Australian Personal Computer

WIN AN EPSON PX-8

ISSN 0725—4115 NZ $3.50 AUGUST 1984
REGISTERED BY AUSTRALIA POST PUBLICATIONS VBP 3691

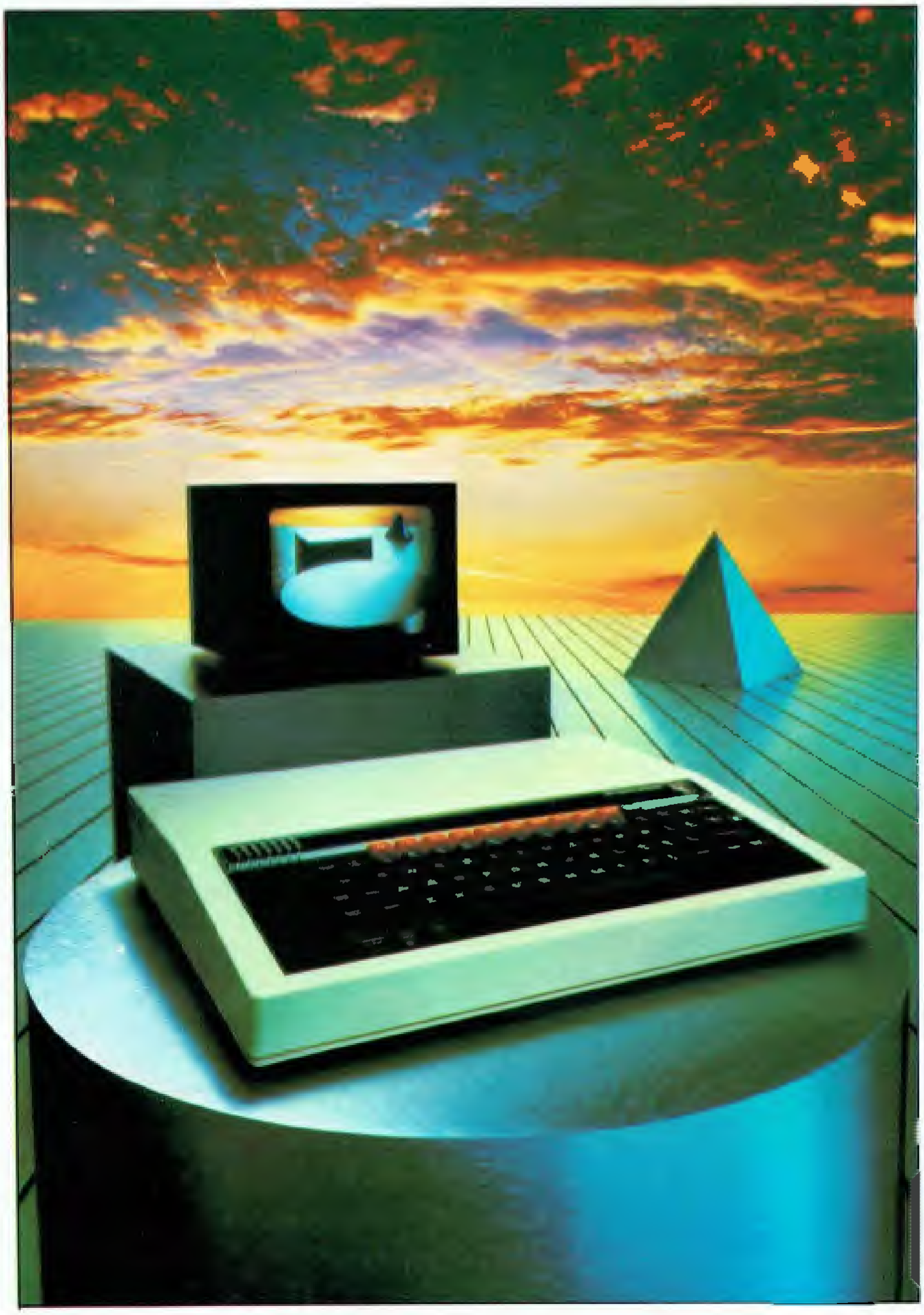
$2.95

AUSTRALIA'S TOP SELLING COMPUTER MAGAZINE

SINCLAIR QL — CHUMP OR CHAMP?

The home computer specialist launches a $1000 business system

BBC Microcomputer
The teaching computer for those who have done their homework



The BBC Microcomputer is the mainstay of the British educational system and will take their youth confidently into the 21st century.

The success of the BBC Computer Literacy Project is spreading rapidly across the world.

In Australia, a very large number of BBC school computer systems have already been installed in every state.

Why? Because 'The BBC' is not just an educational computer. It is one part of the British Government's project to produce the best microcomputer for education, plus the whole range of software and training aids needed to secure for youth the advantages of computer literacy in the coming computer age. Software abounds. The TV 'Computer Programme' has only begun. There is a wide variety of books and teacher aids. And the list grows constantly.

Australia is fortunate to be able to adopt the entire project without change — and to enjoy all the future developments. For the BBC Computer Literacy Project is ongoing. It will still be with us in the 21st century.

Of course, you are probably aware that Barson Computers were selected to distribute the BBC micro in Australia and New Zealand because they have the desired technical expertise, and are capable of giving BBC Microcomputer users a very high level of support indeed.

You see, the BBC did their homework, too.

BARSON
computers

To: Barson Computers Pty Ltd
335 Johnston Street, Abbotsford,
Victoria 3067

Please send me information on the BBC Microcomputer and Software ticked below.

Name ....

Telephone ....

Address ....

.... Postcode ....

☐ BBC Microcomputer
☐ Educational Software
☐ Games Software

1411/APC983

inside

Managing Editor Sean Howard; *Assistant Editor* Maria Bokic; *Consultant Editors* Steve Withers, Ian Davies; *Advertising (NSW and Qld):* Mark Reiss (02) 268 0666; *Advertising (Vic, SA, WA, and Tas):* Philip Treyvaud (03) 531 8611; *Art Director* Mike Northcott; *Production Assistant* Chris Nisbet; *Typesetter* Marcia Brown; *Subscriptions Manager* Valerie Meagher; Printed by Quadricolor Industries Pty Ltd; Subscription rates: Australia $35.00 per annum. Dverseas A$47.00 (surface), A$130 (airmail); Published by Computer Publications Pty Ltd, 77 Glenhuntly Road, Elwood, Victoria 3184. Telephone: (Editorial and Publishing) (03) 531 8411; Tele». AA 30333 AMJ; *Distributors* Network Distribution Co., 54 Park Street, Sydney 2000.

REGULARS

117

Volume 5, Number 8, August 1984

APC

FEATURES AND SERIES

64

BENCHTESTS & REVIEWS

NEWSPRINT

The biggest news of the month is of course the 3rd APC Show. Highlights are presented throughout this month's Newsprint. Also there's news of Adam Osborne's latest explanation for the fall of Osborne Computers; new Microsoft software for the Macintosh and IBM's difficulties in selling two of its major lines of PCs.

Adam Osborne has been telling lies to the press

He reveals this in a book, which tries to analyse why Osborne Computer Corporation collapsed, suddenly and dramatically just at the point where it looked ready to take over the world.

The gist of the book is one which, he is perfectly aware, could take him straight into the libel courts — in fact he starts the book off with a letter from the lawyer of the man named as a possible culprit.

That man is Robert Jaunich — the man Adam himself brought in from Consolidated Foods to become president of his company.

Osborne did not go bust, says Adam, because of pre-announcing a machine called the Executive. That story is one which he told the *Wall Street Journal* and me, and many other journalists, knowing it to be false a year ago.

'It was the only time I have knowingly misled the press,' Osborne says in his book *'Hypergrowth: a study of the rise and fall of Osborne Computer Corporation.*

Osborne's theory is simple, if at first incredible.

He believes that the new president might have wanted to buy a larger part of the corporation's stock than he was entitled to under his contract.

And one way of doing that, says Adam, would be to take decisions which made the company appear to be barely capable of making it through the year.

He could then watch the share price drop through the floor, pick up the shares for a song, and turn the company around. 'He would appear to be the saviour of the company, and nobody, even if suspicious, would have complained,' says Osborne.

And the plan would have worked, had it not been for the unforseen collapse of Atari and Texas Instruments just around the time when Jaunich was about to raise $45 million in private capital to finance Osborne's next few months.

With the withdrawal of TI and the announcement of Atari losses, the investment community wouldn't touch microcomputers, says Osborne. And so the whole plot was undone, and the company went into bankruptcy.

Evidence provided in the book for this theory is thick, but all circumstantial.

Osborne is at pains, throughout the text, to show that his examples of 'mismanagement' could be interpreted as sensible reactions to a genuine crisis. It's just that he doesn't believe there was a genuine crisis.

Osborne is entitled to his opinion and just possibly Jaunich might have been tempted to dismiss it as 'sour grapes' — were it not for one interesting factor, in the character of the co-author of the book.

The book was prepared with the help of John Dvorak, who was, at the time it was all going on last year, editor of the West Coast weekly, *Infoworld.*

Dvorak's reputation as a reporter is very high, and is bound to make many readers take the book more seriously than they would otherwise. He has no obvious axe to grind, and the fact that he has apparently taken Adam's theories seriously must inevitably lead others to think longer before dismissing them.

Nonetheless, there will be many who will not believe a word of it.

One of Osborne's closest colleagues inside the company confided: 'It's just classic — the man cannot accept that he could make a mistake, and now admits that he did — but only the mistake of believing that he was mistaken. He thinks that if he had stayed on as president, all might have been well.

'And, historically, that wasn't an option open to him, even back in November '82: the weight of executive opinion inside Osborne Computer Corporation was already building up behind an 'Adam must go' campaign,' added the inside man.

Nonetheless, no one who knows US stock market habits and big business practices will deny one thing.

That is: the sort of thing which Adam writes about has happened before, is happening somewhere today, and will happen next year. It isn't just wild fantasy, even if it didn't happen at Osborne Computer Corporation.

Guy Kewney

Micro heavies line up

Jack Tramiel, ex-boss of Commodore, has taken over quite a lot of Atari's computer business.

No one uses the word 'fired' about someone as important as Tramiel — not because they are in any doubt about it, but because even if it is blindingly obvious that he wanted to run a micro company, and wouldn't have left Commodore unless under pressure to do so, the word 'fired' is supposed to be rude.

Another AIDS

A new product for those about to follow Dick Pountain's Teach Yourself LISP series: Microsoft has released a version of the language for MS-DOS and CP/M-80 computers.

The package, called the muLisp-82 Artificial Intelligence Development System (sounds like a set of mental exercises), will work on a range of machines including the Apple and IBM PC.

It can cope, says Microsoft, with the bigger (segmented) memory systems of the 8088 or 8086 family of microchips — something a bit beneath one or two LISP compilers which can restrict memory artificially.

As well as two educational games, the package also includes an implementation of the original Eliza or Doctor program, written by Joseph Weizenbaum of the Massachusetts Institute of Technology — the program which always ducked any attempt to ask it a question by turning the question into another. ('Who are you?' — 'Why do you want to know who I am?').

If you can't find a store selling it, contact Microsoft on (02) 450 2522.

I won't use it, either, because although it is *the* word most Commodore insiders use, there is still the possibility that this very ingenious man has bought Atari in order to bring it into the Commodore empire.

Tramiel is, of course, wealthy. He also has access to powerful backers, and the news that he was able to pay $240 million for a complicated package of shares and debt shouldn't be a shock.

But it was — especially to Philips, where talks were still under way preparing for a takeover on the day the Tramiel deal was announced.

The future (in terms of hardware is still unpredictable. In human terms, however, it is becoming clear that many of Atari's top executives are leaving, and that friends of Tramiel are moving in.

Famous for the 'revolving door management style' of quick hire and fire, Tramiel has never been one to respect people who stood up to him and told him he was wrong.

But when it comes to the high technology designers, his record is very different — and many of the brightest chip technologists and systems designers from Commodore and MOS Technology are expected to follow him to Atari.

Guy Kewney

More for dBase

Ashton-Tate products in the dBase II family can now down-load mainframe database information in a format that they understand, with a product developed jointly with Informatics, a mainframe software company.

The new product is called dBase/Answer. The move should give a lot more executives access to mainframe information.

It always sounds a lot cleverer than it is, to offer 'full IBM-compatible mainframe communications' on a micro.

What it does is to let you access the mainframe as if you had a terminal, and get information out of it as if you were an authorised user. Fair enough, except that the cost of the terminal isn't usually the problem — it's the cost of having yet another user (inexpert, occasional, and likely to cause problems) on the mainframe, and the cost of keeping data there, that makes the DP manager postpone the decision.

dBase/Answer deals with the complete task of taking a mainframe file, and turning it into a dBase file. Once that's been done, the micro user can switch to dBase II or dBase III, or Friday! or the latest product, Framework, and create reports, print sorted lists, or whatever.

Guy Kewney

dBase III

The main difference between the old dBase II database, which sold so well, and the new dBase III, which Ashton Tate has just launched, is that the new one assumes you are using a 16-bit processor — or larger.

The main difference between eight-bit systems (generally) and 16-bit systems, is the amount of memory available, and the size of the file descriptors.

So dBase III can store up to two billion records per file, and 128 fields per database.

From the database user's point of view, the biggest change will be the ability to use 10 database files simultaneously — where dBase II could only manage two.

And it also uses colour.

The only doubts about the new product concern its newness. According to one consultant, the new version is nowhere near as stable as dBase II (which is no surprise, of course) — but also, not as stable as it might be at launch.

That's the sort of information which isn't very useful except as a vague warning, since the only way to test a database for stability is to use it heavily for six months. Typically, databases seldom break down when trying to manipulate one or two small files, even if they are known to be flaky.

On the other hand, it's worth passing on somebody's doubts just in case you were planning to use it for something critical — so that you can take the precaution of keeping your old system going at the same time, until the new one is proved stable. And I suppose you could describe the company, Ashton Tate, as having proved that it will stay around and support the product.

Guy Kewney

Mac software at APC Show

Microsoft showed File, Word and Chart, a new software program for the Apple Macintosh, as well as demonstrating the recently announced Basic and Multiplan programs at the 3rd*APC* Show.

File, an electronic filing system, lets the user record and organise any kind of details, retrieve specific information and then present it in any specific order.

Microsoft Chart lets the user graphically explain ideas by choosing from over 40 different chart formats. The choice can be made by

The picture of a printed circuit board on the case of the new 'computer tape' from Agfa is meant to reassure you that the contents are 'high tech'.

Apart from that, feel free to ignore all implications in Agfa's marketing sales pitch that this tape is somehow 'better' than another tape cassette — the requirements of hi fidelity audio are no less exacting than those of computer users.

If you use back-of-truck tape, you may have trouble. Otherwise, tape which can record sounds will record computer sounds, and any problems you have are more likely to be the result of the cassette recorder than the tape.

But at least Agfa has produced this in 15 minute chunks, seven and a half minutes per side, which is handy for most home computer programs.


MicroPro

the user actually looking at a gallery of chart samples.

In addition to the 40 standard charts, the user can design their own and change or edit individual chart components to create a new design.

Microsoft's Word, the word processing package, will soon be available on the Macintosh. Features include the 'Undo' capability, which lets users reverse their latest changes; a special 'Help' reference guide; underlining italics and all general word processing functions.

Microsoft also demonstrated their recently announced Project program. Microsoft Project allows managers to use their personal computers to create schedules, allocate resources and budget costs for their projects.

Details on (02) 450 2522.

More LAN

10-NET, from Fox Research Inc, is a local area network designed to enable the interconnection of IBM PCs to PC look-alikes via a twisted pair wire.

Speed of data transfer is 1 million bits per second up to 3000 feet, after which a repeater is required. Some features of 10-NET include implicit and explicit file and record locking, printer spooling, electronic mail, remote job submission, news broadcast facility and network management utilities.

10-NET for the IBM PC, PC XT and compatibles runs on PC-DOS 2.0, 128k RAM, one or more disk drives, a cursor addressable monitor and twisted pair wire.

The price is $995 plus tax from SCA. Further details on (03) 347 7011.

And yet another Commodore

The new 8296-D microcomputer is specifically designed for business use and features dual 5¼" disk drives built into the Porsche-designed cabinet containing the CPU.

It is based on the 8096 microcomputer, which was released late last year, and has 128k of memory on an 80-column display.

The dual disk drive has a capacity of 1.06 Mb per drive and a maximum sequential file of 1.05 Mb. It takes 5 milliseconds for the disk drive to access from track to track. Up to 224 files can be stored on each of the disks. The disk drive units can also support relative record files and can copy all the files from one disk to another without copying unused space.

Details on (02) 427 4888.

The fully orchestrated Symphony

Madeleine Long, product manager for the new Lotus 1-2-3 trade-up Symphony has just returned from a Lotus New York dealer training course. She will control the introduction of Symphony to the Australian market through Imagineering's dealer training course. "Lotus is already shipping 5 to 10,000 a day in the US", she reports.

Australia will get priority, she predicts. "Europe is screaming for Symphony, but Australia will get first delivery."

The New York training sessions were fully booked. Demos were on Compaq computers. As the Compaqs were not equipped with printers, demonstrations were not given on printing out a combination of word, numbers and pictures.

Because of an agreement with Lotus US, Imagineering couldn't release a review disk to *APC*. A demonstration in Imagineering's Ultimo, Sydney offices showed that Symphony has a lot of power, but is not so easy to use. But it is easier to use than Lotus 1-2-3. Commands are easier and the tree structures are different from Lotus 1-2-3. "It looks more alien than I expected", said one gung-ho 1-2-3 enthusiast. This keen user (who prefers to remain unnamed) was surprised that a demonstration could not be given of the power to print out numbers, graphs and words on the same sheet of paper. "I'm sure it can be done", determined Long, who promised to work late into the night working out just how.

Apart from this yet-to-be-proven integration, Symphony does have some other features worth the $900 plus retail price. The database has a very easy-to-work forms generator, which you can build calculations into like the 1-2-3 spreadsheet. The word processor and mailmerge is worth the price alone, having at first look the power of 1-2-3 and Multimate combined.

The graphs have the added feature of hi-lo-close. This is not included in 1-2-3.

For two months from the date of its introduction at the Imagineering Opera House

Commodore's new range of business machines. From left the 8296-D computer, 6400 printer, 8250 slimline disk drive and 8296 computer.

M1009. Big on features. Small on cost.

Here is a new compact printer with features you would only expect in more expensive machines. Like a 9 pin dot matrix printing head with a 20 million-stroke service life, crisp 50cps bidirectional logic seeking printing for normal characters, unidirectional printing for super and subscripts and graphs, plus low noise operation.

The M1009 incorporates 96 ASC11 type characters with 63 graphic sets and international characters and Centronics parallel interface. Printout is on cut-sheet paper or, optionally, fanfold and roll paper.

All this for only $299!

HR-15. Letter quality printouts for under $799

A compact, letter quality printer, the HR-15 offers features which give it a quality and versatility unmatched for the price; interchangeable letter quality daisy wheel; optional electronic keyboard, tractor feeder and auto cut-sheet feeder; proportional spacing; colour print; 3K buffer memory; Centronics parallel or RS232C interface and many more.

HR-5. The world's most compact printer.

This is the smallest, most versatile, fully functional heat transfer thermal printer available. At only $249, it is fully portable (battery or mains operation) and so small that it will easily fit into your briefcase. The HR-5 is the only printer in its class able to offer the choice of thermal or plain paper and its other features include: full graphics function; roll paper feed; Centronics parallel or RS-232C interface; silent operation; variable printout; 30cps printout.

brother®

Affordable Electronic Printers

Brother Industries Ltd., Nagoya, Japan.

Brother Industries (Aust.) Pty. Limited, 7-9 Khartoum Road, North Ryde. N.S.W. 2113.

Sydney (02) 887 4344. Melbourne (03) 873 3655. Brisbane (07) 52 5257. Adelaide (08) 42 6373. Perth (09) 478 1955.

All prices are recommended retail and include sales tax. They may be slightly higher in W.A.

CSO.4304

TEAR OUT THIS PAGE NOW AND YOU'LL BE IN A MUCH BETTER POSITION TO SELL YOUR PRODUCT NEXT MARCH.

Yes, I realise that space is at a premium, but I would like to be in the best possible position to claim a good slice of the projected $200,000,000 Sydney microcomputer market in 1985.

Please send me by return mail everything I need to know about The Fourth Australian Personal Computer Show – including a detailed floorplan of the Centrepoint, Sydney, display area. I also understand that my name on this coupon in no way represents a commitment of any kind.

To: Australian Exhibition Services, Suite 3.2 Illoura Plaza, 424 St Kilda Road, Melbourne, Victoria 3004. Telex AA39329. Telephone (003) 267 4500

Please print in block capitals:

Company ______________________________

Address ______________________________

______________________ Postcode ____________

Contact ______________________ Title ____________

Telephone ______________________ Telex ____________

General description of products to be exhibited ______________________________

THE FOURTH AUSTRALIAN

PERSONAL COMPUTER SHOW

CENTREPOINT

PC 85

SYDNEY 13-16 MARCH 1985

Symphony launch on July 26, Lotus 1-2-3 users can trade up to Symphony for the cost of $300.

Mini Conqueror hits market

CPM Systems has announced a new small business computer manufactured in Australia. The computer comes with 64k RAM, two 800k (formatted) mini floppy drives, two RS232 serial ports and one parallel (centronics) port.

Every user has a different view as to what constitutes a good terminal, thus, Mini Conqueror is supplied with or without a terminal. It comes with slimline 5¼" double sided, double density drives only. Additional drives may be purchased later. Minimum drive capacity is 780k and can be increased to a maximum of 3.2 Mb.

There are two serial ports, one for the terminal and one for the printer or modem. A parallel port is standard with the computer. Baud rates are changed through software selection.

The entire computer is on a single card which is removable when four screws are removed. CPM Systems promises next day turn around on repairs.

Details on (03) 267 7829.

New printers from Case

Three new printers, a full-width daisywheel called Powertype and two 'near letter quality' dot matrix machines named the Radix-10 and Radix-15, have been released by Case Communication Systems Ltd.

The Powertype is priced at $775 and operates at 18 cps, while the Radix printers, depending on the carriage width required, cost $1280 and $1560 respectively and operate at 200 cps.

Both dot matrix machines have a three-way paper handling system, being able to use roll paper like a Telex roll in addition to continuous paper and semi-automatic single sheet insertion.

Case, an Australian data communications company, now markets seven printers in the Star range made by Star Manufacturing Ltd at Shizuoka in Japan.

For further information call (02) 452 5231.

"But the IBM keyboard already has a numeric keypad," you cry. "Why would anybody want to spend around $230 on an add-on?"

The answer, according to Touchstone Technology Inc, is that "we know from discussions with PC owners that no one uses the numeric keypad on the standard PC keyboard." That's because the keypad is actually a cursor controller, and you have to press the Num Lock key to get the numbers. Details of the 29-key pad on (716) 235 8358 in Rochester, New York.

Guy Kewney

Modem technology for schools

The UDM-1200 is a direct attach modem. It interfaces to the DTE (the terminal or computer) via a standard V24 (RS-232C) 25 pin interface and to the Public Switched Telephone Network via an approved Telecom plug. Additional control lines are available to the DTE for auto dial and other functions.

The modem has automatic call (ie, dial), answer and disconnect features.

Speeds available are 300 bps full duplex. 1200 bps half duplex and 1200/75 bps full duplex (the videotex standard). Most data communications in Australia through the Switched Telephone Network use one of these speeds.

The UDM-1200 conforms to the Australian industry standards for data communications. This means you can communicate with and interface to most systems in Australia. The UDM-1200 can also be switched to Bell, US equivalent standards, so that it is possible to dial the US direct.

Details on (02) 634 3988.

You have been warned

Jodee Rich, the high-powered entreprenurial golden-boy of Australian software marketing is widely rumoured to have $150,000 up his sleeve to make an example out of someone, anyone, caught illegally copying software.

People who should know better, to wit, certain highly-placed academics at a Law Faculty we shall not name, were recently observed to have, let us say, 10 personal computers, one purchased copy of software, and 9 illegal copies, with photocopied manuals to match.

Take care, all ye who copy software!

Panthers also get irascible

Tarbu, the 3½ year old black-as-night Indonesian panther, wheeled out by Interface International at the 3rd *APC* Show took a bite at the leg of an unwary customer on a stand at the Show. Press representatives from *APC* didn't stick around, but left with their legs still intact.

12kg battery-powered teleprinter

Texas Instruments introduced, at the 3rd *APC* Show, a new silent teleprinter with an in-built modem. If there were a telephone booth at the top of Ayers Rock you could use it there and still hear the sun set. But if your remote locations make a more prosaic list then the two portable models may still suit your needs.

The new models 703 and 709 have dimensions of less than 11 x 8 x 3 inches and weigh less than 6 pounds. Add an optional battery pack and they become as easy to use in remote locations as they are to carry there. All that's needed is a telephone.

All models feature the same keyboard and printer features.

Printing speed is 45 cps, providing 30 cps overall. Paper width is 1.5 inches in 100 foot rolls, accommodating 80 or 132 columns of print.

The replaceable printhead features a full upper and lower case ASCII character set, including true underlining and descenders. The full size keyboard has standard key layout, adding colour-coded control keys, travel keys for tactile feedback and LED indicators.

For further information call (03) 267 4677

Electric Desk: A more integrated software

Electric Desk combines four business functions: word processing, database management, spreadsheet analysis and communications. Electric Desk features a window function, context-sensitive help screens and a macro-programming language. It requires an IBM Personal Computer (or compatible) with 256k memory and DOS 2.0.

An Australian designed modem would shortly be released by SCA, designed for general use with the IBM Personal Computer, but also

Peacock's what-if scenario

Andrew Peacock, Federal leader of the Opposition put it this way: ". . . if widespread computer literacy is absolutely essential to Australians and to the future of young Australians, it is my pleasure to declare the 3rd *Australian Personal Computer* Industry open" (sic).

In his speech before officially opening the Show, Peacock presented his anti-protectionist views on the Australian computer industry.

". . . the Australian market for personal computers will top $300 million this year," he said.

"The reality underlying this projection is that there is to be continued growth over the next few years," he said. "The computer is the symbol of what we term the new industrial revolution. A revolution Australia has to be a part of."

"I'm not one who believes this particular industry requires massive direct government assistance. What it certainly requires is assistance from the government, whether it be for venture capital, development, etc."

"People who advocate the widespread use of protectionist devices are ignoring the underlying reality that the market is the only real determinant of whether an industry will grow. Government is to provide tax deductability for investment in high tech and meeting the needs of product development."

Faster than who?

Olivetti PC offers faster processing, expandability, an alphanumeric colour and "shades of grey" controller plus serial and parallel interfaces as standard.

Operating systems shown at the 3rd *APC* Show include MS-DOS and Concurrent CP/M-86, each with a range of programming languages and tools. Compatibility is also guaranteed with the PCOS environment of the M20, which means the M24 has access to over 2,000 packages immediately.

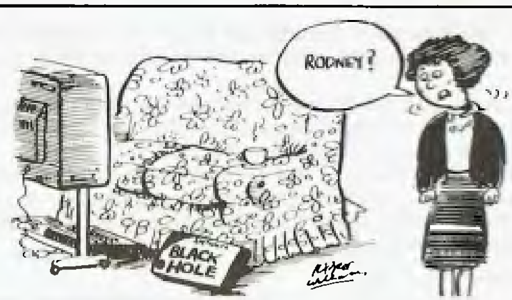
The M24 can be integrated into office automation and distributed data processing environments as an intelligent work station. It can also be connected on-line to remote computers and may be integrated into 3270 terminal networks.

The M24 PC consists of four basic modules — a processing unit, keyboard, display and hard disk expansion unit. The processing unit measures 380mm wide by 160mm high by 370mm deep. It is based on a high speed (8MHz) 8086 CPU and is set up to accept the 8087 as a numeric co-processor. The basic configuration comprises 128k RAM, which can be expanded to 640k. The unit can house either one or two floppy disk drives of either 360k or 720k each. Alternatively, one floppy can be replaced with a slim 10 Mb winchester hard disk.

Two detachable keyboard versions are available. One has industry standard layout, the other an extended 102-key layout with 18 function keys. Both keyboards are available in all major national layouts and include an attachment for an optional mouse.

Details on (02) 358 2655.

The new Olivetti M24 PC



PC 84 The 3rd Australian Personal Computer Show World Trade Centre Melbourne 18-21 July 1984

SHOW REVIEW

"Melbourne was a gamble," said Lee McLean, editor of the APC Show daily newspaper, "and it paid off. Exhibitors tell me the mood of the Show was of quality, and of good hard leads, not just kids," she reports. "As a venue, on the first days, people on Levels 1 and 2 were complaining that they weren't getting the customers. By the end of the Show, exhibitors were grateful that the classrooms of kids couldn't get up in the lifts," she laughs. "Saturday was just bedlam, but traffic flow was not a problem. The Show is definitely on for next year in Melbourne," she says. "Everyone wants to book in Sydney as well, but there won't be enough space in Centrepoint for everybody." Microsoft hosted a Show party at the Melbourne Cricket Club. Here the computer industry let its hair down, to the extent that normally tense IBM people were observed to wear Apple badges.

Melbourne APC Show tops Sydney with 27,000 visitors

Claude Riverso, of Tandy, with the Tandy 100 portable, which he reports is selling well. "Journalists are among the first to buy," he says, "Channel 9 and 7 both now have 100s." The Tandy 100 has been on the market for about eight months. It is still awaiting Telecom approval for its inbuilt CCITT and Bell standard modem. The Tandy 100 is made in Japan by the same factory which makes the NEC portable. "The difference between the two," reports Riverso, "is that the NEC is cheaper, but the Tandy 100 has the benefit of full MS Basic. The NEC has a Japanese version of Basic and is not fully compatible with many programs written in Basic," he claims.

Russell Grimmer, has just arrived from Fiji to take over the Melbourne financial desk for Australian Associated Press (AAP). From the Melbourne APC Show, he filed a news story on computer crime to the Sydney office on an NEC portable from the NEC Show stand.

®C. ITOH
®NATIONAL
®AMUST
®MEDFLY
®DISCOVERY

COMPAK
Compak Computer Centre

SPECIALS FROM COMPAK

For your Apple . . . . .the Software Automatic Mouth

A brilliant voice synthesiser that can interpret intelligible speech from your written text. It even knows the dif ference between words like "through" and "rough". Phonemes too if you want to use them. Complete with manual and disk and hardware . . .

only **$35**
including tax

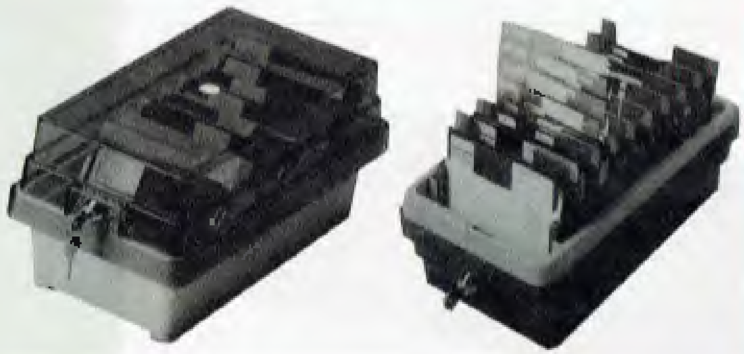
Lockable disk boxes
Concertina display 45 Disks

Plain separators 85 Disks

The same boxes are sold for twice the price elsewhere!

Either Style

From Compak only **$30**
Postage to capital cities **$5**



EXECUTIVE 816

8 bit Menu driven CP/M computer
Comes with software
Brilliantly packaged for the professional and businessman.
Accounting/Wordprocessing/Communications/Spreadsheets/Databases and lots more.
The Crispest style for the businessman on the move.
1.5 megabytes from 2 5¼" drives **$3,550** inc

REMEMBER — COMPAK GIVES TOP VALUE — WE WON'T BE BEATEN ON PRICE

Phone for extra special prices on any of these portables and the C. ITOH 1550 Printer.

350 South Road, Moorabbin,
Vic 3189. Tel: (03) 555 9844

NEWSPRINT

The battle of the showbags; who gave away the most? Barson Computers or Microsoft? Rob Doering and Kathryn Davis of Microsoft spent most of the Show giving demos of Microsoft's new business graphic software package, Chart and Flight Simulator, the world's top-selling software product.

Commodore released a couple of new machines at the 3rd APC Show in Melbourne this July. The Commodore 16 is described as "the brother of the VIC 20" and will retail for around $199. True to form, Commodore has produced yet another version of Basic (version 3.5) for the '16 which means it "is not fully compatible with the VIC 20" which in turn is of course not compatible with the Commodore 64. The C16 has 12k of RAM available for Basic programs and is essentially similar to the existing two Commodore home computers. Commodore assures us the VIC 20 won't be discontinued. We doubt it. Why would you have two machines with virtually the same price tag, virtually the same appearance, virtually the same Basic and virtually the same capabilities but with some hardware and software improvements (which make the machines "not fully compatible") in the same marketplace? Watch for an announcement over the next few months to the effect the VIC 20 is getting the chop.

The second release is the more interesting of the two. It's a machine aimed at the small businessman and enthusiast. The Plus/4, as it is named, includes four applications programs in ROM, a word processor, database, spreadsheet and graphics package. It has 60k of RAM, is compatible with existing Commodore 64 peripherals and has a surprisingly low price tag of $599.

Andrew Peacock made a lightning visit to the Digital Rainbow stand at the Show, reports Keith Reynolds, manager of the Digital Small Systems Group for NSW. "Peacock admitted he didn't know anything about computers," says Reynolds, "and he appreciated the effort DEC had made to make the stand relate to the people." Reynolds commented that you can spot the computer-ignorant. "They stand back, their eyes dart around, looking for something they feel comfortable with. Our non-technical approach has paid off." Digital easily won the prize for the most glamorous stand; towers of glossy black with neon rainbows and rows of life-size perspex people.

NEWSPRINT

The 3rd Australian
Personal Computer Show
World Trade Centre
Melbourne
18-21 July 1984

Framed! Puppeteer, Ron Iliffe, founder of the fast-growing Queensland software distribution company, Arcom Pacific, in a cell, not in his new product, Framework, but in the Old Melbourne Gaol, site of the introduction of dBase III, Framework, and CP/M with DOS to the Australian market at the Melbourne APC Show.

Hidden behind the bars are two desperate types: Bill Boulton, Australia's CP/M expert, and Ron Chernich, both in Melbourne at the APC Show to launch Ashton-Tate's new integrated software package, Framework. The Framework opening was held in the arctic Melbourne environment of the Melbourne Gaol. Here Chernich and Boulton beat the bars with a tin cup, and chant the name of their Imagineering software distribution rival: "Jod-ee Rich, Jod-ee Rich, Jod-ee Rich".

Shirley Elliot, Technical Books Computer Information manager, Melbourne, at their popular Show stand, together with Beth Comber, marketing manager, and Caroline Leslie, advertising manager.

MICROMAIL

Special of the Month:

Tava PC System

System includes: 128k RAM 2 x 320k
Slimline drives, Colour card, 80 columns
1 parallel, 2 serial ports

99.4% IBM compatible
Runs Lotus, Flight Simulator . . .
Motherboard to 256k
One year warranty

All this for only: $3295 plus tax

commodore

Prices include sales tax	RRP	Micromail
Commodore 64	$499.00	$389.00
1541 Disk Drive	499.00	395.00
MSD Disk Drives	Call	Call
Datasette	49.95	43.95
802 Printer	499.00	439.00
1701 Colour monitor	499.00	449.00
SX-64 Portable	1499.00	1375.00
Koala Pad	149.95	139.95
Stack Light Pen	79.95	75.95
Data 20 Parallel interface graphics	119.00	106.95
Advanced Calc Result	200.00	179.00
BC Basic		80.00
Simons Basic	80.00	74.95
Multiplan	149.00	129.00
Intro to Basic 1 2	40.00	32.95
Easy Script	100.00	85.95
The Manager	100.00	85.95
Home Accountant	105.00	89.95
Jumpman	54.95	46.95
Wanda	29.95	25.95
Zork I. II. III	25.95	21.95
Homeword	105.00	89.95
Typing Tutor	19.95	17.95
Bank Street Writer	83.95	75.95
The Hobbit	39.95	35.95
Choplifter	54.95	46.95

IBM

Software includes sales tax	RRP	Micromail
LOTUS 123	950.00	599.00
Sybiz software	Call	Call
Padmede Accounting Modules	390.00	325.00
Spellbinder	795.00	645.00
Symphony	995.00	895.00
Supercalc 3	445.00	369.00
Multiplan	395.00	325.00
Attache software	Call	Call
Microsoft Word	499.00	399.00
Home Accountant Plus	198.00	179.00
Optimum	Call	Call
Rbase	630.00	585.00
Framework	Call	Call
D Base II	825.00	550.00
D Base III	Call	Call
Microsoft Word Mouse	685.00	595.00
Access Manager	450.00	420.00
Visicorp Specials	**Call**	**Call**
Multimate	595.00	495.00
Open Access	940.00	850.00
Wordstar Professional	795.00	695.00
Zork I. II. III	59.95	49.95
More software	**Call**	**Call**
Hardware excludes sales tax		
Microspooler	Call	Call
Quadboards	Call	Call

Six S Accounting	Call	Call
Bank Street Writer	98.95	86.95
Homeword	105.00	89.95
Zardax	324.00	258.00
Bank Street Speller	105.00	95.00
D Base II	825.00	589.00
Home Accountant	105.00	89.95
Ultima III		89.95
Master Type	64.96	54.95
Graphics Magician	84.95	75.95
Wizardry	81.95	69.95
Choplifter	48.95	39.95
Loderunner	48.95	41.95
Flight Simulator II	56.95	48.95
Sargon III	avail.	74.95
Zork I. II. III	60.95	50.95
PFS File. Graph. Report. Write	175.00	149.00
CPM+Wordstar+64K	745.00	545.00
Witness	72.95	65.95
Sandy's w p	225.00	195.00
Crossword Magic	84.95	79.95
Educational software	Call	Call
Mac software	Call	Call

MEMOTECH

Memotech 500		699.00
Memotech 512		859.00
Software	Call	Call

Now available, quality

VIC-20

enhancements

WHY BUY an expensive new computer when your VIC-20 can be economically upgraded with these quality products.

Make your VIC-20 talk.

$66.50

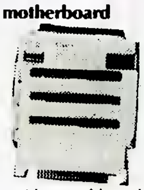
The amazing Adman speech synthesizer. An infinite vocabulary, using the sound amplifier in your TV. Easy to use. Comprehensive instructions included. Major software houses have compatible games.

16K Memory expansion

$79.00

Vixen switchable RAM from Stonechip. Full 16K or 8K or 3K switchable into desired memory block. Use two with motherboard for 28K. Plugs into expansion port. Universal, runs any Basic program without readdressing.

Cartridge motherboard



$65.00

Up to 4 cartridges usable at the same time in this switchable motherboard from Stonechip. Rear positions power and memory location switchable. 4K ROM socket for extra firmware. Housed in protective case.

Stonechip colour lightpen.

$39.95

Draw and paint in colour on the TV screen. Save screen layouts on tape. Superb stability and resolution. Software included.

BASIC programmers aid

18 new commands for the VIC-20, including renumber, autoline, sound, cursor XY, trace, block delete, paper, border, edit, etc. In fast machine code.

ROM (needs motherboard) **$32.00**
Cartridge **$47.00**

Top VIC-20 software.

Write for catalogue of brilliant arcade games. Also test your skill with these excellent challengers from Bug Byte.

VIC-Backgammon (8K) $19.95
Chess (16K) $23.95

Lots more for the VIC-20. Also enhancements for **C64**, Sinclair **Spectrum** and **ZX-81.**
Send self-addressed envelope for catalogue.

Australian distributor:

dolphin computers

99 Reserve Road, ARTARMON, NSW. 2064. Telephone: (02) 438-4933 Telex: AA20677

Enquire at your local computer shop or mail order from Dolphin. Mail orders cert. post $2.00 (software only), o/nite courier $5.00 Sydney. $6.50 Australia, enclose cheque money order/Bankcard details. All items guaranteed 6 months, full instructions included. All prices include Sales Tax.

Dealer enquiries welcome

FOR SALE

BARGAIN PRICE

* * *

One NEC 8201A only 5 months old with all packaging and manuals.

This unit has been used solely for word processing but has recently been made redundant due to an up grade of equipment.

The package for sale consists of one NEC 8201A complete with two 32K RAM cartridges.

This near new model sells for only $1300.

* * *

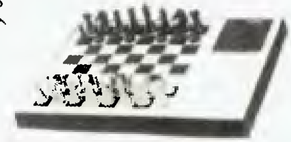
For more information call (03) 523 5947

COMPUTER CHESS from COMPUTER PLAY

EXCLUSIVELY ENDORSED BY THE WORLD CHESS FEDERATION

- VERY STRONG PROGRAMME
- FROM BEGINNER TO CLUB PLAYER
- CAN TEACH YOU TO PLAY CHESS
- FULLY PORTABLE 6 MONTHS BATTERY LIFE OR MAINS ADAPTOR
- EASY TO SET UP, VERIFY AND ADD PIECES

COMPANION 11 SENSOR



- WORLD BEATER 24 LEVELS
- FOR BEGINNER TO CHAMPION
- BATTERY AND MAINS
- MORE POWER AND FEATURES THAN ANY OTHER CHESS COMPUTER

SUPERSTAR 28K SENSOR

AVAILABLE FROM ALL MAJOR DEPARTMENT, ELECTRONIC, GAMES AND VIDEO STORES THROUGHOUT AUSTRALIA.
AUSTRALIAN DISTRIBUTORS: COMPUTERPLAY SCISYS, PO BOX 69, GLEN WAVERLEY 3150.
PHONE: (03) 561 1078. TELEX: AA 30625 ATTN: ME 269.
WA: GAMES WHOLESALE CO. (09) 349 6111
QLD: CHESS SALES QUEENSLAND (07) 52 9633

$500

'INFORMATION BUSINESS MANAGER'

COMPLETE ACCOUNTING PACKAGE FOR ALL 8 AND 16 BIT MICROS

THIS AUSTRALIAN DEVELOPED PACKAGE INCLUDES:

* Accounts Payable
* Accounts Receivable
* Stock Control
* Invoicing
* General Ledger

ALL FULLY INTEGRATED
COMPREHENSIVE MANUAL

DEALER ENQUIRIES WELCOME

ALSO AVAILABLE

*** REAL ESTATE SYSTEM**
*** PAYROLL**
*** INSURANCE BROKER SYSTEM**
*** OTHER CPM SOFTWARE**

UNLIMITED
Computer Solutions
2 PROSPECT HILL ROAD, CAMBERWELL 3124
Telephone: (03) B13 3022

NEWS FROM JAPAN

CP/M-80 and CP/M-86 software to run under MS-DOS . . . a pocket computer for car insurance salesmen . . . the Japanese equivalent of the IBM PC . . . Shinichiro Kakizawa brings you the latest news from the Japanese hot-line.

Shinichiro Kakizawa is a computer technology and applications consultant, and a freelance journalist. He has worked in the computer industry for twenty years, originally on mainframes, and for the last five years on micros. Fujitsu and NEC are among the companies he has worked for in Japan and Singapore. He has been involved in policy setting for the Japanese fifth generation project and he participates in the fifth generation project, Insight.

Multi O/S software

Megasoft is selling a package which allows CP/M-80 and CP/M-86 software to run under MS-DOS. This interesting approach comes in the form of interface software which will run immediately below MS-DOS (versions 1.25 and 2.0). Data can be shared between MS-DOS and CP/M, exchanged, and stored on the same disk. The package is called the EM/3 O/S Integration Adaptor and costs around $300. Manuals are unfortunately only available in Japanese at the moment.

1Mb, 1.6Mb in one floppy drive

There is good news for micro users who have difficulty knowing what to do with ever-increasing piles of diskettes in different capacities and densities for a variety of systems.

The majority of the machines on display in shops in the electronics bazaar, Akihabara, Tokyo are now equipped with a standard 1.6Mb floppy disk drive. Only a year ago, the standard was more like 1Mb. As elsewhere in the world, micro users in Japan are facing difficulties with stacks of incompatible floppy disks.

Matsushita (also known as Panasonic) has come up with a decent solution for this problem. Its new drive, called 'Super mini FDD JU591', can read/write two different capacity disks with a single drive by automatically changing the rotation speed of 5¼in disks between 300rpm (1Mb) and 360rpm (1.6Mb). Diskette type can be identified automatically within one second of insertion. Matsushita's plan is to produce 200,000 units this year at its Hananomaki factory in Northern Japan, and it has been talking to a number of micro assembly makers around the world. The price (note that it's only for OEM sale) is $300.

From black/white to colour in the classroom

Panaboard is a little expensive but a very useful tool for classroom training. What you have written on a black/white board can be hardcopied in colour, transferred to VDU or large video projector, or stored on disks.

Panaboard is from Matsushita, manufacturer of the IBM 5550; price is a little over $15,000. last year Oki developed a black and white hard board copier, and this new machine is certainly a welcome addition to the micro classroom.

Let your watch do the walking

A watch capable of memorising telephone numbers for 10 people is being sold in Japan by Casio. The watch has a one-chip CMOS CPU, stores 10 sets of four alphabetical characters and 10 numeric digits, and provides a calculator function as well. The idea is similar to the Seiko wrist computer, but unlike the Seiko, this watch does not require a separate keyboard for input. How it will sell is yet to be seen, but my feeling is that watch computers have now established themselves firmly in the market.

Many more with a wide variety of features will soon follow, including offerings from Sanyo, Citizen, Ricoh and Seiko.

Calculating the premium

Sharp and Unix Ltd (a Tokyo system house) have developed a special purpose pocket computer for helping car insurance policy salesmen to calculate the premium quickly. The machine is based on the popular CE-1253H with 24k RAM. It has special function keys needed for car insurance, and cannot be used for any other purpose. This trend of developing more dedicated pocket computers suitable for only one task is a forerunner of future pocket expert systems. Sharp expects to sell a lot — over 20,000 — this year.

Fuji diskette drives

Fuji Film Company, best known for films and cameras, has begun shipping 1.6Mb 5.25in diskette drives in limited quantity. IBM has said that it will make 1.6Mb drives its next standard device. Fuji is the fourth Japanese manufacturer to produce a 1.6Mb drive after Y-E data, Matsushita and Hitachi. It seems that anyone in this market who offers a diskette of less than 1.6Mb per sheet will be regarded as a failure.

Industry overview

In the business microcomputer market, the biggest difference between Japan and the rest of the world is probably the non-existance of the IBM PC (I daresay you can buy it somewhere in Tokyo if you really want it). Instead, IBM Japan sells the 5550, something similar to the PC/XT in terms of horse power. The machine has been reasonably successful and large mainframe users are buying it, but unlike the PC, you can't buy the 5550 in the micro shops in Japan.

The major supplier of microcomputers in Japan is, of course, NEC. It has sold over half a million systems, inluding the ever-popular 8-bit PC8000, PC6000 series, and 16-bit PC9000 machines. NEC has now firmly established itself as the leader in Japanese micros, mainly because it had the right machine at the right time when the micro took off in Japan four years ago.

NEC enjoys the same privileged position that the IBM PC has in other markets. Every software house writes packages primarily for NEC.

MSX arrival

There is no clear distinction between machines to be used by home users and business users, unlike Australia.

In Japan, everyone has been using NEC, Sharp, Fujitsu, Hitachi, OKI, Mitsubishi, and Toshiba machines whether it's for home or business.

The only difference is the money spent on peripherals. Home and hobby users spend less, but the CPUs are usually the same. However, this situation is rapidly changing as more and more MSX machines are joining the force. MSX machines are cheap — current offerings are around $230 for an 8-bit 64k average machine. It will be interesting to see if the MSX standard is accepted worldwide.

Among the major suppliers, Sord is one of the first and has a lot of enthusiastic followers mainly among large business users. Sord's PIPS spreadsheet package has been as popular as VisiCalc.

It's rather sad that we don't see many world popular brands like Apple, Tandy, and Commodore. You can get hold of them if you wish but, except at the beginning of the micro fever, they haven't sold well.

Wrong pricing, inadequate advertising budgets and lack of decent support networks were the reasons.

It's very sad indeed that major companies of this stature let themselves down in this way.

YANKEE DOODLES

Apple making its comeback with the IIc, TI waging war against the IBM PC and slow start for the PC Junior. David Ahl Reports.

A bigger byte for Apple

In a full day extravaganza in San Francisco followed by a road trip around the US, Apple Computer introduced the Apple IIc around the theme 'Apple II forever'. In the eyes of many security analysts, Apple has been fading for the past 18 months. Recent events, however, seem to have reversed that opinion.

The Apple II line now consists of the Apple IIe, in appearance identical to the original Apple II, but with a much smaller and more efficient chip set. It has eight expansion slots and a starter kit with 64k, one disk drive and software and sells for under $1000.

The Apple IIc is new from the ground up. It is software compatible with the earlier II family, but packs 128k of memory, a CMOS version of the 6502 CPU, a half-height disk drive, and all interfacing into a briefcase-size package not much bigger than a notebook computer.

It uses only 41 chips in total and can be powered from an external AC transformer, auto battery or portable battery pack. Cables are all labelled with icons and an RF modulator is included. In addition, a set of five instructional disks is included with the $1295 starter package.

The system design, bundled package and colourful packaging suggest that the IIc is aimed at the home market, although Apple's distribution through upper tier retailers means that professionals and businessmen will also be important customers.

On the Macintosh front, many companies have completed testing the machines and the large orders are starting to roll in. Peat, Marwick, Mitchell & Co, one of the big eight accounting firms, recently placed an order for 2000 Macs. Apple has made numerous evaluation sales of 10 to 20 machines, and several Fortune 1000 companies are said to be on the verge of placing orders in the 500 to 1000 unit range.

Apple is backing the introduction of the IIc and Macintosh with enormous advertising campaigns, reportedly $20 million for the IIc and $15 million for the Mac to make sure that they are household names by the end of the summer. Also, Apple is on the verge of patching up its disagreement with the Computerland chain and should be back in within a few months.

Low sales for PC Junior

At the IBM Shareholders' Meeting in late April, chairman John R Opel told shareholders that the PC Junior 'hasn't yet been as successful as I would like it to be'.

His remark indicates that IBM is less than pleased with sales of the IBM Junior, and faced with head-to-head competition from the Apple IIc, IBM appears to be considering some design and pricing changes.

Following the recent introduction of several Japanese supercomputers, several analysts opined that IBM is behind in its plans for introducing its next mainframe generation, the Sierra. Opel countered these notions and said that 'IBM doesn't talk about product plans or experimental work.

Almost concurrently, from its Essex Junction, Vermont research facility, IBM announced that it had produced an operational 150 nanosecond, one M-bit dynamic RAM chip, the first made by an American company. While Hitachi, Fujitsu, and NEC have announced experimental one M-bit chips, 256k is the largest commercially available size. Analysts expressed interest that IBM had bypassed the 256k market and jumped directly to one M-bit.

The chip uses a silicon and aluminium metal oxide semiconductor technology and produces circuit elements as narrow as one micron. Other new processing steps allow adjacent storage nodes to be as close as one micron with the use of 15 nanometer composite dia-electric covering material.

Peachtree gets integrated

Peachtree Software recently announced Decision Manager, a multiple-window integrated software package that includes word processing, spreadsheet, data management, graphics, telecommunications and terminal emulation functions.

Designed primarily for the IBM PC-XT, the package will also run on a dual floppy disk configuration. The package requires 256k and accepts either mouse or keyboard input.

Users can define up to 20 windows, and up to 10 can be displayed simultaneously on the screen. Decision Manager was designed to complement Peachtext 5000 and all files are both data and command-compatible between the two systems. Peachtext 5000 has more comprehensive word processing capabilities than the program included with Decision Manager.

Unlike some other highly integrated packages such as Lotus Development Corporation's Symphony and Ashton-Tate's Framework, Decision Manager is more a collection of stand-alone programs bound together with a windowing umbrella program. However, it is one of the few that offers a micro-mainframe link with its IBM 3270 terminal emulator. According to some analysts, that could be its 'ace in the hole'.

Who dares wins?

In a bold marketing effort, TI 'dares' consumers to compare the TI Professional Computer with the IBM PC. The campaign theme, 'Care to Compare', will appear in a heavy TV magazine, direct mail, and point of purchase advertising programme.

Consumers are invited to visit any of 700 TI dealers, all of whom have a TI Professional Computer set up next to an IBM PC. 'The same demonstration program run on each computer will clearly establish TI's superiority,' TI officials said. TI will give a TI solar-powered calculator to consumers who complete the demonstration.

Although the IBM PC is more expensive than the TI unit, TI officials said that price will not be part of the comparison.

To encourage dealers to set up the test in their stores, TI has even offered to provide an IBM PC-XT for the demonstration if necessary. While the makers of many other PC-compatible computers have taken shots at IBM, none has been so blatant as TI.

Will it work? Stay tuned.

