers' Guide is now being revised and updated in preparation for a new series which will appear on our Prestel pages, please send us information confirming the details contained in the entries and any new facts which you may possess.

ACORN COMPUTERS

Acorn. Single Eurocard-sized microcomputer with 6520 processor, IBK RAM, 16-way I/O. Max size; a second Eurocard adds hex keypad and CUTS cassette interface. Monitor and machine code programming now, Basic and disc operating system in the future. "Highly cost-effective basis for a computer or an industrial development system". Available from Acorn (0223) 312772 or Microdigital (051) 236 0707.

£74.75 kit, £86.25 assembled

APPLE COMPUTERS

Apple II Plus. Min size: 16K memory, 8K ROM, keyboard monitors; mini assembler; colour graphics; Pal card; RF modulator; games; paddles and speakers; 4 demo cassettes. Max size; Expandable to 48K memory; floppy discs and printers are now available. Two versions of Basic, PASCAL; Assembler; games; business packages. An American system regarded as suitable for any kind of applications. Maintenance contracts offered. Microsense Computers is the sole U.K. distributor and has a national dealer network. Tel: (0442) 41191/48151 (24-hour answering service).

From £695.

ATTACHE

Attache: Min size: system with 10 slots, S100 bus, 8080 processor and 16KB housed in desk-top case with built-in keyboard. Max: 64KB, parallel printer interface, two single or double density 8" floppies, video screen. Disc Basic. Full business system includes all software. Sole U.K. distributor is Friargrove Systems Ltd. Tel: 01-353 3196.

From £1737. Full business system £7000

BASF

System 7100. CPU: Z-80A, 64K RAM, internal BASF, 5¼in. discs, single-sided, single-desnity, RS232, BOS BASF operating system, Basic assembler. Sole U.K. dealer MPR, Tel: Paul Curran 01-837 6332.

From £4,937

A PRACTICAL GLOSSARY

Continuing the terminological gamut from S to T

Stepper motor
Sometimes called a stepping motor. It is found frequently in electro-mechanical devices like printers. It rotates by a fixed amount every time it receives an electrical pulse. Stepper motors are generally the best solution where designers want precise, economical movements within minimum wear.

String
A sequence of records, words or characters — usually arranged in some specific order.

STX
Start of Text — a communications control character used to signal the start of a text transmission, but note that some of the message will have preceded STX — there will be a header of control characters before the text.

Subroutine
Some operations will be repeated frequently in a program, like passing results to the printer. You could write a program which repeats the printer call every time, or you could write a short piece of code which does the same thing such that your program can call it up every time it needs to.

That is a subroutine — a sequence of instructions which perform an often-required function and which can be called from anywhere within the body of the main program. Entering a subroutine is described generally as jumping or branching.

Subset
An identifiable bunch of things which belong to an identifiable but larger bunch of things. You'll probably encounter language subsets — which are programming languages with some, but not all, of the features of a well-known language. So Tiny Basic is a subset of Basic, though, as there isn't yet any agreed formal definition of what Basic ought to comprise, maybe the bigger Basics are really a subset of Tiny Basic.

Suite
A suite of programs is generally a collection of separate but inter-related programs run one after another to do a single, major job.

Superset
Something which is an enhanced version of the standard thing. A superset of Cobol, for example, might have all the facilities laid down for the language in the latest ANSI specification and more.

Symbolic
Programming languages are sometimes described as symbolic — programs refer to storage locations and machine operations by symbolic names and addresses which are independent of their literal hardware-determined locations.

A symbolic name is a label used in programs to reference data peripherals, instructions, and the like. In practice, symbolic means representing something by the everyday alpha-numeric symbols.

Synchronous
Often abbreviated to sync. It means two logical processes or parts of a system are kept in step because they are controlled by the same central timing mechanism.

In synchronous transmission, the sending and receiving instruments are operating continuously at the same frequency. That usually means a constant time interval between successive bits of characters sent and received.

Compare asynchronous operation, in which the sender and receiver are not in phase. It means any transmission has to include start-stop commands so that both ends know what is going on.