Random bits

Spectravideo has shut down operations in anticipation of a debt restructuring that would give control of the company to its manufacturing arm, Bondwell Holdings of Hong Kong . . . Zilog, Motorola, and National Semiconductor have put Coleco on credit hold and stopped shipping chips to the company . . . Intellivision Inc, a company set up to market a video game system formerly sold by Mattel, is negotiating with Samsung and others to secure a line of TV sets, VCRs and other consumer electronics products to market under the Intellivision label . . . Pioneer unveiled an MSX computer that loads software from laser video disks and superimposes computer and video disks signals for spectacular games effects . . . IBM offers PC/IX, its version of AT&T's Unix operating system for $900 . . . Although the US marketing strategy of most Japanese computer manufacturers has been to offer a standard operating system (CP/M or MS-DOS), the biggest selling Japanese computer in the US is the Epson QX-10 which uses a unique Valdocs operating system designed by Rising Star Industries.

END

How we avoid it

Will the computer you buy be the best next year? Computers become obsolete at a frightening rate. You can wait (and miss out on the benefits) or buy a computer that will soon be a compromise. The Universe Supercomputer avoids this problem because it is continually being upgraded both in its **hardware** and **software** so that all existing owners benefit.

How do we do it? The Universe is based on World Standard S100 plug-in cards. All major innovations in microcomputer technology (such as 16 and 32 bit microprocessors) are available **first** on these cards.

Plug-in improvements As a Universe owner you can upgrade first. Universe owners were able to upgrade to 16 bit microprocessors (for under $1,000) over 18 months before IBM or DEC released their PCs.

By expanding rather than changing computers you keep all your existing software and data. No traumatic changeovers, no waiting, no need to pay for a computer that's bigger than you need.

Compare the specifications of the Universe today:

☐ **Runs both 8 and 16 bit, single and multiuser.** All standard operating systems including CP/M & CP/M 86 with Multitasking, CP/M 8-16 'PLUS' (8 and 16 bit at the same time), MP/M 8-16 'PLUS' and MP/M 86 (Multitasking, Multiuser 16 bit with ultra high speed 10MHz processor).

An almost unlimited software base plus Multitasking up to 25 terminals, if and when you need them.

☐ **Multiple Program Selection (MPS).** Our version of CP/M can swap between any mixture of up to ten separate programs almost instantly. Conventional computers only let you do one task at a time.

☐ Memory may be expanded up to 1 Megabyte and disk storage up to 200 Megabytes. Start small and expand inexpensively as and when you need to.

Impressed? Astounded? Then call your nearest AED sales engineer for the whole story.

AED COMPUTERS

Sydney: 24 Darcy Street Parramatta (02) 689 1744 (02) 636 7677. Telex AA 70664
Melbourne 53 Waverley Road, East Malvern Vic (03) 211 5542. Telex AA 30624
Canberra 217 Northbourne Ave., Canberra ACT (062) 47 3403. Telex AA 62898

UNIVERSE SUPERCOMPUTER

DEALER, CONSULTANT, VERTICAL MARKET ENQUIRIES WELCOME

Glover & Assoc AED/1


TeleVideo

Perfect Link

More executives are finding that they want to link their micros to mainframes to access both company information and popular public information systems. Peter Bright assesses Perfect Software's Perfect Link, one of a new breed of friendly comms programs.

Data processing departments don't like micros. This simple fact sums up a major problem facing medium to large companies: because DP departments don't usually want to get involved with buying micros, it's often left up to individual users to choose which micros to buy. This has resulted in companies ending up with a wide range of incompatible micros sitting on their executive's desks. The problems start when one executive decides that he would like to transfer data from a colleague's micro.

In order to transfer files you need two things — an RS232 serial port with cable, and appropriate software on each machine to control data flow.

Either of these can give you endless trouble. Although in theory the RS232 is a standard interface, in practice manufacturers use all kinds of different sizes and shapes of plugs, and sometimes they even wire the sockets incorrectly.

Also, if you are directly connecting two micros together, you will need to use a 'null modem adaptor'. This reverses pins two and three and makes sure that data transmitted from one machine ends up on the receive line of the other. Failure to do this can lead to hours of frustration trying to find out why the link won't work.

The problem with the software part of the link is that until recently, communications software hasn't been very friendly. Programs like BSTAM or ASYNC are both very popular comms packages, but are very unfriendly.

Perfect Link is designed to work on the IBM PC or IBM lookalikes (I used an Olivetti M24). The M24 has an RS232 port built in, but if you use a PC you'll need an RS232 card.

Setting up

There are two ways of setting up Perfect Link. Most day to day alterations can be done while the program is running, but to use an auto-dial modem you have to 'install' it for Perfect Link using the PLINSTAL program.

This is where one of the program's two flaws comes to light. The first flaw, as mentioned earlier, is that it presumes you own an IBM or lookalike. The second flaw, which seems to go hand in hand with the first, is that it presumes you are American. It will only accept American phone numbers for auto-dial modems.

Non-auto-dial modems are of course usable, and autos can be used in a manual fashion.

In use

At its simplest level, Perfect Link can be used as a normal dumb terminal to link into a remote mainframe or micro. When in this mode, the screen is divided into two sections. The top 24 lines are used to display data going to and from the remote system, and the bottom line is used as a status line.

Working from left to right along the status line we have: the connect-time clock, flags for carrier detect, printer online, interrupt (XON), and host status. In addition, there are displays for filename (if you are saving to disk), system name, terminal emulation type, and the baud rate you are using. There is a great deal of information displayed on one line, but it's easy to understand what's going on.

The connect-time click is a surprisingly useful feature. It's easy to run up hefty phone bills when you're connected to dial-up services — the clock is an obvious reminder of how much time you're using. The clock also gives a good indication of whether the RS232 link is working properly. It seems to take its timing pulses from one of the lines on the RS232 port, so if the clock doesn't work there's something wrong with the link.

User interface

Perfect Link makes heavy use of the function keys and the 'HOME' key. Most selections can be made with a single keystroke. You can hit the 'HOME' key to call up the main selection at any time, and function keys call up more specific sub-menus. All menus are displayed in the middle of the screen. Sub-menus are displayed on top of their parents, which gives a windowing effect.

Communications settings

As you would expect from a comms program, Perfect Link makes it easy to experiment with the comms line settings. You can change the baud rate, word length, parity and stop bits, and decide if you want full or half duplex. You can also toggle XON/OFF and RTS/CTS. These are fairly standard, baud rate is switchable up to 9600 baud. The only omission is that Prestel 1200/75 baud is not supported.

Perfect Link has the ability to emulate various popular terminals. These are all selectable from the communications settings menus and comprise Televideo 920, DEC VT-52, ADM-3a, IBM 3101 and Teleray, in addition to the normal TTY terminal mode. The terminal emulation facility can be very useful when you are connected to a mainframe system which was designed for non TTY-terminals.

I hooked up the machine to a number of local services and the US-based Official Airline Guide, and they all worked well. I also connected it to my own micro (an Apricot) at 9600 baud, and ran some Apricot programs on the Televideo — with expected problems where graphics were involved (the Apricot's screen handling is quite unusual, and unavailable in Link's range of terminal emulators), but otherwise it worked. If you are online to a system and want hard copy of what's going on, F2 toggles the printer on and off.

XModem

As well as providing terminal emulation facilities, Perfect Link also allows you to transfer disk files to and from other micros. Most terminal packages allow ASCII text files to be transferred, but Perfect Link can also copy .COM and .EXE program files.

This is possible because Perfect Link incorporates the XModem protocols. As long as the machine you are talking to supports XModem you can transfer any disk file. Peter Tootill discusses the XModem protocol in more depth on page 77, and gives details about the technicalities.

Transferring files

When two machines are hooked directly together, it's necessary to decide which machine is host and which is the slave. If you are downloading from a remote mainframe via a dial-up line, the mainframe will usually be the host. Perfect Link can work as either host or slave.

You must then decide whether or not to use the XModem protocols. If you are transferring a program you'll need XModem, but if you are transferring text files it isn't necessary. All the XModem protocols are automatically handled by the program. The main difference between XModem and non-XModem transmissions as far as the user is concerned is that the latter are comparatively fast, whereas the former can be very, very slow.

I tried to download a program from one of the XModem-supported TBBS bulletin boards: it all went without a hitch and the program seemed none the worse for its journey. I also uploaded and downloaded text files to and from various dial-up services and these went without a hitch too.

Disk to disk transfer

One of the most interesting aspects of Perfect Link is that it claims to be able to make the IBM PC read and write disks in alien CP/M disk formats. This is potentially very useful: instead of having to link up the different micros and go through all the problems of file transfer, you just copy the file to the correct disk format inside your PC.

This option can be selected from the main menu where it's to be found under utilities. The rest of the utilities menu allows you to perform fairly mundane tasks while deleting a file, listing a directory, or typing a file.

Kaypro users would be familiar with the UNIFORM program for the Kaypro, which enables it to format, read and write a large number of CP/M disk formats. According to the manual Link does not have the ability to format, but reads and writes to seven formats — IBM CP/M single and double sided, Kaypro II, Osborne double-density, Zenith Z37, NEC 8001A, and Epson QX10. I have combined the two programs in use at home, formatting various disks using Uniform on the Kaypro, and writing information to them, then reading them onto IBM disks using Link. This was far faster, and far less hassle, than using a communications link between the two computers.

Defining a system

When using Perfect Link to log on to remote systems, you might find that you are always having to change the communications settings to match the different systems. Perfect Link gets around this by allowing you to set up the default settings for up to ten different systems. Instead of messing around with all the comms settings, you hit a key and they are all automatically selected.

The process for defining a system is very straightforward and again makes heavy use of menus. Once a system has been defined, it can be called from the main menu by hitting F6 followed by the system number.

In the same way that you can define a system, you can also set up the function keys to return strings. This can be useful in a number of situations: for example, you could program SHIFTed F10 to return your user ID instead of having to type it in every time you logged onto a system. You can also embed special control characters into the string: if you were programming a sign on a sequence and wanted to wait for the host computer to respond, you would embed a '?' into the string. Using this facility, you can build some very impressive one-line mini-programs.

Documentation

The documentation is very good. It makes heavy use of cartoons and is presented as a thick typeset paperback book. The first section is a general introduction to dial-up services and Perfect Link. The following sections go through the Perfect Link functions, explaining clearly what they do. The manual's job is made easier because the system is virtually self-explanatory.

A large section of the manual is given over to explaining how to use some of the more popular American information systems, which means that the manual contains all the information needed for the first time user to learn how to use Perfect Link *and* how to use Perfect Link for dial-up services.

The services covered are: Dow Jones, CompuServe, The Source, The Official Airlines Guide, Newsnet and Knowledge Index.

I was impressed with the manual. It includes not only necessary information, but also presents points that are not vital but interesting. As an added bonus, it has a decent index.

Conclusion

It is becoming much more important to be able to link micros to mainframes and other micros. It is also important that the relative software is friendly and easy to use.

Perfect Link succeeds. It is certainly straightforward to use — you don't get bogged down in layers of sub-menus. The terminal emulation facilities could be very useful for linking into the company mainframe as long as Perfect Link supports the right terminal for your company.

Perfect Link does not pretend to turn your IBM into an all-singing all-dancing terminal, so those of you looking for 3270 emulation and the like will have to look elsewhere.

The XModem protocol is the nearest thing there is to a standard micro comms protocol, so its inclusion is welcome. It may not be the most secure protocol around, but it's uncomplicated and easy to use.

Sinclair QL

Sinclair computers have been known in Australia for several years, first with the ZX80, then the '81 and subsequently the Spectrum — a full colour home computer with a mountain of software as a result of its huge success in its country of origin, England. Sir Clive, as the figurehead of Sinclair is now known, has recently dished up a machine for the business computing sector. The Sinclair QL has just been released in the UK and was seen on the Barson Computers stand at the 3rd APC Show. It should be available in Australia in four or five months time with a price tag of around $1,000. Is it worth the wait? Devid Tebbutt got hold of the first working model and reports exclusively for APC.

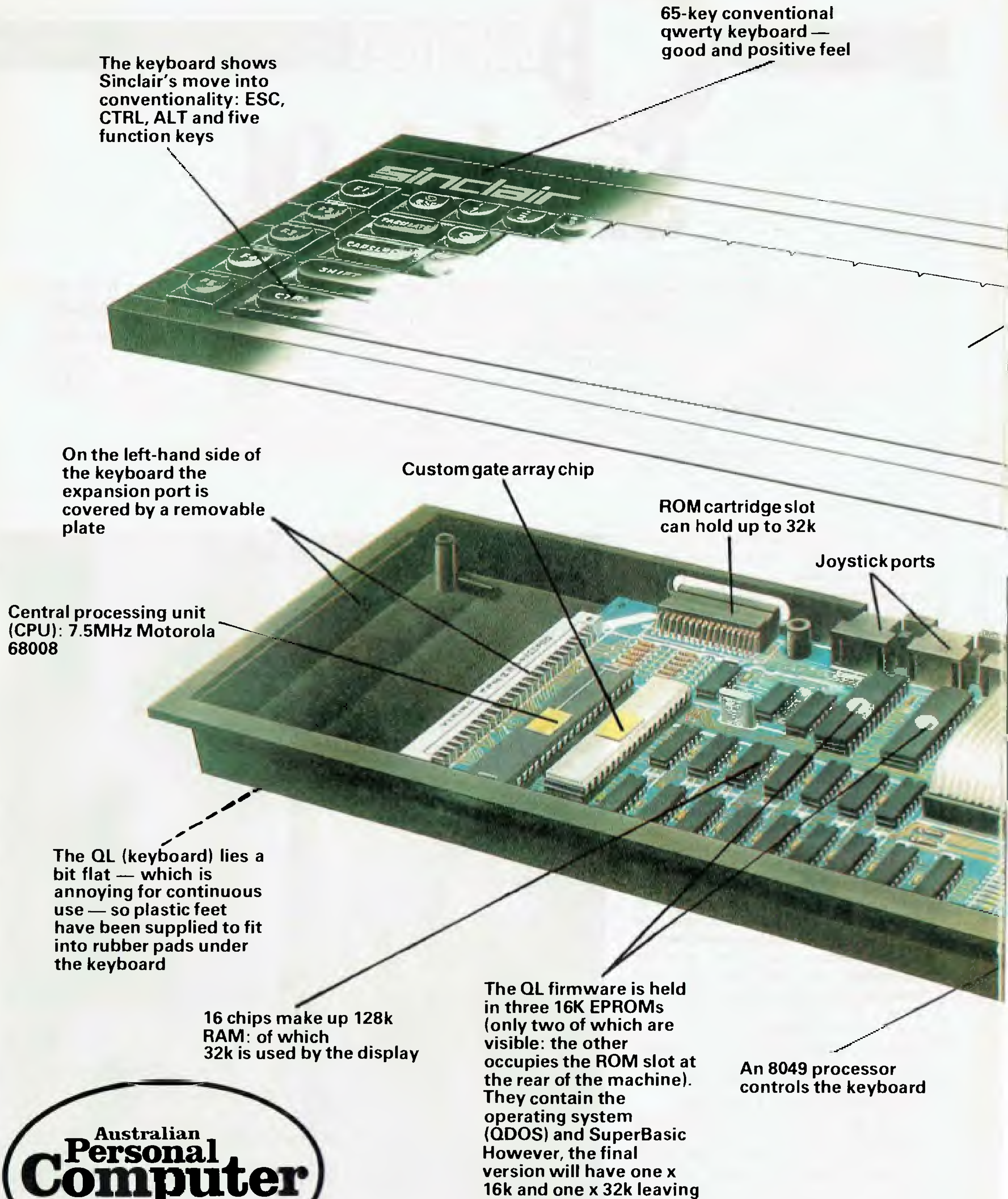
Was I the only person in the world who wondered what all the fuss was about when the QL was announced? People kept on about the Motorola 68000 processor and the 128k of memory, not to mention a souped-up version of Basic. Multi-tasking and windows were thrown at me in a generous attempt to win me round but everything failed. No matter how I tried, I could not get enthusiastic about the QL.

The responses warmed up a little when I read the descriptions of Psion's four programs — Archive, Easel, Quill and Abacus, which compared on paper very favourably with many commercially available packages. But I was still left with doubts; after all, how can anyone do anything serious with a Microdrive?

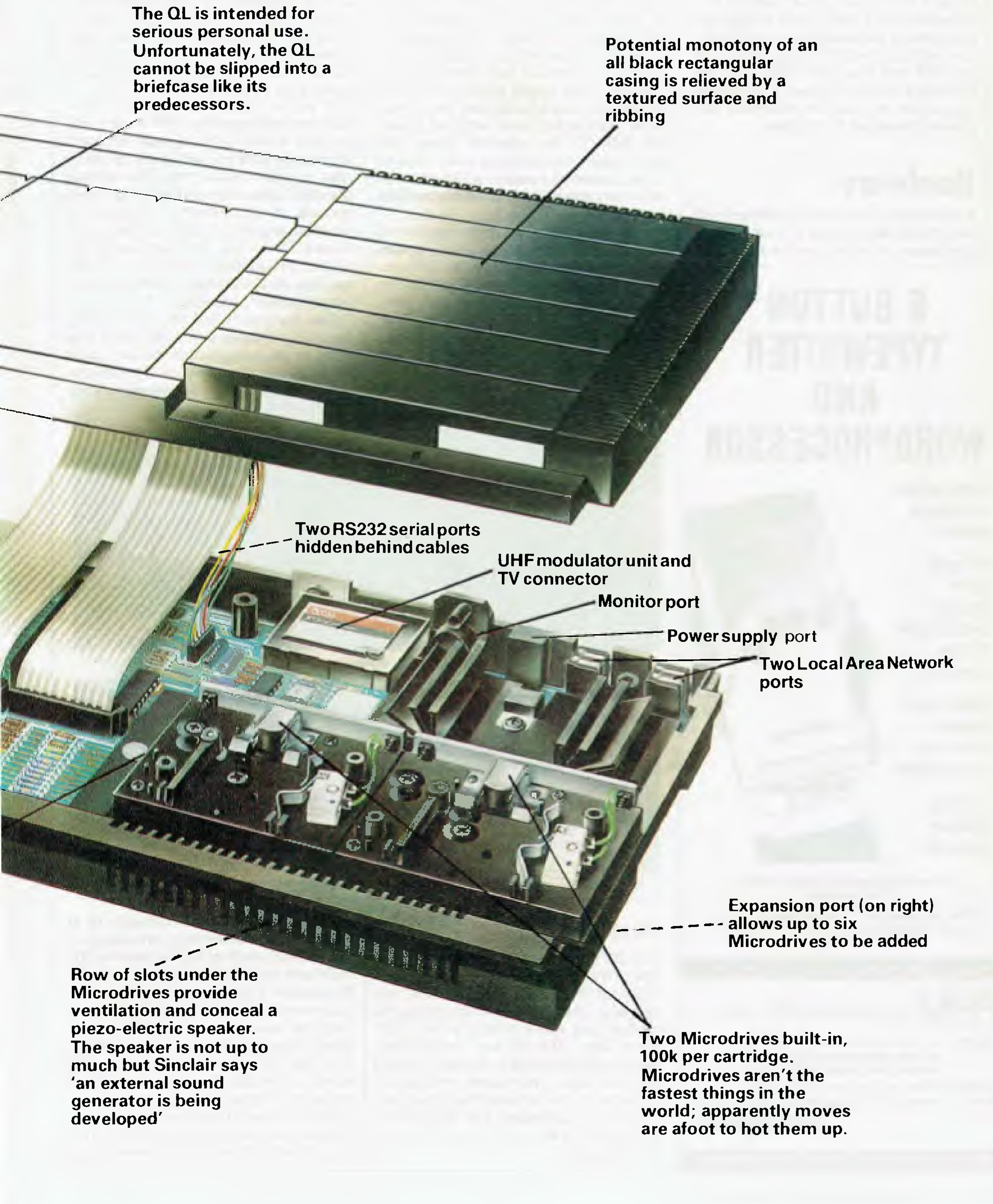
It wasn't until I had been using the QL for many hours that I twigged what it was all about. Like the lap-helds and the Macintosh, the QL has been designed for serious personal use. The supplied applications are just what the professional user needs to get started. Of course, the built-in SuperBasic will appeal to the enthusiast, too, but I suspect the bulk of QL sales will be to people who would like a business computer but cannot afford the cost normally associated with such a purchase.

The QL will cost $1,000 (or thereabouts) but you'll need a printer which will cost from around $500. If you use your existing television, you are ready to go for $1,500. I would strongly advise buying a monitor if you plan to use the QL for hours at a time. This might cost you another $500, so you still get away with a computing facility for less than $2,000.

Julian Barson, the Australian distributor of Sinclair computers, was going to feature on the front cover. But there was a fight, and the monkey won.



Australian Personal Computer


The QL is intended for serious personal use. Unfortunately, the QL cannot be slipped into a briefcase like its predecessors.
Potential monotony of an all black rectangular casing is relieved by a textured surface and ribbing
Two RS232 serial ports hidden behind cables
UHF modulator unit and TV connector
Monitor port
Power supply port
Two Local Area Network ports
Expansion port (on right) allows up to six Microdrives to be added
Row of slots under the Microdrives provide ventilation and conceal a piezo-electric speaker. The speaker is not up to much but Sinclair says 'an external sound generator is being developed'
Two Microdrives built-in, 100k per cartridge. Microdrives aren't the fastest things in the world; apparently moves are afoot to hot them up.

At the moment the firmware is held in three 16k EPROMs which means that one of them must occupy the ROM slot at the back of the machine. You will not be able to run early QLs without this expansion ROM pack. Once the operating system is tucked away on real ROMs, I am told that it will fit inside the QL (on one 32k and one 16k ROM) and the ROM pack slot will be freed. Presumably this will be the case for machines earmarked for export to Australia.

Hardware

A working QL comprises a keyboard with two integral Microdrives, a power supply, a television or monitor and its associated lead. An RS232C cable and a network cable are also supplied with the machine. If you do a lot of typing, you might find the keyboard lies a bit flat. To overcome this, Sinclair has supplied three funny little plastic feet which are supposed to fit into rubber pads under the keyboard. I found that these fell out regularly and in the end I dispensed with them and got used to a new typing position. To compensate for this I found that the printer cable worked first time with my Epson MX 80F/T. The network driver still hadn't been implemented in my version of the operating system, so attempts to talk to my son's Spectrum were doomed to failure.

Just in case you were wondering about the QL's portability, Sinclair boasts that it weighs around three pounds. What it doesn't boast about is that the separate power supply weighs over two pounds! Also, you still need the television and, at eighteen or so inches wide, the QL cannot be slipped into a briefcase like its ZX predecessor.

The keyboard is the 65-key conventional qwerty layout

L to r: sockets for networks, power, 8-pin DIN, UHF, RS232C, joysticks and ROM

Keyboard

The QL has the sort of style the public has come to expect from Sinclair. The potential monotony of an all black rectangular casing is relieved by a textured surface and some ribbing at the right-hand side. The 65-key conventional qwerty keyboard looks very neat. Despite the inevitable membrane mechanism underneath, the keyboard feels good and positive in operation. The Microdrives are hidden under the flat area to the right of the keyboard.

The keyboard highlights the fact that we are not really looking at a traditional Sinclair machine. With its ESC key, five function keys, CTRL and ALT, it would seem to represent a quantum leap by this company into the arms of convention. I must say I'm relieved. I felt immediately at home with the QL and found I could get on with the important business of using it for productive work.

A yellow light at the front of the keyboard tells you when the machine is on and a red light in front of each Microdrive tells you when it's in use. A row of slots under the Microdrives provide a degree of ventilation and also conceal a piezo-electric speaker. A covered expansion port on the right allows up to six Microdrives to be added. Incidentally, they are not the same as the ZX Microdrive although the blank tapes are identical.

The Spectrum and the QL actually format their tapes differently so, if you want to exchange information, you'll need to use the network. The Microdrive cartridges have a capacity of at least 100k. In theory they can take up to 255 sectors, each of 512 bytes on a 200 inch tape loop. A reset key next to the Microdrive expansion port is an improvement on the ZX range which requires you to pull out the power supply lead to obtain the same effect. Like the ZX range, there is no on/off switch on the QL. The left-hand side of the keyboard has an enormous expansion port covered by a removable plate. This will be used for the promised 512k of add-on RAM. No doubt enterprising companies will invent all sorts of devices to plug in here.

Turning to the back of the keyboard, reading from left to right, there are two network sockets, a miniature power socket, an 8-pin DIN socket for the monitor (monochrome or colour), UHF socket, two RS232C sockets, two joystick sockets and a slot with a removable cover for the ROM pack. This pack can hold up to 32k. The RS232C and the joystick sockets are like those newfangled telephone jack sockets. If one RS232C socket doesn't do what you want, you can try the other which has its signalling pins reversed.

Removal of the 10 screws which hold the QL together reveal a very tidy interior. The main PCB is very crowded but neatly laid out. The large Motorola 68008 can be seen on the left, 16 chips make up the 128k RAM, two EPROMs contain the operating system and SuperBasic and an extra 8049 processor controls the keyboard among other things. Four Sinclair-designed ICs control the display, memory, RS232C, network and Microdrives. Over on the right are the Microdrives which look just like the innards of the ZX Microdrives. Behind them is an enormous heat sink which nicely warms the flat panel to the right of the keyboard.

The keyboard is covered by an aluminium plate which when removed reveals the mechanism. The key presses down on a sort of moulded soft plastic dome. The dome has a spike moulded on its inside which in turn presses down on a sheet of plastic printed with metallised tracks.
This presses in turn on another similar sheet thus completing an electrical connection between two tracks — one on each sheet. It sounds nasty but it actually works very well. Once you appreciate how the keyboard works, you can feel it in use but I doubt that most people would notice.

I used the QL with a domestic television; and I must admit that it got a bit tiring after a while. I have seen the machine in operation with a monitor and the picture was a lot better with none of the irritating flicker inevitable with UHF. The computer display wraps off the corners of the television screen, another reason why a monitor may be preferable.

On the other hand, it is possible to select from a variety of character sizes and display resolutions so that you can reduce the problems somewhat. Psion gives users a choice of 80,64 and 40-column displays for its programs. The 64-column display works very well: I found myself using that mode all the time.

Overall, I was very pleased with the quality of the QL hardware; it behaved faultlessly the whole time I had it.

Software

SuperBasic

As I mentioned earlier, the operating system and the Basic were not finished on the machine I tested. They were, however, complete in most important respects.

The Basic is a very powerful language with some additional structures over and above the earlier ZX Basics. Particularly impressive is the ability to define extensions to the language using the procedure definition facilities. One current limitation in SuperBasic restricts overall program size to 32k. This is to do with the internal error checking. Sinclair tells me that this restriction will be lifted in due course.

The other major omission in the review machine's Basic was the full screen editor promised at launch. At the time of writing, the only way to edit a line of Basic is to re-enter it.

That's all the bad news; now let's look at what's actually in the Basic and perhaps see why it has been christened SuperBasic.

The Basic follows a similar pattern to all Basics. I had few problems writing little routines. I know I shouldn't admit this but I must confess to having introduced a couple of GOTOs in my programs. SuperBasic doesn't mind; it just makes them a bit unnecessary. Anyway the point of this confession is that you cannot say IF . . . THEN and a line number, you must say IF . . . THEN GOTO. Basic commands must be typed in full — there is no keyword entry system, although I suppose you could create your own using the procedure definition facilities.

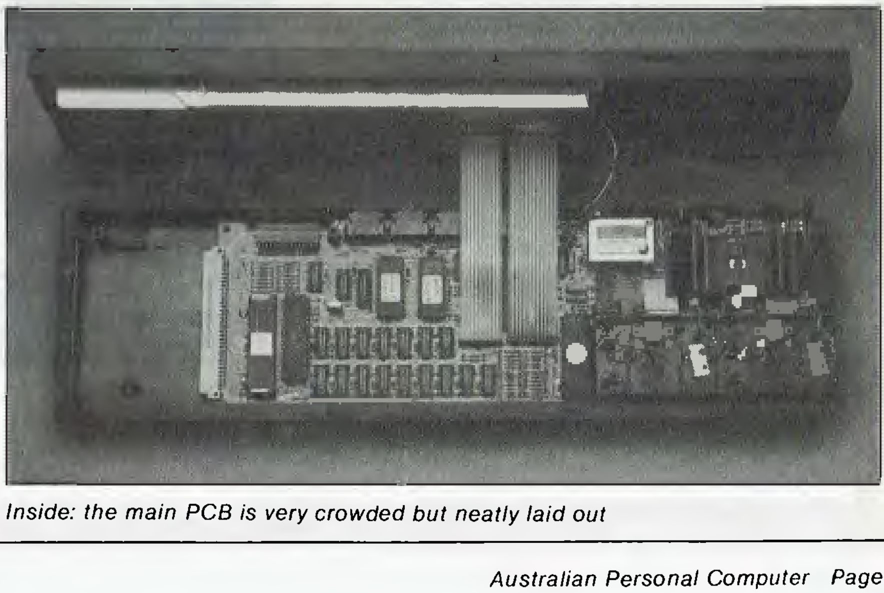
An AUTO line numbering system was missing on my copy but I have been assured that it will be included in the released version. This, coupled with a RENUMBER command, will lead to tidier programs. BAUD sets the baud rate of the two RS232C ports — yes, both of them must be set to the same speed which can be one of eight between 75 and 19200 baud. 19200 is reserved for transmission only. BEEP makes sounds through the grotty speaker. Pitch and duration can be varied, as well as things like second pitch and bounce, which 'bounces' the sound between the two pitches. A fuzzy option adds a random number to the pitch on each cycle causing an appropriate distortion. Fine for fun but not of massive practical benefit. I hear that an external sound generator is under development which should help.

The Basic contains an impressive range of graphics facilities as you might expect. Windows, borders and blocks of colour can all be created on the 512 × 256 resolution screen. Up to four colours are available in this high resolution mode and the lower resolution (256 × 256) gives eight colours plus flashing.

The MODE command lets you switch between 256 and 512 screen widths. When you define the ink and paper colours, you can also define a stipple pattern but, unless you like shimmering, don't use it on the domestic TV. Character sizes can vary in width in four stages from 6 to 16 bits wide and, in height, either 10 or 20 bits high. This would be useful in headings or in applications for young children or partially sighted people, for example. A PAN command allows you to slide the contents of a window sideways. Once you have lost stuff from the window you can't get it back without regenerating it.

Windows are handled by allocating each one a spare channel then you simply address your Basic commands to the chosen channel.

Now for the various control struc-



Inside: the main PCB is very crowded but neatly laid out

Simply the best in quality and price . . .

FREE FREIGHT IN AUSTRALIA

star

★ **AMUST DT80 PRINTER**

- 80 characters second
- Bidirectional printing
- Crisp clear print style from square pins
- Superscript/subscript
- Graphics capability
- Italics
- Sprocket feed for fanfold paper
- Pressure feed for sheet paper

$257 (EXC)

YES THAT'S RIGHT!
OR JUST

$299 (INC)

STAR GEMINI
10X or 15X

Fast 120 CPS
Cloth ribbon (not carbon)
Friction & tractor
Roll paper feed

We cannot be beaten on price.
Over 200 units will go this month.

OUR PLEDGE TO **YOU** . . .

We absolutely guarantee to supply all our mail order customers the very best quality.

All mail order products are clearly and unconditionally covered by the Data Parts 6 months parts and labour warranty!* (That means something —ask our customers).

* Ask about our exclusive Data Parts warranty. Its as good as having a Data Parts store next door!

Data Parts are one of the biggest and oldest mail order houses in Australia. We don't just claim to be the best . . . check our superb quality products . . . but still we are never undersold!

And remember all products are delivered free to your door.

YES — FREE FREIGHT
anywhere in Australia
RING ME PERSONALLY
YOU WON'T BE
DISAPPOINTED

Regards

Boyd DAINTON

SILVER REED DAISY WHEEL PRINTER

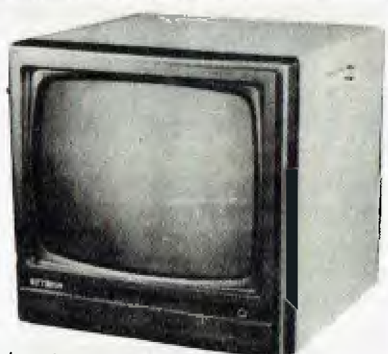
PRICES SLASHED

$895 to $579 (exc) or $679 (inc)

Quality unit, all deluxe features, serial or parallel.

Color monitors that give you more performance and reliability

ADI COLOUR MONITOR
$399 inc
$339 exc
COLOUR MONITOR



Suits Commodore, Apple, etc.
(Manufacturers rec. retail price $620).

NATIONAL • INTRA • BBC • MITISBISHI • XEROX • SILVER REED • DIGITAL • MPI • DEC • TEC

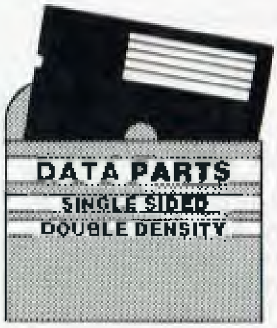
COMMODORE • BMC • AMUST • TEC • COMMODORE • AMUST • TEC • COMMODORE • AMUST • BMC • NEC

• EPSON • OSBORNE • C·ITOH • STAR • ADI • TELEVIDEO • TEAC • VERBATIM • DIGITAL

You'll never buy better than now . . .

FREE FREIGHT IN AUSTRALIA

★ BEST PRICE EVER ★ ★ ★ COMMODORE ★



$29.90
(INC TAX)
$25.90 EXC
In plastic library box

$199.00 (EXC)
ON QUALITY IMPORTED APPLE COMPATIBLE DISK DRIVES
($238 INC TAX)
55 UNITS AT THIS PRICE
SCHOOLS. PHONE FOR YOUR SPECIAL PRICES

ATLAS.8 SLIMLINE
$234 EXC. $264 INC.

• SENDATA • DIGICARD • INTRA • TRIO

SAVE $$$ $28.50

*HOBBIEST ADD ONS	(exc tax)
MPI B51 Disk Drives	189.00
MPI B502 Slim Line DS/DD	289.00
Mitsubishi 4853	329.00
Mitsubishi 4854	349.00
National D/S D/D NEW LOWER PRICE	269.00

CABLES (Printer)	(inc tax)
Serial — serial	$19.95
Centronic — Centronic	29.95
MicroBee Cable	34.95
Centronics plugs solder . . (each)	4.90

*PRINTERS	(inc tax)
DT80	299.00
EPSON RX80	CALL
EPSON FX80	CALL
STAR GEMINI 10X	CALL
AMUST 120	399.00
DIABLO/XEROX 630	1998.00
SMITH DAISY WHEEL	595.00
PRAXIS '40	695.00
C. ITOH COLOR PRINTERS	CALL

PAPER SPECIAL
10" TRACTOR FEED TOP QUALITY BOX 2000 . . . . 39.00

*MONITORS	(inc tax)
NEC 15 MHZ Green	139.00
BMC 18 MHZ Green	174.00
INTRA 20 MHZ Green Steel, Super Hi Res	179.00
INTRA ADI Color (Composite & RGB)	399.00
AMBER MONITOR HIRES with Swivel Base, Super HI-RES	199.00

DOWN UNDER CLOCK
for Apple $99
fully PRODOS and THUNDERCLOCK compatible battery backed

*APPLE COMPATIBLE	(inc tax)
Z80 CARD	54.50
SNAPSHOT CARD	119.00
80 COL 'DIGI SWITCHING'	89.00
16K CARD	60.00
PARALLEL PRINTER CARD & CABLE	89.00
COOLING FANS	49.00
JOYSTICKS TOP QUALITY	39.00

DX-85 for 5¼" Disks **$35**
DX-88 for 8" Disks **$50**
(Holds 50)

LOCKABLE DISK BOX
HOLDS 100 DISKS

***OSBORNE** (inc tax)
- MODEL 01, D/D drives 80/100 col. card and all the software 1895.00
- EXECUTIVE Model unit complete with all the extras 2850.00
- Best deal in Australia on all Osborne products

*COMMODORE	Price (inc tax)
1541 DISK DRIVE	$380.00
1526 PRINTER	425.00
1526 LOOKALIKE PRINTER with Interface	368.00
64 COMPUTER	CALL
DATASET	48.00
KOALA PAD	129.00

*MEMORY COMPONENTS	(exc tax)
6116 RAM	5.68
4116 RAM	1.59
4864 RAM (any quantity)	6.59
2732 EPROM (any quantity)	4.50
2764 EPROM	6.99
2716/2516	3.80

*DISKETTES	(inc tax)
OUR BRAND Box of 10	29.95
Datalife (the best)	525.01
Box of 10 S/S D/D	34.95
DEC Formatted diskettes (10)	55.00

*TERMINALS	(exc tax)
KDS. I ICL Lookalike	790.00
ESPIRIT II	795.00
TATUNG VT100 or VT152	695.00
• MODEMS Sendata 300	199.00

FOR ITEMS NOT MENTIONED PLEASE PHONE

• ARCOM • C.ITOH • ZEUS • ACT • OZISOFT • BBC

TRADE ENQUIRIES WELCOME — CASH/CHEQUE OR BANKCARD WELCOME HERE

DATA PARTS

MAIL ORDER CENTRE
11 EDWARDS STREET, SHEPPARTON VIC 3630
Telephone: 21 7155 (058)

- **SHEPPARTON (As Above) Telephone: (058) 21 7155**
- **ALBURY 658 Dean Street Telephone: (060) 21 8088**
- **BENDIGO 27 Bath Lane Telephone: (054) 43 4866**
- **BALLARAT 307 Mair Street Telephone: (053) 31 3399**
- **WAGGA 33 Thompson Street Telephone: (069) 21 6466**

Opening in Melbourne soon.

tures available in SuperBasic. Firstly there's DEF FN which is probably already familiar to you. It allows you to define a function which returns some sort of value to the line using it. It also allows you to use local variables through the LOCAL statement. These may have the same name as variables outside the function definition but they will not become confused during execution of the program. DEF FN is terminated with END DEF. Similarly DEFine PROCedure is terminated with END DEF and it, too, allows local variables. When defining the procedure, any parameters needed are placed in brackets: for example, DEF PROC F (LIMIT), but when the procedure is used the brackets are not needed: for example, F 23. You can do some super things with this facility and, of course, it does away with GOSUBs and the complications of parameter passing.

And that's not all. The old familiar FOR . . . NEXT has a new twist. It allows you to slip some commands after the NEXT but before the END. REPeat gives you a repeating loop which you escape via an EXIT command embedded somewhere in the structure. Perhaps you need to wait until an input satisfies a test before moving on. Once again this is an attempt by Sinclair to encourage GOTO-less programming.

Archive (the database) is one of four excellent business cartridge programs supplied free with the QL

The final neat structure I'd like to mention is SELect. Used with ON this gives a similar effect to ON . . . GOTO but you can embed all the actions inside the procedure itself. For example, SELect ON a can be followed by ON A=1 and then all the things you want to happen if A=1. The next ON will stimulate a new set of actions for a second value of A. This command ends with END SELect. Very neat.

SuperBasic has an interesting feature which Sinclair calls 'coercion'. This allows you to mix your variables when performing calculations. For example, it will let you add '2' to '2' and still get '4'.

Finally, the Benchmark timings. They put the QL in the top third or so of all machines tested by *APC*. Frankly, the comparative speed of machines is insignificant; it's far more important to take the broader view and decide how the facilities offered compare.

QDOS

The QL operating system, QDOS, is busily working in the background whenever the QL is in use. It makes its presence known, for example, when you need to run jobs in different windows and when you are transferring data to and from the Microdrives.

When the QL is first switched on, the screen is divided into three separate windows. The bottom few lines of the screen is a command entry and message display area where all the direct interactions with QDOS and SuperBasic take place. The top of the screen is then divided vertically with the upper left being used for displaying the developing Basic program and the upper right displaying the user view of the program when you run it. The bottom of the screen is attached to channel 0, the listing area to channel 2 and the program execution area to channel 1. It is a simple matter to direct commands to the appropriate channel.

QDOS handles multi-tasking by dividing processing time between the various jobs being executed, and according to their built-in priorities. As I said earlier I couldn't set up an adequate test system to see how this performed. What I do know is that at the launch it was less than impressive. The fact is that the QL can do several things at once. The performance depends more on the quality of the programs running than on the inherent capability of the operating system.

As you may have guessed from the channel numbers, all I/O is device-independent. You simply choose your channel, attach your device to it by saying something arcane like OPEN $5; CON__200×200a50×50__and you have created a 200×200 bit window starting at location 50×50 and with a keyboard buffer of 32 characters. Easy isn't it?

The really annoying thing about QDOS for me was that I had to put an underline at the end of my Microdrive commands. For example 'DIR MDVI' does not give a directory of Microdrive 1 whereas 'DIR MDV1__' does.

Applications software

Without doubt, the Psion programs tip the scales heavily in the QL's favour when comparing the system against others.

Four packages are provided: word processor, database system, spreadsheet program and business graphics. I used all four programs to a limited extent. Then I chose to plunge into the database program in detail.

I'll start with the others and then take a closer look at Archive, the database.

Easel was the largest and most complicated to program, yet it appears to the user as the most trivial and the easiest to use. It's great for bringing numbers to life. At the simplest level you can literally load the program and start keying in numbers and immediately a histogram appears on the screen. If you go off the scale it automatically rescales itself and lets you carry on. If you want to enter another set of figures, simply choose a new name and start keying the new ones.

Let's say the first lot were called 'TURNOVER' and the second lot 'COSTS'. You could easily create a third set by saying 'MARGIN=TURNOVER-COSTS'. These can then be displayed individually as histograms, line graphs or pie charts or they can be superimposed on each other to show the relationships between differing sets of figures. Text can be added to the charts, and moved around, and everything can be printed — provided your printer is capable of graphics.

You can read in files which have been prepared either by Abacus, the spreadsheet or by Archive, the database. Such data will be displayed according to the currently selected format. Data can be transmitted from this program to others in the form of tables of numbers.

Deeper inside the package there are facilities to manipulate text, open windows, vary column widths and transfer data.

Quill, the word processor seems to have very grown-up facilities. I particularly like the fact that what you see on the screen is what gets printed out. Superscripts and subscripts are handled by Quill's built-in special character set.

Underline is an inherent feature of the QL and this is also used to good effect.

Another thing that impressed me was the fact that the current character position is shown by a highlight on the margin ruler at the top of the screen. Quill keeps a word count as you go along, something many professional writers will find invaluable.

Benchmarks

BM1	2.1
BM2	6.4
BM3	10.7
BM4	10.3
BM5	13.2
BM6	26.1
BM7	61.8
BM8	25.8

All timings in seconds. For a full listing of the Benchmark programs see 'Direct Access'.

PORCHESTER Computers

DELIVERY AUSTRALIA WIDE

• MAIL ORDER • PHONE ORDER • CALL-IN

1.6 MB TWIN FLOPPY AND 10 MB HARD DISK AND NEW DESK-TOP MODELS

EXECUTIVE 816 BUSINESS DESK-TOP AND PORTABLES

is a powerful business system and comes to you complete with a large range of ready-to-run business software including: Debtors, stock control. creditors, general ledger word processing, database, mail merging, supercalc spreadsheet, membership, office costing, job costing, & communications.
The standard 816 has twin floppy disk drives with a large 1.6 Mbytes AND NOW!
The 816-10 has a full 10 Megabytes Hard Disk PLUS an 800K floppy

COMPLETE WITH SOFTWARE AND TRAINING!

1.6 Mb model $3,555 inc. 10 Mb model $5,555 inc.

AUSTRALIA'S BIGGEST SELLING PRINTERS —

BX80/80 DT's

80 column, friction and adjustable tractor feed, bidirectional logic seeking, HI-RES graphics and block graphics, sub and superscripts, condensed and emphasised print, and underlining vertical and horizontal tabs, self test, italic print, etc.

WE SELL HUNDREDS OF THESE PRINTERS AND OUR NEW LOW PRICE IS

$295 inc
$255 ex

Perfect for Apple, Tandy, Commodore 64, Microbee, Executive 816, IBM PC, DEC RAINBOW & many more. Phone us to check cabling for your computer!

NASHUA DISKETTES

$29.95 inc.
$25 ex

(Plastic box of 10)

Apple, Tandy, Microbee, Commodore 64, etc.

PRINTER CABLES!

TO SUIT:
APPLE II/IIe
IBM PC
TANDY COMPUTERS
COMMODORE 64 — 2K BUFFER
MICROBEE
PLUS OTHERS! — PHONE FOR DETAILS

Cables for 80 DT/BX80, Gemini, Logitec, Executive

LOWEST PRICES • FAST DELIVERY • FULL WARRANTY

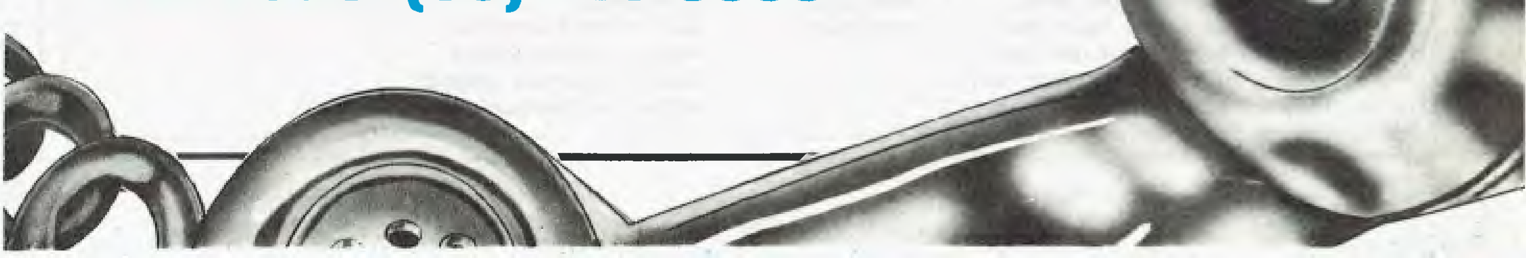
CUSTOMER BENEFIT PACKAGE • Free expert consultation • Technical Staff available to assist with your enquiries • 6 months warranty on all items • $10 delivery Australia wide • Best prices still with quality service and helpful advice.

PHONE ORDERS • Ring (03) 417 6999
MAIL ORDERS • Send cheque/Postal Order, or Bankcard details.

• DEALER ENQUIRIES WELCOME •

6 MONTHS WARRANTY ON ALL PRODUCTS

bankcard MasterCard



Now Archive, the database. I spent hours on this one. It is an extremely deep product. A novice user can use it literally like a card file and be none the wiser after months. Others will find that they prefer to use a special screen layout rather than the default. Some people will want to access multiple files while others will want to build some sort of system around Archive. All these things are possible within the one product.

I started by simply listing the fields into which I wanted to enter data and then creating a simple file of names and addresses. That was easy — it took about 10 minutes. Then I got ambitious and started entering data which exceeded the available space for its entry. The data was accepted although it actually ruined the display. Archive will accept variable length data up to 255 characters per field. In this respect it shows similar limitations to other popular databases.

Since the QL has no character graphics, the design of records is achieved with judicious use of the exclamation mark and the hyphen. The end result is usable but looks a little tacky. Bearing in mind the sort of program size and timescale limitations Psion was working to, I think this is a small price to pay for the power and usefulness of the product.

It is possible to locate records in a file quickly by using the search or locate commands. The first ignores case and the second matches exactly. A sort is built into Archive and it works on the straight ASCII sequence. This means that 'Computer' would come before 'all', for example.

Like SuperBasic you can define procedures within Archive, and the potential for developing applications and new commands is mind-blowing. Suddenly the entire product becomes soft and you can redefine it to suit your needs exactly. For example, do you always open the same files, display them in the same form and then access a particular record. Fine. There's no reason why that shouldn't all be done with two keystrokes — G and ENTER, for example. I chose 'G' for 'Go'.

There's a lot of power in this package and it really will repay diligent study.

All the Psion packages have plain English commands, although this does make them a little long-winded at times. They all have extensive help facilities which can be called up at any stage. Psion is about to launch its products onto other machines and I reckon this is an excellent marketing ploy.

People who become used to using Psion programs on the QL will feel much happier if they also use them on their real machines at work. Conversely, people who use Psion on their real machines may want to buy a QL because they've already learned what Psion's products can do.

Documentation

I was provided with preliminary documentation which was adequate. There was no beginners' guide, although I understand that this will be available in the final version.

A Basic keyword summary is provided which explains every Basic instruction in detail. A concepts section runs through all the concepts regarding the QL which is beyond the scope of the Basic keywords section. And each of the applications packages has a guide to itself which acts both as tutorial reference material. Inevitably, with a preliminary document there were many discrepancies between the manual and the products I was using. Structurally, the documentation is fine and I trust the errors will be put right before the final version is issued.

Conclusion

There's no doubt that the QL is a well made piece of hardware. The operating system, the applications and the Basic look very good, on paper. The review machine was still short of a few facilities. This is one of the occasions you should be thankful Australia won't get a product soon after its launch: the version to be landed by Barson Computers will surely have most of the glitches removed.

If everything were in place, then I would consider this machine very seriously as a truly personal computer but not as something to run a business on. The Psion spreadsheet, database, business graphics and word processing packages coupled with the limited Microdrive capacity define the market very clearly. The Psion programs look very good in terms of the ranges and quality of facilities offered.

The bottom line is that the QL gives you the potential to own a complete serious computing facility, including printer and essential software, for under $2000. Well under that if you're prepared to use a domestic TV rather than a monitor for the display.

END

Technical specifications

CPU	7.5MHz Motorola 68008 plus Intel 8049
RAM	128k (32k used by display). Expansion to 640k coming
ROM	48k QDOS and Super Basic. 32k ROM pack (see review)
Display	512x256 four colour, 256x256 eight colour. UHF or RGB
Keyboard	65-key, normal qwerty plus five functions and cursor
Microdrives	Two drives min 100k each. Can expand with further six
I/O	Two RS232. Two joystick. Two network
Languages	SuperBasic
Applications	Quill — Word processor Abacus — Spreadsheet Easel — Business graphics Archive — Database

The new Sharp MZ5500

A rare combination of features.

1. 16-bit Processor.
For real performance a personal computer must be able to handle lots of memory; the MZ5500 starts at 256 Kbytes and can be expanded to over half a million characters. And it must be fast; the MZ5500 has a true 16-bit processor to handle complex graphics and data processing with ease. If you want real number-crunching power, just plug in the optional 8087 numeric data processor for more power than some mainframe computers.

2. Multiwindow.
Up to four windows can be displayed on the screen simultaneously, allowing you to mix text, tables and graphics. The bit-mapped display simplifies programming and provides smooth scrolling — as little as one dot at a time with no 'jittering'. Other enhancements include colour palettes and a colour-priority function. The MZ5500 comes with 96K bytes and can be expanded to 192K bytes of video memory, ensuring a high resolution screen image.

3. Mouse.
An optional mouse lets you point to shapes on the screen and issue commands. Complementing the MZ5500's graphics capabilities, the mouse is ideal for drawing shapes or for bypassing commands.

4. Software.
The MZ5500 uses CP/M-86* and MS-DOS 2.0** (an option currently under development) operating systems. Used by leading software authors, so you can run the most popular software. Choose from word processing, accounting, database management and others.

5. Sharp engineering.
The MZ5500 sets a new standard in quality of engineering, reinforcing Sharp's reputation as a world leader in computer development.

Complete the coupon or call Sharp today, for your nearest dealer.

To: Mr. Terry Cleary Computer and Systems Division
Sharp Corporation of Australia 64-72 Seville Street FAIRFIELD NSW 2165
Telephone: 728 9111

Please send me some literature on the MZ5500 ☐
Please have one of your dealers contact me ☐

NAME ____________________
POSITION ____________ TEL No ____________
COMPANY ____________________
INDUSTRY ____________________
ADDRESS ____________________
POST CODE ________

SHARP

Simply the best in computers

† *Simulated monitor image*

*CP/M-86 is a registered Trade Mark of Digital Research. **MS-DOS 2.0 is a registered Trade Mark of Microsoft Incorporated.

Sydney 728 9111, Melbourne 763 9444, Brisbane 343 9144, Adelaide 294 7166, Perth 277 7477, Canberra 805 288.

APM 76

SOFTWARE

Friday!

Ashton Tate's Friday! aims to provide simple facilities in a simple manner. Is it on target and is there an easy upgrade path to the company's more powerful dBase II? Kathy Lang reports.

Ashton Tate's data management package dBaseII is still, despite its age and, according to some users, unfriendliness, the market leader. But a good many users do not need the power and flexibility of dBaseII, nor are they prepared to pay $840 for such a package. Ashton Tate has now brought out a data management package called Friday!, aimed at meeting the needs of those who do not need the full power of dBaseII, but who do want straightforward facilities provided in a manner which is very easy to use. At $360, it is competitively priced but how far does it meet the aim of providing simple facilities in a simple manner?

Friday! is a menu-driven system with a reasonably clear structure, and one unusual and highly desirable feature: at almost every stage, it is possible to end your current task and get straight back to the main menu without having to backtrack through several layers. Friday! uses fixed format, fixed length records, so the maximum necessary space is used to store each record regardless of the actual amount of information it contains. A reasonable variety of functions is provided including a simple letter-writing facility. Files can be read by dBaseII, but if they are altered it may be necessary to re-index the file when you next use it.

Friday! can only handle one file of records at a time, although you can use a second file to 'look up' values which regularly recur in order to avoid storing repetitive information (Fig 1).

The major limitations on records are that they may contain no more than 32 fields, and character fields may be no longer than 32 characters. ('Look up' fields, whose values are recalled by looking up the value entered in another reference file, may be up to 60 characters long.) This makes Friday! inappropriate for applications involving text, such as library catalogues. Three types of data field are allowed: you can store items as numbers (called Math fields — ugh!), characters (called Anything) or Logical (containing the values Yes or No). Numeric fields (which may be whole numbers or contain a decimal point) are entered from the keyboard or calculated by Friday!, either at data entry or subsequently.

The first step in setting up a file is to tell Friday! the name, type and length of each item of information in the records. In addition to the conventional data types of character, number and logical item, you can have 'look up' fields. These involve entering an abbreviation for a field, which is filled from a separate file containing a list of abbreviations and their full forms. This feature can be used to ensure that only valid values are entered for a data item, and also as a substitute for two-file processing in some circumstances. Supposing you have a list of products, many of which come from the same supplier; you can avoid having to keep the supplier's full name and address in every relevant product record by storing in the product record a short code for the supplier's name and address, and storing just one copy of the full version of the information in a reference file, which is looked up whenever the product record is retrieved.

Unless you request otherwise, the first field in the record will be used as the 'primary key' — that is, it will be used to determine the order in which records are stored, and to allow fast retrieval of individual records. You can ask for several fields to be used to order

Maximum file size	OS Limit	Data validation	Average
Max record size	999 chars	Screen formatting	Col and row entered
Max no. fields	32		
Max field size	32 chars	Unique keys	No
Max digits	10	Report formatting	Col and row, Letters
Max prime key length	64 chars		
Special disk format?	No	Save calculated data	Input, update
File size fixed?	No		
Link to ASCII files?	Yes, various	Totals & statistics	T+ST
Data types	Numeric, Char, Logic	Save selection criteria	Mandatory
		Combining criteria	And,Or,Not
Fixed rec structure?	Yes	>1 criterion/field?	Yes
Fixed record length stored?	Yes	Wild code selection?	String Within
Amend rec structure?	By copying	Browsing methods	Any Field
Link data files?	No	Interaction methods	Menu
No. data files open	NA	Reference Manual+	***
No. data files open	5	Tutorial Guide+	*****
No. keys	1	Reference Card+	****
Max key length	64 chars, 5 fields	On-Line Help+	****
		Hot-Line?	Dealer support
Subsidiary indexes kept up-to-date?	NA		

Fig 1 Features and constraints

the file (name within department in a personnel records system) but only the first that you specify is available for fast retrieval through the 'quick search' facility.

Once the record format has been set up, Friday! will then create a 'default' format for displaying it on the screen. You may use this, or create one or more formats of your own (up to 15 for any one data file). The record format can be changed to add or delete fields or amend data types within Friday!. Although this involves Friday! in copying the data, it is done without you having to worry about the mechanics, except to allow enough space on the disk to enable the copy to be made. If you want to change field names, you must copy the Friday! file out to a text file and copy it back.

Entry of records is carried out by entering one record at a time on the screen, with full control over cursor movement using the same keys as those used by WordStar. To amend records, you can use the 'quick' search technique with the key, or set up a retrieval rule to select on other fields. When using the key, you get all fields starting with the characters you enter: for example, entering 'Johns' will get you all the Johnsons and Johnstones, too. However, this does mean that you need type only the minimum characters necessary to ensure you find the right record. Whether searching with the index or sequentially, you can restrict the search to a particular range of records, and if it will speed things up to utilise the order into which the file is sorted, then Friday! will do so. When records are deleted, they are simply marked for deletion and may be 'undeleted' until the records are finally purged.

For each file, Friday! automatically creates a screen format called 'Fridayform' which displays all the fields in the record on a single screen. If you prefer, you can design your own screen layouts; you may have up to 15 for any one data file, and these need not all contain every data item. The process of format design is quite straightforward in all respects save one. You are shown a grid representing the layout being designed; you then input the column and row for each field in turn, and these are entered on the grid. For each field, you may also enter a 'typing guide', which allows Friday! to do some validation on the value being entered. For instance, you can specify that a character field is actually to be two letters followed by three digits, or that a numeric field to be used for currency may contain no more than two digits after the decimal point.

The real advantage with the Friday! approach is that if you decide you need to amend the layout of a field, you have to re-enter it completely — you can't edit the format. Worse still, you can only re-enter or delete a field if it's the last one you entered. To erase an earlier field format, you must erase all the ones between the most recently entered field and the one you want to change or delete, and then re-enter all the field formats that were correct as well as the format in error!

In addition to screen formats designed for showing complete records, any report can also be displayed on the screen provided it occupies no more than 20 lines per record.

Printed reports

You may design up to 15 report formats for each file, which may use either the Quick format or a more sophisticated Custom format. The Quick format allows you to specify the items you want to include, the column start, and width for each item. Items may be fields, text, a combination of the two, or expressions combining numeric fields or constants. You can also specify the number of lines per page, whether totalling and sub-totalling is required, and whether a summary or full report is required. The summary report gives totals and sub-totals only. I couldn't persuade Friday! to give me a summary report just showing totals without utilising the sub-total facility. Sub-totals are shown for each numeric field when the value of a single field changes — it's not possible to have more than one controlling field for sub-totals.

Item	Line	Column	Contents
1	21	12	[TRIM(FIRST:NAME)+" "+LAST:NAME]
2	22	15	[ADDRESS]
3	23	15	[TRIM (CITY)+" "+STATE+" "+ZIP]
4	26	15	["Dear New Client,"]
5	28	15	["Thank you for placing your rental home with Fant"]
6	28	63	["asy"]
7	29	15	["Inc."]
8	31	15	["It has been added to our listings, and your home"]
9	31	63	["is now"]
10	32	15	["being offered selectively around the world. Wit"]
11	32	63	["h all"]
12	33	15	["its amenities, I'm certain that we'll have no"]
13	34	15	["difficulty in locating the right kind of tenant"]
14	34	63	["for"]
15	35	15	["your home"]
16	37	15	["If you have any question, or if we have overloo]
17	37	15	["ked"]
18	38	63	["anything in the attached listing, please don't"]
19	39	15	["hesitate to call."]
20	41	15	["I'm looking forward to working with you."]
21	44	15	["Sincerely."]

Fig 2 Custom report table

...other computers cringe

PortaPak comes with its sleeves rolled up. It was designed in Australia to get the most work done in the least time and at the lowest price.

If you want results, sheer computing power, trouble free performance, then PortaPak is the only choice. If you want prestige, if you need to stroke your corporate ego, we suggest you buy something else. Something that costs more, isn't as powerful but has a lot of "image."

Put PortaPak up against imported machines costing $5000, $6000 even $7000. The other machines cringe with embarrassment.

PortaPak has 800K of formatted space on each disk drive. The expensive machines which boast about having 360K suddenly look rather silly. With PortaPak you'll be able to handle much bigger data files and have far more programs on hand without having to fiddle around changing disks.

Reliability is often thought of in terms of machine breakdowns. Nowadays, the big problem isn't with breakdowns it's with hangups – when your machine turns out to be incompatible with your software. This is an enormous, widespread problem.

It's why we teamed up with Australia's software geniuses - Software Source Pty Ltd, the top suppliers of business and professional programs in the country. First, they specially customized PortaPak's operating system so it works with exceptional speed and simplicity.

Then they customized all their CP/M products to run perfectly on the PortaPak – products like dBASE II, SuperCalc, all the major languages, accounting systems, etc. Most importantly, they stake their reputation that all these products will work without hitches. No other computer in Australia can offer this total software support.

As for electrical and mechanical reliability, look inside a PortaPak. There's a striking difference. The PortaPak is completely modular. We didn't scrimp by putting all the circuits on one board. We use *three*. Servicing is simpler, quicker and cheaper. It's why the leading national computer service company, TCG Pty Ltd, is pleased to offer a 12 month service contract on PortaPak in all capital cities.

Take an extra close look at the Canon disk drives. If Rolls-Royce built computers, they'd use Canon drives. See the massive head protection shield? Hear the way the heads lock away every time they deselect? The designers had an unusual attitude to reliability – fanatical.

Now carry out some speed tests. On a standard benchmark test using BASIC routines*, the timings are: PortaPak 12.9 seconds, IBM PC 16.4 seconds, NEC APC 19.7 seconds and Sirius 16.4 seconds. Using a standard dBASE II routine**, the timings are: PortaPak 8 minutes 11 seconds, IBM PC 11m 52s, Sirius 17m 9s and NEC APC 19m 16s.

The expensive imports really cringe at this because they make so much of being "16-bit" machines. PortaPak is an 8-bit machine and proud of it. Not only is an 8-bit machine inherently better suited to jobs like word processing, accounting, spreadsheets, etc, but the 6MHz clock rate ensures it can run rings around the others even in complicated mathematical tasks.

Now to really rub it in, look at PortaPak's stunning additional features:

- Compact portability. It needs only half the desk space of its nearest rival. It goes with you on business trips, at night, on the weekend. *In one hand, you hold the concentrated working power of an entire office.*
- 9-inch screen, 80-characters wide but with 35 lines instead of 24. You see more of your work and the characters are the normal shape – not elongated. Much more readable.
- 640 x 304 high resolution, dot addressable graphics.
- Free software including Spellbinder word processing and office management system (the most powerful available), EBASIC compiler, MENU to make life easy for new users, MODEM for telephone communications, and Speed Print which lets you continue working while you're printing.
- Universal terminal emulation lets your PortaPak mimic the screen handling of other computers and run the programs installed for them. Your PortaPak can act as a terminal for any computer you care to name.
- Ability to read and write to other 5.25-inch disk formats – e.g. Kaypro and Obsorne. Lets you exchange software directly with owners of different computers.
- User definable character set lets you work with foreign alphabets, scientific symbols, etc.
- Five input/output ports for connecting to printers, modems, etc.
- Wide range of peripherals including 5 and 10 Mbyte hard disks, built-in world modem, RAM drive, EPROM programmer, A/D converter, etc.

Radically different, distinctly Australian design using industry standard software.

Two one-megabyte disk drives – total formatted capacity = 1.6Mb.

Z80B processor running at six megahertz.

9-inch screen, 80 characters by 35 lines 640 x 304 graphics.

Intelligent, capacitive, spill-proof keyboard.

Entire system including software and sales tax – only **$3395!**

For the name of your nearest PortaPak dealer contact:
The Portable Computer Co.
80 Beauchamp Rd, Matraville, 2036.
Phone (02) 661 4877. Telex AA71063

*See Australian Personal Computer, Feb., 1984.
**See Australian Micro Computerworld, Nov., 1983.

THE PORTABLE COMPUTER CO.

Custom reports are intended to give you greater flexibility over report layout. They enable you to set up more flexible columnar reports, and also provide a primitive standard letter format. I say primitive because the way in which the letter is entered obliges you to worry about line widths and endings — no fancy wrap-around facilities here. Each item to be printed has a number, column and line number to allow you to print several items on one line. These numbers, and the field names or text, are entered in a table on the screen. As the table only allows 50 characters for each item, in order to print a line of text longer than fifty characters you have to enter two items — one containing the first 48 characters (delimited by quotes) and the second containing the rest of the text line. The example given in the manual is shown in Fig 2 — you can see just how clumsy it is. Certainly it's not easy to find a way to make it simple for the casual user to enter such letters, but surely Ashton Tate could have done better than that!

Report formats of both kinds can be edited, so they may be tried out and then amended to suit. There is a special label-printing feature which allows you to specify the format of your own labels: for instance, labels mounted up to four across the page are permitted. When you come to print your report, Friday! asks you to confirm that the printer is ready before going ahead — a helpful feature. On the other hand, if you abandon printing in the middle, you are returned to the operating system to restart (though no data is lost).

When displaying records on the screen for browsing to view or edit, or when displaying or printing reports, Friday! allows you to set up a retrieval rule which limits the retrieval to records which match the rule. You can have up to 15 such rules stored at any one time for each file. A rule may contain up to 60 characters and consist of field names, constants, comparison operators such as <, > and special operators such as $ (used between two strings to test if the first is contained within the second).

Elements in a rule may be combined with AND, OR and NOT, and brackets may be used to ensure the correct order of evaluation. The syntax of the rule provides an unusually powerful selection tool compared with other packages in Friday!'s part of the market. The scope of the rule may be limited to particular sets of records, on the basis of either the record number (a number assigned to each record in ascending entry sequence), or of the value of the primary key. So, if your invoices are stored in date order and you know that all those you want to process lie between two dates, you can restrict the search to those dates and save a lot of time.

Friday! 'sorts' by producing an index of the fields being used to determine record order and sorting it, a process which is normally much quicker than sorting the whole file, and also saves you having to allow a lot of spare space on your data disk for sort work files. Permanent ordering is achieved by specifying that one or more fields (up to five) should be used to determine the order in which records are to be displayed or printed. If you want a particular report shown in a different order, a temporary sort can be done for that purpose.

Fields are calculated at data entry for storage in the record; this is achieved by attaching a calculation to the field. You can also amend a field definition to attach a calculation to it later, and if you do so Friday! will amend all the existing records accordingly. This feature could be used to raise all prices by 10%, or reset dates in some accounting applications. You can also print the results of calculations in reports, by including expressions as report items.

Only one file can be manipulated at a time (excepting the look-up file facility, which can be used in data entry and in reports). If you need more sophisticated facilities, including tailoring tasks to your specific needs, you can use Friday! files within dBaseII. Friday! uses the same data and index file formats, so you can access files and manipulate them as if they had been created within dBaseII. The manual warns against amending record order within dBaseII, as this could make it

Prompt	Title	Selection	Entry
006	Main Menu	B-Retrieve Data	[B]
200	Retrieve Data	E-Mailing Labels	[E]
200	Retrieve Data	Enter any letter if you don't want default listing	[Z]
009	Data Files	C-Choose a Data File	[C]
013	Data Files	Choose a letter	[ ]
009	Label Format	A-Create	[A]
012	Label Format	Enter Name and Description for your new Label Format	
500	Design a Label	Refer to Prompt 500 instructions to design your label	
500	"	S-Save	[S]
507	Design a Label	Please confirm that you want to save the label	[Y]
600	Search Menu	<RETURN>to Search	<RETURN>
510	Labels	S-Set up Refer to Prompt 510 to change settings	[S]
510	Labels	T-Test Labels P-Print Labels	[T] [P]
502	Design a Label	Add this field to which line 1..5	[ ]
504	"	Jump	[ ]
505	"	Clear Restart which label line	[ ]

Fig 3 Creating a mailing label

Benchmarks

BM1	Time to add one new record	2secs
BM2	Time to select record by primary key	6secs
BM3	Time to select record by secondary key	NA
BM4	Time to access 20 records from 1000 sequentially on 3-character field	13secs/rec
BM5	Time to access record using wild code	14secs
BM6	Time to index 1000 records on 3-character field	14mins 47secs
BM7	Time to sort 1000 records on 5-character field	12mins 24secs
BM8	Time to calculate on one field per record and store result in record	6mins 55secs
BM9	Time to total three fields over 1000 records	2mins 20secs
BM10	Time to add one new field to each of 1000 records	19m 30secs
	Time to import a file of 1000 records	8mins 20secs

Meeting the challenge of change

microbee

Four award winning models to choose from.

microbee really is "meeting the challenge of change" with two ROM and two DISK based models, each guaranteed expandable to any level. From arcade games through to word processing and business management, you can choose the level of development to suit your needs.

Tomorrow looks equally bright for **microbee** owners . . . . with new technology and advances . . . . with reports and literature on new software and peripherals . . . . and with training courses for each level of development.

HI RES Economy Green Screen Monitor

16K Educator $449

Now with WORDBEE!

The **microbee Educator** is specifically designed for the education market and has been chosen by NSW, Western Australia, Queensland and the Australian Schools Commission as a computer for use in schools.

microbee Educator boasts high quality graphics and sound effects capability, exceptional performance at a realistic price plus powerful software designed for Australian curriculum needs.

32K Personal Communicator $499

microbee's top selling portable computer now features:

Telcom 1 firmware WORDBEE, Microworld BASIC, machine code MONITOR, ADM-3A terminal emulation, self-test in 28K of ROM with 32K of CMOS battery backed user memory, high resolution PCG GRAPHICS, SERIAL AND PARALLEL I/O ports, programmable cassette interface and direct monochrome video output.

A vast library of educational, entertainment and utility software is now widely available. Microworld BASIC supports full high resolution graphics, colour if required, music, I/O data can be directed at will and best of all, MW BASIC is a breeze to program yourself.

The low cost BEEMODEM can be added enabling your **microbee** (via your phone line) to communicate with other computers worldwide!

Software and your microbee

Whichever **microbee** model you choose there is a vast and growing array of software to use with it . . . . some built-in and included in the price of your computer. Games and family entertainment, education programmes, communication, information systems, personal and business finance, word processing and much, much more.

Ask at your **microbee** Computer Centre for the current range of software included with each model or ask for the catalogue of "Software Available".

Options and Peripherials

Advanced "State of the Art" peripherals and options are available to interface with your **microbee** such as:

microbee HI RES Economy Green Screen Monitor . . . **$149.50**

Adjustable Monitor Stand **$19.95**

BEEMODEM **$149.50**

microbee MB80 Dot Matrix Printer
- Parallel **$399.00**
- Seriel **$449.00**
- (Includes cables)

microbee High Resolution Anti Glare Ergonomically Designed Amber Screen Monitor . . . **$249.00**

necessary to rebuild the index on returning.

Security & reliability

Friday! uses two levels of password: one on the whole system and the other on individual data files. As to reliability, I had only one small problem when I was creating a Quick report. To do this, you first make any necessary changes to the values provided for printing options (paper length, and so on) and state whether you want totals and/or sub-totals. Then you design the report layout. At any stage during the design step, you can save the report format and edit it later. When I tried this, I was interrupted in the middle and saved the format before I'd entered anything into the design step. When I came back and edited the design to try to add some fields, Friday! crashed with a fairly incomprehensible message and left me back at the operating system level. This was more of a nuisance than a disaster, as I simply started another report and did the design all in one go. But it had me worried until I'd pinned down exactly what was happening.

Friday! can import and export text files; records may be fixed format or may contain fields delimited by commas. The normal file format is the same as that used by dBaseII and can be read by that package.

Files can be copied and deleted. You can also get a directory of appropriate groups of files at any stage, so you don't have to remember file names. The package uses very sensible rules to decide which file should be the 'default' in any particular situation, but you can always override this.

User image

In the main, I thought the approach taken by Friday!'s designers an excellent compromise between verbosity and ease of use. To get to the right part of the package, you use menus; once there, available options are usually shown on the bottom two lines of the screen, so it's possible to use the package virtually without reference to the manual once you have a grasp of its capabilities. If you do need to refer to the manual, your task is made much easier by the fact that on each screen a number is displayed, linking the screen with an explanation in the reference section of the manual.

The defaults used are generally sensible, and you can also change options easily. Confirmation is required only for actions where it would be a real nuisance to make a mistake. There are some good, unusual touches: for instance, when setting up a new file, Friday! displays in the top right-hand corner of the screen the number of characters available for the record — 999 at the start — and decrements this as each field length is entered. The same technique is used whenever the user runs into Friday! constraints (when setting up rules and calculations, for example). When you have finished with a particular task, you always have the option of moving either to the previous menu or to the main menu.

There are, however, a few unfortunate features: the inability to edit a screen format fully, which obliges you to delete several correct field formats in order to correct an early error, is a real pain. It should be possible to abort printing without leaving Friday! altogether, and the report format used to create standard letters is very crude. While these drawbacks are a nuisance, they are quite small problems set beside the mainly excellent features of Friday!'s user image.

Documentation

This comes in two main parts: a tutorial guide to all the basic features, and a reference list of all the screens documented in numeric order. There is also a 'road map' which indicates the range of functions available, a glossary of terms, an index, and a set of 'standard operations'.

The tutorial guide is excellent — a bit chatty for my tastes, but I expect that's just my personal reaction to our US cousins' tendency to folksiness. It uses example files supplied with Friday! to introduce you to each main feature in turn, giving enough information to encourage further exploration without blinding you with detail.

For more information, you use the screen prompt numbers to check the reference manual, which is quite clear too. To carry out a particular task (one of the 'standard operations', for example), you can use the Friday! documentation to help you (see Fig 3, which shows the steps needed to create a mailing label). Note the prompt number beside each step in the operation.

So far, so very good indeed. The difficulty comes when you want to do something which isn't included in the standard tasks, and you can't find anything relevant in the index. As the 'road map' does not show prompt numbers, it becomes difficult to find out exactly where a particular feature is documented in the reference manual. Usually, the best way is to get Friday! going, use the road map to find the part of the package you want, and use the screen numbers to check out what is necessary. Browsing through the more complex features would be much easier if the road map contained references to screen numbers.

Conclusion

Friday! is a straightforward data management package providing basic facilities in a simple way. Provided your records have a simple structure and do not contain long fields, Friday! should provide enough features to satisfy most ordinary data manipulation requirements. If you do need further sophistication, you can graduate to dBaseII and use it on Friday! files.

The package is well designed, apart from a couple of minor wrinkles, and documentation is excellent: it would be even better with the addition of screen prompt numbers to the 'road map'.

At $360, it represents very good value for money.

Summary

Package Type	Suitable for novice users needing basic data management facilities
Strong Points	Powerful selection facilities; good display and report features
Drawbacks	Single file only; crude method of setting up standard letters; limited direct access to records
Ease of Use	Excellent
Error Messages	Mostly helpful
Documentation	Excellent
Costs	$360
Supplier	Arcom Pacific (07) 52 9522

INTRODUCING

Peripherals with Big Advantages...

ATTRACTIVE DESIGNS.. HIGHLY RELIABLE... PRICED TO SELL!

The Slimmer I Disk Drive

The Slimmer I floppy disk drive from Multitech features high quality recording and reliability. Competitively priced to offer higher accuracy and long life performance, the Slimmer I is the answer to the floppy disk needs of the market.

- ★ **TEAC Mechanism**
- ★ **Apple/IBM compatible**
- ★ **164K/320K formatted storage**
- ★ **Direct Drive – higher reliability**
- ★ **Fast positioning**
- ★ **Super slim profiles**
- ★ **MTBF 10,000 power on hours**

ErgoVision Video Monitor

The ErgoVision award winning design represents the latest trends in ergonomic industrial design. Multi position, tilt and swivel feature, stylish housing, high quality resolution available in either green or amber phosphors. Outstanding dot pattern for easy reading, 12" non-glare screen, completely compatible with Apple, IBM and most popular micros on the market today. Resolution of 1000 lines centre, min 800 lines corner.

BDS 12" Data Display Monitor

Ideal for packaging with IBM PC, PC compatible and OEM products. This unit is, perhaps, the lowest priced *high quality* display monitor yet available on the Australian market. Retailing at less than $350 it has to be seen to be appreciated.

- ★ **Used by major American manufacturers and other OEM's**
- ★ **Very high resolution**
- ★ **Synchronization horizontal rate 18.432 KHz; Horizontal retrace 7μ sec.; Vertical rate 47 to 63 Hz; Vertical retrace 800μ sec.**
- ★ **TTL input signals**
- ★ **P42 Phosphor (P31, P4, P39, P134 also available)**
- ★ **RGB monitors available**

Accufeel High Producitivity Keyboard

Convenience at your fingertips...
The Accufeel MAK-II introduces to you a new world of convenient high producitivity computing. In many respects the keyboard is the most important component of your computer since it is your primary input device. The Accufeel plug compatible keyboard, just as the name implies, gives you positive, accurate tactile feedback with the convenience of plug compatibility. Once you plug in the Accufeel computing will never be the same again.

- ★ **90 keys including numeric keyboard**
- ★ **Auto repeat on all keys**
- ★ **Positive tactile feedback**
- ★ **12 user programmable keys consisting of 4 command levels, including preprogrammed DOS commands**
- ★ **10 user definable function keys**
- ★ **Compatible with Apple II and OEM systems**

DEALERS/OEM ENQUIRIES WELCOME. For further information please call or write to Micro Station.

micro station
the Micro Distribution Company

TEL. (02) 290 1514
(02) 449 1853
(AH)

Suite 46, Level 12, 95-99 York Street, Sydney 2000. Tlx 27585.

SOFTWARE

Knowledgeman

After June's Database Benchtest, Steve Withers turns his attention to KnowledgeMan's spreadsheet facilities.

For those who missed the June issue of *APC*, KnowledgeMan is a data management system that includes a spreadsheet. These facilities may be augmented with modules that permit text editing and (on the IBM PC) graphing.

Because KnowledgeMan is primarily aimed at data management, some of the features may seem a bit strange, but I'll do my best to explain how they could be used. Despite that last remark, KnowledgeMan is in many ways a typical spreadsheet. It has numbered rows and lettered columns, you move the cursor around the screen with appropriate keys (by default the package uses Wordstar's cursor control diamond), typing in values, formulae, and text. If you move the cursor off the edge of the screen, the 'window' moves with it, but unfortunately the whole screen is redrawn to achieve this effect — this results in poor performance on Benchmark 1d.

Constraints

The most obvious limitation of a spreadsheet is the maximum number of rows and columns that may be used. While there is an upper limit of 255 in each direction, the real limits are much more restricting. Our Benchmarks are based on a 13 column worksheet and test the programs capacity for formulae, text, and numbers, and in this case the Benchmarks were run on a 256k Sirius. Test 1 (formulae) gave a disappointing result of 42 rows, while 74 rows of text and 76 of numbers were more realistic, but not impressive. 'Capacity' in this review means the largest spreadsheet that could be saved. This is because KnowledgeMan has the nasty habit of not telling you that it is running out of memory — it just says that there is insufficient memory to delete a row or column, save the spreadsheet, or whatever else you try to do to get out of the mess you find yourself in. The only answer seems to be to save your work even more frequently than usual.

KnowledgeMan has the nasty habit of not telling you that it is running out of memory

KnowledgeMan permits the user to specify a maximum sheet size when invoking the spreadsheet module and these tests were carried out on the maximum size (255 x 255), but setting a more appropriate limit of 255 x 13 made no apparent difference. MDBS claims that KnowledgeMan's ability to access data not held in a spreadsheet cell makes it more powerful than a conventional spreadsheet. There is clearly some truth in this, but I believe many people would find these limitations annoying.

Functions and Things

KnowledgeMan has a range of built-in functions, two of which are intended specifically for use within a spreadsheet. These are LOOKUP (to simplify the construction of a look-up table) and SUM (which returns the sum of the contents of a block of cells). These functions are found in most spreadsheets, of course. KnowledgeMan's general purpose functions like ABS (absolute value), LOG, and SIN can also be used when you need them.

Something that sets KnowledgeMan apart from most spreadsheets is that cell contents are not restricted to values, expressisons, and labels — a cell may contain a KnowledgeMan 'program'. The manual suggests two uses for this: the expression of some algorithm that requires a number of steps, and a convenient method for controlling the appearance of the spreadsheet (e.g. to make any negative figures in a particular column appear in red).

This feature is made all the more powerful by the fact that KnowledgeMan is quite happy to treat the sheet as an array — cell #A1 can be referred to as #(1,1), and the subscripts can be variables, expressions, or constants. This makes it easier to write 'programs' inside cells to carry out some function involving references to a number of cells, such as the net present value of a stream of payments.

Security

It may seem peculiar to talk about security when describing a spreadsheet, but it is especially relevant when the program is backed by a data management system. When KnowledgeMan is started, it requests and checks for a username and password. These are controlled by a 'super-user' who is also responsible for assigning the appropriate access codes to each user. In the case of the spreadsheet subsystem, the builder of a sheet can choose to PROTECT certain cells so that alterations may only be made by users with the access codes he or she specifies. Similarly, CONCEAL hides the contents (but not the value) of a cell from unauthorised examination. At least some of the data in the spreadsheet would normally be drawn from a database to which access is controlled in much the same way.

These features would be an advantage in situations where spreadsheets are developed by experienced users for their colleagues, but they are of little interest in an environment where everyone does their own thing, keeping their disks locked away.

Something that sets KnowledgeMan apart from most spreadsheets is that cell contents are not restricted to values, expressions, and labels — a cell may contain a KnowledgeMan "program".

Using the spreadsheet

As always, there is good news and bad news — I'll tell you about the good stuff first.

The idea of integrating a spreadsheet with a DBMS is a good one. There are many occasions where spreadsheet analysis can usefully be applied to data already stored in the system, and this approach saves the chore of converting files from one format to another. If KnowledgeMan looks after your financial records you can play the "what if?" game when drawing up a budget without having to type in the current figures.

Spreadsheets can be linked through the database, which ensures that the data remains consistent throughout the system. Where necessary the definitions used in a portion of a saved spreadsheet may be combined with the current one. Given the relatively small size of the spreadsheet, it would be an advantage if one sheet could reference the values held in another without the need for an intermediate data table.

Most of the features we have come to expect in spreadsheets are present in KnowledgeMan. Going quickly through Mike Liardet's checklist (*APC* April 1983), the features I haven't covered explicitly are row and column insertion and deletion, variable column widths and display formats, formula printout, formula editing, automatic or manual recalculation, long jumps, and searching and logical functions. The shortcomings are poor configuration facilities (in terms of making the best use of a particular system), no graphics (except as an optional extra for the IBM PC), indifferent instructions for the first-time computer user,

I have two main complaints about KnowledgeMan's spreadsheet. The first is that several operations seem to take forever. The creation of the test spreadsheets involves copying blocks of cells, and it was slow to the point of being painful. It's made worse by the fact that cells must be referenced absolutely, so when you do a move or copy, KnowledgeMan asks whether each reference is to be treated as an absolute or relative reference, and then makes any necessary adjustments. The difference may be shown in an example. If cell A3 contains the expression A1+A2 as absolute references, and you copy the column A into column B, then B3 will contain A1+A2. Using relative references, the copy command will interpret A1+A2 as the sum of the two cells above the current cell, and so B3 will contain B1+B2.

The other problem is the 'dumb' way in which KnowledgeMan accesses the and no split-screen facility.
screen and keyboard. Unless you are using an IBM PC, KnowledgeMan uses character-by-character output which is disastrous for spreadsheets and explains the poor performance on Benchmark 1d. If you are accustomed to a spreadsheet that takes advantage of memory-mapped video screens, you won't enjoy this one. There seems to be a minor bug lurking here — when the display window scrolls, values sometimes appear in empty cells (ie, not all cells are erased properly). I saw an earlier version of KnowledgeMan that had the opposite problem, in that cells were erased if they were already blank.

The keyboard problem is that KnowledgeMan throws away any characters you type while it's doing something. This becomes especially noticeable when using the spreadsheet due to the time taken to update the screen.

Documentation

There's not much to add to my comments in the database Benchtest. What's there is ok, but the system cries out for the tutorial manual that I still haven't seen.

Conclusion

My feelings about KnowledgeMan's spreadsheet are best expressed like this: if you want a really good spreadsheet then look elsewhere, but when viewed as an accessory to a very flexible data management system it could be useful, although it can also be aggravating. An obvious comparison is between KnowledgeMan and 1-2-3 — if database management is your prime concern, go for KnowledgeMan, but if you spend most of your time spreadsheeting then 1-2-3 would be a better choice.

Supplier: Database Management Services, PO Box 62, Brighton, Victoria 3186. Telephone: (03) 523 5947.

Benchmarks

See "Which Spreadsheet", *APC* April 1983.

1a	(expression capacity)	42 rows
1b	(integer arithmetic)	1 min, 20 sec
1c	(real arithmetic)	1 min, 20 sec
1d	(scrolling)	40 sec, 15 sec

NB KnowledgeMan claims 14 digit numerical accuracy

2	(text capacity)	74 rows
3	(numeric capacity)	76 rows

Practical Pick

The Pick operating system is not simply a collection of base functions, but one of the few complete software offerings providing all the requirements for the user — a full database management system, enquiry and programming languages, editors and utilities. In part one of a two-part feature, David O'Byrne examines this operating 'environment'.

The term 'operating system' is increasingly used to refer to those elements of software that handle the base functions of a computer system: the routing of I/O to the various peripherals on the system, or the file access mechanism, for example. These functions are effectively transparent to the user in that as long as they operate successfully, the user need not be concerned with their precise mechanics.

However, the user will need to be familiar with those software features which provide a platform on which to build the applications — things like utilities, programming languages and editors. These features tend to be added on to the base operating system as extras, in some cases a variety of different products (and suppliers) being combined to provide all the required functions.

Viewed from this angle, the Pick operating system is badly named. It could more correctly be termed an 'environment', providing a completely integrated set of facilities.

Pick, as shown in Fig 1, is something of a sandwich. At the bottom are the 'base functions', the parts of the operating system which carry out those hidden jobs — managing the virtual memory system, or maintaining the variable length data structure. Above these are the intermediate functions, system-wide facilities providing a security system from initial logon down to file retrieval, update locking mechanisms, and data dictionary facilities for each and every file within the system, plus accounting statistics, maintained by the system and enabling the user to see at a glance the current state of the database.

Above these, and interfacing to the user, are the final tools, ranging from powerful word and text processing options, through programming and procedural languages, file enquiry/report formatting facilities and a comprehensive set of utilities — a computing environment.

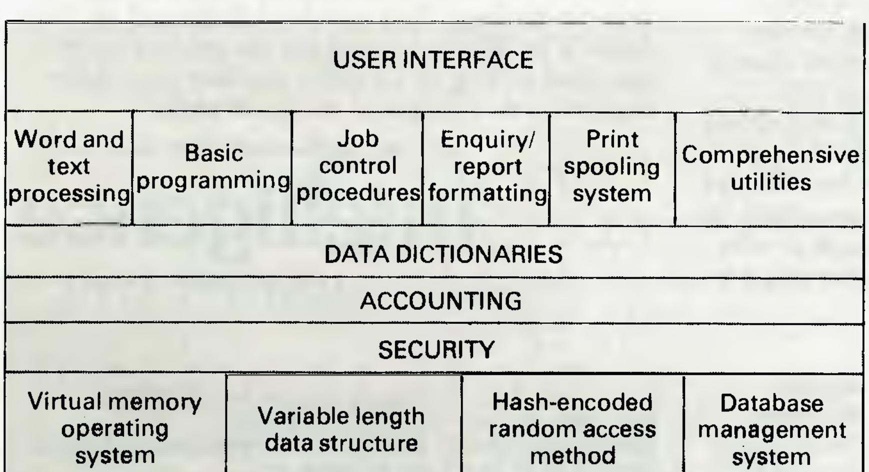


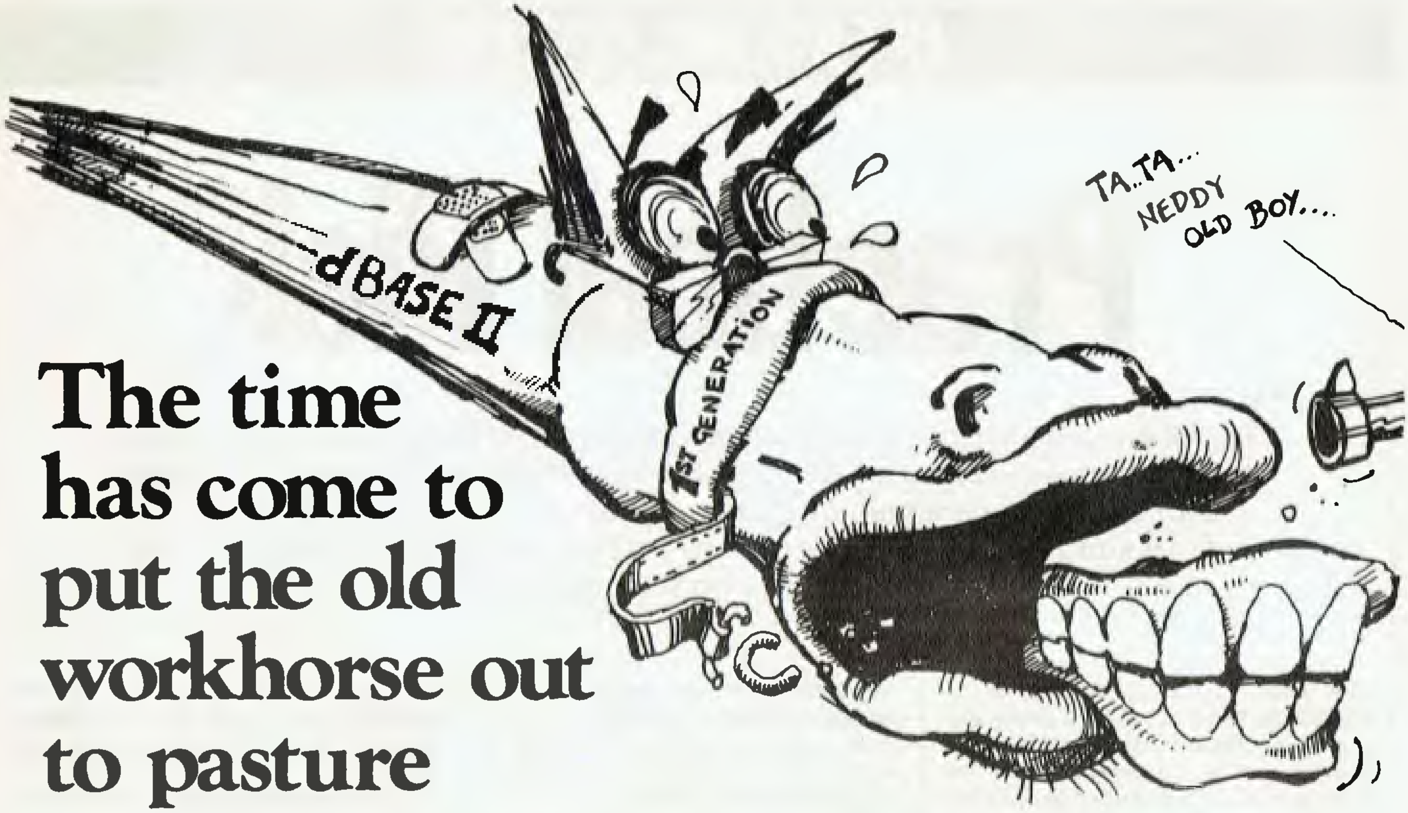
Fig 1 The Pick sandwich

Base functions

Here we see a complete virtual memory system which handles system software, user software and data, with the entire disk being seen as an extension of main memory. The virtual memory manager ensures that programs and information for active users are transferred from disk to memory as required, and that inactive users take up hardly any memory at all. This allows a minicomputer with a relatively small amount of memory (128k) to run as many as 12 active terminals at one time, although response will vary according to system performance and the mix of jobs being run.

All information within a Pick system is held in a completely variable length format, every byte on disk being held in records composed of variable length fields, sub-fields and sub-values. This scheme provides a very high degree of independence between data elements, as well as highly efficient disk storage utilisation. A simple change to the type or size of a field can be effected without having to restructure the file in any way — the records expand and contract according to the size of the actual data.

The third basic function provides a hash-encoded random access file access mechanism. Record's keys are passed through a highly efficient algorithm which provides the address of an area on disk where the record is to be



The time has come to put the old workhorse out to pasture

The days when dBASEII* stood all by itself at the apex of the database management pyramid are past. The old workhorse has been put out to pasture by a new breed of faster, easier, more powerful packages.

It's not that the old horse hasn't done its job: it has, gloriously. Those of you using dBASEII know that it has paid for itself many times over. In its time it was dBest. And now there's no denying that it's slow, it's cumbersome, it's difficult to live with — when you compare it with some of the newcomers on the market.

Fresh out of the starting gate.

So who are the newcomers that have usurped the old master's position?

Delta:

We regard it as the most complete user-oriented database system on the market today and for the forseeable future. Delta caught our attention when it received the UK government's prestigious RITA award for "best software of its kind", followed closely by IBM UK's adopting it as their own distributed product for the IBM PC. Everything we have learned about it since working with it here has served only to increase our enthusiasm. Delta has so much to offer — and you don't have to be a programmer to get it all.

Dataflex:

Dataflex is a true multi-file, multi-user database management system. "Flex" offers you unlimited flexibility in the development of high quality, easy to use database applications – in a minimum of time. "Flex" is for software developers everywhere, as well as for users with complex applications.

Take some advice from experienced punters.

"Dataflex . . . truly the best application development tool on the market today. Its power and flexibility have enabled us to write a total package in less than 50% of the time compared to conventional methods."

Tim Lamberton, M Sc. software consultant to Occidental Life.

"Dataflex allows the developer to devote his energies to good systems design because the nitty gritty of screen, file and report routines are superbly handled by Dataflex. The end user has the knowledge that his system can be easily modified and further enhanced."

Jeff Hooper Talon Systems

Place your bet on a winner.

Delta and Dataflex are available at leading computer stores everywhere. Ring your local dealer and ask for a demo. If he doesn't know about the product you're interested in, ring us. Or better yet, have your dealer ring us. It's time he got on the inside track.

Delta and Dataflex distributed in Australia by

Intelligence

(AUST. PTY. LTD.)

Sydney: 4th Floor, 204 Clarence Street, Sydney, NSW 2000. Phone (02) 267 1711. Telex: MNC AA25026.
Melbourne: Suite 303, 620 St Kilda Rd., Sth Melbourne Vic 3004. Phone: (03) 51 1406 Telex: AA39219.
Brisbane: (07) 343 9122 **Perth:** (09) 322 1677

*copyright Ashton-Tate.

stored; distribution is even and response times high. Additional file space is assigned dynamically as the file grows (and removed if it shrinks), making the system self-maintaining apart from a weekly or fortnightly reorganisation.

Database

The Pick relational database management system can be classified as another base function although, as with most database systems, the interpretation of which elements form the DBMS and which are separate is open to argument. One feature of the system is its suitability for the concatenated keys which form such an essential part of the relational model. These are handled quite happily by the file access system, and can easily be broken down for retrieval purposes into their component parts via data dictionaries.

Within any computer system the security of data is vital, and particularly so in one which prides itself on its ease of use and speedy data retrieval labels. For this reason, there are a number of ways in which the Pick user can restrict the flow of information without impairing service. The first line of defence is the usual 'logon password' requirement, one of the most under-estimated security tools. Stop unauthorised access at the front door, and the chance of a burglary is reduced substantially!

Once logged successfully on to the system, the user is allocated a privilege level which determines the powers available to him. A user of the highest privilege may look at data in different databases, create programs and adjust system parameters. The lower level users are restricted in their ability to change pre-defined commands or use advanced facilities (mag tape, debug, assembler, and so on). The intermediate privilege level allows command alteration and use of the mag tape, but prohibits use of data security utilities, assembler, and debug.

These are system-wide settings and are intended as general options. For more specific protection, sensitive commands can be removed altogether, the user tailoring the available commands to suit the requirements of the relevant user department. Powerful commands and facilities are restricted to those parts of the system used only by those in overall charge.

In addition, individual data, program or parameter files can be retrieval and/or update locked. This restricts access to users with retrieval or update keys which match the lock codes attributed to the relevant file.

Another system-wide feature is the accounting statistics, produced by a variety of tools, and showing the organisation of individual files and records up to a full picture of the system as it stands at any point. User logon and logoff times are automatically recorded, together with an indication of the amount of use they have made of the system.

Data dictionaries

As Pick is a database system, it makes use of data dictionary facilities on every file in the system. The enquiry facility operates via these dictionaries, and allows the interrogation of any file on the system with the same standard techniques.

Data dictionaries are a fundamental part of the Pick system. When a file is created, it is formed in two parts — data and dictionary. It's therefore possible to use Pick's enquiry language on any file in the system, whether it contains data, programs, PROCs or system data; all can be quickly and easily interrogated.

Data within a Pick system is usually stored in its raw form and expanded on output, with elements of the database frequently being concatenated or modified to fit a particular format. These functions are performed via dictionary conversions — operations carried out on a piece of data to transform it into output form.

Conversions take many forms. They can specify data extraction in either text, group or character form; elements of the data field can be extracted by direct reference. For example, T[2,3] returns the characters in positions 2 to 5 of the data fields by group extraction: that is, the code G2$1 returns '3' when processed on the field '1$2$3$4' or by character extraction, where only alphabetic or numeric characters are returned.

The dictionary can specify that output should be pattern, length or range checked, returning only those data elements which fit a specific range of values, match certain patterns or are of a specified length or range of lengths.

Data formatting can also be achieved (with codes enabling the user to convert dates and times from internal to external format or *vice versa*), to convert character strings to upper or lower case, or from their ASCII representation to the hexadecimal equivalent. In addition, numbers can be scaled, justified and formatted via decimal masking routines.

Mathematical functions and expressions can be carried out, albeit in Reverse Polish Notation. These operators permit the use of constants, literals, system parameters and substrings. Arithmetic operations and relational operators are fully supported, together with commands to alter the make-up of the stack itself.

In order to simplify these operations, a second mathematical notation is supported within Pick which allows the user to specify the functions in a much simpler syntactical form (similar to Basic), but with the added advantage of referring directly to other dictionary items, thus permitting relational algebra to be coded simply and efficiently.

These dictionary functions can make use of a file translation correlative to incorporate details from fields held in other files, which can then be processed via conversions to provide full relational operation.

Tools

One of the most obvious requirements is for a programming language, and in the Pick's case, this is a derivative of Basic. This has been modified in a number of areas to make full use of the features of the operating sytem. For example, the handling of variable length strings of up to 32k, magnetic tape, dynamic arrays, pattern matching, external subroutine calls, full screen handling with sophisticated cursor control, and so on.

Programs are input via the standard line editor and then compiled. A number of interesting compile options are available, including the ability to output a listing of the program complete with opcodes and the 'pseudo' assembly code generated by the compiler. The generation of variable and statements maps, so that the programmer can accurately monitor the execution of the program.

Data is represented within the program either as a variable or a constant, with variables containing single or multiple values (arrays), arrays being either dynamic or dimensioned and one (vector) or two (matrix) dimensional. Statements enable the programmer to locate, extract, replace, insert and delete data held in either a vector, a matrix or a variable.

Programs may pass 'common' data elements between them, and may call either internal or external subroutines. They may initiate any other process, be it a PROC, an enquiry language statement, or a utility. Control structures available to the programmer include CASE, FOR NEXT loops, IF THEN ELSE statements, LOOP UNTIL/WHILE and ON GOTO or GOSUB.

One of the most significant enhancements to the Pick Basic is the facility to convert data quickly and easily from one form to another. This is achieved by means of the ICONV and OCONV functions (Input/Output CONVersion).

Conversion

When creating the Pick database, the user is encouraged to store data in its 'raw' format, without decimal points, unnecessary fillers, leading zeroes, and so on. In order to illustrate how the data can be changed into the desired output form, let's take a look at a date conversion. On a Pick system, dates are usually held in 'internal' format: that is, a number representing the number of days since 31 December 1967, so that 27 October 1983 is held internally as 5779, making date calculations a simple arithmetical matter.

Similar conversions are available for a variety of operations, including formatting numeric fields, time (held as the number of seconds since midnight), converting decimal to hex (and *vice versa*), converting to and from packed fields, translating codes to values from other files, or calling user specified assembly routines for special tasks.

Another feature of Basic is its symbolic debugger (available only to users with the highest privilege level), which allows the user to interrupt the program either immediately it commences or at any subsequent time. Once within the debugger, the programmer may 'step through' the execution either in single or multiple steps, temporarily halt on execution of a specific line, or when certain logical conditions are met.

In addition, the programmer may display and/or alter any variable(s), display the actual source code lines, or alter the execution of the program by directing it to continue at a different point. Output can be directed to either screen or printer, and specific variables can be displayed whenever a break in execution occurs.

Advantages

The advantages of such a tool in debugging large programs are very real.

The programmer can identify and subsequently avoid bugs, alter data to test different routines, and investigate live problems without having to set up copy programs, data, and so on.

Next month: an extensive look at Pick commands, and its PROC (stored procedure) language.

Cash & Carry Com

IBM

	Our Price	R.R.P.
Symphony	875.00	995.00
Lotus 123	595.00	895.00
Spell Binder	492.00	695.00
Supercalc 3	367.00	445.00
dBase II	595.00	825.00
Multimate	495.00	595.00
Corporate MBA	1020.00	1200.00
Wordstar Propak	575.00	795.00
Open Access	675.00	940.00
Quickcode	300.00	380.00
Flight Simulator	69.95	87.95
Cardbox	279.00	328.00
Ada	360.00	420.00
C-Compiler CP/M86	315.00	385.00
Fortran IV	499.00	578.00
Turbo Pascal	175.00	187.50
Single User Unix	1987.00	2392.00
Multi User Unix	2499.00	2925.00
Revelation	1499.00	1762.00
Knowledgeman	645.00	895.00
Hercules Graphics	595.00	938.00

FOR ALL SOFTWARE REQUIREMENTS — CALL US NOW FOR SUPER SAVINGS!

Abstat	599.00	742.00
Stat pac	577.00	637.00
Pascal MT +86	585.00	725.00
Concurrent CP/M 86	508.00	610.00
C-Basic 86	314.00	395.00
Basic Program Tools	295.00	370.00

commodore

	Our Price	R.R.P.
Commodore 64	385.00	499.00
Commodore 5X-64	1299.00	1499.00
Commodore Vic 20	224.00	229.00
1541 Disk Drive	385.00	499.00
Datasette	41.95	49.95
802 Printer	429.00	499.00
1701 Colour Monitor	429.00	499.00
Koala Pad	138.75	149.95
Multipan	128.00	149.00
The Manager	83.15	100.00
Easy Script	83.15	100.00
Home Accountant	87.95	105.00

ATARI

Atari 600 XL	329.00	399.00
Atari 800 XL	545.00	599.00
Atari 1050 Disk Drive	625.00	699.00
Atari 1010 Program Rec	147.00	159.00
Atari Koala Pad	137.00	159.00
Atari Writer	103.95	119.95

KAYPRO

Kaypro II
Kaypro IV
Kaypro X

Call For Top Value

MAJOR BRAND COMPUTERS CALL NOW FOR TOP VALUE!!!

TeleVideo

Televideo PC	4695.00	5445.00
Televideo PC XT	7475.00	7975.00
Televideo Portable	4099.00	4444.00
Lotus 123 with each Televideo	400.00	895.00

Cash & Carry Computers

puters & Software

OSBORNE

	Our Price	R.R.P.
Osborne 1	1890.00	2180.00
Osborne Executive	2690.00	3180.00
Osborne Drive C	1149.00	1200.00
Osborne Trantor 10mb	CALL US	
Osborne Comm pac modem	FOR TOP	
Osborne Web Network	PRICES	
DISK DRIVES		
ACT Hard Disks	Call	Call
Tallgrass Hard Disks		
12.5 mb w/ Tape backup	4645.00	5161.00
20 mb w/ Tape backup	5395.00	5994.00
35 mb w/ Tape backup	7122.00	7913.00
IMB PC Disk Drive	395.00	635.00

apple

Apple CED Card par/ser	97.50	135.00
Microsoft Z-80	422.00	545.00
Digitek Printmaster	128.00	202.00
CP/M + Wordstar + 64k.	642.00	745.00
Homeword	89.00	99.65
Zardaxx	256.00	324.00
PFS File Graph Report	146.00	175.00
dBase II	595.00	825.00
ALD System II	182.00	215.00
Microsoft Fortran	279.00	329.00
Microsoft Basic Compiler	559.00	659.00
Uspeed	698.00	897.00
RGB for II+ & IIe	142.00	162.00
RGB & 80 col for IIe	189.00	234.00
Accelerator II (for II+)	811.00	990.00
Perfect Writer	235.00	450.00
5 mb Apple III	2759.00	3456.00
10 mb Apple III	3599.00	4554.00

PRINTERS	Our Price	R.R.P.
Amust	356.00	449.00
C-Itoh 8510 par	993.00	1340.00
C-Itoh 1550 par	1245.00	1635.00
Star Gemini 10x	420.00	599.00
Star Gemini 15x	775.00	955.00
Star Delta 10x	775.00	955.00
Star Delta 15x	998.00	1434.00
Olivetti Praxis 35	650.00	875.00
Olivetti Praxis 40	775.00	995.00
Brother HR-15 par	675.00	895.00
Brother HR-25 par	1035.00	1345.00
HR-15/25 tractor feed	108.00	189.00
HR-15/25 Sheet Feeder	248.00	449.00
MODEMS		
Cicada 300	199.00	250.00
Cicada 300T	228.00	275.00
Cicada 300 (Commodore)	213.00	275.00
MONITORS		
Kaga Green IBM	307.00	327.00
Taxan Kaga Green	346.00	306.00
Taxan Kaga Amber	251.00	311.00
Roland RGB	872.00	995.00
Roland 8 Pen Plotter	1286.00	1395.00
Roland 1 Pen Plotter	795.00	895.00

& Software

P.O. Box 695
DARLINGHURST 2010
(02) 331-3888

If you know what you want. Why pay more. Phone for products not listed for our exceptional prices.

MasterCard bankcard welcome here

Prices subject to change without notice.

Divine laws?

Ever wanted to take the law into your own hands to protest about the lack of definition in computer information? Martin Banks has. He dons a ministerial cap to define his Square Laws of Information Processing.

Legislation and stuff like that is what governments are for; but even though privatisation is the in-word of the decade, there are some things which should remain state-owned. Making laws is definitely one of them.

This obvious fact notwithstanding, I have decided it's time that more legislation, or laws at least, are brought before you; laws that will hopefully clarify and organise your thought processes and help restrain deviant attitudes. These laws I have collectively defined as the Square Laws of Information Processing.

Now, before you yawn and wander off to the pub, these are not the normal type of computing laws that are bandied about. They are (hopefully) a little different. This stems from the fact that information is different from computing, and that the latter is only the most convenient tool for manipulating and storing the former. The trouble arises because the computer is so damned good at these functions that it has created a situation in which information is devalued to the point of worthlessness. There is just so much potentially available that people are fast approaching the point where they know not what to do with it.