Syntax
Syntax means the rules which decide how programming language statements must be constructed — in other words, the grammar of the language. So if you try to key-in a program and the computer answers with the derisory message SYNTAX ERROR, you can look for these kinds of faults: duff composition — like typographical errors, incorrect punctuation, not following the specified form for statements or parts of statements, misuse of variables; inconsistencies — especially statements which are correct in themselves but which conflict with other statements; incompleteness — like GOSUB 9000 when there isn't a subroutine to go to at line 9000, or a FOR-NEXT loop with a FOR statement but no NEXT statement.

System
The word is used so often and so loosely that an exact definition is now very difficult. A practical one might run along the lines of a collection of parts united by some form of regulated interaction.

We might say parts and procedures, in fact, for the methods of using components are sometimes almost indivisible from the system elements themselves.

How about this: an organised whole — a complete assembly capable of functioning according to its defined intention.

System software
Look at the entry on applications. Software in general is in two varieties — applications software and system software. The former does something for the user; the latter comprises the tools which enable a programmer to develop and run applications.

So a basic set of system software generally includes a debugging program, operating system, a text editor, assembler, and an I/O system.

Systems analysis
Work involving the analysis of all phases of the activities of an organisation and developing detailed procedures for collection, manipulation and evaluation of all associated data. A systems analyst then, is a person who defines the applications problem, determines system specifications, recommends equipment changes, and designs dp procedures.

He or she probably passed through programmer training, but systems analysis work doesn't involve writing programs. However, it will mean producing block diagrams and record layouts from which the programmer can prepare flowcharts and subsequently write the programs.

Table
A collection of data items stored in memory, organised such that individual items may be referenced by specifying keys which are part of each item. The key may be very simple — it may just be the position of the data in the table.

Tandy
Pioneer of the mass market for personal computers and now second only to Commodore in total shipments. It sells in the States under its Radio Shack label, which indicates the home and hobby electronics field through which Tandy approached the micro.

Tape
Mag tape is the same as audio reel-to-reel or cassettes, though manufactured to rather higher tolerances — the fast idiot computer hasn't the sensitivity and the ability to compensate for faulty recording that human hearing can boast.

Tape is cheap, so it's good for the low-cost computer and for large archive files; but it must be accessed sequentially.

Tape head
That part of a tape deck or cassette unit which reads to and/or writes information from the tapes. These days, many tape units have to separate heads, one for reading and one for writing.

Among other things that allows a wonderfully logical read-after-write check on some decks ensuring meaningful information has, in fact, been put on to the tape.

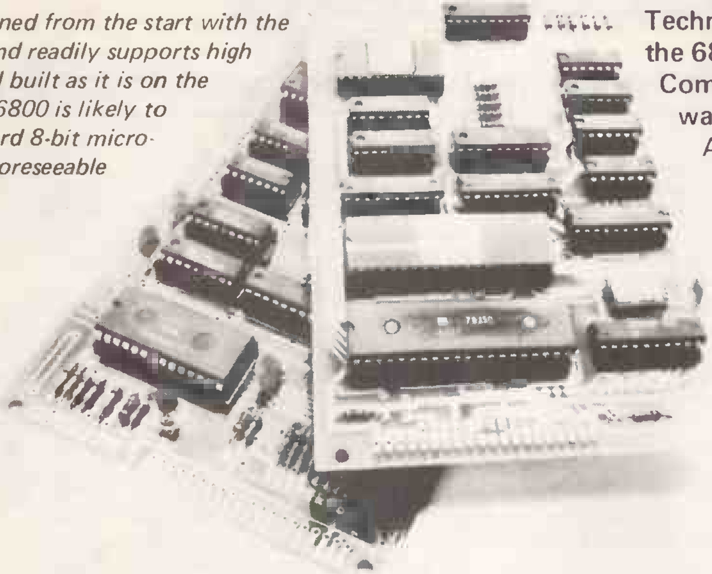
Telecommunications
The transmission and reception of data over radio circuits or telephone lines by means of electro-magnetic signals.