So what we need is a set of ideas, thoughts, epigrams — laws, even — by which we can define our attitudes to the Vast Gobs of information that the dreaded computer is making available to us.

The first law that occurred to me is the following: The amount of understanding anyone has of what is happening in a particular subject area is inversely proportional to the square of the amount of information available on that subject.

This is the most obvious because it's the most fundamental. Information creates misunderstanding. The potential for this increases each time more information is made available, until a point is rapidly reached where no coherent understanding is possible at all. The real danger of this particular law, the one to which it points, is that for many people there exists the illusion that understanding actually increases with the availability of information. But this is a fallacy based on the assumption that all the information is going to somehow 'point' in one uniform direction, so that 'understanding' becomes directional and therefore obvious. As all information is, in its individual form, totally unique, each item will automatically be different from every other. Murphy's Law proves that in any given real situation, each item of information will be directly contradictory to all the others (Law One, Sub-Section One). Confusion and misunderstanding are automatic and inevitable.

The second law is more specific in nature, though it can be broadened to meet more general requirements. It is as follows: The number of mistakes made in a document rises in direct proportion to the square of the number of editing facilities available.

This obviously has its roots in word processing. It's a law that will be familiar to anyone who uses a word processing system with any degree of regularity, and can still remember the days when they used an ordinary typewriter (preferably manual). Though mustaeks, sorry — mistakes, are made with the typewriter (a fact which prompted the use of computers to edit text), it can be shown that the problems caused by correcting those errors manually created an environment of some respect for the tools being used. A certain amount of thought was given over to the task of hitting the right keys.

With the advent of computerised editing, however, the situation changes. Because it becomes so easy to change individual characters, words, sentences or whole paragraphs, two things happen. Firstly, the users promptly become careless and don't try, thus making more mistakes than before. This counteracts any savings in editing time available with a word processor. The users also begin to show tendencies of suffering from verbal diarrhoea because it becomes so easy to write lots. The end result is that more work is created than is actually needed, and in a more inconvenient form than before. This last statement refers to a personally-defined phenomenon of word processing. It's not unknown for me to write something and then hand-deliver it to the publisher while on a trip to the city. The copy (in paper form) can easily be read and corrected while travelling on the train. It may not look pretty, but presentation is usually the least of a writer's or publisher's worries. Try the same trick with a North Star Horizon, terminal and daisywheel printer and see if the conductor can help you.

The third law is similar to the second in that it helps define an area of information surplus, though it specifically considers a different major area of business computing use. It is as follows: The size of the spreadsheet is at least 2.5 times the maximum level of comprehension of the person creating it, and incorporates at least twice the number of column and row headings as are actually required.

This is fairly self-explanatory, for it defines another area of the overall principle: something designed to make life easier actually makes it more complicated. The ability of a spreadsheet to define a multiplicity of different formulae with which to massage business figures means that every business person is obliged to try them all, just in case. One of the best marketing ploys ever developed was the selling of spreadsheets to business people on the 'What if . . .' fear. They don't know, but their job might depend on such knowledge, so they buy it just in case. One day, someone will come up with a 'What if you run out of What If functions' function designed specifically for such paranoias, though two far more useful ones would be 'What The Hell' and 'Does It Matter Anyway?'.

The last law, though obviously there are bound to be more, is as follows: The number of jargon words used to discuss a particular information processing subject is inversely proportional to the square of the misunderstanding of that subject.

This is a more tightly defined version of the old 'blind 'em with science' epigram, and is one which can best be seen in use by politicians, journalists, and senior directors of computer companies.


LAWS
INFO PRO.

SOFTWARE

PlanStar

Mike Liardet looks at MicroPro's PlanStar, a comprehensive and versatile spreadsheet aimed at businessmen, accountants and financial managers.

MicroPro is one of the world's best known software producers, famous primarily for its blue ribbon word processor, WordStar. In fact, the huge success enjoyed by WordStar has eclipsed the rest of MicroPro's product line, but over the years a substantial number of other products have been added to the portfolio. Most of them are easy enough to spot in the catalogues, as they are nearly all named with the 'Star' suffix. Some of MicroPro's later products are adjuncts to WordStar, such as mailing and spelling facilities, and others like InfoStar are fairly unrelated.

Following this long succession of rather 'quiet' products, MicroPro is now hoping PlanStar will take over on the centre-stage. PlanStar is a financial modelling-cum-spreadsheet system, available for CP/M, MS-DOS and PC-DOS systems. It's not MicroPro's first foray into the world of spreadsheet systems: CalcStar has been available for some time, and while not breaking any records it has proved to be a solid and workable system. But PlanStar is far more ambitious.

A quick glance at the checklist of PlanStar's features should set any accountant's heart a-flutter: it provides multiple spreadsheets, consolidation, sensitivity, analysis, equation solving and graphing. All this is in addition to the more mundane(!) facilities, such as help screens, solid documentation, and tutorial software.

PlanStar is presented as a substantial manual with four diskettes, and is fairly daunting at first. You work your way backwards through the manual: installation instructions come at the end, preceded by tutorial material, with the less immediately needed reference material, accounting for a whole three quarters of the volume, right at the front.

Setting up working disks involves the usual ritual of operating system commands. In the PC-DOS version there are specific instructions for doing this. With four disks the entire operation has to be repeated four times. There are two reasons for the considerable volume of software in PlanStar: 1) PlanStar really is packed with a lot of features; and 2) it is implemented in compiled Microsoft Basic, which is notorious for creating large code files.

One disk contains purely tutorial software, the remaining three are for the system. These include no fewer than 15 sub-programs, five language-specific text files (only these need be changed to produce a non-English version), and eight sample models. If you have larger capacity (320k +) disks you can halve the disk count. MicroPro also guardedly advises that the system will work with most of the hard-disk variants of PC-DOS. I used PlanStar across four disks throughout the Benchtest, but this involved rather too many disk changes for my liking.

The idea of blending traditional financial modelling with the newer style of spreadsheeting is not new and most authors attempt to dismiss pure spreadsheet systems as 'not serious', or 'something to be outgrown'. PlanStar is no exception. The first page of the manual provocatively dismisses pure spreadsheet systems because 'a successful plan for a modern business can no longer be expressed on a single piece of paper'.

The best way to get a feel for things is to start with the tutorial disk. This is a separate system entirely, with the sole purpose of teaching the new user. It's not simply a set of 'sample models', but purpose-built software introducing all the major features of PlanStar in eight lessons.

The tutorial software contains very good ideas spoilt by some frustrating aspects to the user interface, but it should be emphasised that MicroPro is almost unique in providing any tutorial software. In spite of my reservations, I did find it more useful than no tutorial at all, which is the norm with most systems.

It's quite clear that MicroPro decided that the way to gain the attention of an indolent user is to get him typing. Many of the lessons involve transcribing material from the lessons section of the manual to the screen. I find this type of thing quite pointless, and somewhat reminiscent of the 'doing 50 lines' punishment from my schooldays.

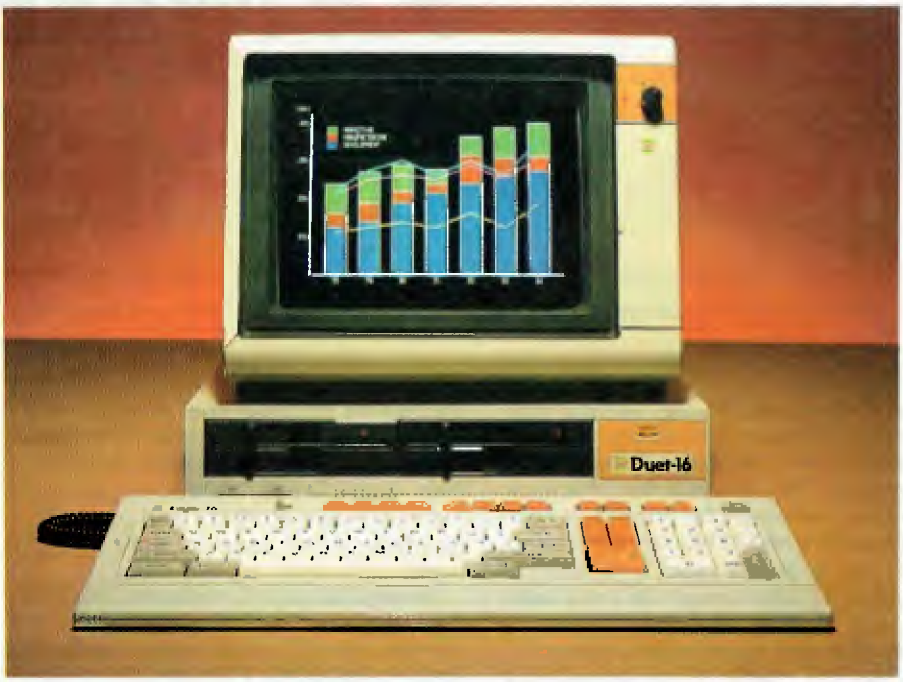
Occasionally there are discrepancies between the manual and the screen which cause a good deal of confusion. I avoided all this bother once I discovered that the '?' key would do the typing 'automatically' and then encountered the next irritation — the screen handling is very slow.

I presume that this slowness is deliberate as the 'real' software oper-

POWERFUL NEW BUSINESS CREATIONS IN AN AFFORDABLE PACKAGE.

PANAFACOM, Japan's best known maker of minis and micros, together with Sakata, brings you truly high-tech in a 16-bit package.

- **Processors:** 8086 main CPU (8MHz) with optional 8087 co-processor
- **Expandable user memory:** up to 512K bytes
- **Large external storage:** dual 720K-byte minifloppy drives plus 10M-byte to 40M-byte hard disks (optional)
- **Powerful operating systems:** MS-DOS™ Ver. 1.25 standard: CP/M-86™ optional; MS-DOS™ Ver. 2.11 available with hard disk support and IBM PC file access capability.
- **Versatile communications:** two RS232C ports, a centronics port, and an optional IEEE-488 port, IBM 3270 emulator software.



See the Duet-16 today.

SBC Duet-16™

You will be amazed at its sharp, clear color graphics. And with the full range of user software packages and programming languages—Multiplan™, WordStar™, MailMerge™, SpellStar™, TSS support, BASIC-86™, Advanced BASIC, L-II COBOL™, FORMS-2™, ANIMATOR™—the Duet-16 is ready to work for you anywhere.

MS-DOS, BASIC-86 and Multiplan are trademarks of Microsoft Corporation. WordStar, MailMerge and SpellStar are trademarks of MicroPro International Corporation. LEVEL II COBOL, FORMS-2 and ANIMATOR are trademarks of Micro Focus Limited. CP/M-86 is a trademark of Digital Research Inc. SBC Duet-16 is a trademark of SAKATA SHOKAI LTD.

We welcome distributors/dealers inquiries *Call or write us for more information*

Supplier:
SAKATA SHOKAI, LTD.
International Division:
6-2, Higashi-Temma 2-chome, Kita-ku, Osaka, 530 Japan
P.O. BOX 597 CENTRAL OSAKA No. 530-91, Japan
TEL: (06) 356-2160 TLX: SAKATA J 63397, SAKATA J64412

Our Sydney Office:
SAKATA SHOKAI, LTD.
29 Bertram St., Chatswood, N.S.W. 2067
P.O. BOX 293 CHATSWOOD 2067, Australia
TEL: 412-1609 TLX: AA25131 SKT

Manufacturer:
PANAFACOM LIMITED
National Bldg., 17-15, Shinbashi 6-chome, Minato-ku, Tokyo, 105 Japan TEL: (03) 438-0311

ates more quickly, but it becomes rather frustrating, particularly when you feel that you've got the message and want to pass onto the next thing. Possibly aware of this, Micropro has included a fast automatic presentation where you type nothing but just watch. Unfortunately, this turns out to be too fast and completely unstoppable. 'Oh for the days when tutorials were on paper', I hear you say. I don't agree, but software tutorials should be adjustable to your own absorption rate and PlanStar misses the boat in this respect. Just the addition of a 'speed-up' keystroke with the normal presentation would have improved things immensely.

The actual content of the lessons is very good, and the review questions at the end of each lesson are useful. Generally these involve questions on the correct order for placing commands, True/False problems, and multiple choice options. I found them particularly taxing, and they really did test the amount I had absorbed, while simultaneously teaching me the bits I had missed.

In short, in spite of my reservations most of my initial learning of PlanStar was achieved with the tutorial disk. With only a few minor improvements to the interaction I would be ecstatic about it instead of griping!

```
PS
Project name? DEMO
100 ROWS Sales Costs Profit
120 COLUMNS January February
  March Quarter
140 Profit = Sales – Costs
160 Quarter = sum of January to
  March
CALCULATE
Sales? 240 260 290
Costs? 160 175 200
END
```

Fig 1 A short introductory session

	January	February	March	Quarter
Sales	240.00	260.00	290.00	790.00
Costs	160.00	175.00	200.00	535.00
Profit	80.00	85.00	90.00	255.00

Fig 2 The resulting model

	1984	1985	1986
Units	50000.00	5500.00	5600.00
SalePrice	5.00	5.00	5.00
Sales	25000.00	27500.00	28000.00
Costs	17500.00	19250.00	19600.00
Margin	7500.00	8250.00	8400.00

Fig 3 Sales model

Initial Values are			
Sale Price	5.00	5.00	5.00
and			
Margin	7500.00	8250.00	8400.00
a – 5% change in			
Sale Price	4.75	4.75	4.75
results in			
Margin	6250.00	6875.00	7000.00
a 5% change in			
Sale Price	5.25	5.25	5.25
results in			
Margin	8750.00	9625.00	9800.00

Fig 4 Sensitivity of Margin for 5% change in Sale Price

Target figure: Margin = 10000

	1981
Selling price	5.50
Margin	10000.00

Fig 5 Seeking goal of 10000 margin (sales model)

In use

To get the real PlanStar running, place disk one in the current disk drive and type 'PS'. Fig 1 shows the shortest session you could possibly have while still actually doing anything. After PS you must supply a model name, PlanStar's only acknowledgement that there is a disk filing system for storing models. All further interactions with the disks are automatic and once you 'END', all changes and additions are recorded automatically.

The outcome of this mini-session is the display in Fig 2. This is built up by just four lines numbered 100 to 160 (Fig 1). The order in which you type lines is irrelevant (they are always ordered by the line-number) and one way of correcting a line is simply to retype it, overwriting the original line with that number. To confirm what you have already entered, type 'LIST'; anyone who has ever worked in Basic will find familiar ground here. The 'ROWS' line specifies that there are three rows in the model; the 'COLUMNS' line specifies the number and names of the columns in like fashion. The other two lines constitute the 'logic' of the model, showing how the Profit row and the Quarter column are to be derived.

All lines starting with a line number are noted by the system as they are typed. If no line number is present, the line is acted upon immediately: this is called an 'immediate command'. CALCULATE is an immediate command, causing a sequence of operations to be performed. CALCULATEing takes two changes of disk for the four-disk version. The syntax of the stored lines is checked, the calculation is made, and finally the display is generated. The first two rows of the spreadsheet (Sales and Costs) are undefined when CALCULATE is started, and values are requested for them accordingly. These values could have been included as part of the model by entering the following before CALCULATEing:

```
125 Sales = 240 260 290
130 Costs = 160 175 200
```

When CALCULATE is finished, PlanStar drops into a spreadsheet-like mode, with a cursor pointing at the top left-hand value (240.00 in Fig 2) of the spreadsheet. The cursor can be moved using the control and E/S/D/X keys which are arranged in a diamond on the keyboard, and each moves it in an appropriate direction. For larger models only a small area of the spreadsheet can be accommodated on the screen, and attempts to move the cursor off the edge of the screen cause a very rapid redraw, so that the destination cell can be accommodated in the new display.

There are also other keystrokes for moving several rows or columns at one time, and facilities for adjusting column width and numeric precision, but that's about all PlanStar offers on the spreadsheet front. It's really a financial planning package, albeit a very powerful one, which pays lip service to spreadsheeting. In particular, it isn't possible to dynamically modify calculations or data at the cursor position, and even if it were, the speed of recalculation (see Benchmarks) would discourage anyone from using it that way.

PlanStar models have up to six different parts: row, column and worksheet definitions, logic, and reporting/graphing instructions. All parts are entered at the keyboard using the line numbering scheme already described, but not all of them need be present. For example, the model in Fig 1 has only row and column definitions and some logic. Even though there is no reporting or graphing, the results can still be viewed because of the spreadsheet facility following CALCULATE.

Other modelling systems separate these different parts of the model into individual files, an approach I have never liked as it greatly multiplies the number of files to consider. With PlanStar, everything is held together. The only requirement is that the user maintains the parts in the right order: in other words, the logic cannot refer to a row or column if it hasn't previously been specified. This is fairly obvious, and it's quite natural to start your model with the columns and rows you are going to work with.

Rows and columns can be given short and expanded names. Short names are used within calculations, whereas expanded names can be more descriptive and are used for display and reporting. PlanStar permits a theoretical maximum of 999 worksheets to be handled in one model, although in practice you'll run out of memory or disk before you reach this limit. Nonetheless, the multiple worksheet facility works very well and is a cornerstone for the consolidation facility.

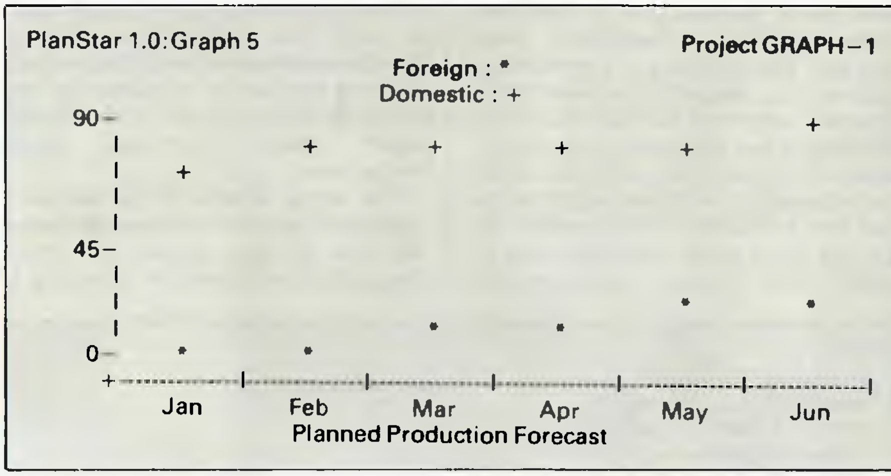


Fig 6 Point-plot graphing

Here's a flavour of the extensive modelling facilities of PlanStar:

5000 Price = 80 90 100 GROW BY 5% sets the Price row to 80, 90, 100, 105, and thereafter increasing by 5% for as many columns as necessary.

5010 Quarter = SUM OF January TO March adds the first three columns of a year, placing the result in the Quarter column.

5020 DEFINE WORKSHEET 1(AUST) 2(USA) 3(Totals) defines the different worksheets to be operated on. PlanStar's multiple worksheet literally and figuratively adds a new dimension to the model. For maximum benefit all worksheets will have very similar logic but different data. It's then very easy to sum together several worksheets and place the result in a fourth, for instance. But summation is not the only option — just about any of the normal arithmetic operations can be used. For example, 5030 WORKSHEET Totals = WORKSHEET AUST * 0.83 + WORKSHEET USA could be used to consolidate US and Australian figures, where each model is expressed in its own currency (and $US 1 =.$AUS to.83).

It's possible to modify all worksheets simultaneously with one command, or to focus attention on just one, or just a small part of one. The CONSIDER command is used to do this:

5030 CONSIDER January
5040 CONSIDER WORKSHEET AUST
5050 Repayment = 75% *Amount

This sequence of commands will only affect the value of January's repayment in Australia. All other repayments in the row are unaffected, as are any repayments in other countries. The global effect can be regained by using CONSIDER ALL.

In general, expressions can become very complex, but there's a natural limit imposed by line-length and by the fact that parentheses are not permitted. There's also a facility for editing lines — it isn't necessary to retype a long line from scratch. PlanStar offers an extensive range of financial functions, such as amortisation, internal rate of return,

Scorecard

Easy to learn:	!!(good)
Easy to use:	?(poor)
Reliability/ error handling:	!!(good)
Facilities:	!!!!(excellent)

Benchmarks

The Benchmarks were run on an IBM PC.
Spreadsheet size: 999 'worksheets' each with 32,714 entries (= rows × columns). 5000-line model.
Max column width: 70+ characters.
Benchmark 1(b) and (c) Recalculation time: 437 seconds for 70 rows (six-plus seconds per row).
1 (d) Vertical scrolling: two rows per second.
1 (e) Horizontal scrolling: 1.3 columns per second.
Benchmarks 2 and 3: Not tested.

SOFTWARE

net present value, and so on, but none of the mathematical functions, such as trigonometry or statistics. This is hardly surprising: the system is unlikely to appeal to the technician and is aimed squarely at the businessman, accountant or financial manager.

Sensitivity analysis

What we have seen so far is a very comprehensive modelling system, well documented and with tutorial software, but PlanStar has even more to offer than that.

Fig 3 shows a simple sales model — a plan for the next three years — where Margins, Sales and Costs are derived from the Units and SalePrice input. But like any model of the future, we may hope it's accurate but we can't be certain. What would happen if we were forced to drop the selling price by 5%? No problem for PlanStar — just type SENSITIVITY OF Margin FOR 5% CHANGE IN SalePrice and out comes the answer (Fig 4). This gives the margins for both a 5% increase and decrease in price.

Sensitivity analysis is very much a case of 'first the good news, then the bad'. A sensitive model can show spectacular increases for a small percentage increase in input values, but just as spectacularly decreases if the input values are lowered. An insensitive model shows more conservative increases, but the decreases are more conservative too. Sensitivity analysis is therefore invaluable for exploring the stability of your business plans, given the inevitable uncertainties of the future. In spite of its use in financial modelling, it is currently a rare feature.

Goal seeking

PlanStar's goal seeking facility enables you to pose questions 'against the flow' of the logic. For example, in the sales model (Fig 3) we may wish to know what selling price is needed to achieve a margin of $10,000 for 1984. On most spreadsheet systems you would be forced to use trial and error, or even (horror of horrors!) work this out by hand. The problem is that the model has been built up with Price as an input and Margin as a calculated output, but with PlanStar you can enter FIND SalePrice GIVING Margin = 10000 IN 1984 and it produces the required result (Fig 5) of $5.50 selling price.

Like sensitivity analysis, this equation-solving capability is a rarity on a modelling system. TK!Solver (*APC* March 1984) is the only other well-known system with this feature.

'The first page of the manual provocatively dismisses pure spreadsheet systems because "a successful plan for a modern business can no longer be expressed on a single piece of paper".'

Graphs

PlanStar has comprehensive graphing facilities which should operate on any computer and printer. This is because the graphs are built up from the ordinary VDU/printer characters and do not need any special hardware facilities. Of course, this means that the final presentation (Figs 6-8) is not of the highest quality and does not compare with the graphing of Lotus 1-2-3, for example.

The three graphs are all derived from the same data. The specification for plotting each of them is concise and easy to use. If certain aspects of the plot are left unspecified the system uses intelligent defaults, so it's fairly easy to get started with graphing; a plot can be precisely tailored to your requirement. For example, it's generally preferable to use as much as possible of your printer paper in the plot, as this effectively increases the resolution (and you can always take a reduction photocopy of it). PlanStar comes equipped with a variety of commands for

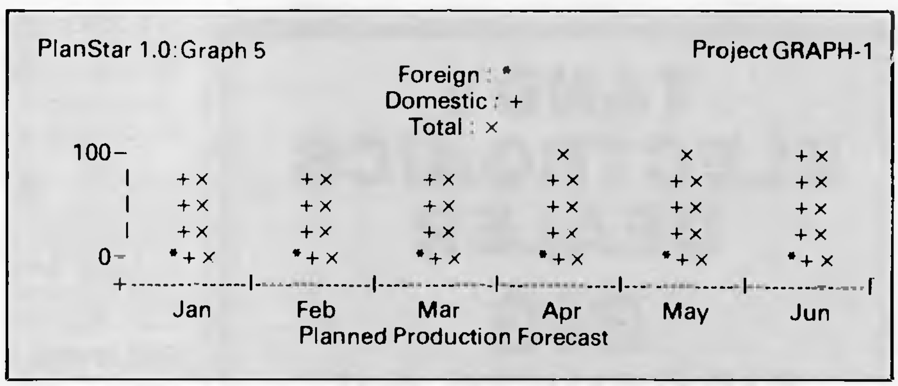


Fig 7 Bar graphing

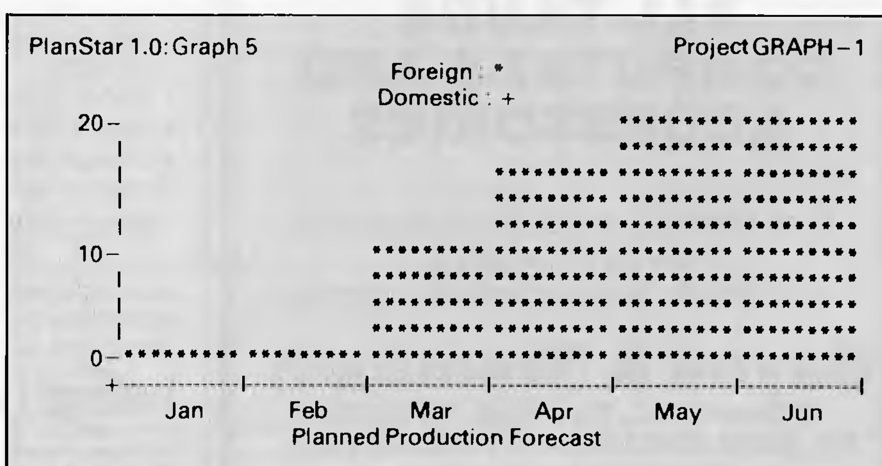


Fig 8 Histogram

A CONSTELLATION OF CHOICES

For the job of your choice

Need faster throughput for those long spreadsheets or near letter quality for a report or proposal? What about firing off some program listings — or even homework? For all jobs, for all people — get STAR Power.

GEMINI-Best-selling 120 cps model for both home and business applications. Tractor feed, friction feed, roll paper, and full graphics all standard. Economical. Reliable. Compatible.

DELTA-Rugged assistance for when the job demands more. 160 characters per second fast, Delta solves small and large business needs. More buffer (8K), more interface (both serial and parallel) — less dollar.

RADIX-The Powerhouse. Feature-packed Radix prints at the speed of light (200 cps) but slows down to excellent near letter quality resolution for when looks count. Tear-off edge and semi-automatic sheet feed included with standard model.

All models available in two sizes — 10 and 15 inch.

*Recommended Retail Price

GEMINI 10X	$449	GEMINI 15X	$795
DELTA 10	$795	DELTA 15	$1195
RADIX 10	$1280	RADIX 15	$1550

*Please add sales tax where applicable.

Please ask for:

SYDNEY (02) 452 5231 — Michele Cahill
MELBOURNE (03) 529 7644 — Kevin Murphy
BRISBANE (07) 221 4727 — Bruce Isbister

STAR

PRINTERS FOR PERSONAL COMPUTERS

Distributed by:
Case Communication Systems Ltd.
1-3 Rodborough Road Frenchs Forest NSW 2086

SOFTWARE

scaling, adjusting paper dimensions, and so on, but needs to be told what to plot, of course: this is done with the SHOW command. As the examples show, more than one variable can be plotted at a time. The specification for Fig 7 included:

8000 SHOW Foreign WITH *
8010 SHOW Domestic WITH +
8020 SHOW Total with ×

All the graphing commands apply no matter which graph you intend drawing, and the only difference arises with the inclusion of the BARCHART command for Fig 7 and HISTOGRAM for Fig 8.

The Benchmark results were the most disappointing aspect of the system. Financial modelling systems have a tendency to be slower than pure spreadsheet software and PlanStar is no exception. There is some validity in the argument that the considerable weight of functionality slows the system down, but a major reason that, for example, Lotus 1-2-3 (*APC* December 1983) knocks spots off it for speed is that 1-2-3 was expertly implemented in compactly coded assembler language, whereas PlanStar uses the rather sluggish and large compiled Basic with much time spent reading the disks.

A new PlanStar user will quickly learn to be very careful to check everything before doing a CALCULATE. Anything but the most trivial model takes it several minutes to work out. If you have used a spreadsheet system previously, forget the 'try it and see what happens' approach.

In fact, the Benchmarks had to be slightly modified as Planstar does not permit parentheses or work in the same way as a typical spreadsheet. However, each stage in the calculation still involved the four common arithmetic operations just once, so comparisons are still valid. The model used was:

1000 COLUMNS Jan Feb etc Dec Tot
2000 ROWS R1 R2 R3 etc
2050 ROWS R69 R70
3000 DEFINE WORKSHEET 1
5000 CONSIDER Jan to Dec
5010 R1=1 GROW BY 1
5020 R2=12*R1/12−1+13
5030 R3=12*R2/12−1+13
etc
5700 R70=12*R69/12−1+13
5800 CONSIDER ALL COLUMNS
6000 Tot=SUM OF Jan TO Dec

Note that PlanStar has no equivalent of the spreadsheet 'replicate' command and the 70 lines 5010−5700 had to be typed in by hand: consequently, I couldn't test the system to capacity.

Conclusion

If you are looking for a fast, easy-to-use modelling system, cross PlanStar off your list. If you are more interested in solid features and facilities, particularly consolidation and sensitivity analysis, then PlanStar takes a lot of beating. But be prepared for a big learning effort: although MicroPro has made a brave attempt with the tutorial material, there really is a lot to learn!

I would say that PlanStar is the most exciting financial planner I have yet seen, but it definitely does not achieve the goal of a decent marriage between traditional financial modelling and spreadsheet technology. It's a great pity that MicroPro did not apply all those years of experience with highly interactive software to this, its latest product. If it had done so, it really would have something to overshadow WordStar.

HI-TECH C

An Australian developed C Compiler that runs rings around the opposition.

Features:

- Compiles Full V7 C — including enums and structure operations
- 50%—100% faster than any competing Compiler.
- Complete I/O library
- Source for libraries included — no royalties
- Easy to use — one command compiles, assembles and links
- Powerful debugging tool included
- Command line I/O redirection
- FAST floating point
- Locally developed and supported
- Supports ROM-based software
- Includes MACRO assembler, linker and librarian

Disk formats:

CP/M: 8" SSSD
Most 5¼" formats
(add $5 for 5¼" formats)
MSDOS: IBM PC format

Pricing: (includes 12 months support)

Z80	$250
MSDOS	$300

Hobbyists introductory special:
Z80 version without support $125
Post and Handling $5 on all orders

System Requirements:

Z80
56K
CP/M2.2

or MSDOS 2.

Z80 and 8086 versions now available.

HI-TECH SOFTWARE

P.O. Box 118, Eastwood, 2122. Ph. (02) 85 4552

Modem protocols

Micro communications has a complex etiquette, an example of which is the XModem, as developed by Ward Christensen. Peter Tootill describes this method of transferring data.

As many people will already be aware, there has been a tremendous increase in the number of computer bulletin boards recently. The number of situations where two or more computers are available has grown rapidly, and both these facts have led to an increasing interest in the transfer of files between two computers.

There are several methods of doing this, whether via the telephone system or by connecting the RS232 ports together (remembering to use a 'null modem' or to reverse pins two and three at one end of the connection).

'. . . . set of protocols for improving the reliability of transmission of data by using a checksum technique was developed by Ward Christensen for inclusion in his 'Modem' and 'XModem' series of intelligent terminal programs written for CP/M systems.'

Files

One of the most important considerations is the type of file that you wish to send across. There are two main types, usually referred to as 'binary files' and 'ASCII files'. The latter are files that contain nothing but printable characters from the standard ASCII 7-bit alphabet. A list of ASCII codes (in your computer's manual, for example) will provide codes between 32 and 127, plus carriage return and linefeed.

The simplest type of ASCII file is ordinary text (the text of this article, for example) without any special printer control codes of the type that can be inserted by some word processing packages. Any file that contains other characters (that is, characters whose ASCII code is over 127) is usually called a binary file. The distinction is quite important, as ASCII files can be transferred between systems by using the common standard word length of seven bits. Binary files will need eight bits for transfer. It's possible to translate a binary file into an ASCII file by translating each byte into two printable characters. For example, the byte 'D6' (hex) would be sent as the letter 'D' followed by the number '6'. This means that the transfer takes twice as long as it would if we could use the full eight bits, but it's a very common way of doing things.

Examples of binary files are machine code and Basic programs that use a tokenised form of the Basic keywords, that is, Basic programs from most micros unless they have been saved with an ASCII option.

The simplest method of transferring an ASCII file (or a binary file that has been translated into ASCII) is to send it one character at a time with no error checking. The problem with this way of doing things is that a noise on the line can cause the data to be corrupted. This may not be a serious problem with a text file, as the human brain is very good at compensating for such errors from clues in the surrounding text. However, in a computer program a small error could be disastrous and may not be easy to find, especially in a machine code program.

One way to reduce the probability of such errors is to rely on parity errors to highlight corrupted characters. A better method is to transmit the file in blocks, and to calculate the 'checksum' of each block by adding all the ASCII values in the block. The receiving computer requests retransmission if the checksum it calculates doesn't agree with that sent by the transmitting system. This is not a complete solution as it's possible for errors to compensate for each other and still produce the correct checksum, even though the data is wrong. It is, however, widely used and generally produces very reliable results. More advanced methods use more complex checks on the data, and can give virtually 100% error free transmission.

Protocols

One particular set of protocols for improving the reliability of transmission of data by using a checksum technique was developed by Ward Christensen for inclusion in his 'Modem' and 'XModem' series of intelligent terminal programs written for CP/M systems. (These are available from the CP/M user group library, the latest versions being Modem 7.65 and XModem 5.0.) These protocols, often referred to as 'CP/M' or 'XModem' protocols, have been applied to other sytems and have become widely used in North America. They have been incorporated in many smart terminal packages and are supported by the

RCP/M and some of the CBBS systems operating in Australia. They provide much more reliable downloading of programs than would otherwise be possible. There are also some packages that support the XMODEM protocol for systems that don't use CP/M, such as the Commodore 64 and the BBC Micro. The protocols use an 8-bit standard (with no parity), and can transmit machine code files without the need to translate them into ASCII code first.

<soh>	01H
<eot>	04H
<ack>	06H
<nak>	15H
<can>	18H

Fig 1 Protocol definitions

The protocols are 'in the public domain' and no licence fees are required to use them (see Fig 1 for details). If you are writing terminal software, I strongly recommend that you consider including support for XModem protocols in the package.

The protocols work at three levels — transmission level, message block level, and file level.

Transmission level

Data format: asynchronous, eight data bits, no parity, one stop bit.

There is no restriction on the contents of the data being transmitted. Any kind may be sent — binary, ASCII, and so on. To maintain compatibility with the CP/M file structure — that is, to allow the transfer of ASCII files to or from CP/M systems — the files should adhere to the following:

* ASCII tabs used (09H): tabs set every nine characters.
* Lines should be terminated by CR/LF (0DH 0AH).
* End-of-file should be indicated by one or more Control-Zs (1AH). (A CP/M peculiarity is that if the data ends exactly on a 128-byte boundary, a subsequent sector containing the Control-Z EOF character(s) is optional, but is preferred. Some programs still do not handle EOF without Control-Zs.)
* The last block sent is not different in any way from others: there is no 'short block'.

Message block level

Each block of the transfer looks like the following:

<SOH><blk no><255-blk no><..128 data bytes..><chksum>, in which:

<SOH> = 01 hex.

<blk no> = block number, starts at 01H, increments by one, and wraps from 0FFH to 00H (not to 01).

<255-blk no> = the 'one's complement' of the block number: that is, each bit in the 8-bit block number complemented with itself. (For example, the block number after going through the 8080 'CMA' instruction.)

<cksum> = the sum of the data bytes only, ignoring any carry.

File level

a) Common to both sender and receiver: all errors are retried 10 times. For versions running with an operator, a message is output on the terminal after 10 errors asking the operator whether to 'retry or quit'.

Some versions of the protocol use ASCII <can> (Control-X) to cancel transmission. This is undesirable, as having a single 'abort' character makes the transmission susceptible to false termination due to corruption of a control byte.

b) At the receive end: the receiver has a 10-second timeout. It sends a <nak> every time it times out. The receiver's first timeout, which sends a <nak>, signals the transmitter to start. (Optionally, the receiver could send a <nak> immediately, in case the sender is ready. This would save the initial 10-second timeout. However, the receiver must continue to timeout every 10 seconds in case the sender wasn't ready.)

Once into receiving a block, the receiver changes to a one second timeout for each character and the checksum. If the receiver wishes to <nak> a block for any reason (invalid header, timeout, and so on) it must wait for the line to clear.

Synchronising: if a valid block number is received, it will be:

1 The expected one, in which case everything is fine.

2 An unexpected repeat of the previously received block. This should be considered OK, and only indicates that the receivers <ack> became corrupted, and the sender re-transmitted.

3 Any other block number indicates a fatal loss of synchronisation, such as the rare case of the sender getting a line-glitch that looks like an <ack>. In this case the transmission is aborted, either by sending a <can> or by waiting for a timeout.

c) At the senders end: while waiting for transmission to begin, the sender needs only a single, long timeout (one minute, for example). In most implementations, the sender has a 10-second timeout before retrying, but this is not necessary as the protocol can be completely receiver-driven; this will be compatible with existing programs. When the sender has no more data, it sends an <eot> and awaits an <ack>, resending the <eot> if it doesn't get one.

A sample of the data flow, sending a three-block message, is shown in Fig 2. It includes the two most common line problems — a corrupted block, and an <ack> reply getting corrupted. <xx> represents the checksum byte.

```
SENDER              RECEIVER
                    times out after 10 seconds
      <-- <nak>
  <soh> 01 FE -data- <xx> -->
      <-- <ack>
  <soh> 02 FD -data- <xx> --> (data gets corrupted)
      <-- <nak>
  <soh> 02 FD -data- <xx> -->
      <-- <ack>
  <soh> 03 FC -data- <xx> -->
      <-- <ack> (but <ack> gets corrupted)
  <soh> 03 FC -data- <xx> -->
      <-- <ack>
          <eot> -->
      <-- <ack>
```

Fig 2 Data flow sending three-block message

Programming hints

The character-receive subroutine should be called with a parameter specifying the number of seconds to wait. The receiver should first call it with 10, then <nak> and try again, 10 times. After receiving the <soh>, he should call the character receive subroutine with a one-second timeout for the remainder of the message block and the <cksum>. As they are sent as a continuous stream, timing out of this implies a serious line glitch that can cause 127 characters to be seen instead of 128, for example.

When the receiver wishes to <nak>, it should call a 'purge' subroutine to wait for the line to clear. The sender should ignore any characters in its UART buffer immediately upon completing the sending of a block, to ensure that no glitches were misinterpreted.

The most common technique is for 'purge' to call the character receive subroutine, specifying a one-second timeout, and looping back to purge until a timeout occurs. The <nak> is then sent, ensuring the other end will see it. END

SOFTWARE

Exploring WordStar

John Lee and Timothy Lee describe some simple but useful ideas which can be implemented on the WordStar word processing package.

Although WordStar is a very powerful and widely used word processing program for microcomputers running under CP/M and 16-bit machines, there are a number of simple things that ordinary users would like to do which are not explained or documented.

WordStar has a function to allow text to be underlined. This is carried out by putting ^PS before the first letter to be underlined, and ^PS after the last letter to be underlined. Many users find it annoying that this underlines the words satisfactorily, but does not underline the spaces between the words (this certainly spoils the look of headings). The easy way to underline both words and spaces is to use the WordStar underlining feature as usual, but to type the underline character in between words rather than a space:

^PSThe_easy_way_to_underline_both _words_and_spaces^PS

It should be noted that WordStar treats the underline _ character as a letter, and hence the underlined text is considered as one large word. This may be noticeable if WordStar needs to add a lot of spaces to pad the line, as there are few words between which WordStar can insert space.

WordStar provides a wide variety of cursor movements. The usual diamond of keys ^E, ^S, ^D and ^X move up a line, one letter left, one letter right and down a line, while ^A and ^F move one word to the left or right. The command ^QR moves to the beginning of the document and ^QC moves to the end of the document, but it would be useful to be able to move to the end of a sentence. Since a sentence should always end with a full stop, the WordStar Find instruction can be used to locate the next full stop:

^QF. RETURN RETURN

If you use the above command, the cursor will be moved to the next full stop in the text. This full stop may be the end of a sentence, the decimal point in a number, or the dot in a WordStar dot command. However, there is an alternative command to move to the end of a sentence. As there is always a space after the full stop between sentences, and there is not a space after the decimal point in numbers or after the dot in dot commands, by looking for a dot followed by a space the cursor is moved more reliably to the end of a sentence. This can be achieved by typing:

^QF. SPACE RETURN RETURN

This command makes mistakes occasionally. For example, it incorrectly finds three combinations of dot and space in the following, even though they are not the end of a sentence:

Dr. John D. Lee
and
Timothy D. Lee

A small problem remains with this command. The last sentence in a paragraph will not be found, because immediately following the full stop at the end of the sentence is a RETURN which marks the end of the paragraph. Consequently we use the first of the two commands to 'move to end of a sentence'. If this command finds a dot that is not the end of the sentence the command can be typed again, or alternatively ^L can be typed to look for another dot. The second command makes fewer mistakes, but may move too far.

In a similar way, WordStar can move the cursor to the end of a paragraph — this is indicated by a full stop and a RETURN. The WordStar Find instruction can be used to locate the next full stop followed by RETURN. In ASCII, a carriage return is actually a Control-N character (stored in the file as ^N), so the required sequence is:

^QF.^N RETURN RETURN

Unfortunately this command will skip over any paragraph that has a space typed after the full stop and before the RETURN. One way of getting round this problem would seem to be to look just for the RETURN, but if this is tried, WordStar moves the cursor to the start of the next line. The best solution is to type RETURN immediately after a full stop at the end of a paragraph.

END

64
apple
VIC 20

BIG
LITTLE BITS

1. GRAPPLER CD Printer Interface

FOR COMMODORE C64 A NEW ERA IN COMMODORE PRINTING POWER.
The Grappler CD may be the most intelligent printer interface available for ANY MICROCOMPUTER, and is available only to users of the Commodore 64. Over 20 commands, accessible either from the keyboard or software, provide unbelievable versatility in printer operation. Using Grappler CDs 1525 Emulation Mode, your printer will be compatible with virtually all Commodore Software. Special character translations permit easy reading of program listings. There's even a Transparent Mode to pass data to the printer exactly as the Commodore sends it. It also offers the revolutionary dimension of Screen Dumping. A wide selection of commands provides full control overtext and hi-res graphic screens. No other Commodore printer interface offers all this. NOTHING ELSE EVEN COMES CLOSE:-

2. UNIVERSAL PARALLEL Printer Interface

FOR VIC-20 & COMMODORE C64.
Use your Vic-20 & C64 with any standard Parallel input printer. You don't have to give up the use of your user port (MODEM), or use special printer commands, or load any special software driver programs to do it. Plug in PRINTRON and print away!, with access to all of the advanced functions of today's printers.

3. GRAPPLER + Printer Interface

FOR APPLE THE MOST INTELLIGENT APPLE INTERFACE AVAILABLE!
The original Grappler was the first graphics interface to give you hi-res screen dumps from your keyboard. The new Grappler + with Duel Hi-Res Graphics adds flexibility with side-by-side printout of page 1 and page 2 graphics. 4K of exclusive firmware makes the Grappler + the most intelligent, fully-featured Apple Printer Interface made. And, the Grappler + is Apple III compatible.*

Requires additional software driver.

Australian Distributors:

OziSoft™

Suite 33/8-24 Kippax Street, Surry Hills NSW 2010. Australia.

THE MAT OS 2081 APC

Four teachers for Sinclair ZX Spectrum and BBC Microcomputer users available now in Melbourne

Arnold Wheaton, Sulis, Artic and Widgit

These four established educational software publishers have seen the light and have selected Micro-Allsoft to manufacture and distribute their excellent ranges of teaching software throughout Australia.

Programs available now, for classroom or home use, are designed for all student ages. Many come with varying degrees of difficulty.

There are even pre-school 'starters' for two-year-olds, and packages for adults.

For those who want to get the most from their Sinclair ZX Spectrums, there are Pascal, Forth and Assembler now available, plus Disassemblers, Toolkits and 64 column display.

Our educational programs cover a lot of subjects, like . . .
• Alphabet • Spelling • Vocabulary
• Reading • Punctuation • Grammar
• Shakespeare • History • French • Time
• Counting • Angles • Logical Thinking
• Spatial Reasoning • Science Horizons
• Magnets • Diet

There are books available on elementary RAMs and ROMs, Basic programming, machine language and networking capability with the use of micro drives.

Micro-Allsoft is the new name in educational software. It will pay you to phone us or mail the coupon below for more information.

micro-allsoft

23/210 Toorak Road, South Yarra, Vic 3141
Telephone 10am-5pm: 240 0156
Hello, Micro-Allsoft. Please send me details on Sinclair ZX Spectrum and BBC Microcomputer educational software.

Name____________________

Address____________________

________ postcode________ 'phone________

MICRO PRO COMPUTERS

The Medfly is not dead. It is still the best buy — complete with disk drive and monitor. $1850

Talk to the world. Modem, direct connect. Auto answer. Auto dial.
300 to 1200 full duplex operation. Complete with self test & power supply.
Introductory offer.

$625

Full range of peripheral cards to suit Apple — Vectorio — Wombat etc.

MINI CONQUERER CP/M COMPUTER

COMPRISING

- 12in green screen terminal, on swivel base with detachable keyboard. 95 keys, separate numeric key pad. 10 function keys, built-in graphics capability.
- Z80A running at 4MHz.
- 64K bytes of RAM, 4K bytes of ROM.
- 2-RS232 serial ports.
- 2-RS232 serial ports. Software selectable baud rates to 38.4K baud.
- Mass storage: two double sided, 80 track disk drives. 800K bytes (formatted) per diskette. 1.6M-bytes total internal storage.
- CP/M 2.2 with enhanced BIOS & CCP.
- Integrated accounting software: General Ledger, Accounts Receivable, Accounts Payable, Stock Control & Invoicing.
- Reads/writes other popular computer formats.

Complete with business package software of 100 cps. 9x11 Dot Matrix printer. **$4850**

Disk Drives

$285

12" Monitors 18 meg from

$185

TELEPHONE (03) 568 6911
MICRO PRO COMPUTERS
43 ATHERTON RD, OAKLEIGH 3166 MELBOURNE AUST

PLEASE SEND ME MORE INFORMATION ON

NAME ....

ADDRESS ....

SUBURB ....

STATE .... Postcode ......

Input/output control

If you've ever tried to do graphics in machine code on your Atari, access data files on disk or cassette, or dump a screen to the printer, then you'll appreciate just how difficult it is. By changing a few parameters, it's possible to send an unspecified amount of data to a device. Anthony Roberts shows you how.

Input Output Control Blocks ('IOCBs') and Central Input Outputs ('CIOs') allow the user to control input from and output to various devices. The devices that are controlled using CIOs are principally cassette recorders, disk drives, printers and the screen.

Before you can open an IOCB, it must first be closed. This is to prevent errors occurring when you attempt to open an already open IOCB.

To close an IOCB, choose which one you are going to work with. There are five possible choices which correspond to the numbers 1,2,3,4 and 5 in Basic. In machine code, you choose the number by loading it into the 'X' register. The number you load is '#$10' for IOCB 1, '#$20' for IOCB 2, and so on. When you have decided, the routine in Fig 1 will close that IOCB.

Line 10 tells the computer which IOCB to close. Line 20 loads the accumulator with the number #$0C (this is the number which tells the routine you want to close the IOCB). Line 30 stores the number #$0C in location $342 offset by 'X'. As 'X' contains the value of #$10, the number will be stored in $352. The 'X' register is used because the computer uses the value in the 'X' register to decide which one to close. Line 40 calls the operating system routine that actually performs the close operation.

Now that the IOCB is closed, it should be opened for a device. You need to declare the device name, which is best done by storing it in an ASCII string. Let's use the conventions employed by the Syn-Assembler, but there are conversions for the Atari Assembler/Editor. Fig 2 will open an IOCB: see Fig 3 for the full list of device names.

```
00010     LDX #$10     ;  IOCB #1
00020     LDA #$0C     ;  COMMAND FOR CLOSE
00030     STA $342,X   ;  COMMAND LOCATION
00040     JSR $E456    ;  CALL OS ROUTINE
```

Fig 1 Routine to close IOCB

```
00050     LDA #$03     ;  COMMAND FOR OPEN
00060     LDX #$10     ;  IOCB #1
00070     STA $342,X
00080     LDA #LAB     ;  LOW BYTE OF DEVICE NAME
00090     STA $344,X   ;  BUFFER ADDRESS (LOW)
00100     LDA /LAB     ;  HIGH BYTE OF DEVICE NAME
00110     STA $345,X   ;  BUFFER ADDRESS (HIGH)
00120     LDA #$08     ;  OPEN FOR OUTPUT
00130     STA $34A,X
00140     LDA #$00     ;  JUST TO BE SAFE BUT NOT NEEDED
00150     STA $34B,X
00160     JSR $E456
```

Fig 2 Program to open an IOCB

The program opens IOCB 1 for the screen editor (E:). Lines 50, 60 & 70 tell the computer to open up IOCB 1. Line 80 loads the accumulator with the low byte value address of the label 'LAB': for example, if the address of the label 'LAB' was $1234, the low byte value of the address would be #$34. Line 90 stores this value in a location where the computer can find it for later use. Lines 100 and 110 do the same thing, except they load and store the high byte value of the address: for example, #$12. These four lines are necessary because when the computer comes to perform the open, it can look in locations $348 and $349 to find the address where the name of the device is located (E: is the name of the device). Lines 120 and 130 put a #$08 into $34A,X. The contents of location $34A,X tell the computer in which direction the data will be travelling (that is, to or from the device). In this case, the 8 means we will be sending data. Lines 140 and 150 are just to be tidy; the 0 in the location has no effect on this particular IOCB. Line 160 calls the routine to do the open. Fig 4 shows the critical memory locations.

'E:' .... Screen editor (see basic manual)
'S:' .... Screen graphics for graphics modes
'P:' .... Printer (output only)
'K:' .... Keyboard (input only)
'C:' .... Cassette recorder
'D:*.*' .... Disk directory
'D:filename.ext' .... Disk files

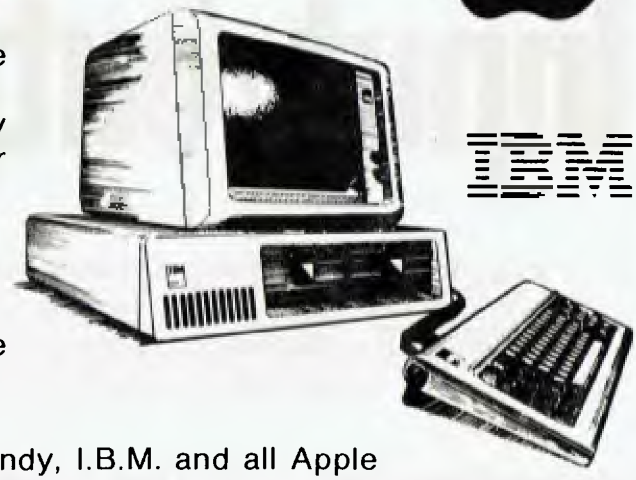
Fig 3 Device names specified when opening a device.

In use

Now that the IOCB is open, you'll want

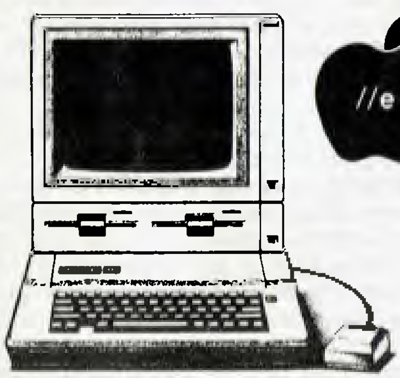
COMPUTER.
SERVICE

The personal computer back-up system you've been looking for.
Unfortunately, most people find out the hard way that there's one question even a personal computer can't answer.



WHERE CAN YOU TAKE IT FOR SERVICE?

We have a suggestion. Come to our new service centre.
Right now we're equipped to handle:
Apple II, II+, IIe, III, Osborne, Commodore, Tandy, I.B.M. and all Apple compatibles. In addition Epson, Itoh and many other printers.
And in the future, we'll be servicing even more brands of personal computers and related items.
We also keep our parts well stocked. Which means your problem can be fixed a lot faster than you'd imagine.
But one of the best things about bringing your equipment to us is that you get quality personal service at a very affordable price. Pick-up and delivery available.



Service Contracts

We are also introducing a new type of service contract. The new contract is valued at 10% and 8% of total hardware cost for on-site and in-store respectively. In addition the contract will have a No-claim bonus system of up to 40% per annum. The above contract includes a quarterly preventative maintenance on the system covered, FREE of charge. This ensures your system is operating to the manufacturer's specifications.

We offer

- 24 hour turnaround on Apple II, II+, I.B.M., Apple disk drives and peripherals.
- All original equipment repaired will carry a one month warranty.
- ALL I.B.M. equipment repaired will carry a three months warranty.
- Open on Thursday nights and Saturday mornings.

SCARLAN PTY LTD (Inc in N.S.W.)
T/A **GENERATION MICRO SERVICES**
REAR 229 BURNS BAY ROAD, LANE COVE WEST 2066. Telephone: (02) 427 0314

Location	What it is used for	Possible values	Result of using value
$E456	Calls the routine to act on IOCB.	None	Execution of IOCB.
$342	Sets the way the IOCB will be used	3	Open the IOCB.
		12	Close the IOCB.
		7	Get binary record.
		11	Put binary record.
		4	Input string.
		8	Output string.
$344	Low byte value of buffer address	0-255	Tells the computer the low byte address of where to get or put data.
$345	High byte value of buffer address	0-255	Same as above only high byte.
$348	Low byte value of buffer length	0-255	Sets the amount of data to be moved. (low byte).
$349	High byte value of buffer length	0-255	Same as above only high byte.
$34A	Sets the direction of data transfer	4	Read and write data.
		8	Open for directory.
		12	Read and Write data.
		6	Open for directory.
$34B	Used mainly for graphics.	0-255	See graphics table.

Fig 4 I/O critical memory locations.

```
00170        LDA #$08      ;  GOING TO SEND STRING
00180        LDX #$10      ;  IOCB #1
00190        STA $342,X
00200        LDA #MES      ;  LOW BYTE OF MESSAGE ADDRESS
00210        STA $344,X
00220        LDA/MES       ;  HIGH BYTE OF MESSAGE ADDRESS
00230        STA $345,X
00240        LDA #$FF      ;  MUST BE MORE THAN MESSAGE
                              LENGTH
00250        STA $348,X    ;  BUFFER LENGTH HELD HERE (LOW)
00260        LDA #$00      ;  JUST TO BE SAFE
00270        STA $349,X    ;  BUFFER LENGTH HELD HERE (HIGH)
00280        JSR $E456
00290        LDA #$0C      ;  CLOSE IOCB #1
00300        LDX #$10
00310        STA $342,X
00320        JSR $E456
00330        BRK           ;  END OF PROGRAM
00340 LAB    .AS "E:"      ;  DEVICE NAME
00350 MES    .AS "ATARI COMPUTERS ARE GREAT"
00360        .HS 9B        ;  END OF LINE CHARACTER
```

Fig 5 Routine for sending data to the device.

```
00010  ;  Graphics mode 2 program
00020  ;
00030        LDX #$10      ;  CLOSE IOCB #1
00040        LDA #$0C
00050        STA $342,X
00060        JSR $E456
00070        LDA #$03      ;  OPEN IOCB #1
00080        LDX #$10
00090        STA $342,X
00100        LDA #SNAME    ;  DEVICE NAME (LOW)
```

to do something with it. In this case, because we put a #$08 into $34A,X, we want to send data. The routine in Fig 5 is used.

The routine in Fig 5 will print the message on the screen. Lines 170 to 190 put a #$08 into the command location ($342); the #$08 tells the computer to expect an undetermined amount of data. This is comparable to printing a string in Basic, because you don't need to know how long the string is to print it. The computer will stop printing data when it reaches a #$9B. Lines 200 to 230 instruct the computer where to find the data for what it has to print; lines 240 to 270 tell the computer how much data to send. As we have used a #$08 in $342,X, the number in these two locations has to be more than we want to send. If you were to send a known amount of data, these two locations would contain this number. $344,X is the low byte and $345 is the high byte. Line 280 executes the operation.

If you are not familiar with the low byte, high byte notation, it simply means the storing of numbers greater than 255 in two consecutive locations. As the maximum value in one location is 255, we have to store numbers greater than this in a special way. The high byte location contains the number of 256's in the number, and the low byte location contains the number of 1's in the number. To store the number 1027, place a 4 in the high byte ($4\times256=1024$) and a 3 in the low byte ($3\times1=3$): the result is $1024+3=1027$. The same principle is used when storing an address.

When the program has been written, close the IOCB by using the first part again. More than one IOCB can be open at a time, so you can read data from a disk using one IOCB and print it to the screen using another. Here is an example of some conversions:

Syn-Assembler	Atari Assembler/Editor
#LAB	LAB&255
/LAB	LAB/256
.AS "ATARI etc"	.BYTE "ATARI etc"
.HS 9B	.BYTE #$9B

If you are using the Atari editor, you'll need a '*=$4000' at the start of the program.

As an example of IOCB use, the program in Fig 6 will open the screen for graphics mode 2 and print a message.

When the graphics screen is opened, the mode number goes into location $34B,X. Location $34A,X contains details of the type of screen you want — that is, split screen configuration. Fig 7 shows how to obtain the different types of screen.

```
00110          STA$344,X
00120          LDA/SNAME    ;  DEVICENAME(HIGH)
00130          STA$345,X
00140          LDA #$18     ;  SPLITSCREEN + OUTPUT
00150          STA$34A,X
00160          LDA #$02     ;  GRAPHICSMODE
00170          STA$34B,X
00180          JSR$E456
00190          LDA #11      ;  PUTBINARYRECORD
00200          LDX #$10
00210          STA$342,X
00220          LDA #$06     ;  AMOUNTOFDATATOSEND(LOW)
00230          STA$348,X
00240          LDA #$00     ;  AMOUNTOFDATATOSEND(HIGH)
00250          STA$349,X
00260          LDA #WORD    ;  LOWBYTEADDRESSOFMESSAGE
00270          STA$344,X
00280          LDA/WORD     ;  HIGHBYTEADDRESSOFMESSAGE
00290          STA$345,X
00300          JSR$E456
00310          BRK          ;  ENDOFPROGRAM
00320SNAME     .AS"S:"      ;  DEVICENAME
00330WORD      .AS"MODE2"
```

Fig 6 Graphics mode 2 program

128	64	32	16	8	4	2	1
		C	S	W	R		

Fig 7 Obtaining different screen types

If bit 'C' is set (equal to 1), the current display will not be cleared when the screen is opened.

If bit 'S' is set, the screen will be set up for a split screen arrangement. This is the same as if you were to open a screen in Basic without putting the '+16' on the end, (GRAPHICS 2, for example).

If bit 'W' is set, it instructs the screen to expect data. This is set for 'plots' and 'drawtos'.

If bit 'R' is set, the screen will be set up for data retrieval from the screen; this is used in 'locate' statements.

END

1984 READER SURVEY

Your chance to win an Epson PX-8

Reader Survey time is with us again. This is your chance to influence the development of *APC* and perhaps win a sparkling new micro while you're at it. This year we've got Epson's latest lap-held, the PX-8, for the lucky name out of the bag. The PX-8 (Benchtested *APC* July '84) has virtually all the capabilities of a desktop CP/M machine but weighs just 2.3kg. It has a variable angle eight-line by 80-column display and 64k RAM. WordStar is bundled in as is a spreadsheet and scheduler, and the basic machine costs around $1300.

Don't despair if you're not the kind of person who wins first prizes — there are also 10 annual subscriptions to *APC* to be won. If you've already got one, then in our great magnanimity we'll extend it for another year.

But the point of all this is to keep us in touch with what you the readers want from the magazine. Not that we're totally out of touch, of course! Lots of you phone in or write, or even come and say hello at the *APC* Show. But this Survey gives us the chance to analyse the views of as many readers as possible. If there's anything you particularly love or hate about *APC*, new ideas you'd like to see included or regular items you think we should dispense with immediately, here's your chance to let us know.

The Survey is used to plan the future of *APC*, so this is your chance to take an active part in choosing the direction we move in over the next year or so. The prize draw will take place in mid-September so the questionnaire should be returned to us as soon as possible. Our thanks to Epson for providing the star prize and to you for completing the questionnaire.

Please use BLOCK CAPITALS throughout and/or tick appropriate boxes.

1 Name: ____________________

2 Address: ____________________

____

3 Age: Under 15 ☐ 15-18 ☐ 19-25 ☐ 26-35 ☐ 36-45 ☐ 46-65 ☐ Over 65 ☐

4 Sex: Male ☐ Female ☐

5 Occupation:
- Professional & Business ☐
  - Less than 10 in firm ☐
  - 10-99 in firm ☐
  - Over 99 ☐
- Education ☐
- Student ☐
- Computer industry ☐
  - Sales ☐
  - Service ☐
  - Support ☐
  - Other ☐
  - (please specify) __________

6 Which of the following publications do you read? Please indicate which one you think is best

	Regularly	Sometimes	
Australian Personal Computer	☐	☐	☐
Your Computer	☐	☐	☐
Today's Computers	☐	☐	☐
Australian Microcomputerworld	☐	☐	☐
What's New in Computing	☐	☐	☐
Australian PC World	☐	☐	☐
Electronics Australia	☐	☐	☐
Electronics Today International	☐	☐	☐
Byte	☐	☐	☐
Creative Computing	☐	☐	☐

Other — state which __________

7 Please indicate your interest in the following sections of *APC*:

	A lot	Some	Little	None
Newsprint	☐	☐	☐	☐
Communications	☐	☐	☐	☐
Banks' Statement	☐	☐	☐	☐
SubSet	☐	☐	☐	☐
Network News	☐	☐	☐	☐
TJ's Workshop	☐	☐	☐	☐
Hardware Benchtests	☐	☐	☐	☐
Hardware Checkouts	☐	☐	☐	☐
Software reviews	☐	☐	☐	☐
Language & operating system features	☐	☐	☐	☐
Teach yourself Assembler	☐	☐	☐	☐
Bibliofile	☐	☐	☐	☐
Yankee Doodles	☐	☐	☐	☐
Beginners' Guide to Program Conversion	☐	☐	☐	☐
Micro Chess	☐	☐	☐	☐
Newcomers Start Here	☐	☐	☐	☐
Numbers Count	☐	☐	☐	☐
Lazing Around	☐	☐	☐	☐
Programs	☐	☐	☐	☐
User Groups Index	☐	☐	☐	☐
Diary Data	☐	☐	☐	☐
Chip Chat	☐	☐	☐	☐
Advertisements	☐	☐	☐	☐

8 What other topics would you like to see in *APC?*

9 What other comments (if any) would you like to make about *APC?*

10 Do you use a micro:
- At home ☐
- At work ☐
- both ☐

11 Do you use a micro for:
- Games ☐
  - Arcade style ☐
  - Adventure ☐
  - Strategy ☐
  - Simulation ☐
- Education ☐
- Word processing ☐
- Database ☐
- Spreadsheet ☐
- Accounts ☐
- Mainframe ☐
  - Communications ☐
- Scientific/Engineering ☐
- System development ☐
- Graphics ☐

The Epson PX-8: quality and portability

12 Do you own a micro:
Yes ☐
Which make and model? ____________
No ☐

13 Do you own disk drives?
Yes ☐
Which make and model? ____________
No ☐

14 Do you own a printer?
Yes ☐
Which make and model? ____________
No ☐

15 If you upgraded your system:
Which micro would you buy? ____________
Which peripherals would you buy? ____________

16 Have you owned a micro for:
- Under six months ☐
- Six months-two years ☐
- Over two years ☐

17 How much money do you spend on software a month?
- Under $25 ☐
- $25-$49 ☐
- $50-$99 ☐
- $100-$499 ☐
- $500-$1000 ☐
- Over $1000 ☐

18 Do you use:
- CP/M ☐
- MS-DOS ☐
- PC-DOS ☐

19 Do you read *APC:*
- Every month ☐
- Occasionally ☐
- Rarely ☐

'... the most capable lap-held so far...'

20 Apart from yourself, how many other people read your copy of *APC?* ____________

21 Do you have an *APC* subscription?
Yes ☐ No ☐

22 Do you object to your name and address being placed on a mailing list:
For *APC* use only:
Yes ☐ No ☐
For other commercial users
Yes ☐ No ☐

Australian Personal Computer

Compatible with your environment.

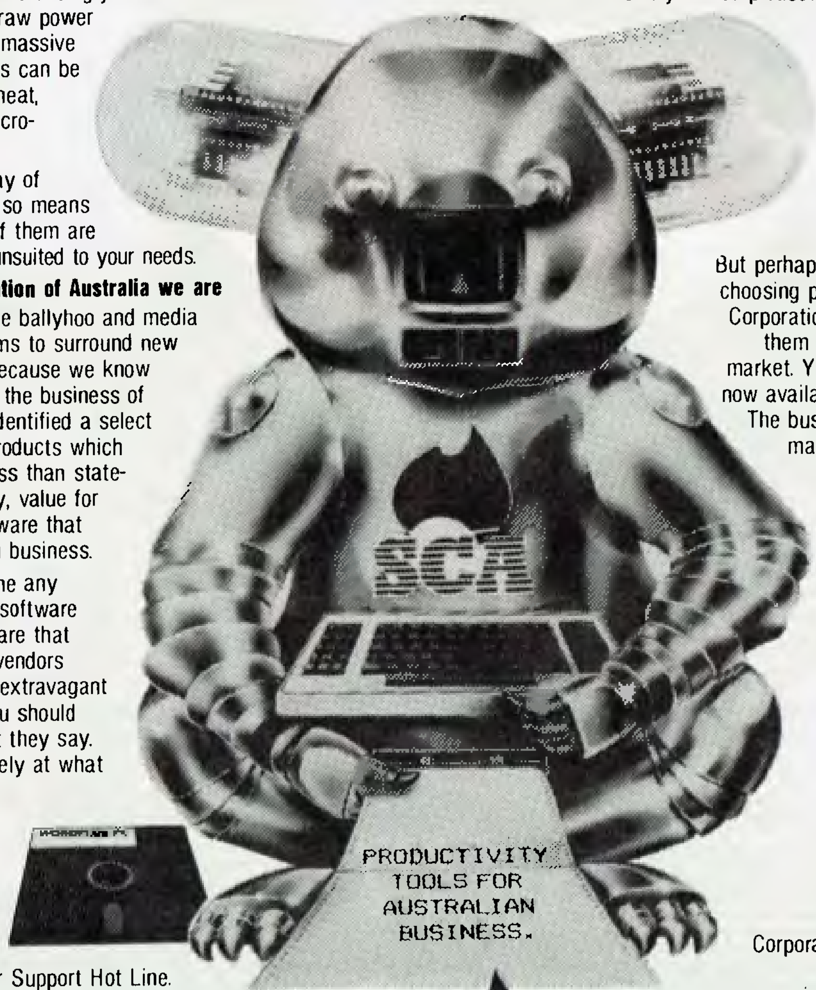
In case you hadn't noticed it, there is a revolution going on. Computer hardware is getting more and more similar: compatibility has seen to that. But software is growing increasingly diverse and distinctive. And the raw power which used to drive massive mainframe computers can be packaged into very neat, small bundles for micro-computers.

But the growing array of software products also means that vast numbers of them are going to be entirely unsuited to your needs.

At Software Corporation of Australia we are unimpressed with the ballyhoo and media hype that often seems to surround new software marvels. Because we know business as well as the business of software, we have identified a select group of software products which represent nothing less than state-of-the-art technology, value for money . . . and software that works for Australian business.

Now if you have done any investigation of the software market you'll be aware that there are plenty of vendors making all kinds of extravagant claims. We think you should look closely at what they say. And even more closely at what they don't.

For example, user support. As a registered user of an SCA product, you will have access to our acclaimed Customer Support Hot Line. So that when difficulties occur, help is never far away. And we will also mail SCA Software User Notes to keep you in touch with what other people are doing with software like yours. And to let you know when upgrades become available. And when an upgrade becomes available, we'll make it available to you at a fair price. You may be surprised at how rare that kind of support can be.



And user training: we have applied our skills in that field to our successful and growing SCA Institute. Not every product needs intensive user training, of course. But you'll be pleased to know that we can provide it at a professional level.

Technical support: even if you never need it, we make sure that when your dealer buys products from us, he can rely on our skilled tech staff. And with the growing sophistication of available products, that's a big issue.

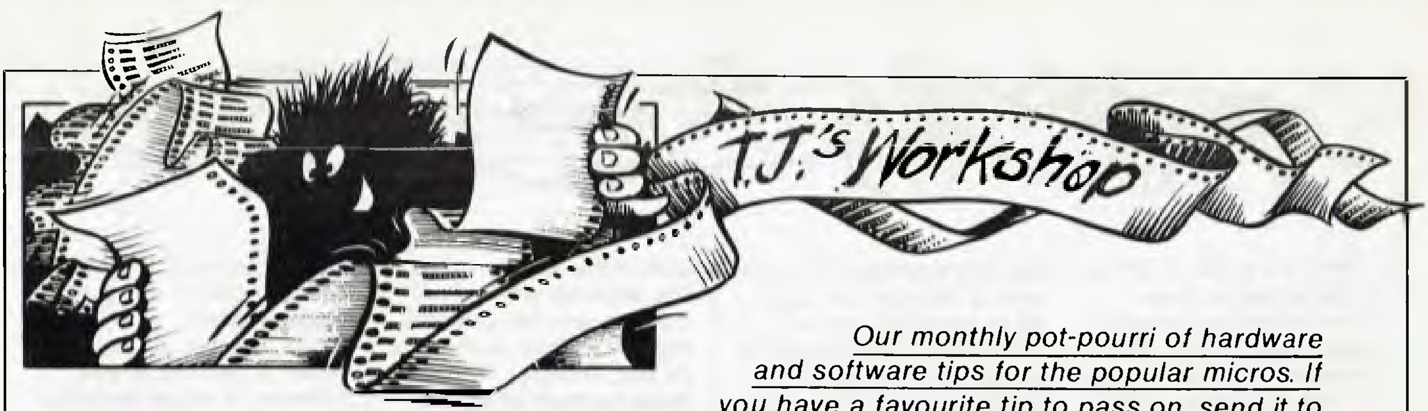
But perhaps the most important reason for choosing products distributed by Software Corporation of Australia is that we select them very carefully for the Australian market. You see, many software products now available just don't work in Australia. The business assumptions under which many of them operate simply don't apply here. Others come with documentation that defies comprehension. Or – worst of all – some are just murder to learn. Our products have to meet the rigorous selection criteria of our Software Evaluation Team. So that before we put our name on the box, we make absolutely sure that we can support the product – and that it is worth supporting.

Support. That's an important concept in a go-go market. And that's why it is important for you to look at the name behind the name. At Software Corporation of Australia, we know that long after the show is over, it's going to be support that makes the difference.

SCA

Software Corporation of Australia Pty. Ltd.

449 Swanston Street, Melbourne, Victoria 3000. Telephone: (03) 347 7011.
Suite 501, 203 New South Head Road, Edgecliff NSW 2027.
Telephone: (02) 328 7074. Telex: AA 30625 ME1293.



Our monthly pot-pourri of hardware and software tips for the popular micros. If you have a favourite tip to pass on, send it to 'TJ's Workshop', 77 Glenhuntly Road, Elwood, Victoria 3184. Please keep your contributions as concise as possible. We will pay $10–$30 for any tips we publish. APC can accept no responsibility for any damage caused by using these tips, and readers should be advised that any hardware modifications may render the maker's guarantee invalid.

Commodore 64 ON ERROR routine

One of the features missing on the Commodore 64 is an ON ERROR command. The idea behind this is that when an error occurs, rather than the standard error message (for example,'?OVERFLOW ERROR IN 100') and the program halting, control is handed over to a particular line (specified by the program) which will deal with it as appropriate. This has many uses, such as suppressing invalid data messages: for example, '?REDO FROM START' and 'EXTRA IGNORED'.

How to use it

The routine at line 500 sets up the machine code, so that a simple call of 'GOSUB 500' is the only initialisation necessary.

To use this new routine, just call
545 867, line number.
So, after
545 867, 200
when the next error occurs, there will be a GOTO 200. This line number may be changed at any time, for the ON ERROR routine only remembers the last line given.

For example, 545 867, 100
545 867, 200
will GOTO 200 when an error occurs.

Great care must be taken to ensure that the specified line number does exist, and is less than 63999. This is because a call to an invalid line will itself produce an error call, resulting in an infinite loop. The only way to escape from such a condition is to press RUN STOP and restore.

The demo section of the program gives an example of error handling.

To restore the error handling back to normal, use the following:
Poke 768, 139:Poke 789,227.

How it works

The machine code routine is split into two parts. The first is a patch into the normal error routine: this is achieved by altering the pointer at $0300/$0301, and forcing a 'GOTO' to the line number store at location $00FB/$00FC.

The second part of the routine called by '545 867' actually links the patch in and gets the line number, storing it in lcoations $00FB/$00FC.

David Gristwood

```
10 REM
20 REM
40 GOSUB 500:REM  SET UP M/C ROUTINE
50 ERR=867
60 SYS ERR,200:REM  ON ERROR GOTO 200
70 A=10/0 :REM  ERROR 1
80 A=A(55):REM  ERROR 2
100 STOP
200 PRINT"ERROR 1 IN LINE 70"
210 SYS ERR,250:REM RESET ON ERROR
230 GOTO 80
250 PRINT"ERROR 2 IN LINE 80"
260 END
279 REM
500 REM  SET UP ON ERROR M/C
520 FOR T=828 TO 889
530 :READ A:POKE T,A
540 NEXT T
560 REM  DATA FOR M/C
580 DATA 138 , 48 , 3 , 76 , 69 , 3 , 76
 , 116 , 164 , 234 , 165
600 DATA 251 , 133 , 20 , 165 , 252 , 13
3 , 21 , 76 , 163 , 168
610 DATA 234 , 0 , 0 , 0 , 0 , 0 , 0 , 3
2 , 253 , 174
620 DATA 32 , 138 , 173 , 32 , 247 , 183
 , 96 , 234 , 169 , 60
630 DATA 141 , 0 , 3 , 169 , 3 , 141 , 1
 , 3 , 32 , 88
640 DATA 3, 165 , 20 , 133 , 251 , 165 ,
 21 , 133 , 252 , 96 , 0
660 RETURN
```

Printer problem

Owners of Commodore printers may find the following information useful. It seems that sometimes part or all of a line of text or data may be missed out altogether from a listing. The problem seems to be at its worst when printing the contents of arrays. The cause of the missing line lies in the Commodore printers (as with most other brands).

When data is transmitted to the printer it goes into a character buffer. This holds the incoming information until it is full, at which time it "dumps" the data to the printer. This system works well until this particular problem occurs: what if the buffer does not fill or only partly fills? The answer is that the data is simply not printed! To see this action enter the following program and watch.

```
1 OPEN 4,4:CMD4
2 FOR T=1 TO 255
3 PRINT "[S]"T:
  PRINT#4,CHR$(T);
4 NEXT T
```

As the program is running watch the counter in the top left hand corner, you will see that it goes up to 117 and then stops. While it is pausing the computer is filling the buffer with the CHR characters from 1 to 117.

When the buffer is full the printer prints-out these characters and the process repeats itself. On the third attempt the counter stops at 255 but the printer prints nothing. The buffer has not been filled and therefore the printer is not presented with any data.

To see the missing line type ' PRINT#4," " '

The buffer is emptied and the data is printed. This process is described on page 29 of the 1525 printer manual and I assume that it is also contained within the 1515 and the 1526 printer manuals under the heading *AUTOMATIC PRINTING*.

I hope this answers any questions pertaining to this problem.

Darren Crocker

Saving user-defined graphics on VIC 20

When defining characters to produce a game with defined graphics it's not necessary to run two programs every time or use up valuable program space.

The following shows how it's possible to save and load the whole defined character set along with the program using it. This is achieved by loading and protecting the characters, loading the game program, then fooling the computer into thinking it has a Basic program that fills all the memory before saving the program on tape.

When the resulting tape is loaded, it loads not only the program but the characters as well.

First a character loading program (such as that shown) is required. This protects the character set from being overwritten.

The rest of the program POKEs the characters into place.

This program defines only one character; in practice there would be a lot more characters. The −1 at the end of the data list ends the sequence and the program. In line 30 POKE 7168+X,A assumes that the character pointer (location 36869) is being changed from 240 to 255 during the game to access the defined characters.

Once the loading program has run, the characters are in place and protected. It is now possible to load the game program from tape (that is, the program that will use but not define the characters) without disturbing the character set. Once the game has loaded, it is necessary to add the following line at the very beginning. This is important: it must be the first line and it must be typed in exactly as shown.

```
10 POKE51,0:POKE52,28:POKE55,0:POKE56,28:CLR
20 READ A:IFA=-1 THEN END
30 POKE7168+X,A:X=X+1:GOTO 20
40 DATA1,2,3,4,5,6,7,8,-1
```

Example

The first two POKEs in line 10 reset the bottom of strings pointer to 28 x 256, that is, 7168, and the second two POKEs reset the top of free RAM pointer to the same location. The CLR statement executes this before anything else is done.

```
1 POKE45,***:POKE46,***
  :POKE51,0:POKE52,28
  :POKE55,0:POKE56,28
  :CLR
```

Do not run the program. Locations 45 & 46 point to the end of the program. This must be changed before saving and it must be set to the correct value before the program really does anything. The asterisks are length of the program; the numbers that will replace them are at present unknown. To find these numbers type:

?PEEK(45),PEEK(46)

Now replace the asterisks with these numbers. If any of the numbers has less than three digits, add a preceding zero (45 045) to maintain the length of the program.

At this point the end of program/start of numeric variables pointer must be reset to the top of user memory. Type:

POKE45,0:POKE46,30

The computer now thinks that the program occupies all user memory. It doesn't care that part of this is not a Basic program.

The program can now be saved in the normal way. However, it would be just as well to verify it and if extra copies are required make them now before running the program: it will not be possible to name copies made later.

The saved tape now contains the program and the graphics and will run normally. However, if any changes are made to the program the process must be repeated.

MJ Curtis

VIC 20 program merge routine

(1) Load main program from tape.

(2) Position tape at beginning of routine to be tagged on to the end of the resident Basic program.

(3) Type in immediate mode.

```
POKE43,PEEK(45)-2
POKE44,PEEK(46)
LOAD
POKE43,1
POKE44,16
```

(Press RETURN after each line!)

(4) Now LIST your program, and you will find the subroutine tagged on to the end.

Note

Before merging with the main program, make sure that the version of the routine being added on has line numbers starting at a greater figure than those at the end of the main program.

A good figure to use is 60000.

Richard Bhanap

Atari cassette salvage operation

The routine below will allow you to salvage programs that have been saved to cassette on any of the Atari machines. It works for programs that have been saved using LIST"C:" only, and not those that have been saved using CSAVE.

It also works for programs that have been saved using the Assembler cartridge.

Rather than halt with an error message if the program has been corrupted, the routine simply skips the corrupted block and goes on to read the next block. In this way you can load most of your program even if part of it is unloadable.

To use the routine, first insert the cassette containing the program you are having difficulty loading and press Play on the recorder. Then run the routine and press any key when you hear the beep. When you hear two beeps insert a new cassette and press Play and

60 Sharp Calculators To Be Won . . .
See inside for details.

Bulletin BOOK NEWS

No 1. AUGUST 1984

McGraw-Hill Book Company Australia Pty Limited

4 Barcoo Street
Roseville NSW 2069
PO Box 2[illegible] Roseville
Telephone (02) 406 4288
Telex 20849

Dear Computer Enthusiast,

If you're not a member of the McGraw-Hill Computer Book Club, then you'll need to know the answers to these important questions.

Q: What will it cost me to join the club?
A: Membership is free – you may resign at any time – simply tell us.

Q: If I joined the club, what sort of books will I be offered?
A: New books on . . . Programing Languages, Introductory and Advanced Computers. Computer Literacy, Data processing, Information Systems, Guides, Games and Projects, Word Processing, Data-Analysis and more.

Q: What's so special about your books?
A: McGraw-Hill has built an enviable reputation as one of the largest education and information organizations in the world. Our books are built upon that reputation.

Q: How many books will I be expected to buy?
A: Unlike other clubs, there is no commitment. The decision is yours.

Q: What sort of savings will I make?
A: You are guaranteed substantial savings over current recommended retail prices. Read on and let this brochure convice you.

Q: Can I see the books before I make up my mind?
A: Yes. We've designed for your convenience, a **FREE** 21 day inspection program.

Q: And if I decide to join?
A: Then you've made a wise decision. All of the above benefits are yours.

Let these answers guide you in joining the McGraw-Hill Computer Book Club. A **FREE** inspection of any book from this brochure, automatically makes you a member.

Yours sincerely,

Rod Elliott

Rod Elliott
EDITOR

CONTENTS

McGraw-Hill Book Company Australia Pty Limited
Incorporated in NSW

4 Barcoo Street, East Roseville, NSW 2069
Telephone **(02) 406 4288**

WHEN COMPUTER MINDED PEOPLE WANT TO KNOW WHAT'S NEW, THEY CHECK WITH US FIRST.

USER GUIDES – Hardware

Your Commodore 64™: A Guide to the Commodore 64™ Computer

by John Heilborn & Ran Talbott

Master the use of your C-64™ with this fully illustrated teaching guide. Packed with all the information you need to get your computer up and running, this guide includes step-by-step operating instructions for the C-64 and its peripherals, plus an introductory tutorial on programming in BASIC. You'll also learn about the special graphics and sound features of the C-64. Detailed coverage of BASIC statements and functions and an extensive memory guide make this book an invaluable reference source for more experienced users too!

464pp. **50A 142**

Your IBM® PC: A Guide to the IBM PC (DOS 2.0) and XT

by Lyle Graham & Tim Field

This popular, comprehensive user guide has been updated to include both the IBM PC (DOS 2.0) and the PC XT!

592pp. **50A 207**

CBM™ Professional Computer Guide

by Adam Osborne with Jim Strasma & Ellen Strasma

". . . jam-packed with goodies for the . . . CBM user." (Desktop Computing)

Beginners and experts alike will find a wide range of information in this guidebook to Commodore Business Machines (CBM®) model computers. From start-up procedures through mastering the use of random numbers, graphics, the machine language monitor and the built-in clock, this book is organized for quick reference or in-depth study and will be indispensable for users of the Commodore CBM™ Professional Computer.

514pp. **50A 756**

Spreadsheeting Manuals

H. Anbarlian

This series of manuals is designed for software Spreadsheet users. It makes using software packages fast, easy, and accurate. Here is the practical guidance the package manuals do not provide. Harry Anbarlian's manuals show how to create reusable matrixes tailored for your personal and business needs – on three levels of complexity. You get stock protfolios, P/E ratios, credit card expenditures, payroll records, creating three-dimensional bar graphs, and more.

BYTE Books

INTRODUCTION TO VisiCalc MATRIXING for the Apple® and IBM® **001605**

VUCALC SPREADSHEETING on the ZX-81 and TIMES/SINCLAIR 1000 **001699**

Apple® II User's Guide for Apple® II Plus and Apple® IIe (Second Edition)

by Lon Poole, Martin McNiff, & Steven Cook

". . . must reading for every Apple® II owner." (Personal Computing)

Now the ever-popular Apple® II User's Guide has been revised and expanded to include the Apple® IIe. An exhaustive hands-on resource book, this guide will explain more about the Apple® than any other single source. A complete BASIC programming tool, all the special features of the Apple are covered.

482pp. **50A 045**

Your IBM PC: A Guide to the IBM Personal Computer

L.J. Graham

The information and resources packed into this book will smoothly glide you through all the steps you need to know to get your computer up and running – from easy-to-grasp instructions on how to operate your computer, its peripherals and software, to lessons in beginning and more advanced programming. You'll also learn about color graphics and sound. An indispensable book for all users of the IBM Personal Computer.

An OSBORNE Book

400pp. **50A 853**

The Osborne/McGraw-Hill Guide to Your Apple® III

by Stan Miastkowski

The only complete, one-volume guide to purchasing, installing, and using the Apple® III system. *The Osborne/McGraw-Hill Guide to Your Apple® III* will quickly teach you about the Apple III system, files and utilities, and the Sophisticated Operating System (SOS). A comprehensive guide to business BASIC is included and a variety of special topics are covered in the appendixes.

276pp. **50A 010**

Your IBM® PC Made Easy (Includes IBM PC (DOS 2.0) and PC-XT)

by Jonathan Sachs

A basic operating guide for the beginning IBM® PC user, *Your IBM® PC Made Easy* covers all the fundamentals of your new system. Detailed explanations of each major feature, step-by-step running instructions, software, use and care, tips on how to trouble shoot computer breakdowns – even a guide to IBM supplies, services, and user groups – it's all here in this easy-to-understand tutorial.

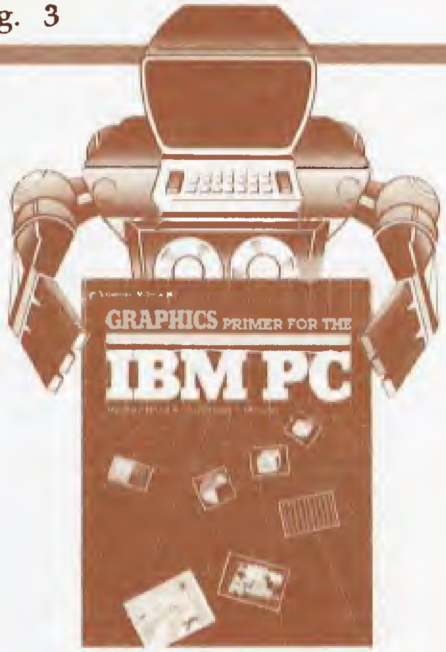
250pp. **50A 126**

Your ATARI Computer: A Guide to ATARI® 400/800™ Personal Computers

by Lon Poole with Martin McNiff & Steven Cook

"Your Atari Computer is unquestionably the best book about Atari computers published to date. It is so complete, so informative, and so well written that a copy should be packed with every Atari computer sold." (Popular Computing)

458pp. **50A 659**



Graphics Primer for the IBM® PC

by Mitchell Waite & Christopher L. Morgan

For beginning and experienced IBM® PC users, this beautiful, full-colour teaching guide will show you all you need to know to create dynamic graphics on your computer. With this book, you'll learn to use your system's graphics commands to produce useful color graphics for business, engineering, education and entertainment applications. Complete example sets show programming steps together with photographs of actual graphic displays: pie charts, graphs, maps and animation, and more. *Graphics Primer for the IBM® PC* contains all the vital information you need to make compelling graphics of your own.

430pp. **50A 993**

BASICS A guide to the Timex/Sinclair 1000 (2K)

H. Mullish

A comprehensive, hands-on guide to the least-expensive, fastest-selling personal computer in the world. It takes the absolute beginner through the first steps of BASIC programming. Then adds to the Timex/Sinclair's power by giving you greater familiarity with it. You'll be able to use it for household record-keeping, personal finance, graphics displays, and instructive and entertaining games. An ideal introduction to the world's most popular computer.

160pp. **044041**

VIC 20™ User Guide

J. Heilborn & R. Talbot

You'll learn to solo promptly on your VIC 20 with this reference tool. It offers clear instructions for operating the computer and its peripherals, and easy-to-track lessons in VIC BASIC. You'll quickly learn to use the color graphics and sound capabilities of the VIC 20. Experienced users will learn advanced color graphics and animation techniques, with programming tips and custom character sets. Loaded with tables, charts, and resource listings.
An OSBORNE Book

250pp. **50A 861**

The HHC™ User Guide

J. Sachs with R. Meyer

Information on the new Panasonic HHC, Quasar HHC, and Olympia HHC Portable Computer is here at last. You learn to use the HHC thermal printer, memory extender, video adapter, color plotter, and more. You'll get a look at some of the HHC packaged applications programs – electronic mail, financial forecasting, and custom sales order entry. And you'll get details of MBASIC, SnapBASIC, and SnapFORTH. All made clear even if you have no previous knowledge of computers
An OSBORNE Book.

200pp. **50A 87X**

PET™ Personal Computer Guide

by Adam Osborne with Jim Strasma and Ellen Strasma

Packed with information on all PET® products, accessories, compatible software, peripherals, and resources; here is literally all the information needed to get your computer up and running.

532pp. **50A 764**

54 SuperCalc® Models: Finance-Statistics-Mathematics

by Robert H. Flast

Analyze loans, investments, taxes, and solve over 30 different statistical and mathematical problems with this ready-to-use collection of SuperCalc® models. Presented in a standardized format, these models can be keyed directly into any microcomputer that runs the popular SuperCalc program.

288pp. **50A 185**

VisiCalc: Home and Office Companion

D.J. Castlewitz & L. Chisausky

Both beginner and expert will return to this reference book again and again for information on this popular software program. For the novice, it contains 50 models that can be used immediately for personal and business applications. For the experienced user, it's a source of new ideas and methods. Included are models for loans and investment, inventory control, household aids, and more. Each model includes the VisiCalc entry grid. Accommodates most micros and every VisiCalc version. An OSBORNE Book.

184pp. **50A 500**

USER GUIDES – Software

Using dBASE II™

by Carl Townsend

Here's a manual for business professionals who want to design customized programs with dBASE II™ software. Informative, clear and non-technical in style, *Using dBASE II™* is a compendium of techniques presented to help you master this remarkable software package. After learning about installation, you'll be introduced to information file organization, records, and databases. Then you will move on to system design, structured programming, and other advanced topics. Using dBASE II™ provides you with a virtual encyclopedia of information, much of which is unavailable elsewhere

250pp. **50A 088**

Data Base Management Systems: A Guide to Microcomputer Software

by David Kruglinski

". . . a welcome addition to the personal-computer software bookshelf".
(Personal Computing)

You'll be able to evaluate and select a data base management system to meet your business requirements after reading this excellent book. The author defines the capabilities of the three

categories of DBMS and reveals tested criteria for evaluating commercially available packages. In addition, he examines, in detail, a number of these packages which run under CP/M® including: Condor Series 20™, dBASE II™, FMS-80™, MDBS III™, Pearl Level 3®, DataStar™ and more.
260pp. **50A 845**

The VisiCalc® Program Made Easy
by David M Castlewitz

The exercises presented in *The VisiCalc® Program Made Easy* will teach you all the basic skills you need to design a professional worksheet and all the advanced skills required to tap the special capabilities of VisiCalc® You'll quickly maximize the quality of your output by reading this fundamental, computer-side tutorial.
160pp **50A 896**

WordStar® Made Easy (Second Edition)
by Walter A. Ettlin

". . . a quantum improvement over the MicroPro manual." (PC Magazine)

In just a few easy, time-saving lessons, you'll learn to produce sales reports, legal documents, business letters, manuscripts and more. Also featured is a useful pull-out card for quick reference and computing ease. Simple to learn and handy to use, *WordStar® Made Easy* is highly recommended.
132pp. **50A 90-X**

The Microcomputer Software Buyer's Guide
T. Webster

The Microcomputer Software Buyer's Guide provides in-depth coverage of many of the top-selling word processing, spreadsheet, data base management and integrated software packages currently available Detailed reviews are provided on important packages with tabular summaries on a wide range of additional products
V68967

54 VisiCalc® Models: Finance-Statistics-Mathematics
by Robert H Flast

Manage investments, loans, taxes, and solve over 30 different statistical and mathematical problems with this comprehensive collection of ready-to-use Visi Calc® programs. Just key them directly into your computer! Whether you're a beginning or experienced VisiCalc program user, you'll get more out of your software by using this anthology of powerful programs.
278pp **50A 002**

Spreadsheeting Software
H. Anbarlian

This series is designed for software Spreadsheet users to make the programs fast, easy, and accurate. Harry Anbarlian provides the practical hands-on guidance the VisiCalc manual doesn't offer You learn how to create reusable matrixes tailored for your own personal and business needs — stock portfolios, P/E ratios, credit card expenditures, payroll records, creating three dimensional bar graphs and more No programming experience is required

INTRODUCTION TO VisiCalc SPREADSHEETING for the TRS-80 Model III (book & diskette) **001597**

The Diskguide Series

Life with your computer becomes a lot easier if you have a DiskGuide next to it It's a specially designed computer-side reference that contains the vital commands, keys, and functions of computing systems and software packages that you need for everyday use Ideal for the times when you want a quick reminder but you don't want to wade through the manuals Each is in a wire-bound format that lets you flip instantly to the necessary items

CHECK ORDER FORM FOR PRICE LIST.

Datapro/McGraw-Hill Guide to CP/M Software
Datapro

Aimed squarely at the information needs of all CP/M software owners and users The most comprehensive guide to CP/M software available. Culled from Datapro's data banks, this volume provides objective details about vendors, utilities and application programs available for the CP/M software
015404

Datapro/McGraw-Hill Guide to IBM/PC Software
Datapro

Culled from Datapro's data banks, this volume provides details about vendors, utilities and application programs available for the IBM/PC
015424

Datapro/McGraw-Hill Guide to Apple Software
Datapro

A directory of all known software for the Apple machines, based on their massive databases of software developers, publishers and suppliers. A well-indexed dsvolume, the information prospective software buyers need to evaluate packages before ordering.
015403

The Osborne/McGraw-Hill Home Computer Software Guide
by Steve Ditlea

Choosing the right software for your home computer is no easy task Considering the wide variety of packages on the market and the price tag attached to them, it's crucial to know what's available, and to learn which software is best for your needs. This is the book to help you make those decisions Every major home computer software package, including personal finance, investment, education, word processing and games is listed, summarized, and analyzed in this comprehensive reference guide. If you own or use an Apple® II, ATARI® home computer, IBM PC®, TRS-80® Model III or Color Computer, VIC 20™, TS 1000™, or TI-99/4A, *The Osborne/McGraw-Hill Home Computer Software Guide* is an essential time- and money-saving investment!
200pp **50A 07-X**

SuperCalc® Home & Office Companion
by Elna Tymes & Peter Antoniak

Designed to help you take complete advatnage of your powerful SuperCalc® software program, these 63 models are ideal for a broad range of applications, including general business, inventory control, advertising and sales, loans and investments, household management, personal finance, and much more Each model gives you a clear and simple introduction, a sample model run or worksheet with sample data entered, and the listing for the worksheet Simply key in the listing and replace the sample data with your own. An excellent collection of models for both practical and educational use.
192pp **50A 134**

OPERATING SYSTEMS & PROGRAM LANGUAGES

Learning With LOGO and Learning With Apple LOGO

D.H. Watt

LOGO is a fascinating computer language that has become a standard one taught to children, and it's time for you to learn it, too. Both books cover programming and activities and are filled with pictures and examples, clear descriptions of bugs and pitfalls, and many projects to enhance understanding. *Learning with LOGO* is keyed for use with the Terrapin and Krell version of LOGO software; *Learning with Apple LOGO* is keyed to the Apple Computer Company's own software version of LOGO.

BYTE Books
1983 (paper)
Learning with Apple LOGO **068571**
256pp.
Learning with LOGO
208pp. **068570**

The Programmer's CP/M® Handbook

Andy Johnson-Laird

An exhaustive coverage of CP/M-80(TM), its internal structure and major components is presented in *The Programmer's CP/M® Handbook*. Written for the programmer, this 750-page volume includes subroutine examples for each of the CP/M system calls and information on how to customize CP/M — complete with detailed source codes for all examples. A dozen utility programs are shown with heavily annotated C-language source codes. An invaluable and comprehensive tool for the serious programmer.

750pp. **50A 037**

Discover FORTH

Thom Hogan

Whether you are a beginner seeking information on this multi-faceted programming language or a serious programmer already using FORTH®, this book is a reference that should not be overlooked. Long considered a computer language of building blocks, FORTH® has been optimized for speed and requires little computer support.

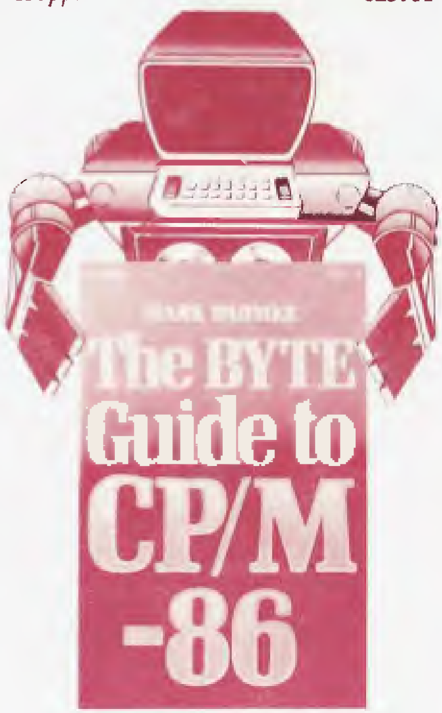
146pp **50A 799**

The C Primer

L. Hancock & M. Kreiger

As the popularity of UNIX-like operating systems continues to rise, this manual will be increasingly important. Here is a full description of the high-level programming language C, developed at Bell Laboratories in support of UNIX. The manual uses many examples to illustrate programming techniques, and will lead you through the steps of C language, showing how it can be used as an alternative to other high-level languages.

256pp. **025981**



The BYTE Guide to CP/M

M. Dahmke & S. Ciarcia

By simple assembly of standard parts, you can build your own sophisticated computer! This book shows you how to put together single-board microcomputers based on 8088 microprocessor, using the CP/M operating system. Whether you build one or not, you gain a greater understanding of CP/M, its features, facilities, and the roles of the basic disk operation system (BDOS) and basic input/output system (BIOS).

A BYTE Book
216pp. 015072

A User Guide To The UNIX(TM) System

by Rebecca Thomas, PhD, & Jean Yates

Specially designed to ease the novice through the difficult learning stages, while also being an excellent reference for those already familiar with the UNIX(TM) operating system. Includes basic commands, a list of UNIX(TM) system resources, a glossary for beginners, comprehensive appendices and a bibliography for further study.

508pp. **50A 713**

Hands-On BASIC for the Apple II

H. Peckham

Even if you have a limited background in mathematics, you will emerge from this book competent at programming the Apple II personal computer. It uses a proven hands-on training method to explain how computers perform practical tasks. It emphasizes programming rather than theory to let you develop your skills as quickly as possible.

320pp. **049179**

The MBASIC(TM) Handbook

by Walter A. Ettlin & Gregory Solberg

Gain a better understanding of programming while you learn to develop and customize programs with this fundamental guide to Microsoft(TM) BASIC. After reading *The MBASIC(TM) Handbook*, you'll be able to develop and document useful business applications programs to fit your special needs. Commands, statements, functions and operators are completely covered, and all programs have been fully tested to run directly on any microcomputer using MBASIC(TM)

457pp. **50A 029**

CBASIC® User Guide

Adam Osborne, Gordon Eubanks & Martin McNiff

". . . this is an excellent user's guide and . . . a helpful addition to a programming library." (Journal of Electrophysiological Technology)

Co-authored by the creator of CBASIC® Gordon Eubanks, this is more than a self-teaching textbook – it is the definitive reference on the CBASIC® language.

212pp. 50A 616

Introducing the UNIX System

H. McGilton & R. Morgan

This right-to-the-point introductory guide to the UNIX operating system will get you over the initial hurdles and hazards in using this powerful new computer tool. And for those with some UNIX experience, a better understanding of the system and its sometimes cryptic documentation. Thorough, detailed coverage, including shell programming, the ex-text editor, the vi display editor, text manipulation, the directory structure and file system, commands, and more.

A BYTE Book

480pp. **045001**

FORTRAN 77 A Structured, Disciplined Style, 2/e

G.B. Davis & T.R. Hoffmann

Why another FORTRAN book? First, the 1977 changes in the American National Standard (ANS) FORTRAN made many existing texts obsolete. Second, recent developments in programming style and discipline have made most existing works inappropriate. Third, in the past, the teaching of FORTRAN has emphasized individual statements. Here, the authors stress that the best method of learning is by writing complete programs in a clear, disciplined style.

416pp. **015903**

Hands-On BASIC For the ATARI 400® and 800® and 1200® Computers

H. Peckham, W. Ellis, Jr., & E. Lodi

Atari personal computer users can now learn programming with Peckham's famous hands-on method. It demands no mathematical background, and uses step-by-step, guided exercises to bring you to proficiency with gratifying promptness. All material is geared specifically to the Atari, with programming emphasized over theory.

308pp. **049194**

Top-Down BASIC for the TRS-80 Colour Computer

K. Skier

If you already know BASIC commands and now want to create your own well-structured programs, follow this friendly course in more advanced programming skills. It guides you through the design process of 11 color BASIC programs, demonstrates concrete methods for achieving top-down design, structuring maintainable and extensive programs, and creating user-friendly programs that are easy to work with and hard to misuse.

A BYTE Book

256pp **057861**

GENERAL INTEREST

Armchair BASIC An Absolute Beginner's Guide to Programming in BASIC

A. Fox & D. Fox

If BASIC doesn't seem all that basic to you, here's a book that will give you a firm understanding of what programming is all about. Even if you don't have or can't get to a computer! Filled with illustrations and examples, it's an unintimidating introduction to programming and BASIC, the most popular computer language. It will make you feel totally at ease when you do find yourself around computers.

An OSBORNE Book

272pp. **50A 926**

Word Processing Buyer's Guide

A. Naiman

Don't buy a word processor until you buy this book. It's a very personal but professional evaluation of virtually every word processor on the market. Neither too elementary nor too technical, it describes what word processors do and how they do it. Covers processors, software, printers, storage systems, keyboards, monitors, and specific information on each system. Plus helpful appendices: conversion tables, bibliography, glossary

A BYTE Book

304pp. **045869**

An Introduction to Microcomputers: Volume 0 – The Beginner's Book

Adam Osborne & David Bunnell

". . . one of those rare introductory books . . . outstanding and informative." (Personal Computing)

One of our most popular titles, expanded and revised to include new sections on software and communications. If you're a beginner in the world of computing, this book introduces you to all the basics: construction, operations and uses of microcomputers.

233pp. **50A 640**

How to Buy (and Survive!) Your First Computer A Guide for Small Business Success

C.N. Kolve

If you are about to be one of the more than one million business people to buy a small computer this year, you'll profit from this painless, nontechnical introduction to computers for business applications. It's the first book of its kind to offer a no-nonsense cost analysis of the computer's benefits. It'll take you through installing and integrating the computer into your business. You'll avoid disaster and excess costs.

256pp. **035130**

An Introduction to Microcomputers: Volume 1 – Basic Concepts

Adam Osborne

". . . unequalled as a single reference on how microcomputers work." (Microcomputing)

This best selling textbook details concepts that are common to all microprocessor systems. It is the most comprehensive and up-to-date introduction to microprocessor systems available anywhere.

320pp. **50A 349**

The Business Guide to Small Computers

L. Calmus

Ready to bring a small computer into your business? This book will show you how to determine what jobs you want it do do, what kinds of computers do those jobs and what they cost, what software is, what it does, and how to evaluate it, what extra equipment is available and what it can do for you, how to set up your system and how to get the most out of it. Even what space your system will need. You'll avoid a lot of mistakes, a lot of expense, and get more out of your computer.

240pp. **009662**

ASSEMBLY LANGUAGE PROGRAM–MING

The 6809 Primer

Assembly Language and Subroutines for the TRS-80 Colour Computer
K. Skier

Written in clear, concrete style, this is an excellent primer for the computer user who wants to understand and program in assembly language. Exploring the 6809 microprocessor it describes assembly language functions, then provides commonly used assembly language subroutines that are building blocks for virtually any application, including delays, character-conversion, string handling, block moves, sorting, and input/output for keyboard, CRT, RS-232, joystick, audio
A BYTE Book
280pp **057862**

6502 Assembly Language Programming

Lance A Leventhal

"The book that will probably get the reputation as being the 6502 bible" (Interface Age)

Provides an extensive range of programming examples, from simple memory load loops to complete design projects. A straight-forward, self-teaching book that's precise yet easy to understand.
640pp. **50A 276**

Z80® Assembly Language Subroutines

L.A. Leventhal & W Saville

Save yourself valuable programming time by using the more than 50 useful subroutines in this overview of assembly language programming for the Z80. They include array, bit, and string manipulation, code conversion, arithmetic operations, sorting and searching techniques, and much more. All routines have been thoroughly tested and debugged.
An OSBORNE Book
550pp. **50A 918**

6809 Assembly Language Programming

Lance A Leventhal

"Leventhal appears to have a formula for producing programming manuals If so it's a good formula ... and has produced another clear and thorough manual for the serious programmer" (American Mathematical Monthly)

This book covers 6809 assembly language programming in unmatched detail, and contains many fully debugged, practical programming examples with solutions in both object code and source code. Assembler conventions, I/O devices and interfacing methods are also covered.
576pp. **50A 357**

8080A/8085 Assembly Language Subroutines

Lance A. Leventhal & Winthrop Saville

"The text is beautifully designed with many many examples complete with source code. The routines included can actually be used to save hours of tedious development time Whether you program in assembly for fun or profit check this one out. You'll be impressed" (Lifelines/The Software Magazine)
500pp. **50A 586**

8080A/8085 Assembly Language Programming

Lance A. Leventhal

". . . an excellent encyclopedia of assembly language programming." (Byte)

More quality programming examples and instruction sets than can be found in any other book on the subject. Information on assemblers, program loops, code conversion and more. A must for 8080A/8085 programmers.
448pp. **50A 101**

Z80® Assembly Language Programming

Lance A. Leventhal

"There may never be a better book on the Z80 assembler than this one . . . " (Creative Computing)

Programming examples illustrate software development concepts and actual assembly language usage. More than 80 sample programming problems with solutions and a complete Z80® instruction set reference table.
640pp. **50A 217**

The 8086 Book

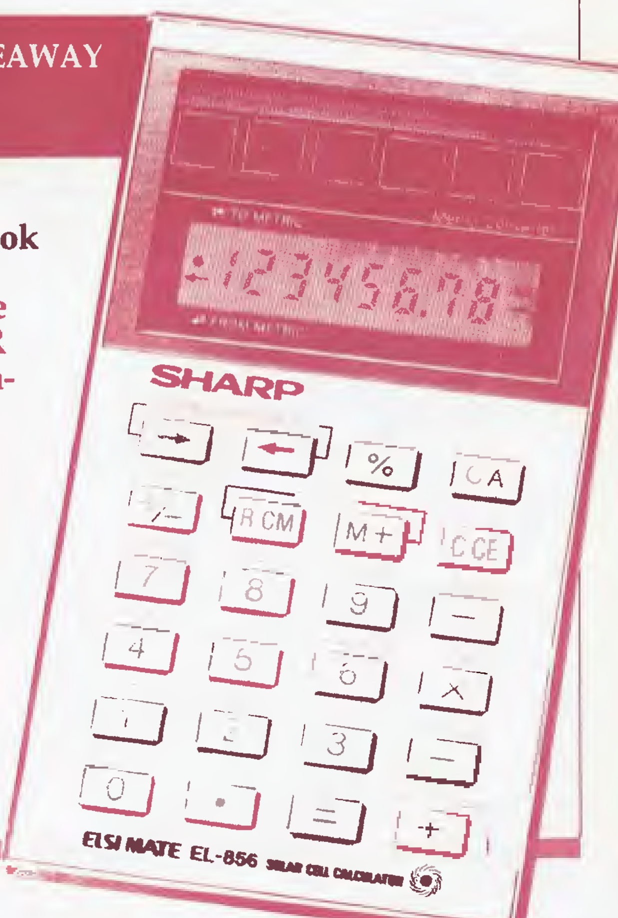
Russell Rector & George Alexy

"... is far superior to any other book about the 8086." (Dr. Dobbs Journal)

Anyone using, designing, or simply interested in an 8086-based system will be delighted by this book's scope and authority. As the 16-bit microprocessor gains wider inclusion in small computers, this book becomes invaluable as a reference tool which covers the timing, architecture and design of the 8086, as well as optimal programming techniques, interfacing, special features and more.
624pp. **50A 292**

SPECIAL BONUS GIVEAWAY HERE'S YOUR CHANCE TO WIN.