Teleprinter
Really it's just the typewriter-like device at one end of a telegraph line, but the term is used more generally to refer to any printer-plus-keyboard device on any telecommunications link — which is more accurately described as a Teletypewriter.

Teleprocessing
IBM invented the word to describe systems in which remote locations are connected to a central computer by data transmission circuits — usually telephone lines.

Maintain the lead with Acorn 6809

. . . the 6809 designed from the start with the programmer in mind readily supports high level languages and built as it is on the experience of the 6800 is likely to become the standard 8-bit microprocessor for the foreseeable future. . . .



Technology leader in microprocessors the 6809 is now available from Acorn Computer with full supporting hardware on a eurocard paired with the Acorn VDU interface at a special introductory price.

SPECIAL OFFER £160 + VAT

Acorn is offering their two most powerful modules as the basis for a 6809 development system requiring the addition of keyboard, power supply and monitor. For existing owners of Acorn Systems, the 6809 CPU card is a direct plug-in replacement for the 6502 CPU and can be used with all the supporting cards presently consisting of 8K memory, tape interface, VDU interface, Floppy disc drive, Analogue to digital/digital to analogue and Universal interface.

For newcomers to Acorn the two card system can readily be linked to terminals printers, etc., the operating system firmware is designed for modularity and has disc bootstrap.

ACORN 6809

- 1K RAM
- Direct printer drive
- ASCII keyboard input
- Fully buffered address and data bus
- 2K operating system with
  - Printer routine
  - VDU routine
  - Cassette load and save
  - Disc bootstrap
  - Trace
  - All usual debug facilities

VDU
- Seven colours
- Teletext compatible
- Upper and lower ASCII and graphics
- Memory mapped
- Transparent access
- Programmable screen format
- RGB & Sync output
- Light pen facility
- Single 5v rail

- Fastest execution time of any 8-bit processor
- More addressing modes than any other processor
- Position independent code
- Re-entrant programming
- High level language efficiency
- Indirect, indexed and auto increment/decrement
- Four 16-bit registers
- Super powerful instruction set including multiply
- 6800 software compatibility

All Acorn modules are covered by a money-back guarantee and a fast repair/advice service, kits are supplied with full assembly instructions and operating details including where necessary sample programs.

ACORN COMPUTER

Acorn Computer,
4a Market Hill, Cambridge CB2 3NJ

Please send me the following

qty ☐ 6809/VDU kit Special offer @ £160 + £24 VAT £184
☐ 6809 CPU kit @ £95 + £14.25 VAT £109.25
☐ VDU Interface kit @ £88 + £13.20 VAT £101.20
☐ Details on all Acorn cards

I encose Cheque/PO to value . . . .

Name

Address

Acorn Computer 4a Market Hill Cambridge CB2 3NJ VAT No. 215 4002 20

• Circle No. 287

COMPUKIT UK101

EUROPE'S FASTEST SELLING ONE BOARD COMPUTER

★ 6502 based system — best value for money on the market. ★ Powerful 8K Basic — Fastest around ★ Full Qwerty Keyboard ★ 4K RAM Expandable to 8K on board. ★ Power supply and RF Modulator on board. ★ No Extras needed — Plug-in and go. ★ Kansas City Tape Interface on board. ★ Free Sampler Tape including powerful Dissassembler and Monitor with each Kit. ★ If you want to learn about Micros, but didn't know which machine to buy then **this is the machine for you.**

Build, Understand and Program your own Computer for only a small outlay.

KIT ONLY **£199** + VAT NO EXTRAS NEEDED

AVAILABLE READY ASSEMBLED & TESTED READY TO GO FOR **£249** + VAT

Specially designed case for Compukit in orange/black. With room for accessories. **£29.50** + VAT

6502 Assembler/Editor for Compukit £14·90 + VAT

The **Compukit UK101** comes in kit form with all the parts necessary to be up and working, supplied. No extras are needed. Ater plugging in just press the reset keys and the whole world of computing is at your fingertips. Should you wish to work in the machine code of the 6502 then just press the M key and the machine will be ready to execute your commands and programs. By pressing the C key the world of Basic is open to you.

This machine is ideal to the computing student or Maths student, ideal to teach your children arithmetic, and is also great fun to use.

Because of the enormous volume of users of this kit we are able to offer a **new reduced price** of £199 + VAT

NEW MONITOR FOR COMPUKIT UK101 ● In 2K Eprom 2716 ● Allows screen editing ● Saves data on tape ● Flashing cursor ● Text scrolls down £22.00 + VAT

THE NEW TRS80 SURPRISE — MODEL II

Fast and expandable! Model II operates at twice TRS-80's high speed.

In addition to either 32 or 64 thousand characters (bytes) of Internal Random Access Memory, one built-in 8" floppy disk stores an additional one-half million bytes, including the Disk Operating System. And you can easily expand up to a four-disk system for up to two-million bytes of storage.

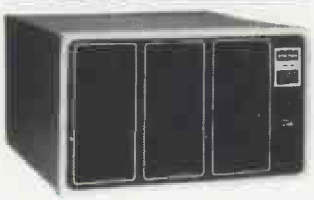
Model II features upper and *lower* case letters. Its built-in 12" high-resolution video monitor displays 24 lines of 80 normal characters. The profesional 76-key keyboard (with "calculator" keypad) includes advanced functions such as Control, Escape, Caps, Hold, Repeat. The keyboard is detachable and moveable for convenient data entry.

You get the enhanced Level III version of TRS-80's already-famous Level II BASIC language and "TRSDOS" operating system, automatically loaded in memory when you "power up." (About 24K of RAM is used by this software.)

Each time you power up, Model II thoroughly tests itself to insure proper operation. Your chosen program can appear immediately, without any intermediate steps or questions to answer.

Versatility ... plug-in expandability
Built-in input/output capabilities include two RS-232C channels, and one Centronics parallel port. Future expansion is provided for through four plug-in slots for optional PC boards.

64K 1-Disk Model II **£2250.00** + VAT



1 DISK EXPANSION Room for 3

Just Plug In

WE ARE NOW EX-STOCK

500K per Drive gives total of 1.5M Byte — 1 Drive plus Cabinet **£799** + VAT

EXTENDED WARRANTY BY COMPUCARE

Why do people buy more from COMPSHOP than anywhere else?

★ **LARGER STOCKS** — we hardly ever run out
★ **GOOD SERVICE** — we give extended warranties on all our products.
★ **EXCELLENT REPAIR SERVICE** — Through Compucare we repair and maintain most makes of personal computers.

LAST YEAR WE SUPPLIED TO THE PUBLIC — LARGE & SMALL
16,000 Television Games & 7,000 Computer Systems

NEW TV GAME BREAK OUT

as got to be one of the world's greatest TV games. You really it hooked. As featured in **ETI**. Has also 4 other pinball games id lots of options. Good kit for up-grading old amusement imes.
IINI KIT — PCB, sound & vision modulator, memory chip ind de-code chip. Very simple to construct. £14.90 + VAT
OR PCB £2.90 MAIN LSI £8.50 Both plus VAT

THE ATARI VIDEO COMPUTER SYSTEM £99.00 + VAT

Atari's Video Computer System now offers more than 1300 different game variations and options in twenty great Game Program™ cartridges!

Have fun while you sharpen your mental and physical coordination. You can play rousing, challenging, sophisticated video games, the games that made Atari famous.
You'll have thrill after thrill, whether you're in the thick of a dogfight, screeching around a racetrack, or dodging asteroids in an alien galaxy. With crisp bright colour (on colour TV) and incredible, true-to-life sound effects. With special circuits to protect your TV.

Cartridges now available All at £13.90 each + VAT
Basic Maths, Airsea Battle, Black Jack, Breakout, Surround, Spacewar, Video Olympics, Outlaw, Basketball, Hunt & Score*, Space War, Sky Diver, Air Sea Battle Codebreaker*, Miniature Golf.