The McGraw-Hill Book Company offers you the chance to win one of 60 SHARP SOLAR CELL EL – 865 calculators. The first lucky sixty people to purchase any of the books displayed within this bulletin will receive this special bonus gift. THIS OFFER WILL NOT BE REPEATED.



Top-Down Assembly Language Programming for Your 6502 Personal Computer

K. Skier

Targeted at first time users of Apple, Atari, Ohio Scientific and PET computers, this book offers a guided tour of their computers and their 6502 microprocessors. Easy to understand, the discussions offer a thorough grounding in assembly language programming. The book also introduces subroutines and other programming techniques to make the computer more useful.
A BYTE Book
434pp. **057863**

Assembly Language Programming for the Apple® II

Robert Mottola

This comprehensive, easy-to-understand introduction provides many subroutines written in assembly language – and most explanations are also shown with equivalent examples in BASIC. An excellent section on hexadecimal arithmetic is included, as well as appendices for further study.
143pp. **50A 519**

68000 Assembly Language Programming

Lance A. Leventhal
Gerry Kane, Doug Hawkins

"This book is one of the most complete assembly-language manuals available . . . and it's written so that a person experienced with other assembly languages can quickly grasp all the concepts unique to the 68000." (Byte Magazine)

Each of the 68000's instructions is individually presented and fully explained. For new programmers, *68000 Assembly Language Programming* is an assembly language tutorial. For experienced programmers it is a complete reference to the 68000 instruction set and programming techniques.
624pp. **50A 624**

READY TO USE PROGRAM SERIES

Commodore 64™ Fun & Games

Ron Jeffries, Glen Fisher & Brian Sawyer

A collection of 35 games and puzzles guaranteed to provide hours of enjoyment for any C-64™ user. Play *Boswain Mad, Godzilla, Yahtzee, Leap, Zap, Blackjack,* and much more. Includes games that make use of the C-64's special graphics and sound capabilities. Each game is presented with complete instructions and BASIC program listings. All you have to do is key in the program listings and . . . PLAY!
192pp **50A 169**

Practical BASIC Programs

edited by Lon Poole

". . . a welcome sequel to the popular Some Common BASIC Programs. *The checkbook reconciliation, home budgeting and income averaging programs alone make this book a useful investment." (Interface Age)*

Useful in small business and household applications, here are 40 fully documented programs designed to run on most micro-computers. Easy-to-follow format requires no previous programming knowledge
171pp.

Text	**50A 38-1**
IBM PC Edition	**50A 80-2**
Apple® II Edition	**50A 66-7**

Science and Engineering Programs for the Timex/Sinclair 1000

C.R Lewart

These 25 useful and original programs designed to run on the TS1000, cover a wide range of topics such as electrical engineering . . . data transmission . . . number theory . . probability and statistics . . . operations research . . . pure and applied math and more. The documentation included makes it simple to alter any program to suit your own needs.
120pp. **037444**

Some Common BASIC Programs/Apple® II Edition

Lon Poole, Mary Borchers & David M. Castlewitz

You can key these 76 short programs directly into your Apple® II and own a powerful collection of applications that will provide a wealth of resources for any programming library.
200pp **50A 683**

Some Common BASIC Programs

Lon Poole & Mary Borchers

76 well-designed and painstakingly documented programs that help you solve a variety of problems in statistics, finance, and math. Easy to use at any level of experience Generalized BASIC edition.
196pp. **50A 063**

Atari® 400/800 Edition	**50A 535**
IBM® P/C Edition	**50A 837**

Science & Engineering Programs/ Apple® II Edition

edited by John Heilborn

An important collection of programs written in Applesoft® BASIC which allows researchers to run highly complex, often tedious mathematical calculations on an Apple® II with a minimum of 16K memory. For those who don't have an Apple® II, there is a section that discusses methods of modifying the programs to run on other computers.
223pp. **50A 63-2**

Practical Pascal Programs
by Lon Poole

These programs take full advantage of the structured programming capabilities of Pascal. Written in UCSD Pascal® and designed to conserve your valuable programming time, the included programs have primary applications in the areas of finance, science, math, management, and statistics.
206pp. **50A 748**

Some Common Pascal Programs
Lon Poole, Mary Borchers & Greg Davidson

These Pascal conversions of our 116 popular BASIC programs for the home, school, and office are also useful as Pascal learning tools. The programs will run with any version of Pascal you now use.
235pp. **50A 73-X**

TECHNICAL REFERENCE

Practical Aspects of Data Communications
P.S. Kreager

A network manager or supervisor who wants to improve both the networking and facility environment for data communications will find a wealth of information here. Avoiding technical theory, the book plunges into practical information for implementing and supporting local communications networks. Coverage includes special cable and adapter assemblies, network and facility components, and common-sense conventions and practices. There's helpful discussion on support personnel, too.
256pp. **035429**

McGraw-Hill's Compilation of Data Communications Standards, 2/e
H.C. Folts

Here is an orderly, usable consolidation of current data communications standards required to make individual user applications, equipment, and transmission systems work together smoothly. The standards represented here come from the most important organizations in the field including CCITT, ISO, ANSI, EIA and others. This resource will save all organizations using data communications equipment or networks both time and money.
1923pp. **021457**

Computers Today
D.H. Sanders

Whatever you want to learn about computer literacy you'll find here, in this thoughtful discussion of the impact of computers on people. Thorough coverage of software management, hardware, the computer and society, applications, and flowcharting. Bonus: 30 simple programs in BASIC, COBOL, and FORTRAN involve you in the business basics of accounts receivable, accounts payable, and inventory control.

Contents. Computer Concepts: An Introduction. Computers at Work: An Overview. Putting the Computer to Work A summary. The Impact of Computers at Work A Preview. Computer Hardware Systems. Central Processor Concepts Codes, and Components Computer Data Entry. Computer Secondary Storage. Computer Output. Micros Minis, Mainframes and "Monsters". Data Communications and Distributed Data Processing Networks. Word Processing and Electronic Mail/Message Systems Programming Analysis. Preparing Computer Programs: An Overview. Programming in Basic. Operating Systems: Concepts and Functions Information Systems Concepts System Analysis Design and Implementation Considerations. System File and Processing Alternatives Management Information Systems The Impact of Computers on People. The Impact of Computers on Organizations. Computers Tomorrow
672pp. **054681**

The McGraw-Hill Computer Handbook
H.L. Helms, Jr., Ed.

Here's the everything-between-two-covers reference tool the computer world has been waiting for! Written by a staff of experts, it answers virtually any question of either the experienced user or the beginner. Covers mainframes, minis, micros, hardware and software, languages, I/O, operating systems, databases, files, and much more, including voice recognition, graphics plotters, and local networking. Inexhaustible and indispensable.
1200pp. **027972**

Data Communications for Microcomputers
Practical Experiments for Z80 Based Microcomputers
E.A. Nichols, J.C Nichols & K.R. Musson

If you want to set up communications between your microcomputer and peripheral equipment or another system, you'll welcome this information-packed guide. It covers everything from constructing a cable to writing input/output software that's virtually foolproof. Included are hands-on experiments, charts, tables, standards, specifications, and examples. Systems-dependent experiments are based on the TRS-80 and Z80 CP/M systems but can easily be converted for use on others.
352pp. **046480**

The Local Network Handbook
G.R. Davis, Ed.

From the pages of *Data Communications* magazine come these 37 articles divided into six sections to address every aspect of local networking. Read about technology software, equipment, implementation, application, and selection. A must for every serious practitioner or student of data communications, these carefully selected articles will help narrow the information gap between the real world and the marketing claims by vendors.
256pp. 015823

Osborne 16-Bit Microprocessor Handbook
Gerry Kane & Adam Osborne

An authoritative reference book on virtually every 16-bit microprocessor, including the 2900 Chip Slice Family, this book permits objective evaluation and comparison of these new devices.
800pp. **50A 438**

An Implementation Guide to Compiler Writing

J.P. Tremblay & P.G. Sorenson

This useful and versatile supplement to *The Theory and Practice of Compiler Writing* offers an extended case study of the development of a simple programming language, GAUSS, and the implementation of a compiler for this language. This practical book illustrates a type of documentation similar to what you should produce for your own compiler. The book also helps you anticipate areas of potential difficulty that are often encountered when implementing a compiler.

160pp. **065166**

Microprocessors for Measurement and Control

D.M. Auslander & P. Sagues

You'll learn to design mechanical and process equipment using microprocessor-based "real time" computer systems. This book presents plans for prototype systems which allow even those unfamiliar with machine or assembly language to initiate projects.

310pp. **50A 578**

Interfacing to S-100 (IEEE 696) Microcomputers

Sol Libes & Mark Garetz

"It is a goldmine of useful information . . ." (Microsystems)

This book is a must if you want to design a custom interface between an S-100 microcomputer and almost any type of peripheral device. Mechanical and electrical design is covered, along with logical and electrical relationships, bus interconnections and more.

322pp. **50A 373**

The McGraw-Hill Encyclopedia of Electronics and Computers

The Staff of the McGraw-Hill Encyclopedia of Science and Technology

This unique, authoritative, and comprehensive reference contains 477 alphabetically arranged articles that treat the full spectrum of information pertaining to the theory, devices, and applications of electronics, electricity and computers. Entries span such diverse and important topics as: microcomputers, programming languages, computer storage technology, and electronic circuit theory. It also covers: electronic displays, character recognition, artificial intelligence, lasers, electroacoustics, and much more.

960pp. **045487**

Interfacing to Microprocessors

J.C. Cluley

If you design with microprocessors, here's a complete guide to putting them to work for you. Details on data transfer, device addressing, a microcomputer data-bus standards, serial and parallel interfacing, and memory-mapped I/O methods. Special attention is paid to transducers and signal conversion methods. Interfacing devices such as Intel 8255 and Motorola M6821 are discussed and used in design examples. You'll find the latest data on the 16-bit M68000 and I8086.

160pp. **011409**

A Programmer's View of the Intel 432 System

E. Organick

You can harness the power of the Intel 432, the microprocessing unit with the computing ability of a mid-range mainframe. This book shows you how the 432 system caters to systems development based on modular programming; and how its object-based architecture and hardware, iMAX, the I/O system, and the object filing system reduce the cost of building and maintaining correct systems.

252pp. **047719**

Structured Analysis and Design of Information Systems

K.O. Wigander et al

The growing complexity of today's information systems can be intimidating, but you can simplify your professional life with this structured development model. It is practical and user-oriented, encompassing the entire system-development life cycle, from feasibility study to programming. Developed in Sweden, its graphical presentation is powerful and simple to use, and its techniques have earned top praise from many major U.S. companies.

352pp. **015061**

Interactive Programming Environments

D.R. Barstow, H.E. Shrobe & E. Sandewall

Bring yourself quickly up to date on major advances in interactive programming environments. Focus of this volume is on the interplay between artificial intelligence, programming methodology, and software engineering. Written for programmers, software engineers, and managers of software engineering projects, this volume includes several reports published here for the first time, plus structure editors, program refinements systems, and display-oriented programming environments.

480pp. **003885**

The Small Computer Connection

Telecommunications for the Home and Office

N.L. Shapiro

Link your home or small-business computer to other small computers and mainframe computers through telephone lines – even if you're not a computer professional. Step-by-step you will learn how to use a computer as a terminal, setting up your own telecommunications network with easily obtainable hardware and software. It's the only how-to guide of its kind – and it's written in clear, easy-to-follow jargon-free language!

A BYTE Book

256pp. **056412**

8086/8088 16-Bit Microprocessor Primer

L. Morgan & M. Waite

Learn why the power-packed 16-bit microprocessor (four to eight times faster than previous 8-bit units) is significant, and the impact these devices are having. You get an overview of the typical 16-bit microprocessor, complete with basic concepts, structures, and processes, and detailed information on coprocessing, alter-ego processors, and supply chips. There are pinouts, instruction sets, block diagrams, and a detailed description of the Intel 8086/8088 chips.

A BYTE Book

224pp. **043109**

Software Engineering Design, Reliability, and Management

M. Shooman

Escort yourself through a valuable learning process with this lucid account of the philosophy and techniques of software engineering. You'll be introduced to all the analytical techniques useful to developers of software designs, including models and examples previously available only in research literature. From the many design representation schemes that punctuate the book, you'll be able to express designs in several widely used notations.

700pp. **057021**

VLSI Technology

S.M. Sze, Ed.

Offering the most current and thorough coverage of all major VLSI topics, this book discusses all the important steps in the fabrication of VLSI circuits from crystal growth and film deposition through lithography and dry etching to packaging and reliability testing. For use as a reference by process and device engineers, material scientists, and students, it contains 400 technical illustrations and photos, and 40 tables of parameters.

672pp. **062686**

McGraw-Hill
COMPUTER BOOK CLUB

21 DAYS FREE EXAMINATION ORDER FORM

IMPORTANT: When ordering, please return this complete page.

THE Mc-GRAW-HILL COMPUTER BOOK OFFER GUARANTEE

It's simple. The Mc-Graw-Hill Computer Book Club offers you the opportunity to build your own computer book collection AT SUBSTANTIALLY REDUCED RATES. If you are not already a member, simply complete the details below and return this page to us FREE POST (No postage stamp required).
YOUR MEMBERSHIP BEGINS IMMEDIATELY.

☐ **YES!** Yes please enrol me now as a member of the McGRAW-HILL COMPUTER BOOK CLUB. I understand that my membership is FREE and that I am immediately eligible for all exclusive membership benefits. I also understand that I may resign at any time I wish. Please let me inspect the book(s) I have ticked ☑ FREE AND ON APPROVAL FOR 21 DAYS. At the end of that period I will remit for the book(s) I keep and return unwanted books post paid.

HOW TO CALCULATE YOUR ORDER

DEDUCT YOUR OWN DISCOUNTS

10% OFF – Purchases under $50
15% OFF – Purchases Over $50
20% OFF – Purchases Over $100

OR

You may buy any book(s) from the "MAIN SELECTION" and select FREE OF CHARGE, another book from the "PREMIUM SELECTION" on reverse of this pg

MAIN SELECTIONS

AUTHOR'S NAME	TITLE	CODE	PRICE
Anbarlian	Spreadsheeting Software	☐ 001597	$86.45
Anbarlian	Spreadsheeting Manuals	☐ 001605	$41.45
Anbarlian	Spreadsheeting Manuals	☐ 001699	$41.45
Auslander	Microprocessors for Measurement & Control	☐ 50A578	$27.20
Barstow	Interactive Prog. Environments	☐ 003885	$66.10
Calmus	The Bus. Guide to Small Comps.	☐ 009662	$38.00
Castlewitz	Visicalc Program Made Easy	☐ 50A896	$22.05
Castlewitz	Visicalc: Home & Office Companion	☐ 50A500	$27.20
Cluley	Interfacing to Microprocessors	☐ 011409	$45.60
Dahmke	The Byte Guide to CP/M	☐ 015072	$30.55
Datapro	Datapro/McGraw-Hill Guide to CP/M Software	☐ 015404	$34.55
Datapro	Datapro/McGraw-Hill Guide to IBM/PC Software	☐ 015424	$34.35
Datapro	Datapro McGraw-Hill Guide to Apple Software	☐ 015403	$35.95
Davis	Fortran 77: A Structured	☐ 015903	$41.35
Davis	The Local Network Handbook	☐ 015823	$50.90
Ditlea	The Osborne/McGraw-Hill Home Computer Software Guide	☐ 50A07X	$20.35
Etlin	Word Star Made Easy (2nd Ed.)	☐ 50A90X	$22.05
Ettlin	The M Basic Handbook	☐ 50A029	$30.55
Flast	54 Visicalc Models: Finance – Statistics – Mathematics	☐ 50A002	$27.15
Flast	54 Supercalc Models: Finance – Statistics – Mathematics	☐ 50A185	$27.15
Folts	McGraw-Hill Compilation	☐ 021457	$519.75
Fox	Armchair Basic: An Absolute Beginner's Guide to Prog. Basic	☐ 50A926	$20.35
Graham	Your IBM PC: A Guide to the IBM PC (DOS 2.0) XT	☐ 50A207	$30.55
Graham	Your IBM PC: A Guide to the IBM PC	☐ 50A853	$30.55
Hancock	The C Primer	☐ 025981	$30.55
Heilborn	Science & Engineering Programs	☐ 50A632	$27.20
Heilborn	Your Commodore 64: A Guide to the C 64 C	☐ 50A142	$25.45
Heilborn	Vic 20 User Guide	☐ 50A861	$27.15

AUTHOR'S NAME	TITLE	CODE	PRICE
Helms	The McGraw-Hill Computer Handbook	☐ 027972	$144.70
Hogan	Discover Forth	☐ 50A799	$25.50
Jeffries	Commodore 64 Fun & Games	☐ 50A169	$20.35
Kane	Osborne 16-Bit Microprocessor Handbook	☐ 50A438	$37.35
Kolve	How to Buy (and Survive) Your First Computer	☐ 035130	$26.95
Kreager	Practical Aspects of Data Commun.	☐ 035429	$61.30
Kruglinski	Data Base Management Systems: A Guide to M S	☐ 50A845	$28.85
Johnson	The Programmer's CP/M Handbook	☐ 50A037	$37.35
Leventhal	68000 Assembly Language Prog.	☐ 50A624	$32.25
Leventhal	6502 Assembly Language Prog.	☐ 50A276	$32.25
Leventhal	6809 Assembly Language Prog.	☐ 50A357	$32.25
Leventhal	8080A/8085 Assembly Language Subroutines	☐ 50A586	$30.55
Leventhal	Z80 Assembly Language Programming	☐ 50A217	$32.25
Leventhal	Z80 Assembly Language Subroutines	☐ 50A918	$30.55
Leventhal	8080A/8085 Assembly Language Programming	☐ 50A101	$32.25
Lewart	Science & Engineering Programs for the Timex/Sinclair 1000	☐ 037444	$25.15
Libes	Interfacing to S-100 (IEEE 696) Microcomputers	☐ 50A373	$25.50
McGilton	Introducing the Unix System	☐ 045001	$34.15
McGraw-Hill	The McGraw-Hill Encyclopedia of Electronics & Computers	☐ 045487	$95.25
Miastkowski	The Osborne/McGraw-Hill Guide to Apple III	☐ 50A010	$30.55
Morgan	8086/8088 16-Bit Microprocessor Primer	☐ 043109	$37.75
Mottola	Assembly Language Prog. for the Apple II	☐ 50A519	$27.15
Mullish	Basics: A Guide to the Timex/Sinclair 1000 (2K)	☐ 044041	$17.95
Naiman	Word Processing Buyer's Guide	☐ 045869	$28.75

CR01

MAIN BOOK SELECTION CONTINUED

AUTHOR'S NAME	TITLE	CODE	PRICE
Nichols	Data Commun for Micro-comp	☐ 046480	$30.85
Organick	A Programmer's View of the Intel 432 System	☐ 047719	$62.30
Osborne	An Introduction to Micro-computers: Vol 0 The Beg. Book	☐ 50A640	$25 45
Osborne	CBM Professional Computer Guide	☐ 50A756	$25.50
Osborne	An Introduction to Micro-computers: Vol 1 Basic Concepts	☐ 50A349	$28.85
Osborne	C Basic User Guide	☐ 50A616	$30 55
Osborne	Pet Personal Computer Guide	☐ 50A764	$25 50
Peckham	Hands-On Basic for the Apple II	☐ 049179	$34.55
Peckham	Hands-On Basic for the Atari 400, 800 & 1200	☐ 049194	$36 35
Poole	Some Common Basic Programs Apple II Ed	☐ 50A683	$28.85
Poole	Some Common Basic Programs	☐ 50A063	$28 85
Poole	Some Common Basic Programs/ Atari 400/800 Ed.	☐ 50A535	$28.85
Poole	Some Common Basic Programs IBM Personal Computer Edition	☐ 50A837	$25.50
Poole	Practical Basic Programs IBM PC Ed.	☐ 50A802	$28.85
Poole	Some Common Pascal Programs	☐ 50A73X	$28.85
Poole	Apple II User's Guide for the Apple II and IIe	☐ 50A045	$30.55
Poole	Practical Basic Programs/ Apple II Ed.	☐ 50A667	$28.85
Poole	Your Atari Computer: A Guide to	☐ 50A659	$31.05

AUTHOR'S NAME	TITLE	CODE	PRICE
Poole	Practical Basic Programs (Text)	☐ 50A381	$28.85
Poole	Practical Pascal Programs	☐ 50A748	$28.85
Rector	The 8086 Book	☐ 50A292	$28.90
Sachs	The HHC User Guide	☐ 50A87X	$25.45
Sachs	Your IBM PC Made Easy	☐ 50A126	$22.05
Sanders	Computers Today	☐ 054681	$46.75
Shapiro	The Small Computer Connection:	☐ 056412	$28.75
Shooman	Software Engineering, Design, Reliability and Management	☐ 057021	$66.55
Skier	The 6809 Primer	☐ 057862	$21.55
Skier	Top-Down Assembly Language Prog. 6502 Personal Computer	☐ 057863	$30.55
Skier	Top-Down Basic for the TRS-80 Colour Computer	☐ 057861	$26.95
Sze	VLSI Technology	☐ 062686	$66.15
Thomas	A User Guide to the Unix System	☐ 50A713	$30.55
Townsend	Using D Base II	☐ 50A088	$32.25
Tremblay	An Implementation Guide to Compiler Writing	☐ 065166	$23.35
Tymes	Super Calc Home & Office Companion	☐ 50A134	$27.15
Waite	Graphics Primer for the IBM PC Disciplined Style 2/e	☐ 50A993	$37.35
Watt	Learning with Logo	☐ 068570	$41.35
Watt	Learning with Apple Logo	☐ 068571	$26.95
Webster	Micro Computer Software Buyer's Guide	☐ V68967	$24.95
Wigander	Structured Analysis & Design	☐ 015061	$56.65

☐ **YES!** Having calculated my order value, for one or more books I have elected to opt for the applicable discount.

☐ 10% OFF – Purchases under $50
☐ 15% OFF – Purchases Over $50
☐ 20% OFF – Purchases Over $100

OR ☐ I have chosen to forgo my discount, and have selected this book FREE OF CHARGE from the selection below. ☑ Tick your book selection.

PREMIUM SELECTIONS

AUTHOR'S NAME	TITLE	CODE	PRICE
Buchsbaum	Microprocessor-based Elec. Games	☐ 008722	$17 95
Charlton	Bogglers T/S Basic	☐ 023959	$17 95
Chiu	Crunchers T/S 1000 (2K)	☐ 010831	$16 15
D'Ignazio	How to Get Intimate With Your Computer	☐ 016901	$12 65
Foyt	The ZX81/TS1000 Home Comp. Book	☐ 50A061	$13.55
Gaby	Gosubs	☐ 022677	$17 95
Gifford	Apple II Disk Guide	☐ 50A969	$13.55
Helms	The Basic Book	☐ 027959	$13 75

AUTHOR'S NAME	TITLE	CODE	PRICE
Hohenstein	All About Hand-Held & Briefcase PCs	☐ 029452	$17.95
Ingraham	CP/M Disk Guide	☐ 50A977	$15.25
Kane	CRT Controller Handbook	☐ 50A454	$16.95
Kane	68000 Microprocessor Handbook	☐ 50A411	$16.95
Mullish	Basics A Guide to the Timnex/S	☐ 044041	$17.95
Osborne	8089 I/O Processor Handbook	☐ 50A39X	$17 95
Taylor	Atari 400/800 Disk Guide	☐ 50A950	$13.55
Wilson	IBM PC Disk Guide	☐ 50A942	$15 25
Wilson	Visicalc Disk Guide	☐ 50A985	$11.85

NAME ....

POSITION ....

COMPANY ....

....
(PLEASE INDICATE STREET ADDRESS, NOT BOX NO.)

CITY/SUBURB: ....

STATE ....POSTCODE: ....

IF AFTER 21 DAYS INSPECTION I DECIDE TO KEEP ANY OF THESE BOOKS I PREFER TO PAY BY ...

☐ CHEQUE ☐ AMERICAN EXPRESS ☐ BANKCARD

CARD NO ....

EXPIRY DATE: ....

SIGNATURE ....

Having calculated your order, now add a total of $3 50 for Freight/Handling

Please envelope this complete page and address as follows.

NO STAMP REQUIRED IF ADDRESSED TO: **FREEPOST NO. 2.**
McGRAW-HILL BOOK COMPANY AUSTRALIA PTY LIMITED*
PO BOX 239 4 BARCOO STREET ROSEVILLE NSW.
2069 PHONE: (02) 406 4288. *Incorporated in NSW.

Maybe their Cat does eat Apples . . . but our LINGO licks CATS!!

$2160
INCLUDING MONITOR
TAX PAID COURIER
DELIVERY AVAILABLE

	Cat	Lingo 128	Apple IIe
Processor	6502a	6502	6502
Z80 processor for CP/M built-in	No	YES	NO
RAM memory	64K	128K	64K
Enhanced microsoft basic	YES	YES	NO
Size of interpreter in ROM	24K	24K	12K
Numeric keypad	YES	YES	NO
80 column text display built-in	YES	YES	NO
Voice synthesizer included	NO	YES	NO
RGB colour output standard	YES	YES	NO
RS232 serial adapter included	NO	YES	NO
Centronics printer port included	YES	YES	NO
Twin drives built-in	NO	YES	NO
Whisper quiet Chinnon Japanese drives	NO	YES	NO
Numeric keypad	YES	YES	NO
Separate processor for keyboard	YES	YES	NO
RF modulator included	NO	YES	NO

SO LET'S COMPARE PRICES

Apple IIe: Cost of computer with 80 column facility, RGB color and printer port, single disk drive controller DOS and hi-res screen monitor. (average of quoted prices)			$3170
Cat: Configured to same level	$1485		
Extras to bring Cat to level of Lingo:			
Second disk drive	$349		
Emulator cartridge	$99		
RF modulator	$35		
Z80 second processor	$449		
Extra 64K memory say	$120		
RS232C serial adapter	$129		
Cost of Lingo		$1950	
Amber high res monitor		$210	
	$2666	$2160	$3170

SAVING LINGO OVER CAT: $506

And of course the Lingo also comes with 128K DOS 3.3 and 128K CP/M and Pascal and software to use it's powerRAM disk. Oh yes, and about that souped up 6502A. May it runs faster than the 6502 but we decided that nearly all business programmes of consequence run on CP/M with the Z80. As far as leisure programmes go, have you ever tried to fight off space invaders coming at you at twice the speed? **Good Luck!**

COME TO COMPAK WHERE YOU CAN CHOOSE THE BEST!!!

COMPAK
Compak Computer Centre
THE HOME OF THE 816

**350 SOUTH ROAD,
MOORABBIN 3189
Telephone: (03) 555 9844**

Record, then press any key. The routine creates a fresh copy of your program which should load more easily.

F O'Dwyer

```
10 DIM FILE$(FRE(0)-100):FILE$="":CLOSE
   #1:OPEN #1,4,0,"C:":TRAP 30
20 GET #1,C:FILE$(LEN(FILE$)+1)=CHR$(C):
   PRINT CHR$(27);CHR$(C);:GOTO 20
30 IF PEEK(195)=5 THEN PRINT: PRINT
   "OUT OF MEMORY - SORRY":END
40 IF PEEK(195)=136 THEN CLOSE #1:OPEN
   #1,8,0,"C:":PRINT #1;FILE$;:END
50 TRAP 30: GOTO 20
```

Commodore in quotes

When using INPUT# to pick up previously recorded text data on Commodore computers, problems may arise if the text contains colons or commas. The problem also arises from using INPUT from the keyboard. In this case the computer responds with the extra ignored message.

The solution with INPUT is to type a set of quotes in front of any typed data and for INPUT# to record the data with preceding quotes: PRINT#X, CHR$(34)D$ Where X is the file number and D$ is the string to be recorded.

S Rodgers

VZ-200 trace

In the July edition of *APC*, J Williams suggested a method for printing a moving message across the bottom of the Comodore 64 screen. I modified this for the VZ-200:

```
5  CLEAR 1000
10 A$="PUT MESSAGE
   HERE":REM LET A$ BE
   MESSAGE
15 PRINT@480," "
20 PRINT LEFT$(A$,31);
25 PRINT CHR$(27);:REM
   MOVES CURSOR UP
30 FOR I=1 TO 40:
   NEXT:REM: DELAY
35 A$=MID$(A$,2)+
   LEFT$ (A$,1):GOTO 25
```

A friend also told me of a tracing function for the VZ-200:

POKE 31003,175 starts trace function and prints line numbers

POKE 31003,0 disables this function.

The only problem is with MODE(1), the screen returns to MODE(0) to print line numbers and you don't get to see what is happening in high-res graphics.

Jay Batterson

Giving Atari text colour independence

Atari graphics modes 1 through to 8 are split screen displays consisting of a graphic window above a 4-line text window.

Generally, the graphics window makes use of colour registers 0 through 4; in addition, the text window uses register 1 for character luminance and 2 for its background colour.

This restricts the use of registers 1 and 2 because any change in them affects both windows.

The program here sets up a display list interrupt routine which changes registers 1 and 2 for the text window thus making its colours independent of the graphics window.

In the routine: location 1540 holds the text window's background colour, and location 1542 holds the text window's character luminance.

N Pearce

```
900 ? "GRAPHICS MODE (1-8)";:INPUT G: GRAPHICS G
910 DLIST =PEEK(560)+256*PEEK(561)
920 FOR I=DLIST+6 TO DLIST+201
930 IF PEEK(I)<>66 THEN NEXT I:END
940 POP :POKE I-1,PEEK(I-1)+128
950 FOR I=0 TO 19 : READ J:POKE 1536+I,J:NEXT I
960 DATA 72,138,72,162,50,169,12,141,10,212
970 DATA 141,23,208,142,24,208,104,170,104,64
980 POKE 512,0:POKE513,6:POKE 54286,192
```

ML embeding in Basic

This tip is mainly for TRS-80/System 80 users but could possibly, with the right modifications, suit other computers. Any suggestions?

Basically, I have developed a neat way of including machine language routines within Basic programs, thus eliminating the need to either load them separately (as with the 'APC-80' routine), or wait while the program creates the routine from data statements. I did this because I often require the speed of machine language for such functions as graphics handling, sorting or communications, but prefer to write the program in Basic to allow easier modification.

The big advantage of this method is that I can store popular routines on tape, and when I wish to write a program which will make use of the routine, I firstly load it from tape and then add the program to it. The combined program and routine can then be saved, reloaded and/or modified at will, with the routine instantly accessible.

The concept is simple — instead of POKEing the routine data into high memory (and remembering to reserve that memory before loading the program), it is instead POKEd into line 0 of the Basic program, 0 being chosen such that subsequent modification of the program will not relocate the routine. Now for the catch. The TRS-80's Basic interprets any byte of 0 in the program storage area as being the end of a program line, thus any byte of 0 in the routine will play havoc with the program. But this can be overcome by following these simple guidelines:

(1) Do not use the instructions 'NOOP" (=00 H) or "RLC B" (=CB00 H)

(2) Do not use any single-byte value of 0 in an instruction. eg, LD A,0 , CP 0 , ADD A,0

(3) Do not use any dual-byte value containing 0, ie, any dual-byte value less than 256, or any exact multiple of 256. eg, LD HL, 15360 (=3C00 H) , CALL 002B H , JR NZ,18944 (=4A00 H)

(4) Do not address IX or IY directly. ie, LD A,(IY+0)

(5) Do not do a relative jump to the next instruction. (Pointless but possible) eg, JR NZ,.+2

(6) If the routine itself contains data statements, such as graphics data, obviously no data byte can be 0.

Thus it can be seen that for the most part there are no worries except where single or dual byte values containing 0 are concerned. Instructions like "CP 0" or "ADD A,0" can be ignored as they are of no use anyway. The solution for those like "LD B,0" is to substitute either "LD B,255 ... INC B" or "XOR A ... LD B,A". The same trick can also be used on dual byte values. Note that if the flags are to remain unchanged it is necessary to save the AF register pair prior to doing so.

A second technique is especially useful for screen handling. As the screen on the TRS-80 resides from 15360 to 16383 (3C00 H to 3FFF H), it is not possible to address the start of the screen without implementing the trick just mentioned. Therefore, when a full screen of data is to be accessed or modified in some way, it is often easier to begin with the last byte of the screen and work backwards: "LD HL,16383 ... ... DEC HL"

The only other point to watch for is a call or jump to a memory location whose address contains 0. eg, "JP 18944" (=4A00 H). This is merely a matter of watching where your subroutines lie, and not calling any ROM routines below 255 (=00FF H)

Having overcome the hard part, I will now demonstrate the technique using a practical example, in this case a "reverse video" graphics demo. This simple routine reverses all the graphics on the screen while leaving any alphanumerics unchanged — very useful for "flashing" game titles, etc. The original routine is shown, followed by the routine modified according to the above rules. Fig 1 shows how for simple routines, simple solutions are possible.

```
                00000 ;ORIGINAL GRAPHIC REVERSE VIDEO DEMO BEFORE MODIFICATION.
                00010 ;
42F1            00020        ORG    17137       ;----------------------
42F1 21003C     00100 REVERS LD     HL,15360    ;GET READY TO MOVE SCREEN
42F4 113075     00110        LD     DE,30000    ;TO FREE MEMORY SPACE.
42F7 010004     00120        LD     BC,1024     ;----------------------
42FA E5         00130        PUSH   HL          ;SAVE ALL REGISTERS AS
42FB D5         00140        PUSH   DE          ;ALL THEIR VALUES WILL BE
42FC C5         00150        PUSH   BC          ;USED LATER TO MOVE BACK.
42FD C5         00160        PUSH   BC          ;----------------------
42FE EDB0       00170        LDIR               ;MOVE SCREEN TO SPACE AND
4300 C1         00180        POP    BC          ;RETRIEVE SCREEN LENGTH
4301 1B         00190 LOOP   DEC    DE          ;& WORK BACK THROUGH THE
4302 1A         00200        LD     A,(DE)      ;SCREEN BYTES IN SPACE
4303 FE80       00210        CP     128         ;----------------------
4305 3802       00220        JR     C,ALPHA     ;IF BYTE IS GRAPHIC, THEN
4307 EE3F       00230        XOR    63          ;REVERSE VIDEO. REPLACE
4309 12         00240 ALPHA  LD     (DE),A      ;BYTE IN FREE SPACE.
430A 0D         00250        DEC    C           ;----------------------
430B 20F4       00260        JR     NZ,LOOP     ;COUNT DOWN SCREEN LENGTH
430D 10F2       00270        DJNZ   LOOP        ;IN BC TILL ALL REVERSED.
430F C1         00280        POP    BC          ;----------------------
4310 E1         00290        POP    HL          ;RETRIEVE VALUES SO DATA
4311 D1         00300        POP    DE          ;CAN BE RETURNED TO VIDEO
4312 EDB0       00310        LDIR               ;----------------------
4314 C9         00320        RET                ;& RETURN TO BASIC
42F1            00330        END    REVERS      ;----------------------
00000 TOTAL ERRORS

ALPHA    4309
LOOP     4301
REVERS   42F1


                [illegible] ; UPDATED ROUTINE WITH MODIFICATIONS ACCORDING TO RULES.
                00010 ;
42F1            00020        ORG    17137       ;----------------------
42F1 21FF3F     00100 REVERS LD     HL,16383    ; **** MODIFICATION ****
42F4 113075     00110        LD     DE,30000    ;
42F7 01FF03     00120        LD     BC,1023     ; **** MODIFICATION ****
42FA 03         00125        INC    BC          ; ****   ADDITION   ****
42FB E5         00130        PUSH   HL          ;
42FC D5         00140        PUSH   DE          ;
42FD C5         00150        PUSH   BC          ;
42FE C5         00160        PUSH   BC          ;
42FF EDB8       00170        LDDR               ; **** MODIFICATION ****
4301 C1         00180        POP    BC          ;
4302 13         00190 LOOP   INC    DE          ; **** MODIFICATION ****
4303 1A         00200        LD     A,(DE)      ;
4304 FE80       00210        CP     128         ;
4306 3802       00220        JR     C,ALPHA     ;
4308 EE3F       00230        XOR    63          ;
430A 12         00240 ALPHA  LD     (DE),A      ;
430B 0D         00250        DEC    C           ;
430C 20F4       00260        JR     NZ,LOOP     ;
430E 10F2       00270        DJNZ   LOOP        ;
4310 C1         00280        POP    BC          ;
4311 E1         00290        POP    HL          ;
4312 D1         00300        POP    DE          ;
4313 EDB8       00310        LDDR               ; **** MODIFICATION ****
4315 C9         00320        RET                ;
42F1            00330        END    REVERS      ;----------------------
00000 TOTAL ERRORS

ALPHA    430A
LOOP     4302
REVERS   42F1
```

Figure 1

The assembled routine converts to the following decimal data statements: 33,255,63,17,48,117,1, 255,3,3,229,213,197,197, 237,184,193,19,26,254, 128,56,2,238, 63,18,13, 32,244,16,242,193,225, 209,237,184,201.

To enter the routine into the program, it is first necessary to create a line with the space to house it. To do this, write line number 0 as follows, with the number of asterisks equal to the length of the routine (in this case 37).

0 GOTO10:**************** *********************

10 INPUTA:POKE17 137+B,A:B=B+1: GOTO10

Now run the program, typing in the data statements one by one, and press "BREAK" when all 37 have been entered. Line number 10 can now be deleted, and the program written from line 10 onwards. If all has gone well, the routine now resides from 17137 onwards; the familiar "X=USR(N)" is used in your program to call the routine after telling the computer where it resides by including "POKE 16526,241: POKE16527,66" somewhere, preferably at line 10.

One word of warning now that the routine is finished — although the rest of the program can be modified at will, never edit line 0, as the editor routine could destroy your routine as it attempts to compact what it considers to be a Basic program line. If you must see the garbage you have created, use "LIST 0". A second point arises from that — you may find that listing the program firstly fills the screen with garbage from line 0 and then distributes the rest of the program listing through it, making the whole lot unreadable. To alleviate this, either list from line 10 onwards, or do as I do. I add two data bytes to the end of my routine which home the cursor and clear the screen (data bytes 28 & 31), remembering to make line 0 two asterisks longer initially. Listing will then "begin" from line 10.

Well, that's about all there is to it. Oh, you want a demo of *MY* routine? OK then, here's a quickie . . .

Ian Fieggen

```
0 GOTO10:*I<A?0u|<APEEK-INSTRINSTRMKS$CLEARUSR!85END8MKO$?
 STR$FIXUSRCOS(MKS$CLEARINKEY$
10 POKE16526,241:POKE16527,66:CLS
20 POKERND(1024)+15359,RND(255):X=USR(0):GOTO20

THE ABOVE PROGRAM SHOWS WHAT THE ROUTINE DOES. WHILE BELOW IS AN APPLICATION.
NOTE THAT THE GARBAGE IN LINE #0 IS THE BASIC INTERPRETATION OF THE DATA.

0 GOTO10:*I<A?0u|<APEEK-INSTRINSTRMKS$CLEARUSR!85END8MKD$?
 STR$FIXUSRCOS(MKS$CLEARINKEY$
10 POKE16526,241:POKE16527,66
20 FORA%=15360TO16383:POKEA%,128:NEXT:A%=45:B%=36
30 FORN%=1TO130:SET(48*SIN(N%/A%)+64,23*COS(N%/B%)+24):NEXT
40 X=USR(0):FORA=1TO30:NEXT:GOTO40
```

64 is ahead of its type

A useful facility found on the Commodore 64 is 'type-ahead', the ability to store up to ten characters in the keyboard buffer, until an INPUT is requested.

This can, however, have disadvantages for games which use the keyboard to

T.J.'s Workshop

move objects around the screen — unwanted keystrokes are stored up in the buffer. This can be avoided by restricting the size of this buffer to only one character, using the following POKE: POKE 649,1

And for games which require joystick control only, you can use: POKE 649,0 which will stop any input from the keyboard.

The buffer may be returned to normal with: POKE 649,10

David Gristwood

Simulated 'REPEAT UNTIL' function

Mr Sheppard's solution of a simple problem (February, page 113) is much too long-winded. Try this one for size:

```
10 X=0
20 X=X+1
30 IF X*X<150 THEN 20
40 PRINT X
```

This 2-line loop finds the lowest integer with a square greater than 150. It can be nested within one or more FOR-NEXT loops and will exit to line 40 without disrupting the outer loops.

Unlike the TRS-80, many machines, including the VIC 20, let us exit a nested FOR-NEXT loop early and safely by setting the FOR variable equal to the end value within the loop:

```
10 FOR J=4 TO 8 STEP 4
20 PRINT J
30 FOR X=1 to 100
40 IF X*X<150 THEN 70
50 PRINT TAB(J);X
60 X=100
70 NEXT X
80 NEXT J
```

I Edmundson

Commodore garbage

Commodore programs which use a lot of string manipulation will sometimes appear to hang up, or pause inexplicably. This is caused by garbage collection of unused strings when the micro needs to free some string storage spaces.

You can often alleviate the problem by forcing a garbage collection during a non-critical part of the program. Simply add a line: XX=FRE(0).

Y Hall

Brun's Constant

The sum of reciprocals over the twin primes converges to a finite limit, known as Brun's Constant. Ed Rosenstiel decided to attempt the calculation on a micro, and made some interesting discoveries in the process.

$25,000 Prize

Worldwide Computer Services is offering a $25,000 prize until 31 March 1987 to prove or disprove that there are infinitely many twin primes (the twin prime conjecture).

Ed Rosenstiel's article illustrates how far down the road you can get with a micro today; previously the calculations shown have been done with minis and mainframes.

One learns at school that the so-called harmonic series '1 + 1/2 + 1/3 + 1/4 + 1/5 + . . . and so on' diverges to infinity, *but so does* 1/2 + 1/3 + 1/5 + 1/7 + 1/11 + 1/13 + 1/17 . . . , that is, summing similarly but only over the primes.

Schur demonstrated this in a lecture in 1932 in Germany as follows:
Assume the contrary: that is, that the sum of the prime reciprocals converges to some limit, say, K.

Then, by a formula due to Euler, we have $1 + 1/2 + 1/3 + 1/4 + 1/5 + \ldots + 1/n < (1 + 1/p_1 + 1/p_1^2 + 1/p_1^3 + \ldots) * (1 + 1/p_2 + 1/p_2^2 + 1/p_2^3 + \ldots) * \ldots * (1 + 1/p_m + 1/p_m^2 + 1/p_m^3 + \ldots)$
where the p_i on the right-hand side are just the m prime factors of all numbers from 1 to n.

A little bit of simple calculus then shows that *for all n:*
$$1 + 1/2 + 1/3 + 1/4 + 1/5 + \ldots + 1/n < \prod_{i=1}^{m} 1/(1 - 1/p_i) < \prod_{i=1}^{m} e^{2/p_i} < \exp[2*(1/2 + 1/3 + 1/5 + 1/7 + 1/11 + 1/13 + 1/17 + \ldots \text{to infinity})] = e^{2K}$$
by the assumption, so the RHS is finite.

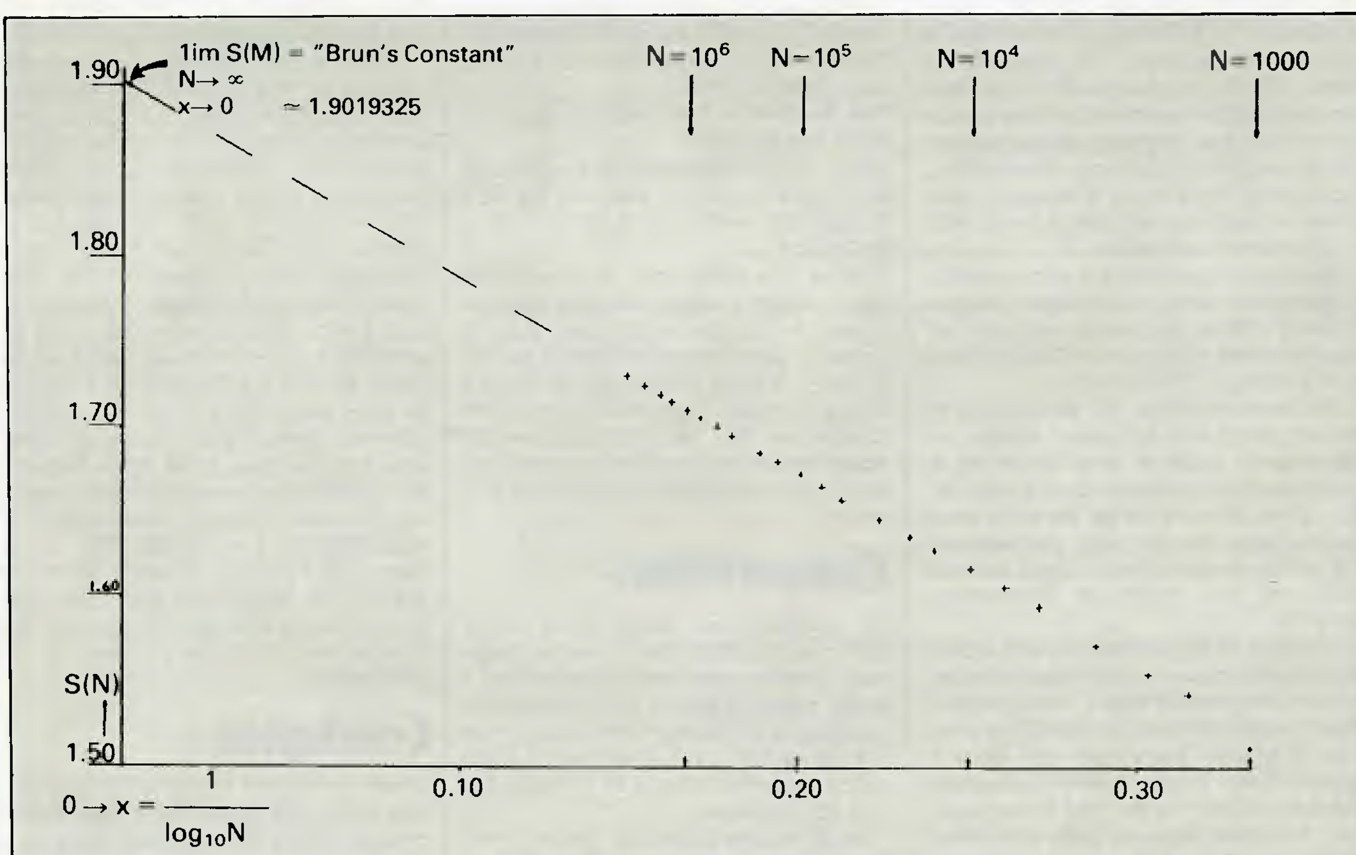


Fig 1 Graph of $x = 1/\log^{10}N, y = S(N)$ for N = 1000 to 5000001 gives an estimate for Brun's Constant S ≃ 1.9019325.

Thus the sum of the reciprocals of all positive integers is also finite, which is false. Hence, so was the assumption. Therefore the sum of the reciprocals of *all* the primes also diverges to infinity!

Then Schur tantalised his audience by mentioning some of the problems connected with the so-called **twin** primes (3,5), (5,7), (11,13), (17,19), namely:

(i) it was an unsolved problem (*and still is!*), as to whether the list of twin primes ever ends; and

(ii) in 1919 Viggo Brun (who died only recently at the age of 92) stunned the mathematical world with a proof that the sum of reciprocals not over *all* the primes, but only over the twin primes (even if their number could be shown to be infinite) converges to a finite limit which is now known as **Brun's Constant** = say, S.

This much I remembered when, as part of a computer course in Pascal, I embarked on a project to calculate Brun's limit.

Writing a program in Basic to list twin primes and to evaluate the sums of their reciprocals is not difficult. The problem is that to find all the twins there is no other way but to compute almost *all* the primes, and this is a slow business on any computer. On a Commodore PET (since the machines operating Pascal were too busy most of the time), I went up to the last pair under 3020001, (later extended to 5000001), then made a graph of necessity in logarithmic scale: that is, in powers of 10. It looked irregular at the lower end, but the gentle curve for the higher values looked promising and I also remembered that, according to Brun's Theorem, this curve would approach some horizontal line for very high values.

It seemed a good idea to eliminate the logarithmic scale, so I plotted 1/log N instead of N on the X-axis, and also left out the lower values under 10000 (Fig 1) and a *straight line* appeared.

It is remarkable in the wilderness of prime numbers, to come across an *apparently* straight line. Ignoring a professional mathematician's remark: '. . . if you take any kind of data and keep taking logs often enough, you will end up with a straight line. . .', my instinct told me this might be something original.

Using a TI-59 program which works out the least squares fit of a polynomial, I soon confirmed that I had found a much more accurate straight line than had it merely been deduced from a graph (Table 1a). And some extrapolations to values higher than those used for the least squares approximation were later found to agree with their computer counterparts to four significant digits!

Looking seriously at what was behind these findings, I decided to retrace the steps which had led me to such an extraordinary result: the 'gentle curve' prompted me to look for some closed mathematical expression to graph it and I had noticed that:

a) it was convex; and

b) it was asymptotic to a line parallel to the x-axis by Brun's Theorem, so I had thought of curves which might fit. By chance I had hit on the right answer straight away, namely on $y = S - 1/x$, the 'upside down hyperbola', although I had meant to consider also $y = S - 1/\exp(x)$ if $y = S - 1/x$ would not work.

The next step was to make a thorough literature search. Brun's Constant had indeed been calculated by several workers (3,4), and the most recent *probable* value given (4) was:

$1.9021604 \pm 5 * 10^{-7}$

However, all the calculations had assumed that the famous conjecture made in 1923 by Hardy and Littlewood (6) is true. This says that the number of twin prime pairs up to some number X is closely approximated by:

$L_2(X) = 2c_2 *_2 \int^X dt/(1n\, t)^2 \sim 2c_2 X/(1n\, X)^2$

that is, neglecting terms of order $X/(1nX)^3$, where $c_2 = 0.66016181$ . . is the 'twin prime' constant as given by Brent (4).

Furthermore, Brent estimates, making the assumption that twin primes are randomly distributed with density $2c_2/(1nx)^2$, (which implies that Brun's series is an infinite series):

that $1im_{n \to \infty} S(n) - S(X) \sim 4c_2 * {}_x\int^\infty dt/t*(lnt)^2 = 4c_2/1nX$

which is the 'Straight Line Conjecture' that I had come up with on the PET (Table 1b), with $c_2 \simeq 0.25*k*1n10 = 0.6596417$ . . .

Does this show that, 60 years after two brilliant mathematicians had deduced a (so far unproven, but, in practice, very accurate) formula for the number of twin primes, by taking the opposite route, from the Straight Line Conjecture to the Hardy-Littlewood approximation, a mere tyro *could* have discovered this celebrated formula on a micro?

Computations

All computations were done on a Commodore PET with a simple program. These were cross-checked on a faster 'sieve' program which leaves out division by multiples of the first primes 2,3,5,7, and 11, and other checks were made against printouts of primes from a TI-59 calculator.

Most results were just copied from the VDU, but a complete printout of all twin primes less than 100000 allowed a manual count of 1224 in agreement with figures previously published by Brent (4). It was interesting to compare the calculation speed of the sieve with that of the simple program: it took the latter 25.3 days to reach the twins up to N = 1700000, while the sieve program needed only 12.2 days, a saving of ≃52%! (The sieve program took 54 days for a complete run up to N + 5000001.)

From the least squares fit (Table 1a) it will be seen that the value derived for S, on the assumption that the Straight Line Conjecture is true:

that Brun's Constant

$S = 1im_{N \to \infty} S(N) = S(N) + k/\log N + \text{error}(N)$

is $S = 1.90074$ . .

which agrees with Brent (4) for three significant digits,

while from $k = 1.1396 .. \simeq 4c_2 * 1n10$

we have $c_2 \simeq 0.6560$ . . .

However, there is something rather unsatisfactory in the above approach, where values below some arbitrarily chosen N are ignored for the extrapolation to S, and it is then observed that all higher values appear to lie on a straight line — not exactly, but to a high degree of 'accuracy'. (This mimics the quite different situation in physical experiments, where data is inevitably tainted due to to observational errors.) I was thus led to consider the question whether 'better' estimates for Brun's Constant might be obtainable by using a statistical approach to curve fitting.

With the help of the Applied Statistics Module for the TI-59 (7), I re-evaluated the results obtained, and also computed the correlation coefficient 'r'. Next I tried to *improve* 'r' by excluding in turn one value, arguing that because of the locally irregular distribution of primes one particular value might perhaps unduly influence the final result. As was not altogether surprising, the coefficient was improved by omitting either of the two *lowest* values for N, so I felt justified to omit both and to start calculating from N = 100001 upward, using higher values for S(N), which had come to hand. From Table 1b N = 734001 was omitted when calculating the final figures. These were: S = 1.901932526, k = 1.14591496., therefore $c_2 \simeq 0.6596417$., where c_2 differs by 0.079%, S by 0.012% from the published results already mentioned. (The correlation coefficient was: r = 0.9999908.).

Conclusion

What I called the Straight Line Conjecture is not new, but during simple micro computations it suggested itself in a most obvious way; yet there was no hint about how to estimate *indepen-*

$3099
INC TAX

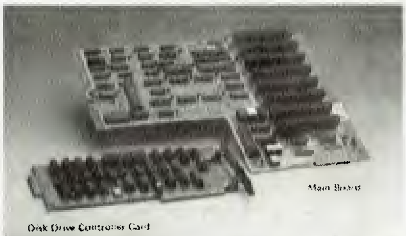
256 k Twin Drives with the lot

Winchester Hard Disk Drive $2100 TAX PAID

BUFF "16" 16 BIT PERSONAL COMPUTER

- IBM PC compatible system
- Disk drive card: Either for internal or external use, can control up to 4 drives
- Microprocessor: 8088 Microprocessor
- Color/Graphic Adapter: Can drive RGB monitor, color monitor, B/W monitor
- Operation system: MS-DOS 2.0
- Low-profile detached keyboard
- MS-DOS Licensed by Microsoft Corporation

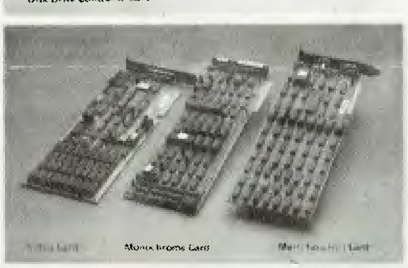
Low Profile Key Board
- Function key — 10 keys
- Numerical key — 15 keys
- Alpha/num key — 57 keys
- Cursor control key — 8 keys
- Other keys — 5 keys



Disk Drive Controller Card
- Can control 4 pcs 5¼" floppy disk drive
- Use SMC 9216 chip more reliable and access data quicker

Main Board
- XT upgrade version, 8 expansion slots, 8k ROM (Bios)
- MS-DOS licenced by Microsoft Corp.

Switching Mode Power Supply
- 135W output for main board and 2 pcs
- 5¼" floppy disk drive with fan inside
- Overload and short-circuit protection



Monochrome Card
- Dot font 14 x 9 with parallel printer port
- 4-layer board

Video Card
- For RGB monitor, composite color and B/W monitor
- 100% IC socket 4 layer PCB

Multi-function Card
- 128k Ram built in, expandable to 256k
- 2 serial asynchronons communication ports (RS-232 interface)

MICRO ~ BUFF
TELEX AA 139718
TELEPHONE 277 7754 (ASK FOR RAY POPE)

499 HIGH STREET ROAD, MT WAVERLEY 3150 MELBOURNE, VICTORIA, AUSTRALIA

Table 1a
Plotting S(N) against $\log_{10}N$

N	$\log_{10}N$	S(N)
51	1.708	1.2700
71	1.851	1.3032
101	2.004	1.3310
151	2.179	1.3969
201	2.303	1.4286
301	2.479	1.4602
501	2.700	1.4861
701	2.846	1.5061
1001	3.000	1.5180
1501	3.176	1.5426
2001	3.300	1.5549
3001	3.477	1.5722
5001	3.699	1.5947
7001	3.845	1.6067
10001	4.000	1.6169
15001	4.176	1.6279
20001	4.301	1.6359
30001	4.477	1.6462
50001	4.699	1.6585*
70001	4.845	1.6652*
100001	5.000	1.6728*
150001	5.176	1.6806*
200001	5.301	1.6858*
350001	5.544	1.6953*
500001	5.699	1.7007*
734001	5.866	1.7064
1020001	6.009	1.7111
1142001	6.058	1.7127
1420001	6.152	1.7156
1500001	6.176	1.7164
1700001	6.230	1.7180
1800001	6.255	1.7188
2000001	6.301	1 7201
3020001	6.480	1.7251
......		

RESULTS:
From the starred values by the TI-59 pakette(2) program:

$S \simeq 1.90074..$
$k = 1.139594148$
$c_2 \simeq 0.65600..$

Table 1b
Plotting S(N) against $1/\log_{10}N$
where $S(N) = \sum [1/p + 1/(p+2)]$
$p<N$, (p and p+2 prime)

N	$1/\log_{10}N$	S(N)	least squares fit to S(N)
100001	0.1999998263	1.67279958	1.672750.
150001	0.1931958674	1.68055034	1.680546.
200001	0.1886425074	1.68584216	1.685764.
350001	0.1803729262	1.69527377	1.695240.
500001	0.1754702774	1.70071693	1.7008585
734001*	0.1704827337	1.70642789	1.706574.
1020001	0.1664281031	1.71108006	1.711220.
1142001	0.1650800688	1.71268937	1.712765.
1420001	0.1625411382	1.71564571	1.715674.
1500001	0.1619146983	1.71635648	1.716342.
1700001	0.1605020716	1.71802810	1.718011.
1800001	0.1598651315	1.71877363	1.718741.
2000001	0.1587042065	1.72013171	1.720071.
3020001	0.1543208189	1.72513665	1.725094.
5000001	0.1492766778	1.73097675	1.730874.
....			
$2*10^{10}$	0.0970776709	————	1.7906898.
$1*10^{99}$	0.0101010101	————	1.8903576.
....			

RESULTS:

$S \simeq 1.9019325..$ [cf. Brent (4) who gives a probable value for S as: $1.9021604 \pm 5*10^{-7}$]

$k = 1.14591496$, hence
$c_2 \simeq 0.6596417...$ and $r = 0.9999908...$

where r is the correlation coefficient computed by the TI-59 Bivariate Data Transform Program ST-12 (6).
(The starred value 734001 was not used for calculating these results,cf.p.7)

IN BOTH TABLES: $S = \lim_{N \to \infty} S(N)$

$S - S(N) \sim k/\log_{10}N$

$k \simeq 4c_2/\ln 10$ and $c_2 = 0.660161181$

[$\simeq$ is used for 'approximately equal to',
$\sim$ means 'asymptotically equal to' in the strict mathematical sense (cf.LeVeque (5)) and
c_2 is the 'twin prime constant' as given by Brent (4).]

P.102

Cash & Carry Computers & Software

ONE TIME OFFER ONLY

Perfect Link

For IBM & compatibles

R.R.P. ~~$149~~ **Our price** **$129**

Limited Numbers

Perfect Link

Supermarket Special

Perfect

For Apple IIE

Perfect Writer/Speller is a fully featured word processor that is fully integrated with **Perfect Speller™** which is an automatic spelling checker program. This offers all the standard features including facilities such as multiple file buffers, virtual memory facilities and 80 sophisticated documents commands.

R.R.P. ~~$450~~ **Our price** **$195** **One time offer only**

1st 20 Callers only $125

The Integrated Software

If you know what you want — why pay more call us.

For super savings on your choice of hardware/software ring Cash & Carry Computers & Software (02) 331 3888

Cash & Carry Computers & Software

Perfect Link, Perfect Writer/Speller, Perfect are trademarks of Perfect Software Inc.

dently the errors with these methods. If one uses the most recently published estimates for S and c_2 to calculate error terms for each N of Table 1b; that is, error (N) $= S - S(N) - 4c_2/\ln N$, then by a simple calculator exercise we have: $|error(N)| < 2/N^{0.66}$, so $k/\ln N$ dominates the approximation.

An essential difference between Brun's and other converging series is seen when comparing it with Gregory's well-known series (which was also discovered independently by Leibniz): $\pi = 4[1 - 1/3 + 1/5 - 1/7 +- \ldots - 1/(2n-1)] + 1/n + error(N)$, where the error consists of terms of the form $constant/_n(2k+1)$ with $k>0$.

Now the square-bracket expression converges to $\pi/4$ with any desired number of decimals, (although much too slowly without the correction 1/n to be of any practical use), provided that a sufficient number of terms is computed (8). To show that the same is true for Brun's series still requires proofs of conjectures of one kind or another, even if better estimates were obtained for Brun's Constant by the use of more powerful computers. It will be remembered that to determine S to only three significant figures by computing its partial sums, requires a program to 'look' at *all* prime numbers up to 10^{1000}.

Until new theories are discovered, one can still only make 'plausible' estimates, — however well these might seem to fit with computation carried out so far.

Thus the mysteries of Brun's series still beckon: only one of the many unsolved problems of The Theory of Numbers.

It is not known whether Brun's converging series $S = 1/3 + 1/5 + 1/5 + 1/7 + 1/11 + 1/13 + 1/17 + 1/19 + 1/29 + 1/31 + \ldots + \ldots$ has an infinite number of terms, but if so then it probably converges *very slowly* indeed with the largest error term $\simeq 2.64/\ln N$. This has been compared with Gregory's infinite series for π which has as largest error term 1/N, thus converging too slowly for practical computation, but still much faster than Brun's series. A more well-behaved series (although a rather trivial example) is the geometric series $G = 2 = G(N) + 1/2_n$ with $G(N) = (1 + 1/2 + 1/4 + 1/8 + \ldots + 1/2_n)$ where the error term is exactly $1/2_n$ and convergence is correspondingly fast.

References

1 **C L Baker and F J Gruenberger,** *Primes in the thousandth million,* P-1245, The Rand Corporation, Santa Monica, 1958

2 **J Walston,** *Least-mean-square fit of a polynomial,* Texas Instruments Inc, 1978

3 **D Shanks and J W Wrench, Jr** *Brun's Constant,* Math Comp, v28, 1974 pp293-299

4 **R P Brent,** *Irregularities in the distribution of primes and twin primes,* Math Comp, v29, 1975, pp43-56

5 **W J Leveque,** *Fundamentals of Number Theory,* p134, Addison-Wesley Publishing Company, London, 1977.

6 **G H Hardy and J E Littlewood,** *Some problems of partitio numerorum,* Acta Math, v44, 1923, pp1-70

7 **Applied Statistics,** *Solid State Software Module,* Texas Instruments Inc, 1977

NEWCOMERS START HERE

This is our unique quick-reference guide, reprinted every month, to help our readers pick their way through the most important pieces of (necessary) jargon found in APC. While it's in no way totally comprehensive, we trust you'll find it a useful introduction. Happy microcomputing!

Probably the first thing you noticed on picking up this magazine for the first time was the enormous amount of unintelligible-looking jargon. In the words of *The Hitch-hiker's Guide to the Galaxy:* Don't panic! Baffling as it may sound, the jargon does actually serve a useful purpose. It's a lot easier to say VDU, for example, than 'the screen on which the computer's output is displayed.' This guide is intended to help you find your way around some of the more common 'buzzwords' you're likely to come across in the pages of *APC.*

For those completely new to computing, let's start with the question: What is a microcomputer? We can think of a micro as: a general-purpose device in contrast to a typewriter, which can only be used for typing; a calculator, for performing calculations; a filing cabinet, for filing information, to name just a few of its functions. A micro can do all these things and more.

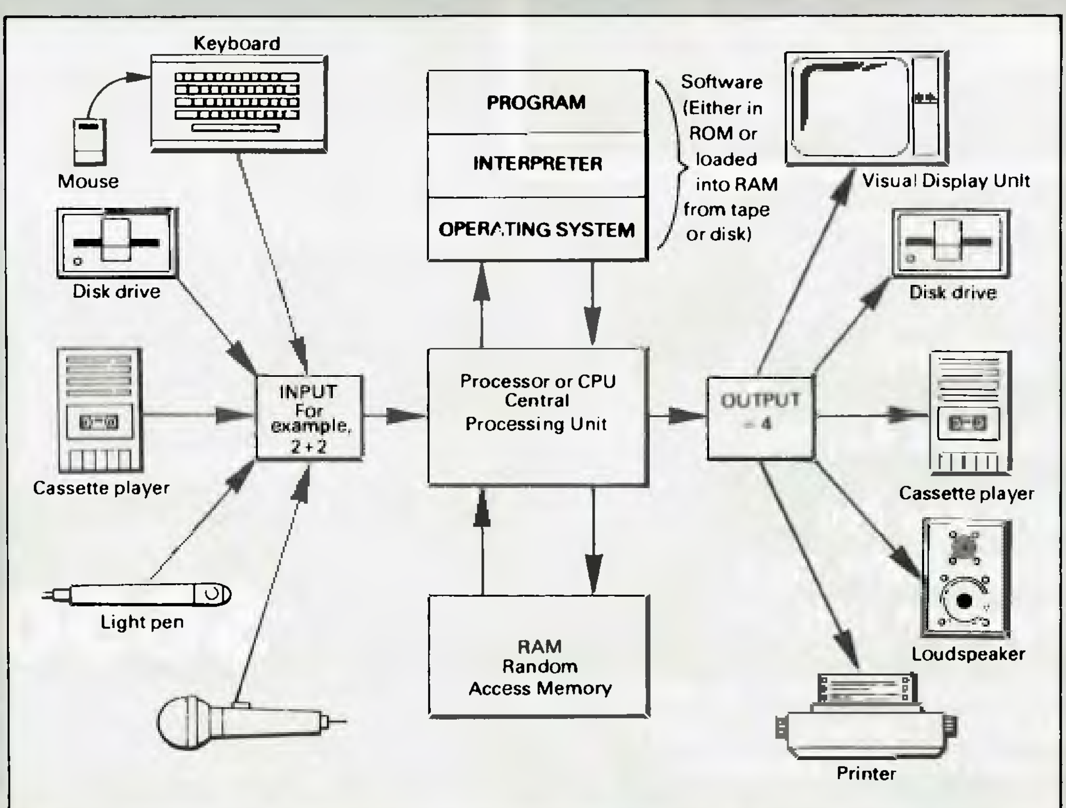
If it's to be of any use, a general-purpose device needs some way of knowing what to do. We do this by giving the computer a set of logical instructions called *a program.* The general term for computer programs is *software.* Every other part of a microcomputer system is known as *hardware:* 'If you can touch it, it's hardware.'

Programming

Programs must be written in a form the micro can recognise and act on — this is achieved by writing the instructions in a *code* known as a *computer language.* There are literally hundreds of different languages around, the most popular of these being *Basic.* Basic is an acronym of *B*eginners' *A*ll-purpose *S*ymbolic *I*nstruction *C*ode. Although originally intended as a simple introductory language, Basic is now a powerful and widely used language in its own right.

Other languages you're likely to come across in *APC* are *Forth, Pascal, Logo, C* and *Comal* to name but a few. These are known as *high level* languages because they approach the sophistication of a human language. You'll also see references in *APC* to the *low level* languages, *assembly language* and *machine code.* We'll look at these in a moment.

The heart of a micro, the workhorse, is the *processor* or Central Processing Unit *(CPU).* The processor usually consists of a single silicon chip. As with computer languages, there are a number of different types of processor available, *Z80, 6502, 6800* and *8088* being just a handful (literally) of the types in common use. The processor is nothing magical — it's just a bunch of electronic circuits. It's definitely not a 'brain'.



A schematic view of a microcomputer system

As it's electronic, the processor's circuitry can be in one of two states: on or off. We represent these two states by *binary* (base two) notation, the two binary digits (known as '*bits*') being 0 and 1. It's possible to program computers in binary notation, otherwise known as machine code (or machine language) programming.

Machine code is called a low level language because it operates at a level close to that 'understood' by the processor. Languages like Basic are known as high level languages because they are symbolic, operating at a level easily understood by people but not directly understood by the processor.

Between high level languages and machine code is a low level language known as assembly language or, colloquially, *assembler*. This is a mnemonic code using symbols which the processor can quickly convert to machine code.

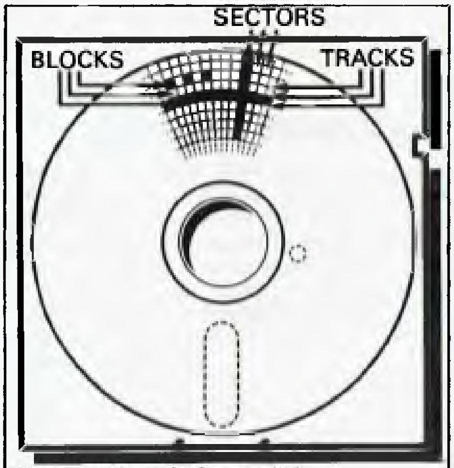
Since everything has to be converted into binary form before the processor can make sense of it, we need some sort of code to represent each character to be processed by the computer. In order to simplify communication between computers, a number of standard codes have been agreed on. The most widely used of these codes is the American Standard Code for Information Interchange, *ASCII*. This system assigns each character a decimal number which the processor can then convert to its binary equivalent.

A program written in a high level language must be converted into binary before the processor can carry out its instructions. We could of course do this manually, but since this is exactly the sort of tedious job computers were designed to do for us, it makes much more sense to write a program to do it.

There are two types of program to do this translation for us.

The first of these is a *compiler* which translates our whole program permanently into machine code. When we *compile* a program, the original high level language version is called the *source code* while the compiled copy is called the *object code*. Compiled programs are fast to run but hard to edit. If we want to change a compiled program, we either have to edit it in machine code (extremely difficult) or we have to go back to a copy of the source code. For this reason there is a second translation program: an *interpreter*. An interpreter waits until we actually *run* (use) the program, then translates one line at a time into machine code — leaving the program in its original high level language. This makes it slower to run than a compiled program, but easier to edit.

There are two unusual Basic words you're likely to come across: *POKE* and *PEEK*. When you program in a high level language, you are normally unable to choose in which part of the machine's memory the processor will store things. This makes programming easier as you don't need to worry about memory locations, but slows down the program since the processor has to 'look up' addresses for you. Using the POKE command, however, you can 'poke' a value directly into a desired memory address. 'POKE 10000,56', for example, puts the value 56 into memory location 10000. PEEK allows you to examine the content of a particular memory address. If you were to follow the above poke with 'PEEK (10000)', the computer would respond by displaying the value 56. POKEing and PEEKing is normally done to increase program speed, but may also allow us to do things which could not be done through Basic.



Cross-section of a floppy disk

Memory

So far, we have a processor and a program. Since a computer needs somewhere to store programs and data, it needs some kind of *memory*. There are two types of memory: *Read Only Memory (ROM)* and the badly named *Random Access Memory (RAM)*. ROM is so-called because the processor can 'read' (get things out of) its contents, but is unable to 'write to' (put things in) it.

ROM is used to store *firmware*, the name given to software permanently available on the machine. An interpreter is a typical example of firmware (stick with it: it gets easier!).

RAM differs from ROM in two important ways. Firstly, you can write to it as well as read from it. This means that the processor can use it to store both the program it is running and *data* (information). The second important difference is that RAM needs a constant power supply to retain its contents: as soon as you switch the computer off, you lose your program and data.

There is a type of RAM, known as *CMOS RAM*, which requires only a tiny amount of power to retain its contents. This is found in portable computers like the Tandy 100. It is usually powered by small ni-cad batteries so that programs and data are retained even when the main power is switched off. At present, CMOS RAM is extremely expensive and is not likely to be used in desktop machines for a little while yet. (CMOS stands for Complementary Metal Oxide Semiconductor).

Memory is described in terms of the number of characters we can store in it. Each character is represented by an 8 bit binary number. 8 bits make one *byte* and 1024 bytes make one *Kilobyte* or *1k*. 32k, for example, means that the computer can store about 32000 characters in its memory. If 1024 sounds like an odd number, remember that everything is based on the binary system, thus 1,2,4,8,16 . . . 1024 being the nearest binary multiple to 1000.

While we're on the subject of bits, you'll often see computers and their processors described in terms of their *bit power: 8-bit, 16-bit, 32: 16-bit* and so on. This is a means of describing how large a binary number the processor can handle in one chunk. A binary number, incidentally, is known — confusingly — as a *word*. An 8-bit processor, for example, can handle 8-bit words, that is, up to 11111111 (255 in decimal). Anything larger than this has to be broken down into manageable chunks before it can be processed.

A 16-bit machine can handle bigger chunks of data at a time. This means it can handle ('*address*') larger amounts of memory at one time. This is why most 8-bit machines have a maximum of 64k RAM while 16-bit micros usually have 128k upwards.

As 16-bit processors can handle larger words than an 8-bit machine, they ought to be twice as fast. In practice, however, there is a little more to it than that. While it may take a 16-bit machine half as long to work out that 2+2=4, the actual processing is only part of the story.

The result of the calculation has to be placed into the appropriate memory location, passed to the screen or whatever is required. The transfers to and from the processor are often made in 8-bit form; this is why you'll hear people arguing that certain processors are not 'true' 16-bit. If the problem has to be handed to the processor in 8-bit form, turned into 16-bit, calculated and then the result turned back into 8-bit for transfer elsewhere, there may be little or no saving in time over an 8-bit system.

The other factor affecting speed is that the actual processing may form only a small part of the overall operation. A word processor, for example, spends most of its time passing files to and from disk and waiting for the user to type the next character. The processing itself consumes very little time. And if you look at the Benchmarks summary (*APC*, February 1984, pp 59-60), you'll see some 8-bit machines beating their 16-bit rivals — even in processor-bound operations like the *APC* Benchmarks.

Returning to the subject of RAM for a moment, a word of warning: Don't rush out with your new-found understanding to buy the machine offering you the most RAM for your money. Quite aside from the fact that the amount of RAM is by no means the only consideration when buying a micro (no matter how much manufacturers may stress it), different machines use differing amounts of RAM for things like graphics. Always check how much RAM is actually available to the user for program storage. Machines which proudly proclaim '64k' may well leave you with less than half of this in which to store Basic programs and data.

Back-up storage

There are numerous forms of *permament* or *back up storage*, but by far the most common are *floppy disk*, *floppy tape* and *cassette*.

Floppy disks or diskettes are circular pieces of thin plastic coated with a magnetic recording surface similar to that of tapes. The disk, which is enclosed in a protective card cover, is placed in a *disk drive*. Disk drives comprise a high-speed motor to rotate the disk and a

NEWCOMERS START HERE

read/write head to record and 'play back' programs and data.

The disk is divided into concentric rings called *tracks* (similar to the tracks on an LP) which are in turn divided into small *blocks* by spoke-like divisions called *sectors*.

There are two methods for dividing the disk into sectors. One method is called *hard sectoring*, where holes punched in the disk mark the sectors, and the other is *soft sectoring* where the sectors are marked magnetically. The reason that disks from one machine can't be read by a different make is that each manufacturer has its own way of dividing up the disk. Recently, however, manufacturers have apparently begun to acknowledge that this situation can't go on forever, and they are working on making their disks compatible.

Since the computer needs some way of organising the disk, we have a program called a *Disk Operating System (DOS)*, usually known simply as the *Operating System (OS)*. The operating system does all the 'housekeeping' of the disks, working out where to put things, letting the user know what is on the disk, copying from one disk to another and so on. As you might expect by now, there are lots of different operating systems available, each with its own advantages and disadvantages. The three most popular OSs are *CP/M* (Control Program for Micros), *MS-DOS* (MicroSoft Disk Operating System) and *PC-DOS* (Personal Computer Disk Operating System). MS-DOS and PC-DOS, incidentally, are all but identical.

Disks can support what are known as *random access files*. That is, you can randomly choose a point in a file and the drive head will move directly to that point. You can then edit the file, and only the blocks affected will be rewritten. The rest of the file remains unchanged.

Floppy disks provide a reasonably fast and efficient form of secondary storage and are cost-effective for business machines. For home computers, however, the usual form of program and data storage is on ordinary cassette tape using a standard cassette recorder. This method of storage is slow and unreliable, but is very cheap and adequate for games, for example.

Cassettes can support only *serial access files*. That is, whenever a file is to be edited, the whole file must be written back to the tape. This makes certain applications — word processing being a prime example —extremely tedious.

Floppy tape drives are a compromise between speed and cost. They use a small continuous loop tape which, like a disk, is divided into blocks. Floppy tape drives rely on serial access files, but by rotating the tape at high speed and using the block markers, they can simulate random access files.

Another type of disk you'll see referred to is the *hard disk*. This is an extremely efficient method of storing large amounts of data. Hard disk capacity generally starts at around 10*Mbytes* (10 million bytes) and rises to . . . well, you name it. Besides offering a much greater capacity than floppies, hard disks are more reliable and considerably faster. They are, however, much more expensive than floppy drives.

Input/output

Since computers need some way of communicating with the outside world, we need *input* and *output* devices. Input and output devices include all manner of things from hard disk units to light pens, but the minimum requirement for most applications is a typewriter-style *keyboard* for input and a TV-like *Visual Display Unit* for output. The Visual Display Unit is variously referred to as a *VDU*, *Cathode Ray Tube (CRT)* and monitor.

The various component parts of a computer system (processor, keyboard, VDU, disk drives, and so on, may all be built into a single unit or they may be separate, connected by cables.

Take this paragraph slowly and it will make sense! When a computer communicates with an outside device, be it a printer or another computer, it does so in one of two forms — *parallel* or *serial*. Parallel *input/output (I/O)* requires a number of parallel wires. Each wire carries one bit, so with eight wires we can transmit/receive information one byte at a time (8 bits = one byte, remember). Serial I/O, in contrast, uses a single wire to transmit a series of bits one at a time (that's why it's called serial), with extra bits to mark the beginning and end of each byte.

To enable different devices to communicate with each other in this way, standards have been agreed for different *interfaces*. An interface is simply a piece of circuitry used to connect two or more devices. The most common standard serial interface is the *RS232* (or *V24*) while the Centronics standard is popular for parallel interfaces.

Networks

When two computers want to communicate with each other over a distance, there are again two ways of doing it (nothing is ever clear-cut in the world of micros — you'll get used to it). Both methods use the public phone network. The first is known as an *acoustic coupler*. This simply plugs into your computer, and has a receptacle into which you place your telephone handset. The acoustic coupler is convenient in that you can unplug it from one computer and plug it into another one in a matter of seconds. They are generally slow, however, and prone to interference.

The alternative method is to use a *modem*. Unlike an acoustic coupler, a modem is wired into the telephone system and you should get permission for this from Telecom.

A term you'll hear used in connection with acoustic couplers and modems is *baud rate*. The baud rate is a measure of the speed at which a device can transmit and receive data. You can safely think of the baud rate as being bits-per-second, though the accurate definition is a little more complex. Therefore, a 300-baud modem can transmit/receive data at the rate of 300 bits (about 50 characters) per second.

A 1200/75 modem means that it receives at 1200 baud but transmits at 75. Most modems are 1200/75 and acoustic couplers 300/300. By way of comparison, saving programs to cassette is normally done at between 300 and 1500 baud.

Finally, communications between computers is either *full* or *half duplex*. Full duplex is when the machine receiving the data echoes it back to the machine transmitting it and says 'This is what I think you said — is that right?'. If it's wrong, the section will be transmitted again. Half duplex is where no checking is made. If you're ever unsure of which to use, start with full duplex. If everything you type appears on your display twice, then you should switch to half duplex.

Computer-to-computer communications using acoustic couplers.

Database

A database allows you to store, process and report on structured information. Most of the cheaper packages are based on a traditional card index where each card about an individual, order or item of stock is stored in a single record and a group of like records is stored in a file (corresponding to the index card box). Sophisticated packages can relate several files together, so that you can process groups of dissimilar but related records.

Spreadsheet

Spreadsheet software is useful to anyone who regularly uses a calculator. The VDU acts like a 'window' on a large sheet of numbers —neatly laid out in rows and columns, occasionally interspersed with text headings. The user is able to shift the window to the point of interest and so enter text. The rest of the calculation is displayed immediately with automatic recalculations throughout.

ATARI 800XL

ALSO AVAILABLE — PRINTER INTERFACES — SUIT ALL ATARI COMPUTERS

PHONE NOW AND PLACE YOUR ORDER
WITH AUSTRALIA'S OLDEST ATARI DEALER.

FOR SUPPORT GO WITH
THE EXPERIENCED DEALER

CALCUTRONIC PTY LTD
CALCULATOR AND COMPUTER CENTRE
**797 GLENFERRIE ROAD,
HAWTHORN, VIC 3122
Telephone: (03) 818 6631**

SEE US NOW!! FOR COMPUTER BOOKS

THE Co-op BOOKSHOP

- **RANGE**
  Biggest in Australia. Always in stock — from Robotics to Sinclairs.
- **QUALITY**
  Good service, trained staff, best titles we can find on every subject.
- **PRICE**
  5% Discount to members.
- **AVAILABILITY**
  Check us out now at your nearest branch.
  1. Sydney Technical Branch, 831 George St., Railway Square, Sydney. Ph: 211-0561.
  2. Bay St., Broadway (Nr. Grace Bros.) Open Saturday mornings. Ph: 212-2211.
  3. University of NSW on campus. Ph: 663-4024.
  4. Institute of Technology, Broadway. Ph: 212-3078.

ALSO AT: Macquarie University, Ph: 888-6523. Sydney University, Ph: 692-3705; And at ANU Canberra, Armidale, Wagga, Wollongong, Newcastle, Bathurst, etc. **VISIT US NOW!**

Good News... gametronics

202 PROGRAMS FOR

sinclair

ZX SPECTRUM

Beat That! We've tried, but we can't find anyone else with as great a variety of Spectrum software. Maybe we're bigger because we specialize in Spectrum. Maybe we're cheaper because we sell more than any other retailer. Whichever reason, you benefit with more to choose from, lower prices and our special offers.

THIS MONTH – 10 TOP GAMES REDUCED *• 16/48K Naanas • 16/48K Space Zombies • 48K Knock-Out • 48K Cruise Attack • 48K Pat the Postman • 16/48K Cosmic Raiders • 48K Laser Warp • 48K Deffendar • 48K Mad Martha (Adventure) • 16/48K Gold! (Adventure).*

now only

$16.95

THIS MONTH'S SPECIAL OFFER

So that's the one to buy: Sinclair ZX Spectrum, with full guarantee and service back-up, for only

16K—$225
48K — $315

AUSTRALIA'S SINCLAIR ZX SPECTRUM SPECIALIST

gametronics

6/177 Toorak Road, South Yarra 3142. (03) 241 3031

Mail Order/Cheque, Money Order or Bankcard it.
Please send me ☐ your catalogue, and:

ITEMS .... @ $..........
.... $..........
.... $..........
Add $8.50 P&P H/ware or $2.50 S/ware: $..........
$..........

Name ....
Address ....
....
Signature ....
Bankcard No. .... *Exp. Date* ....

MPA/APC634

TEACH YOURSELF ASSEMBLER

Paul Overaa discusses the arithmetic operations of addition, subtraction, multiplication and division on the 6502, Z80 and 8080 processors.

This is part six of APC's Teach Yourself Assembler series. It's unique in using Basic as its point of reference, and avoiding the 'drop you in it' approach often used on this subject. Three processors, the Z80, 6502 and 8080 are covered in detail, but enough information is provided to enable users of other processors to follow the course. Copies of earlier articles in the series, which started in March 1984, may be obtained from our Back Issues dept.

The basic arithmetic instructions available on the 8080, Z80 and 6502 processors are for addition and subtraction. The 6502 operates on 8-bit operands only, but both the 8080 and Z80 have certain instructions that enable 16-bit operands to be dealt with.

Addition Z80

On the Z80, addition instructions take the form ADD A, operand. The specified operand is added to the value present in the accumulator, and in symbolic form we can write A ← A + operand. Various forms of addressing are possible, as follows:

ADD A,8: adds the immediate value 8 to the accumulator — that is, is performing the function A ← A + 8.

ADD A,B: adds the contents of the B register to the accumulator, thus performing the function A ← A + B.

ADD A,(HL): adds to the accumulator the contents of the byte whose address is specified by HL — that is, A ← A + (HL).

ADD A,(IX+d): in the indexed addressing form, the address of the byte to be added is found by adding a specified displacement to the address held in index register IX. The symbolic representation is A ← A + (IX+d).

Instructions for 16-bit operations use HL, IX or IY as destination registers. Typical examples are as follows:

ADD HL,DE; adds the contents of the DE pair to the contents of HL, thus performing HL ← HL+DE.

ADD IX,BC: in a similar fashion, this adds the contents of BC to the index register IX.

On the Z80, the instruction 'add with carry' (ADC) will include, in the 'addition', the carry flag value: ADC A,B will perform the function A ← A + B + Carry. The usefulness of this instruction can be seen from the example in Fig 1. We add two 'two byte numbers' — 255 and 257 — by adding the two low bytes first and then adding the two high bytes.

The addition of the low bytes causes a 'carry' to occur: the ADC instruction takes it into account when the high bytes are added. As a general rule, multibyte addition is performed by using a normal addition instruction for the first (least significant) bytes, and using the 'add with carry' instructions for succeeding bytes. The program in Fig 2 adds the contents of two 'two byte numbers' held in locations labelled FIRST$NUMBER and SECOND$-NUMBER.

Because of the existence of double register addition instructions, it's possible to write a much simpler 16-bit addition program on the Z80. DE and HL can be loaded directly with the numbers to add, and an ADD HL,DE instruction used to perform the 16-bit addition with one addition instruction (Fig 3).

```
HIGH BYTES            LOW BYTES
 0000  0000           1111  1111          00FF hex = Decimal 255
 0000  0001           0000  0001          0101 hex = Decimal 257
 ----------           ----------
 0000  0010    (1)    0000  0000     ←    Result of addition
       ↑        ↑           ↑             Low byte addition causes
       |        |                         the carry to be set
       |     Carry flag
       |
High byte addition
with ADC includes
the carry value
```

Fig 1 Z80 'add with carry' instruction

```
LD    HL,SECOND$NUMBER     ;HL points to low byte of second number
LD    A,FIRST$NUMBER       ;Get low byte of first number in Acc
ADD   A,(HL)               ;Add low bytes
LD    (RESULT),A           ;Store low byte of result
LD    A,FIRST$NUMBER+1     ;Get high byte of first number
INC   HL                   ;Now points to high byte of second number
ADC   A,(HL)               ;Add high bytes + carry
LD    (RESULT+1),A         ;Store high byte of result
```

Fig 2 Z80 16-bit addition

```
LD    DE,(FIRST$NUMBER)    ;Load DE with first number
LD    HL,(SECOND$NUMBER)   ;Load HL with second number
ADD   HL,DE                ;Performs HL ← HL + DE
LD    (RESULT),HL          ;Store result
```

Fig 3 Z80 alternative 16-bit addition

Addition 8080

Immediate loading of 8080 register pairs uses a LXI instruction. LXI H, SECOND$NUMBER will load the HL pair with the 16-bit address equivalent to the label SECOND$NUMBER. LDA is a direct loading of the accumulator from the byte whose address is FIRST$NUMBER. 'M' is the 8080 assembler convention to specify an

```
LXI   H,SECOND$NUMBER     ;HL points to low byte of second number
LDA   FIRST$NUMBER        ;Get low byte of first number in Acc
ADD   M                   ;Add low bytes
STA   RESULT              ;Store low byte of result
LDA   FIRST$NUMBER+1      ;Get high byte of first number
INX   H                   ;Now points to high byte of second number
ADC   M                   ;Add high bytes + carry
STA   RESULT+1            ;Store high byte of result
```

Fig 4 8080 16-bit addition

```
LHLD  FIRST$NUMBER        ;Load HL with first number
XCHG                      ;Swap to DE
LHLD  SECOND$NUMBER       ;Load HL with second number
DAD   D                   ;Performs HL ← HL + DE
SHLD  RESULT              ;Store result
```

Fig 5 8080 alternative 16-bit addition

```
CLC                       ;Clear.carry flag
LDA   FIRST$NUMBER        ;Low byte of first number
ADC   SECOND$NUMBER       ;Add low bytes
STA   RESULT              ;Store low byte of result
LDA   FIRST$NUMBER+1      ;High byte of first number
ADC   SECOND$NUMBER+1     ;Add high bytes
STA   RESULT+1            ;Store high byte of result
```

Fig 6 6502 16-bit addition

```
LD    HL,SECOND$NUMBER    ;HL points to low byte of second number
LD    A,FIRST$NUMBER      ;Get low byte of first number in Acc
SUB   (HL)                ;Subtract low bytes
LD    (RESULT),A          ;Store low byte of result
LD    A,FIRST$NUMBER+1    ;Get high byte of first number
INC   HL                  ;Now points to high byte of second number
SBC   A,(HL)              ;Subtract high bytes with borrow
LD    (RESULT+1),A        ;Store high byte of result
```

Fig 7 Z80 16-bit subtraction

```
LD    DE,(FIRST$NUMBER)   ;Load DE with first number
LD    HL,(SECOND$NUMBER)  ;Load HL with second number
AND   A                   ;Clear the carry flag
SBC   HL,DE               ;Equivalent to HL ← HL + DE
LD    (RESULT),HL         ;Store result
```

Fig 8 Z80 alternative 16-bit subtraction

```
LXI   H,SECOND$NUMBER     ;HL Points to low byte of second number
LDA   FIRST$NUMBER        ;Get low byte of first number in Acc
SUB   M                   ;Subtract low bytes
STA   RESULT              ;Store low byte of result
LDA   FIRST$NUMBER+1      ;Get high byte of first number
INX   H                   ;Now points to high byte of second number
SBB   M                   ;Subtract high bytes with borrow
STA   RESULT+1            ;Store high byte of result
```

Fig 9 8080 16-bit subtraction

```
SEC                       ;Set carry flag
LDA   FIRST$NUMBER        ;Low byte of first number in accumulator
SBC   SECOND$NUMBER       ;Subtract low bytes
STA   RESULT              ;Store low byte of result
LDA   FIRST$NUMBER+1      ;High byte of first number in accumulator
SBC   SECOND$NUMBER+1     ;Subtract high bytes
STA   RESULT+1            ;Store high byte of result
```

Fig 10 6502 16-bit subtraction

indirectly addresed memory location, and it refers to the byte whose address is contained in the HL register pair. Thus, ADD M on the 8080 is performing the same function as ADD A,(HL) on the Z80. STA is the 8080 'store accumulator direct', the contents of the accumulator are stored at the address specified. INX is a 'double register increment'. After the INX H instruction, HL is pointing to the byte after that labelled SECOND$NUMBER — that is, it is pointing to SECOND$NUMBER+1. Typical 8080 code is shown in Fig 4.

An equivalent version of the second Z80 form using the HL and DE register pairs can be written, the only difference being that on the 8080 it's not possible to load the DE pair directly. Instead, we load HL with the contents of the byte labelled FIRST$NUMBER, then use an exchange instruction XCHG to 'swap' the contents of the HL and DE registers. The first number is therefore placed into DE, leaving us free to re-load HL with the second number. A double register DAD D instruction is then used to perform the function HL ← HL+DE. The instruction SHLD will store the contents of the HL register pair in the two bytes RESULT and RESULT+1 (Fig 5).

Addition 6502

The only addition instruction available on the 6502 is an 'add with carry' (the mnemonic is ADC). This is no real disadvantage, but it does mean that if you wish to perform 'normal addition' you must 'clear' the carry flag before using ADC. The 6502 can be conditioned to operate in one of two modes, Binary or Decimal. The operations we are discussing are related to normal binary operation and we'll assume that the processor has been placed in binary mode by using a CLD (clear decimal mode) instruction (Fig 6).

Z80 subtraction

As with the addition instructions, it's useful to have two types of subtraction — normal subtraction and 'subtraction with borrow'. Normal subtraction (mnemonic SUB) is used for the 'low bytes' (least significant bytes), and subtraction with borrow (mnemonic SBC) is used for the succeeding bytes (most of the instructions in Fig 7 are identical to the earlier addition program). If, after the subtraction of the least significant bytes the carry flag has been set, this indicates that the value subtracted from the accumulator is greater than the accumulator value itself — a borrow has occurred. The SBC instruction allows for this 'borrow' by including the carry flag in the subtraction.

A more compact version using HL and DE can also be written. The only subtraction instruction available for the

double register operations is a subtract with carry. This being so, we clear the carry flag by ANDing the accumulator with itself, thus producing a 'normal subtraction' (there is no explicit 'clear carry Z80 instruction' that could be used). The code in Fig 8 gives the general idea.

Subtraction 8080

The mnemonics are SUB and SBB. The 8080 does not have double register subtraction instructions, and the example in Fig 9 uses the accumulator as in the first 8080 addition example.

Subtraction 6502

The 'subtract with borrow' instruction on the 6502 performs the function A ← A – operand – $\overline{\text{Carry}}$, with the bar over the carry indicating the 'complement' of the carry. Borrow is thus defined as the carry flag complemented. The 6502 equivalent for a 16-bit subtraction starts by SETTING the carry flag using a SEC instruction. As with Z80 and 8080 forms, the least significant bytes are dealt with first. The equivalent 6502 program for a 16-bit subtraction is shown in Fig 10.

These ideas can be expanded to any number of bytes and the general principles remain unchanged, but for now we'll turn our attention to the slightly more complicated problem of multiplication and division.

Multiplication

Consider the base 10 product shown below:

```
   2 5    ← Multiplicand
   1 2    ← Multiplier
 ------
  2 5     ← Partial products
   5 0
 ------
 3 0 0    ← Result
 ------
```

Let's take this simple product and do the same calculation using base 2 — that is, binary arithmetic:

```
        1 1 0 0 1    ← Multiplicand
                       (25)
          1 1 0 0    ← Multiplier
                       (12)
    ---------------
  1 1 0 0 1          ← Partial
                       products
    1 1 0 0 1
      0 0 0 0 0
        0 0 0 0 0
    ---------------
1 0 0 1 0 1 1 0 0    ← Result (300)
    ---------------
```

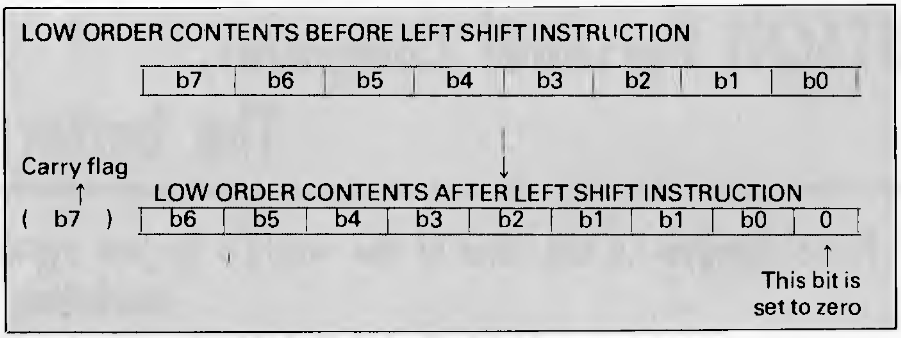


Fig 11 Normal left shift on low order byte

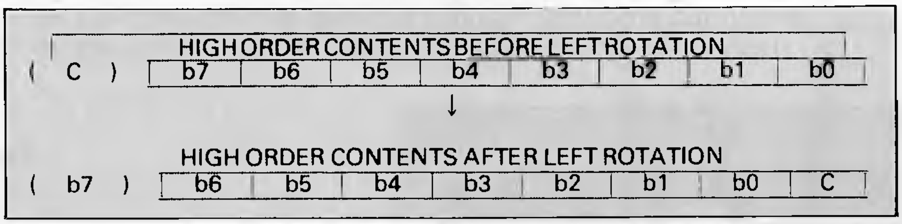


Fig 12 Rotation to the left

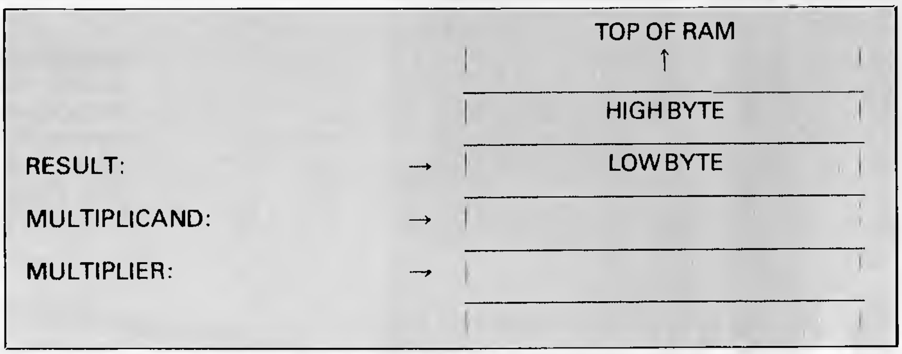


Fig 13 Layout in memory of 8-bit multiplication

The important point is that the partial products are either zeros, or a 'shifted' version of the multiplicand; we can use this knowledge to devise an algorithm for binary multiplication. For each 'Bit' in the multiplier, we ask: 'Is this bit set to 1?' If it is, we add the shifted equivalent of the multiplicand to the result. Two approaches are possible: we can either 'left shift' the multiplicand during the operations, or we can 'right shift' the bytes or registers that are storing the result.

Before showing some typical code for an 8-bit multiplication, we need to understand the general ideas behind creating '16-bit shifts'. Generally, the left shift operations available on our microprocessors will push bit7 into the carry flag. When attempting to left shift a 16-bit (2-byte) value, we can use a normal left shift on the low order byte as shown in Fig 11.

Bit7 falls into the carry flag, and to obtain a 16-bit shift we must shift this bit, now in the carry flag, into bit8 of the 16-bit number. In other words, we want to push this carry value into bit 0 of the high order byte. We need an instruction that performs a left shift and includes the carry, and the most commonly implemented instructions that perform this are rotation instructions. Rotation to the left has the effect shown in Fig 12.

By utilising a combination of left shift on the low order byte and a left rotation (through the carry) on the high order byte, we can left shift a 16-bit number held in two bytes or in two 8-bit registers; the principles can be extended to any number of bytes as required. Instructions are usually available for the equivalent right shifts and right rotations. Occasionally, you will find 'tricks' being used to create 16-bit left shifts. One favourite on the Z80 is to use the double-register addition instructions to add a register pair to itself. For example, ADD HL,HL results in a 16-bit arithmetic left shift.

Let's see how these ideas help to produce a simple multiplication program that takes an 8-bit number held in a location labelled MULTIPLICAND, mul-

HITACHI

HITACHI Personal Computer,
The better Business Solution

Nissei Sangyo Co Ltd, One of the world's largest system, marketing and sales organisations!

It's not so important that you know our name.
Although, Internationally, we are well-known marketing and sales specialists of electronic products — and a turnover of more than A$1,400 million speaks for itself.
However, it is much more important for us that you are impressed by our products.

Professional, High Performance Business Computer
HITACHI MBE-16000 SERIES PERSONAL COMPUTER

Specification: Intel 8088, 8087 Option, memory (320KB RAM expandable upto 512KB), Interface (Monochrome & R.G.B. Colour Monitor Interface, Parallel & RS232C Interface, Light Pen Interface), Expansion Slots (Four Slots), Built in Two 320KB Mini Floppy Drives, Popular Software on MS-DOS is available.

Greater Performance Personal Computer
HITACHI MB-6890K PERSONAL COMPUTER

Specification: MC-6809, memory (24KB ROM, 32KB RAM standard), Interface (Monochrome & Colour R.G.B. Monitor Interface, Light Pen Interface, Parallel & RS232C Interface), Expansion Slots (Five Slots), Popular Software, including CP/M Card* available (Optional).

For your local dealer, please contact:

Nissei Sangyo Co.,Ltd.
(INC. IN JAPAN)

200 PACIFIC HIGHWAY, CROWS NEST
N.S.W. 2065 AUSTRALIA

TELEPHONE (02) 923 1522 (4 lines)
TELEX AA70941 FACSIMILE (02) 922 2498

nsa

```
          LD      HL,MULTIPLIER        ;HL points to multiplier
          LD      C,(HL)               ;Get multiplier in C register
          LD      B,8                  ;B is used as a 'bit' counter
          INC     HL                   ;Now HL points to
                                        multiplicand
          LD      E,(HL)               ;Get multiplicand in E
                                        register
          LD      D,0                  ;Now DE = multiplicand !
          LD      HL,0                 ;HL will be used to hold
                                        result
-----------------------------------------------------------------
MULTIPLY: SRL     C                    ;Least sig but (multiplier)
                                        into carry
          JR      NC,SKIP              ;Indicates least sig bit is
                                        zero
          ADD     HL,DE                ;Add partial product to
                                        result
SKIP      SLA     E                    ;Left shift multiplicand low
                                        byte
          RL      D                    ;Left rotate high byte
                                        through carry
          DEC     B                    ;Decrease bit counter
          JP      NZ,MULTIPLY          ;Do next bit
          LD      (RESULT),HL          ;Store result
```

Fig 14 Z80 8-bit multiplication

```
          LD      HL,(MULTIPLIER—1)    ;Get multiplier in H register
          LD      L,0                  ;Clear to zero
          LD      B,8                  ;B is used as a 'bit' counter
          LD      DE,MULTIPLICAND      ;Get multiplicand in E
                                        register
          LD      D,0                  ;Now DE = multiplicand !
-----------------------------------------------------------------
MULTIPLY: ADD     HL,HL                ;16-bit left shift
          JR      NC,SKIP              ;Indicates least sig bit is
                                        zero
          ADD     HL,DE                ;Add partial product to
                                        result
SKIP:     DJNZ    MULTIPLY             ;Do next bit
          LD      (RESULT),HL          ;Store result
```

Fig 15 Z80 8-bit multiplication version two

```
          LXI     H,(MULTIPLIER—1)     ;Get multiplier in H register
          MVI     L,0                  ;Clear to zero
          MVI     B,8                  ;B is used as a 'bit' counter
          LXI     D,MULTIPLICAND       ;Get multiplicand in E
                                        register
          MVI     D,0                  ;Now DE = multiplicand!
-----------------------------------------------------------------
MULTIPLY: DAD     H                    ;16-bit left shift
          JNC     SKIP                 ;Indicates least sig bit is
                                        zero
          DAD     D                    ;Add partial product to
                                        result
SKIP:     DCR     B                    ;Decrease counter
          JNZ     MULTIPLY             ;Do next bit
          SHLD    RESULT               ;Store result
```

Fig 16 8080 8-bit multiplication

tiplies it by a second number held in location MULTIPLIER, and places the result into the two bytes starting from the lowest byte, which has been labelled RESULT (Fig 13).

Z80 multiply

The code in Fig 14 is split into two parts. Firstly, we load the registers with the following data: HL is loaded with the address of the multiplier, and register C is then loaded with the multiplier itself (using indirect addressing through HL). A 'bit count' of eight is loaded into the B register, and this will be used to count how many times we have gone through the 'multiplication loop'. The HL pair are then incremented so that they point to the multiplicand, which is placed in the E register using a LD E,(HL) instruction. Register D is set to zero because, although the multiplicand is only eight bits, we'll need 16 bits available as in the 16-bit left shift operation explained earlier. Finally, HL is set to zero and will be used to collect the result prior to storing it in locations RESULT and RESULT+1.

The second section of code is the actual multiplication. We use a right shift operation on the C register so that the least significant bit goes into the carry. This means that if the carry becomes 'set', then the least significant bit was a '1'. The carry flag is tested and if it has not been set, the partial product is zero and we skip the addition. Before moving on to the start of the loop again, the DE pair are shifted using a left shift followed by a left rotation, and the 'bit counter' B is decreased. If B is not zero we repeat the loop again, otherwise the final result is stored in RESULT and RESULT+1.

This 'first attempt' code can be shortened and improved in several ways. The Z80 has a combined 'decrement and relative jump on not zero' instruction. It operates using the B register as the counter and decreases the B register by 1, and if B<>0, the relative jump is performed. Another improvement is also possible, but is less obvious. If the Multiplier is placed

```
                LDA  #0
                STA  RESULT
LDX             #8
MULTIPLY:       LSR  MULTIPLIER
                BCC  SKIP
                CLC
                ADC  MULTIPLICAND
SKIP:           ROR  A
                ROR  RESULT
                DEX
                BNE  MULTIPLY
                STA  RESULT+1
```

Fig 17 6502 8-bit multiplication

in the H register and the L register set to zero, the instruction ADD HL,HL will perform a 16-bit left shift. As the multiplier is shifted out during processing, we create room to store the result in HL.

To take advantage of this arrangement we must shift the multiplier to the LEFT, meaning that we deal with the most significant partial product first. We can also 'tighten up' the initial loading code by loading HL as a register pair starting one byte *below* the multiplier (so that the multiplier goes into the H register). The L register can be cleared after this 16-bit load in readiness for receiving the result. A similar 'trick' can be used to load the multiplicand into the E register.