Extra Paddle Controllers — £14.90 + VAT
*Keyboard Controllers — £16.90 + VAT

COMP SHOP LTD
"Europes Largest Discount Personal Computer Store"

Please add VAT to all prices — Delivery at cost, will be advised at time of purchase. Please make cheques and postal orders payable to COMPSHOP LTD., or phone your order quoting BARCLAYCARD, ACCESS, DINERS CLUB or AMERICAN EXPRESS number.
CREDIT FACILITIES ARRANGED — send S.A.E. for application form.
14 Station Road, New Barnet, Hertfordshire, EN5 1QW Telex: 298755 TELCOM G
Telephone: 01-441 2922 (Sales) 01-449 6596
OPEN · 10 am · 7 pm — Monday to Saturday
Close to New Barnet BR Station — Moorgate Line.
NOW in IRELAND at: 80 Marlborough St., Dublin 1. Tel: Dublin 749933

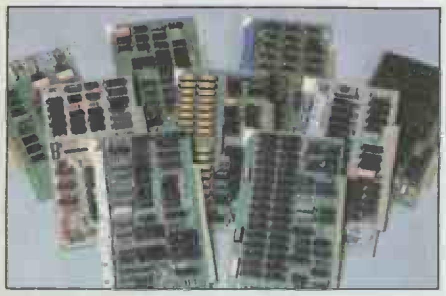
COMP COMPUTER COMPONENTS (Part of the Compshop Ltd. Group)

DINERS CLUB INTERNATIONAL
BARCLAYCARD VISA

At Intersystems, "dump" is an instruction.

Not a way of life.

(Or, when you're ready for IEEE S-100, will your computer be ready for you?)



We're about to be gadflies again.

While everyone's been busy trying to convince you that large buses housed in strong metal boxes will guarantee versatility and ward off obsolescence, we've been busy with something better. Solving the *real* problem with the first line of computer products *built from the ground up to conform to the new IEEE S-100 Bus Standard.* Offering you extra versatility in 8-bit applications today. And a full 16 bits tomorrow.

We call our new line Series II.™ And even if you don't need the full 24-bit address for up to 16 megabytes (!) of memory right now, they're something to think about. Because of all the performance, flexibility and economy they offer. Whether you're looking at a new mainframe, expanding your present one or upgrading your system with an eye to the future. (Series II boards are compatible with most existing S-100 systems and *all* IEEE S-100 Standard cards as other manufacturers get around to building them.)

Consider some of the features: Reliable operation to 4MHz and beyond. Full compatibility with 8- and 16-bit CPUs, peripherals and other devices. *Eight* levels of prioritized interrupts. Up to 16 individually-addressable DMA devices, with IEEE Standard overlapped operation. User-selectable functions addressed by DIP-switch or jumpers, eliminating soldering. And that's just for openers.

The best part is that all this heady stuff is available *now*! In our advanced processor—a full IEEE Bus Master featuring Memory Map™ addressing to a full megabyte. Our fast, flexible 16K Static RAM and 64K Dynamic RAM boards. An incredibly versatile and economical 2-serial, 4-parallel Multiple I/O board. 8-bit A/D-D/A converter. Our Double-Density High-Speed Disk Controller. And what is undoubtedly the most flexible front panel in the business. Everything you need for a complete IEEE S-100 system. Available separately, or all together in our new DPS-1 Mainframe!

Whatever your needs, why dump your money into obsolete products labelled "IEEE timing compatible" or other words people use to make up for a lack of product. See the future now, at your Intersystems dealer or call/write for our new catalog. We'll tell you all about Series II and the new IEEE S-100 Bus we helped pioneer. Because it doesn't make sense to buy yesterday's products when tomorrow's are already here.

InterSystems™

Ithaca Intersystems,
58 Crouch Hall Road,
London, N8 8HG. U.K.
Telephone: 01-341-2447 / Telex: 299568

• **Circle No. 310**