These improvements have been made in the version shown in Fig 15.

Multiplication 8080

Translation to 8080 form is straightforward. All the improvements made in the second Z80 version can be implemented on the 8080 except for the automated DJNZ instruction. Relative jumps are not supported, so normal jump instructions are used in the loop (Fig 16).

6502 multiply

On the 6502, we cannot use any 16-bit 'paired registers', but we can create similar effects by considering the accumulator as the high byte of such a pair, and a memory location as the equivalent low byte. Such a combination can be shifted in the same way as explained earlier. The X register can be utilised as a 'bit counter', and an LSR (logical shift right) instruction can be used to push the least significant bits of the multiplier into the carry flag; this is used to decide whether or not to add the multiplicand.

In the example shown in Fig 17 the multiplicand is not shifted, it is just added to the accumulator. We right shift the 'accumulator memory byte' 16-bit pair using ROR instructions, and this provides an equivalent alternative.

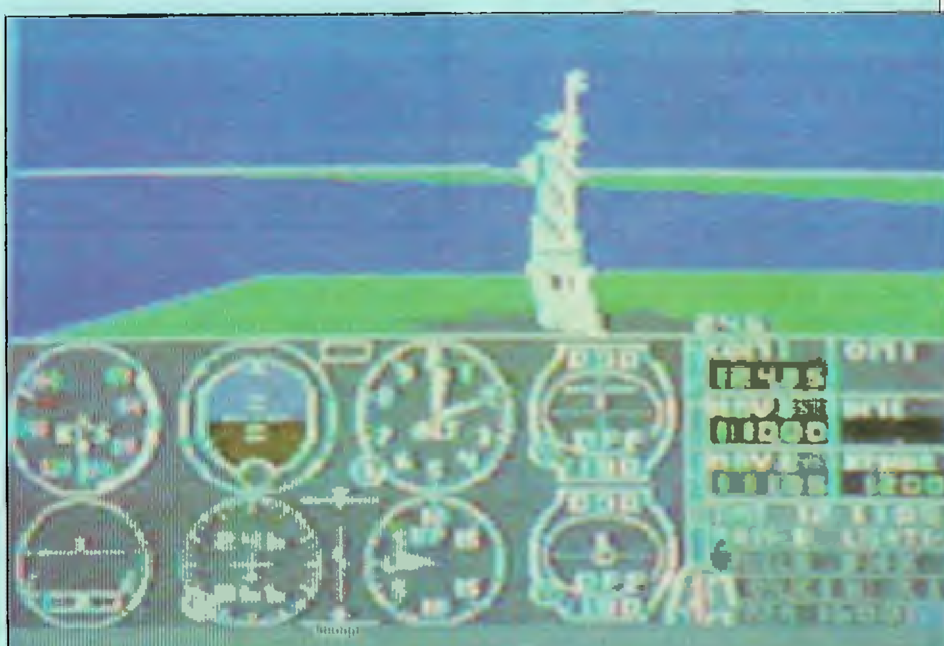
Did you try the left shift experiment suggested last month? If you did, you will have found that shifting a number to the left is equivalent to multiplying the number by 2. Similarly, two left shifts are equivalent to multiplying by 4. In general, an 'n bit' left shift will multiply the value by 2 raised to the power 'n'. END

'Well, we've made a start. We've agreed to cut back the distribution of video games.'

Poised for flight

The fact that flight simulators are in demand for the Commodore 64 is proved by the virtual impossibility of finding a copy of Flight Simulator II in any computer store. (Believe us, we've tried to get more than the review copy for several *APC* staff members hooked on this program!). Until recently the 64 was the most poorly served of the home micros in this area — but that has changed. With the release of FSII from Sublogic, the 64 is now *the* micro if you have the urge to head for the wide blue yonder.

FSII is a relative of the highly acclaimed IBM PC simulator from Microsoft which produced its version under a deal with Sublogic. The 64 version is not only streets ahead of the PC package, I would go so far as to say it outshines every other piece of 64 software available. It is a masterpiece of the programmer's craft — author Bruce Artwick has squeezed every last bit of performance out of the 64's 8-bit processor and graphics chip.



Features

The most instantly impressive feature is the full colour, high-resolution display. A few of the best flight simulators give out-the-cockpit views; FSII gives full 360 degree vision in 3D perspective and the detail is astounding. There are skyscrapers, road networks, towns and mountains but the most staggering experience is a flight across Manhattan Island where you can fly around the Statue of Liberty, the Empire State Building, the Manhattan suspension bridge or between the World Trade Centre towers!

These are all in 3D with hidden-line removal and if you try to fly through them your trip will end with shocking speed.

The instrumentation is superb — no digital readouts here but lots of dials that work like real instruments matching the view out of the window. In addition to speedometer, turn and bank indicators, VSI and gyrocompass, there are two VORs (for instrument navigation) that allow you to select frequencies for different radio beacons, and a communication radio. Set the frequency for your destination airport and you get a message from air traffic control advising on weather, visibility and your take-off and landing runway.

Some airports have ILS (instrument landing systems) so you can practise flying and landing blind.

There is also a real-time clock — a necessity since time in the simulator matches reality and if you're in the air when night falls you'll have to learn instrument navigation in a hurry. It really does get dark outside although you can see the airport lights if the weather is fine.

Weather is also handled realistically. You can choose any of the four seasons with suitable changes in weather. Fly the northern states in winter and there'll be snow on the ground, and you can set wind direction and strength at three altitudes, and two levels of cloud. Select low cloud and the sky changes from bright blue to an oppressive grey and if you enter the cloud bank you go to zero visibility. You can fly above it though.

All the features are available through an editor which covers everything mentioned so far and much more. You can select easy or reality flight mode, set your aircraft's position anywhere within the simulator's world, and even set a reliability factor to make life even more interesting: how would you cope with instrument failure at night in thick cloud?

DIRECT CHANNEL TO ASIAN

Micro-Buff combines direct imports &

Twelve years managing hi-fi businesses gave Ray Pope the contacts and skills necessary to turn his interest in personal computers into a new business venture, able to import the best available equipment from several countries and sell it direct to the public.

Dealing with Ray is a refreshing change from many computer sellers as he does not immerse you in technical details, having learnt from his own experience just what is necessary to get his wide range of equipment and software up and running usefully.

He emphasises the importance of after sales service by providing systems support staff for both hardware and software.

Ray started the business in his own home, but that was soon overrun with computers and night and day enquiries, so earlier this year he shifted it into a homely little shop in Mount Waverley and painted his new business name, MICRO-BUFF, on the window.

To keep up with the latest offerings from overseas suppliers, Ray travels regularly, selecting the best offerings to add to his range.

He has built up an excellent reputation with his overseas suppliers and so is constantly deluged with mail offering him distribution rights to new products, enabling him to be very selective as to what is best for the Australian market.

As well as the imported computers, peripherals and accompanying software, Ray has established local sources for software and consumables to complement his range of imports and provide a complete service for the personal computer user.

MICRO-BUFF is centrally located with regard to Melbourne's population distribution, being within half an hour's drive of more than a million people. Located in a traditional roadside strip shopping centre, there is almost always parking available right outside the door.

Ray even commissions original products that he identifies a need for, such as his MICRO-BUFF perspex printer stands which provide space for incoming continuous stationery under your printer so that it does not mix with the outgoing printout. The use of perspex enables visual monitoring of the paper supply, without wasting any

This month's special: Slimline Japanese built Teac or Chinon slim-line disk drives suitable for Apple or compatible only $269 inc tax.

COMPLETE DISK SYSTEM UNDER $1000

The QCAL Z80A-6502 dual processor with 64K-bytes RAM, slim line disk drive, green screen monitor and CP/M operating provides the most economical entry into useful computing.

QCAL gives access to a vast body of widely available software, including Pascal, Basic, Logo, Fortran, Cobol and Z80 Assembler programming languages.

It has a four function numeric key pad, and all keys are able to be programmatically redefined, with special definitions already included to give single keystroke keyword entry for Basic programming.

For $999 (including sales tax) from MICRO-BUFF, the QCAL starter system is the easiest possible entry into disk based computing.

N SAVINGS
& local service

space. The stands are available in two widths for 10 and 15 inch printers, costing $28 and $38 respectively.

Other bargain priced accessories include lockable disk storage boxes for $30 and joysticks for $35.

One of the side benefits of bringing in the BUFF-16 PC compatible (see box) was its introduction to a supply of $399 multi-function disk controller cards, controlling 8 inch and high-density 5¼ inch floppies and up to 30M-bytes of 5¼ inch Winchesters with the one card.

Amongst the extensive range of economical peripherals brought in by MICRO-BUFF are Micron 22MHz monitors which provide a particularly high resolution display suitable for most home computers. The $199 Micron is especially suited to the Microbee computer because it has a 12 volt power supply output plug in the back which can be used to power a 'Bee without a separate transformer.

Special value amongst the printers is the Logitec 15 character per second daisywheel for only $669.

For anyone wanting a very economical entry to two drive computing, the QCAL 1000, bigger brother of the QCAL starter system (see box), provides two internal floppy disk drives, built in 80 columns, ten programmable function and four cursor movement keys, plus the other features of the starter system, all for $1450.

Ray was at first a bit hesitant about bringing in the high performance Japanese Duet 16 MS-DOS system, but, having taken the plunge, it has sold well, with the people who have bought it "raving about it". He has established a new price on the Duet expansion box, giving a 10M-byte Winchester plus eight expansion slots for $2800. Using three of the slots for a 512K-byte RAM card, a multi-serial card and a 68 000 processor card provides an interesting and economical route to a multi-user Unix system.

The success of Duet has proved that MICRO-BUFF has something to offer the serious computerist as well as its established record of looking after many small users. Part of keeping up to date requires Ray to bring in single units for evaluation so he can personally verify suitability before bring in bulk order quantities. On the odd occasion that such evaluation units do not prove suitable for volume importing, the computerist with specialized requirements may find Ray the source of an invaluable once off special.

To keep overheads at a minimum, Ray uses his skills as an importer to have the next shipment land, Customs permitting, the day the previous shipment is sold out. This may occasionally mean that a popular item is temporarily out of stock, but it also guarantees that all products sold by MICRO-BUFF are the latest versions available. For printers, this is particularly important, as desirable new features are being added almost monthly, giving a succession of different model numbers for what is generically the same product.

PC PRICE BREAKTHROUGH

2 Disk 256K under $3000

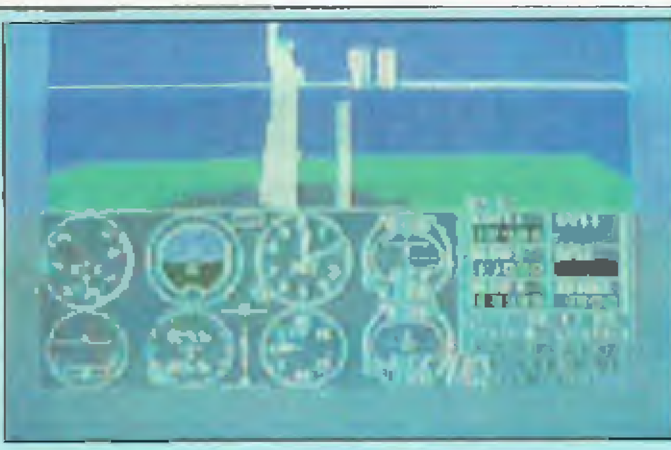
Standard BUFF-16 features include: video output and monochrome screen, clock-calendar with battery backup, parallel and serial ports, and a floppy disk controller capable of handling four drives, all of which can be built in.

A 10M-byte Winchester drive including the controller, suitable for attachment to an IBM Personal Computer or compatible is available for $2100, or packaged with a BUFF-16 which still retains both its floppy drives, for only an extra $1900.

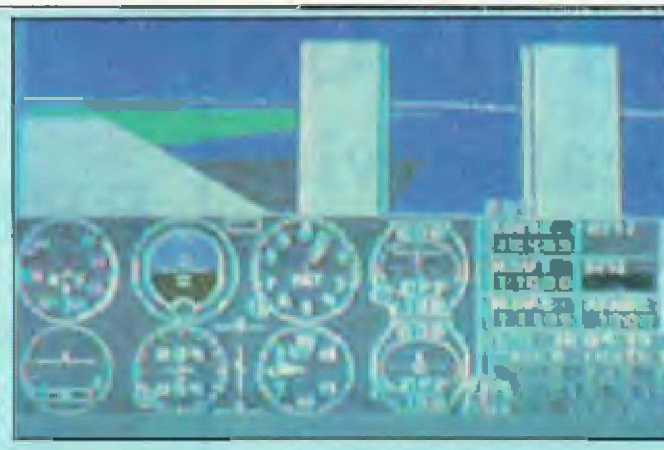
The lowest standard configuration of the PC compatible BUFF-16, including 128K-bytes RAM and two 320/360K-byte floppy drives, sells for $2919, with the 256K-byte version costing $3099.

Use of slimline floppy and Winchester drives enables up to two 10M-byte Winchesters and two floppies to be included in the system box, with the Winchester controller having a second port for external 1.2M-byte 8 inch or high density 5¼ inch floppy drives for high capacity backup.

MICRO ~ BUFF
499 High Street Road, Mount Waverley, Vic., 3149.
Telephone: (03) 277 7754.



A view over Manhattan



Flying dangerously close to the towers of the World Trade Centre

In use

FSII takes the idea of real-time simulation to breathtaking limits and actually flying the simulator is a joy. The sound of the engine changes realistically with changing revs and the scenery is a delight.

Loading from disk takes nearly three minutes but even here there is nice attention to detail. The screen border flashes to let you know things are happening and should you get a disk error it's not a question of starting from scratch. You can choose to ignore the error, which works sometimes, or try again from the last successful disk access. Two or three retries were always enough to get over the problems. You also get to specify whether you're using monitor or TV in colour or black and white.

You then find yourself in easy mode, in user mode 0. One of the most exciting features of FSII is the opportunity to build up a mode library of different locations in different weather and time settings. User mode 0 puts you at a small airport on the shore of Lake Michigan with Chicago off to the left.

Take-offs are quite easy, regular flight a little tricky, and landings frustratingly difficult. After a while you can put it on the ground without crashing but getting it onto a runway takes hours of practice. In reality mode, landing on runways becomes crucial if you want to avoid becoming bogged down in mud, or tearing off your undercarriage.

A smoke trail feature leaves a 3D track of your progress in the sky. You can then fly around it and see how your turns and approaches look.

You control the aircraft either from the keyboard or joystick, or a combination of the two. If there is any criticism of the package it is in the choice of keys for some of the secondary controls. I would have preferred to see the function keys used for some controls instead of the actual CTRL combinations. However, the system is workable and this is very much a minor quibble.

The aircraft in the simulator (a Piper Cherokee Archer) is not rated for aerobatics and I failed to get it to loop but you can have a lot of fun with the rolls, stall turns, spins and the like. You can even fly upside down for a while but it usually ends in a fatal dive.

However, the real fun of the simulator is the scenery. You can undertake marathon cross country flights in real time which forces you to plan your course with refuelling stops, taking in some of the sights along the way. In Illinois there are large towns and rivers, skyscrapers in Chicago, and Lake Michigan; New York features the stunning Manhattan Island trip; Los Angeles has the harbours of LA and San Diego and the Santa Ana mountains; Seattle features lakes and bridges and Mount Rainier. Much of the scenery is taken from aerial photographs and Sublogic promises the release of new library disks with further areas. You may one day undertake transatlantic or round-the-world flights.

Finally, when you think you've really mastered the aircraft you can try the World War I air ace game. Here you're at the controls of a 1917 biplane where you have to bomb enemy fuel dumps and

South of Chicago — radar scan over the Great Kanakakee River

WW1, Europe 1917 — enemy air ace approaching. Note snow on mountains

factories and shoot down six German fighters. Two of them are aces and will quickly demonstrate the difference between a good pilot and a turkey.

You need to shoot down five to be classed an ace; after hours of practice I've never done better than two before having my wings shot off.

Documentation

The documentation matches the standards set by the programming. FSII comes with a 90-page glossy booklet which covers not only the program, but provides a simple guide to small aircraft flight in general. Instrumentation, flight controls, radio equipment and navigation are all covered thoroughly, as well as use of the editor. There is also an extra sheet of late changes to the program and a handy reference card for the controls and instruments.

Finally, four maps cover the main areas in the simulator's world: Chicago, Los Angeles, New York and Seattle. They contain the necessary information about airport altitudes, runways, latitude and longitude settings for the editor, navigation and communication radio settings, plus crucial bits and pieces like whether your target airport has refuelling facilities. You can't just press a refuel key.

Verdict

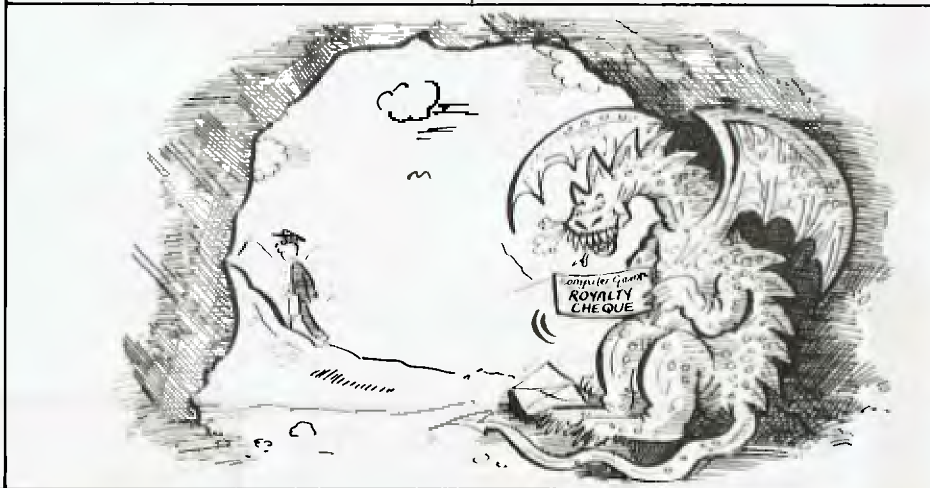
If there is ever going to be a better piece of software for the 64 I can't wait to see it. This is superb programming that realises the potential of the 64 to the full, in effect turning the machine into a dedicated 64k colour flight simulator.

By the standards of entertainment software it is expensive, but it's still good value for money.

At present, only Apple owners can share the delights (although a version for the 64k Ataris may be forthcoming) and it is ironic that 'boring, old' 8-bit technology has produced this kind of masterpiece.

Owners of other machines can only look on in envy and I suspect that Flight Simulator II will sell an awful lot of disk drives and quite a few Commodore 64s.

FSII sells for $64.95 and is distributed by Imagineering.



commercial and professional microsystems

9th Floor, 505 St Kilda Road, Melbourne 3004, Aust. Telephone (03) 267 7829

C.P.M. MINI CONQUEROR

THE COMPLETE DESKTOP SOLUTION — POWERFUL, EASY-TO-USE, FRIENDLY, RELIABLE

COMPRISING:

- 12in green screen terminal, on swivel base with detachable keyboard. 95 keys, separate numeric key-pad, 10 function keys, built-in graphics capability
- Z80A running at 4MHz
- 64K bytes of RAM, 4K bytes of ROM
- 2-RS232 serial ports
- 2 RS232 serial ports Software selectable baud rates to 38.4K baud
- Mass storage: two double sided, 80 track disk drives, 800K bytes (formatted) per diskette 1.6M-bytes total internal storage.
- CP/M 2.2 with enhanced BIOS & CCP
- Integrated accounting software: General Ledger, Accounts Recievable, Accounts Payable, Stock Control & Invoicing.
- Reads/writes other popular computer formats

OPTIONAL: (1) T Maker III software **$365.00****, wordprocessor/spreadsheet/database management/bar charting. **(2)** Friendly — operating system overlay — **$79.00****. (You won't have to learn how to use CP/M

SPECIALISED BUSINESS SOFTWARE: For nearly all fields, e.g., motel, insurance, real estate, medical/dental, farming, video shops, etc., etc., also available.

$2595.00**

**Prices quoted are without tax

Introductory offer.

COMPUTER MEDIA TRANSCRIPTION/COPYING

- Software duplication, copying and formatting for virtually all popular computers
- Downloading 8" CP/M formats to over 60 5¼" CP/M formats.
- Conversion and downloading from CP/M to MS-DOS/PC-DOS.
- Copy protection for most popular formats.
- Rapid turn around.
- Reasonable charges

SOFTWARE AUTHORS/DEVELOPERS & DEALERS

At last a computer that lets you read/write/format virtually all soft-sectored CP/M formats, whether single or double density, single or double sided, 5" or 8", 48 TPI or 96 TPI.

INCLUDES: • Z80A at 4MHz.

- 2 serial ports for printer/modem/ & terminal
- 1 5¼" DSDD 40 track 48 TPI drive.
- 2 5¼" DSDD 80 track 96 TPI drive
- 1 8" DSDD 77 track drive
- CP/M 2.2 & all disk format conversion software and utilities

ALSO AVAILABLE: 16-bit computer (IBM PC compatible) for format conversion, or buy software separately, see details in this ad.

SPECIAL IBM PC (AND COMPATIBLES) UTILITIES

- **OPERATING SYSTEM CONVERTERS:**
  - CP/M 80 to MS-DOS/PC-DOS — Run your 8-bit software on the IBM PC without any modifications
  - CP/M-86 to MS-DOS/ PC-DOS — Permits execution of object code written for CP/M-86 under MS-DOS/PC-DOS.
  - MS-DOS PC-DOS to CP/M-86 — As above in reverse.
  - Z80/8086 Cross Assembler.
  - 8086/8087/8088 Assembler.
- **DATA/TEXT FILE CONVERSION:** Convert any data/text files from CP/M xx to MS/PC-DOS, CP/M xx to CP/M xx, MS/PC-DOS to CP/M xx, running on MS-DOS 2.0
- **TRANSCRIPTION UTILITY:** Copy, Format, Read/Write dozens of different formats in your PC
- **FORMATTING SOFTWARE FOR ALMOST ALL ADD-ON DRIVES:** From DSDD 80 track 96 TPI drives to the whopping 2.5M-byte 5¼" floppy drives

STOP PRESS WE CAN NOW READ, WRITE AND FORMAT APPLE DISKS IN THE IBM PC OR ELSE TRANSFER APPLE DOS FILES TO IBM PC-DOS FILES.

THE CONQUEROR

POWERFUL BUSINESS/SCIENTIFIC COMPUTER

INCLUDES ALL THE SOFTWARE YOU NEED TO WORK BETTER AND FASTER WITH WORDS, WITH NUMBERS AND WITH IDEAS, NOW!

$2165.00**

Not including terminal

- **POWERFUL MICRO:** Z80A CPU, 4MHz, 64K RAM, 2¼" 80 track 96 TPI drives, 1.6M-bytes (formatted) total storage, 2 RS232 ports Compact 6" x 6" x 12" cabinet.
- **POWERFUL SOFTWARE:** Complete software package including:
- CP/M compatible operating system.
- Spellbinder word processor
- DBase II from Ashton-Tate
- Business Master II Accounting.
- PBasic
- Protype touch-typig tutor
- System utilities including multiformat read/write utility.

• **DATA LOGGING SYSTEM** also available. • **CONTROL MODULES** (input/output to other devices & appliances also available.

OAKLEIGH (03) 568 6911

Star constellation

Novag's Constellation is already a force to be reckoned with in computer chess, but even better things are on the way, as Tony Harrington discovers . . .

Novag has beefed up the standard Constellation's processor speed from 2MHz to 3.6MHz, which, for very little more money, offers a definite increase in power over the standard 2MHz version.

In playing terms, the increased processor speed has resulted in a slight gain in strength at tournament level and an appreciable gain at the blitz level. Ken Thomson (of Belle fame) reckons that one ply (one move by either side) is worth 250 ELO points. You have to take account of the branching that occurs in the search tree when a program considers a further ply. With an efficient pruning algorithm, the branching factor is about six, so six times as fast gives you 250 ELO points, 1.8 times as fast gives you 75 points — thus putting the 3.6 version 75 ELO points ahead of the standard version.

What this actually means to the player attempting to beat Constellation is difficult to define. It is also a little academic, since Novag is about to launch the Super Constellation, which has already proved itself much stronger than even the beefed-up 3.6 version.

Nevertheless, the 3.6MHz machine with its 16k program will be around for a while yet, despite the imminent launch of its bigger 52k program, 4 MHz brother. And since it will, in all probability, be considerably cheaper than the Super Constellation it should have no difficulty in finding takers.

The games section contains the results of one tournament level game played by the 3.6 version against the Mephisto III (with annotations by Grandmaster Dr John Nunn).

In a one off game anything can happen. My interest in this particular game is twofold. Firstly, the game shows that the Constellation, even in its 3.6 MHz format, has a tendency to over aggressive tactical plays which can get it into self-inflicted difficulties. Secondly, it produced, by computer chess standards, a very interesting ending which both machines played better than might have been expected.

While the Mephisto III seemed to be able to hold its own at tournament level, at speed chess (five seconds a move) there was no contest. The Constellation 3.6 scored a clean sweep here with five out of five.

So, from the buyer's point of view, if you like playing speed chess, the 3.6MHz machine looks good.

The Super Constellation looks like being even better. At the Commonwealth Championship (sponsored by Novag) and held in Hong Kong recently, the Super Constellation beat 18 out of 29 tournament players at blitz chess (and all the players in the tournament had ratings of 2200 or over). Its victims included a couple of international masters; only the Grandmasters seemed to be able to cope. The Super Constellation's rating for the tournament proper (as opposed to the speed chess event) was 190.

This still has to be confirmed, but the ELO equivalent is around 2120.

David Kittinger, the program's designer, reckons that it could be the first master level microcomputer program. Enrique Irazoquil, who carried out a substantial review of the Constellation's play for *Computer Chess Digest*, sums it up thus:

'Constellation is tactically better and positionally weaker than most human players. It plays some very nice games and some rather poor ones. At speed chess, it is the best thing on the market, with the possible exception of the Fidelity Prestige.'

Novag has entered an experimental form of the Super Constellation in several other open competitions, including the 1983 US Open.

It recorded the first win of any microcomputer over a rated master under actual tournament play conditions, beating Jerry Simon, rated 2207 in 55 moves. One Al Goncer, rated 2037, also fell victim, as did Strayer, rated 2138, and one or two others who should have known better.

This level of performance is much stronger than most casual players can expect to beat. So it opens the question, once again, as to the point of developing even stronger chess programs.

The real benefits of further development will be felt in better play at the faster response times. Casual chess players like to have virtually instant responses to their moves. Hanging about for five or ten minutes while the computer grinds out a reply at tournament level time settings is tedious; Super Constellation promises to take a large step towards solving that particular problem.

Its arrival on the market will send shivers through Novag's competitors, since if the commercial version lives up to this preview when it is released in

September or October this year, there will be nothing in its class.

Games section

White: Mephisto Y. Black: Super Constellation. Budapest 1983. Sicilian Defence. Notes by David Levy.

1 e2-e4 c7-c5
2 Ng1-f3 d7-d6
3 Bf1-c4 e7-e5
4 0-0 Ng8-f6??

(A terrible move, but it requires an 11-ply search to realise that White can win a pawn in reply.)

5 Nf3-g5 d6-d5
6 e4xd5 Bc8-f5

(Now Black can see that 6 . . .Nf6xd5 allows 7 Ng5xf7! Ke8xf7 8 Qd1-f3+ Kf7-e6 9 Nb1-c3, winning back the piece with an overwhelming game.

7 Nb1-c3 Bf8-d6
8 Bc4-b5+ Nb8-d7
9 d2-d3 0-0
10 f2-f4?

(Unnecessarily opening up a diagonal to White's king, and by allowing the trade of the d6 bishop immediately increasing the pressure on the d5 pawn. White was a safe pawn up and need not have taken any such risks.)

10 ... e5xf4
11 Bc1xf4 Bd6xf4
12 Rf1xf4 Bf5-g6
13 Ng5-e4

(No matter how White plays, the d5 pawn is looking precarious.)

13 ... Qd8-b6
14 Ne4xf6+ Nd7xf6
15 Ra1-b1 a7-a6
16 Bb5-c4 Qb6-d6
17 Qd1-f3 b7-b5
18 Bc4-b3 Ra8-d8
19 Rb1-e1 h7-h6
20 a2-a3 Nf6-h5
21 Rf4-h4 Qd6-b6

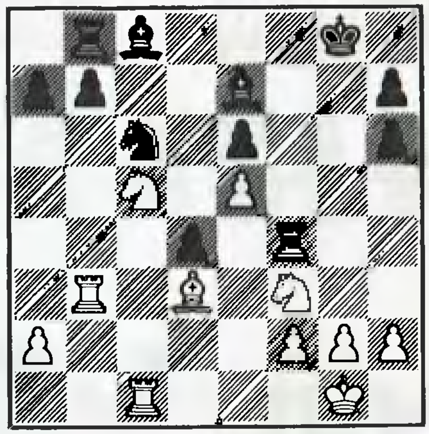
(Threatening 22 . . . c5-c4+

22 Qf3-e3 Nh5-f6
23 Nc3-e4 Rf8-e8
24 Ne4xf6+ Qb6xf6
25 Qe3xe8+ Rd8xe8
26 Re1xe8+ Kg8-h7
27 Rh4-h3 Qf6xb2

For the time being the complications have come to an end, and once White loses the a3 pawn the position will be materially level (a queen and pawn are worth roughly the same as two rooks). White's rooks are not particularly well coordinated in this position, but with correct play I would not expect White to lose. On the other hand, it is easier in a computer game to play with an active queen than with two rooks.)

28 Rh3-e3 a6-a5
29 d5-d6

(Losing the d-pawn, but the threat of . . . a5-a4 could not be met.)



Position after 10 . . . Nf6xe4

29 ... Qb2-a1+
30 Kg1-f2 Qa1-f6+
31 Re3-f3 Qf6-d4+
32 Kf2-f1 c5-c4
33 d3xc4 b5xc4
34 Bb3-a4 Qd4xd6
35 h2-h3 c4-c3
36 Ba4-b3 Qd6xa3
37 Kf1-e1

(After 37 Kf1-g1 a5-a4 38 Bb3xf7 Bg6xf7 39 Rf3xf7 Qa3-c1+ 40 Kg1-h2 Qc1xc2, Black's passed pawns look very menacing.)

37 ... Qa3-c1+
38 Ke1-f2 Bg6xc2
39 Bb3xf7 Qc1-d2+
40 Kf2-g1 Bc2-a4
41 Bf7-g8+ Kh7-h8
42 Re8-c8 Ba4-d7
43 Rc8-f8??

(43 Rc8-d8 appears to save the game: for example, 43 . . . c3-c2 44 Bg8-b3+ Kh8-h7 45 Bb3xc2.

43 ... Qd2-d4+
44 Rf3-f2?

(As a matter of principle I would prefer not to walk into a pin, and would have moved the king instead.)

44 ... h6-h5
45 Bg8-b3+ Kg8-h7
46 Bb3-c2+ kh7-h6
47 Kg1-f1??

(Correct was 47 Kg1-h1. See how much better things would be for white had it played 44 Kg1-h1 instead of 44 Rf3-f2.)

47 ... Qd4-e3
48 Rf2-e2 Bd7-b5
49 Rf8-h8+ Kh6-g5
50 h3-h4+ Kg5xh4

(White resigns.)

White: Constellation. Black: Mephisto III. Notes by Grandmaster Dr John Nunn.

The Constellation is the 'fast' version, running at 3.6MHz. Mephisto is reputed to be better at strategy than tactics, but it quickly exploited an opening error by Constellation to win a pawn. A further dubious knight adventure by Constellation should have cost the game, but Mephisto erred and a fascinating ending resulted.

1 c2-c4 e7-e6
2 Ngl-f3 c7-c5
3 e2-e4 Nb8-c6
4 Nbl-c3 Ng8-f6
5 d2-d4 c5xd4
6 Nf3xd4 Bf8-b4
7 Bcl-g5??

(A move capable of sending a shudder down the spine of a hardened chess master. Material loss is unavoidable after Black's reply. 7 Nd4xNc6 was essential.)

7 ... Qd8-a5!

(With threats to c3, g5 and e4.)

8 Nd4xNc6 Bb4xNc3+
9 b2xBc3 Qa5xBg5
10 Nc6-d4 Nf6xe4

(Black has not only won a pawn but also inflicted serious weaknesses on White's queenside.)

11 Nd4-b5 Qg5-e5?!

(11 . . . 0-0 was a simpler and better way of meeting the threat 12 Nb5-c7+.)

12 Qd1-d4 Qe5xQd4
13 c3xQd4 0-0

(White's pawn structure has been improved by the exchange at d4 and although Black should still win, he now faces a much harder task.)

14 Bf1-d3 f7-f5
15 Nb5-c7

(The start of a pointless knight manoeuvre. 15 f2-f3 followed by Nb5-d6 would have occupied a useful outpost.)

15 ... Ra8-b8
16 0-0 b7-b6
17 Nc7-b5 a7-a6
18 Nb5-a7?

(Suicidal. The knight has no way to escape from a7 and should soon be rounded up.)

18 ... Bc8-b7
19 f2-f3 Rb8-a8?

(Black sees the chance for temporary material gain and takes it. However, 19 . . . Ne4-f6 would have left White helpless against the threat of 20 . . . Rb8-a8 winning a piece.)

20 f3xNe4 f5xe4
21 Rf1xRf8+ Kg8xRf8
22 Bd3-e2 Ra8xNa7

(Thanks to his 19th move Black has won a second pawn, but now White forces the recapture of one of the lost pawns.)

23 Ra1-b1 b6-b5
24 c4xb5 a6xb5
25 Be2xb5 Kf8-e7
26 a2-a4 Ke7-d6!

(Mephisto understands that the king should be used actively in the end game.)

27 Rb1-f1 Bb7-c6
28 Rf1-f7 Kd6-d5

(Black's passed pawn at e4 will become a powerful weapon when aided by his king. With a potential queen at stake, Black doesn't mind the loss of his kingside pawns.)

29 Rf7xg7 Bc6xBb5
30 a4xb5 Kd5xd4?

(Black shouldn't take this pawn because it can be used as a shield to defend his king against checks from White's rook; for example, he could have won easily by 30 . . . e4-e3! 3 Kg1-f1 Kd5-e4 32 b5-b6 Ra7-a1+ 33 Kf1-e2 Ra1-a2+ 34 Ke2-f1 Ke4-d3 — notice that the pawn on d4 prevents Rg7xd7 with check — and the e-pawn will promote.)

31 b5-b6 Ra7-b7?

(Throwing away the win. One of the basic rules of rook and pawn endings is that the rook should be deployed to the rear of enemy passed pawns. 31 . . . Ra7-a1+ 32 Kg1-f2 Ra1-a2+ 33 Kf2-f1 d7-d5 34 b6-b7 Ra2-b2 would still win.)

32 Rg7xh7 e4-e3
33 Rh7-h4+?

(A misguided move. After 33 Kg1-f1! Kd4-d3 34 Kf-e1 White can draw because the pin along the 7th rank prevents the advance of black's d-pawn.)

33 ... Kd4-d3
34 Rh4-b4 d7-d5
35 Kg1-f1 d5-d4

(The advancing mass of Black pawns is more than enough to overwhelm White's king and rook.)

36 h2-h4 Kd3-c3
37 Rb4-b1 Kc3-c2
38 Rb1-a1 d4-d3
39 Ra1-a2+ Kc2-b1
40 Ra2-a7 Rb7xb6
41 Ra7-a5 Rb6-b2
42 Ra5-a7 Rb2-f2+
43 Kf1-g1 d3-d2

(The space invaders are about to touch down.)

44 Ra7-b7+ Kb1-a2
45 Rb7-a7+ Ka2-b2
46 Ra7-b7+ Kb2-c2

(Getting the right idea second time around.)

47 Rb7-c7+ Kc2-d3
48 Rc7-d7+ Kd3-e2
49 g2-g4 d2-d1=Q+
50 Rd7xQd1 Ke2xRd1

0-1

LAZING AROUND

BY J J Clessa

Quickie

No prizes.

There are 10 socks in a drawer — five blue and five grey. Assuming it's too dark for me to see, what is the least number of socks I would have to take out to be sure of getting a pair the same colour?

Prize Puzzle

The Andersons have only one child
The Browns have two children
The Coopers have three children
The Dawsons have four children
The Edwards have five children
The Firths have six children

At Xmas, each family gave gifts of money to their children and altogether exactly $1000 was given to the 21 children.

The children of the two Roman Catholic families each received an equal amount.

The children of the four remaining families also received an equal amount each, but it was twice as much as the amount given to the Roman Catholic children.

Which two families are Roman Catholic?

Answers, on postcards only please, to: Prize Puzzle August 1984, Lazing Around, *APC*, 77 Glenhuntly Road, Elwood, Victoria 3184 — to reach this office not later than 15 August 1984.

May Prize Puzzle

5	7	6
1	?	7
2	1	6

The winning entry came from Mr KJ Ralph of Hobart. Congratulations.

Rapid access

How many micro users out there, working in large data processing departments, wish they were more effective in solving the small, everyday problems encountered by the typical user department? DB Piper suggests an in-house Information Centre to provide end users with the required personal computing facilities.

Suppose that all you want to do is 'take this print, turn it around a bit, knock out the page, re-sort the rest and sub-total here instead of there'. In comes the Systems Analyst, who talks for an hour and concludes the whole discussion with the words: 'About $2000 and six months time, sign here . . .'

This type of interchange creates the typical concept of a slow corporate data (DP) department, very expensive and incapable of providing answers at all. Small wonder then that many large companies find microcomputers being used in rapidly increasing numbers.

Problems

The problems which this uncontrolled use of micros can create are sometimes hard to visualise from the departmental point of view. On a corporate basis, it will be desirable to maintain at least a monitoring role, if not some form of active control, of the micros being introduced, allowing rationalisation of the data being held and maintained (reducing the risks of inconsistencies in the data), and preventing users reinventing the wheel.

Users are unlikely to accept this interference (as many will see it) with any relish, and the DP department may not even be consulted during the introduction of a new micro. The worst possible situation arises when all communication stops between the user and the DP department; users will be unaware of existing facilities which may fulfil their requirements, and the DP department will fail to recognise the increasing demand for 'personal computing'.

The requirements which the DP department must fulfil to make a success of personal computing on a corporate basis are:

1 To be seen as an approachable centre of expertise on methods, and on hardware and software.

2 To find a method of spreading this expertise among the end users without alienating them.

3 To provide education to users on 'how to do it', and on designing efficient solutions to problems.

The concept being put forward as the solution for this problem is that of the 'Information Centre' (IC). In order not to be classed under the same banner as the rest of the DP department, the IC must be visibly separate from it (although functional links must exist and be maintained.

The IC must have a high profile and be visible to end users.

Help, education & support

The IC is made up of three major parts. The Help Desk provides a point of contact for end users. Personnel have minimal technical expertise, but act as a distribution and monitoring point for problems.

The Education Team is responsible for running all educational courses, ranging from simple introductory seminars designed to raise user awareness of computers and their capabilities, to advanced courses designed to teach users how to design and write their own systems. Seminars are also run by the Education Team for departmental managers to illustrate how personal computing can be used effectively in their own departments.

Support is the third major area provided by the IC. The Support Group gives advice and expertise to end users wishing to implement their own systems. Programming support can also be provided, but users are encouraged to do their own programming as much as possible. Technical Support is provided by other personnel within the Support Group, providing information on potential problem areas, new facilities and features, and how to use existing facilities more effectively.

A major feature of the IC is 'Vertical Integration' — the IC is designed to facilitate personal computing of all kinds, from systems implemented on micros to personal computing on mainframes, providing a single source of expertise.

'Copy Management' can be broadly defined as the controlled copying of data from one place to another. Users with micro-based systems will wish to access data from existing mainframe systems. Re-keying data already resident on one computer system is inefficient, but happens frequently in an uncontrolled environment. The important facets of copy management are:

Security: only authorised users should be allowed to copy data from the mainframe databases.

Sources: data for one user may be collected from a variety of sources on the mainframe.

Frequency: a user may require data to be refreshed once a month, or on a daily basis. If access is to be more frequent than this, then direct access to the data is a better solution.

Formats: with the variety of data formats used by micros and micro packages, attention must be paid to the format of the copied data.

Hardware integration exists now, but why not take the whole process a step further and include software integration? The major advantage of this concept is that a user only has to learn one set of rules to access data rather than two or three. Several producers of the 'fourth generation' languages (English-like data extract/query languages) are already experimenting with this approach.

The Information Centre is a facility being introduced by many data processing departments to provide end users with the personal computing facilities they require. By providing these in a controlled manner, efficiency can be increased without decreasing the rate of response to end users. The facilities are provided centrally, ensuring effective, consistent support for both hardware and software.

Facilities can be made available either on micros or mainframe computers, or both. Advances in hardware and software will give rise to integrated environments on all sizes of computers, which will further encourage the use of microcomputers in large businesses. In the longer term, intelligent communications between micros and mainframes will allow users to access mainframe data as though it were resident on their own micro.

END

Creating a program

Last month Mike Liardet gave many useful hints on the strategy to adopt when making mass-market software. He concludes his lesson this month with a thorough grounding in tactics.

Last month I looked at how you might get an idea and schedule your time accordingly. Imagine you have done this — you are sitting at the computer keyboard, you know exactly what you want to do and estimate that you're just a few months of coding away from the best computer product since Babbage's Difference Engine. Where do you go from here?

This is the moment when you have that unique opportunity, seldom available to most of us, to completely and utterly ruin a good idea. In the long run, it won't matter how smart or clever your product is: if no-one can understand how to use it, no-one will want to buy it.

Clearly, some careful thought is needed before you start coding. If you develop a good framework for the software right at the start, then everything that follows will slide easily into place and the user will find it much easier to follow.

Structured programming

A major ally for getting the overall framework correct is to adopt the structured programming technique. A great deal has already been written on this subject, much of it making far more of it than necessary. The basic idea is quite simple: you sacrifice a theoretical 2-3% of program efficiency in the interests of clarity. Instead of one monolithic program, you create lots of self-contained but inter-linking modules. These are normally arranged in a hierarchy, with the top-level module calling on lower ones, and so on, down to very primitive modules such as 'read a character'.

Once the amount of code exceeds a few pages, this is the only way you can keep track of what's happening. As a bonus, it also guarantees that your low-level code is very thoroughly tested, as it's called upon from many places and in all types of situations.

I have found it particularly useful at the preliminary stage of a project to construct (on paper or in my head) a 'world'. If you are planning an adventure game this is a fairly natural thing to do, but it might seem less obvious for an accounting system or spreadsheet.

Now an exercise that may come easily to some of you: imagine you are God. In fact, why restrict yourself? Imagine you are a *lazy* God. You have complete power of creation, destruction and direction of the world's population, but you also have an underlying purpose which you wish to achieve with minimum effort. This purpose might be to find all index cards with a particular match, perform calculations, and so on, depending on the application (or rather, depending on the 'world'). But whatever it is, you expect the world and its population to be maximally cooperative. If your world's population is lined up and you eliminate one of them, then you expect everyone else to automatically shuffle up to close the gap.

If you do this exercise for some time with different scenarios, you'll get a good feel for the operations that you want to perform on the population, and the answers you expect back.

Following this enjoyable session of megalomania, it's natural to start wondering about the form your commands and edicts will take, and also how you'll be told what's going on as a result of them. You must next design the interaction to support the dialogue needed to manipulate this world.

The original task of implementing software has now become one of implementing a world, and organising communications to and from the world. The communications medium should place as small a barrier as possible between you on the outside and the world on the inside. If you can arrange this, and the world is easily understood and well-behaved, then you are 90% of the way to a clear, ideal 'user interface'.

The output language should incorporate clear, brief, non-jargonistic, readable text. Regrettably, at least until Artificial Intelligence research delivers the goods, the input language will be more terse: you'll probably decide to direct the software through some form of menu control or command language. Using menu control is particularly attractive as it naturally imposes a structure on the underlying code, with menus branching into sub-menus, and so on. I prefer menus where selection is based on the initial letter of the option, rather than a number: the number of keystrokes are reduced and it's easier to remember which keystrokes are needed. In fact, it's a non-trivial exercise to invent 10 or more option names, each with a clear meaning but starting with a different letter of the alphabet. The latest software uses graphic icons instead of written messages, which are selected by a pointing device such as a mouse, but they can still be regarded as a menu nonetheless.

Some software operates from a command language, frequently a type of pseudo-English. The problem with command languages is that they involve a lot more typing. There are problems with mis-spellings and it's not usually certain which commands are invalid in particular situations. In summary, they give the user every opportunity to make an error, whereas menus provide very little scope for this. Menus make things easier for the programmer too — a rare situation in software design.

The program can also be controlled by function keys. On older computers, this usually involves using the CONTROL key in conjuction with alphabetic characters. On newer systems, there is

usually a range of arrow keys and numbered function keys available, enabling ordinary input and commands to be freely interleaved. For example, if you type ordinary letters into a word processor it simply echoes them onto the screen, whereas the function keys don't type anything on the screen, but move the cursor or delete words — it's possible to ruin the smoothness of interaction if you get this wrong. Microsoft's MultiPlan insists that actual data entry be preceded by a frequently forgotten function keystroke, and the program interprets what you type as a command sequence instead.

There are various useful rules for organising prompts from the program. Once a user has responded to a prompt, wipe it out — don't leave it polluting the screen to cause confusion. The easy way to do this is to write over it with the next prompt or, if there's a delay for processing, write over it with an apology for the delay.

One of the most important aspects of the user interface is the validation. Whatever key is pressed, the system must be equipped to deal with it and should never crash if the wrong type of input is given. Most programming languages supply a numeric input procedure, which can universally be relied upon *not* to do the right thing if the user enters letters or other rubbish. This means that you must write your own input validation software. A good approach is the 'dead key' technique — any illegal keys are quietly ignored. You may decide that a 'beep' is called for, but this can be irritating to the user or his colleagues if he's in a crowded office. If the input must be in upper case, don't wait for the user to type a line before telling him — do the case conversion for him as he types.

General design

The basic requirement for the software is that it's easy to learn, easy to use, and robust and reliable for a (hopefully) huge range of users. The one key fact to bear in mind is that software serves the user, not the other way round, so when coding always imagine yourself in the driving seat and try to make things as easy as possible for the driver.

Be on the lookout for any inconsistencies in jargon or interactive techniques. They are easily introduced but equally easily corrected. For example, decide on one key (ESCAPE, for example) as a quick exit keystroke and stick to it throughout. New users frequently make an incorrect choice of operation, and if they know that this one key will always 'unselect' for them it makes things easy. (This implies that the keystroke is permanently available, no matter what is happening. All too often, software forces you through a series of irrelevant questions when you know that you have selected the wrong path after the first question: very frustrating.)

Another important aspect is 'reversibility'. If a user does something, can he as easily undo it? In a word processor he might have accidentally moved a block of text, but can he move it back? Some operations, by their very nature, are irreversible ('Delete all Files', for example) and should be specially prompted with an unambiguous message ('Enter 'OK' to do it').

Some activities may take longer than a split second to perform, and if this is the case a pause message, preferably with a count*down* (not a count-up) should give an appropriate indication. Very long delays, as in lengthy printouts, should be signalled with a time estimate so that the user can go to lunch. Lengthy procedures should also be interruptible and restartable, in case of paper jams.

It's worthwhile putting some effort into the program's presentation. An attractive layout may not add anything to the functionality, but will make it look much more appealing to publishers, customers, dealers, and anyone you are trying to interest in it.

A lot of recent software has a help facility, much as you always have an ESCAPE key. Unfortunately, much of this software is insensitive to the context under which help is being requested. I have mixed feelings about help screens; if the screen prompts are properly organised initially, the user should not need any further help. Imagine he is attempting to enter a number, which the software will not accept. The HELP key is pressed, and the information that only integers in the range 0 to 99 are acceptable at this point is given. This information should have been supplied in the first place!

Screen display

Most micro owners take for granted the highly-interactive screen and instantaneous responses to their commands, but this style of interaction with computers is comparatively new. For many years, interactive computing was based around the Teletype (interactive keyboard, paper output) and before that much computing was non-interactive batch processing (input from cards or paper tape with paper output).

The computing world has been slow to exploit the new medium. Some software still has that Teletype look about it, with the screen simulating a piece of paper and scrolling before your very eyes (rather like a primitive tribe who use a brand new lorry by towing it with their oxen).

Most screens today have a variety of control mechanisms which would be meaningless on a Teletype, so it's possible to clear the screen, direct characters onto it at any selected point, and select special display modes such as inverse or flashing. You should definitely use all this without too much detail, and it's not too difficult to implement, in a sufficiently general way, a means for your software to operate with any number of different screens with little or no modification.

Some highly innovative software has emerged recently, enabling software developers to implement very sophisticated screen operations. These systems (Apple's Lisa, Microsoft's Windows and Visicorp's VisiOn) provide windowing facilities, where the screen can be split with different tasks executing in each part, and all under the control of a pointer (such as a mouse). The important point to note here is that these systems took decades to develop, and there is no point in trying to compete by implementing equivalent facilities yourself. They are actually designed to be used by software developers, so it's more a case of putting them on your shopping list if you want to arrange software interaction in this fashion.

Keyboard

The keyboard is likely to be the major source of input for your software, although the mouse and other devices can be particularly easy to use for pointing and moving operations.

Make sure that the keyboard input is still accepted when something else is being done. You don't need to write interrupt-driven software for this (arrange that at any time): consuming activity polls the keyboard at regular intervals.

If your software is driven by function keystrokes, give some thought to the keyboard ergonomics. If control keys are to be used, arrange that most of them are keys near the CONTROL key. That way, the user will only need one hand to use them. Watch out for keystrokes with some local effect on the screen unknown by the software — at least provide the user with a redraw facility should this happen.

Most software, at some stage, requires the use of peripherals such as disk drives or printers. There are two main things to consider here: coping with user errors, and dealing with different types of hardware.

Once the software has been thoroughly debugged it can be assumed to be error free, but errors can still occur if, for example, the user asks it to write to disk and leaves the drive door open. For some operating systems like CP/M, this causes a major failure untrappable by your software, so there is little to be done. But most operating systems return an error condition to the program which stops attempting writes to disk, tells the user what has happened and returns to a sensible point in the interactive sequence, the point at which he can elect to try again. It's a good idea when writing files to first write to a temporary file, which can be deleted if a 'disk full' condition occurs. When the writing has been completed, a sequence of renaming can be attempted. The back-up version (if it exists) of the file can be deleted, the previously current version is renamed as a back-up, and the temporary file is renamed to the current version name.

The other major peripheral you are likely to be concerned with is the printer. Printers have all kinds of special features: some provide colour, graphics plotting, special fonts, and so on. If you want your product to be as general as possible, you'll have to ignore most of this and stick to differing paper dimensions. The user should be able to interrupt the software in the middle of printing, in case the paper is wrongly positioned or the ribbon has run out. Arrange for the space bar to suspend printing — it's the easiest key to hit in a hurry! If the operating system permits, it's a good idea to test that the printer is switched on before attempting a printout. If you don't, you risk having a very puzzled user with a 'dead' keyboard and nothing happening.

Documentation

A bug can be defined as a discrepancy between the documentation and what the system actually does. The documentation defines the system, and until it's written the product does not exist. Ideally, the documentation should contain both tutorial and reference material, although there is a growing trend for tutorial software to replace the former. Some information on package contents and start-up instructions is invaluable, particularly if placed right where you can find it as soon as you open up. Also, a reference card is invaluable for expert users and costs little to produce, and an index is worth its weight in gold!

When you do get round to documenting, make sure it's well proof-read. Don't spoil good documentation with poor quality control — spelling mistakes and references to nowhere.

If you've been using some form of structured programming techniqe, much of the software will have been tested as you wrote it. Each module should be checked as it's written, as it makes the final testing much easier.

A popular misconception is that a naive user will be the best bet for testing your software. I disagree with this. Such a user would certainly test the tutorial aspects of your software, but not the overall reliability. What you really need is an expert (he will need the documentation too).

Bugs can be corrected by changing the manual — you may add: 'If the filesize is divisible by 256, the program will crash.' This may not be a desirable feature, but it's not a bug if it's documented!

A particularly ripe area for bugs is what I term the zero-case and the infinity-case. The zero-case happens if you delete the last thing left, and the infinity-case occurs when you fill things up to capacity. This may involve entering the largest numbers possible for a calculation, or by filling the system with data. The latter may involve a lot of typing, so it's usually easier to create a huge data file with specially written software.

If you have a conscience about these things and want to have a few sleepless nights, then ask yourself if there are parts of the code that have *ever* been executed. There is probably some dark corner of the software that's used once in a blue moon. Organise a thorough test for it, and if it doesn't fall over I'll send you a buck (well, a metaphorical buck).

If you have several testers, don't let them get together. If they work independently the testing will be more effective. While they are at it, they might as well give you some feedback on what they think of the whole thing. Don't argue if they criticise, just note it all down. If several of them say the same thing, then you might conclude that there's some validity in the criticism.

Security & anti-piracy

Having gone to all this trouble, you will be fairly upset if someone illegally copies and sells the fruits of your labours, so you might consider preventative action before they get the chance.

The most obvious point is not to release your software in source form. If a software pirate gets his hands on your source code he can easily change a few bits, making it harder for you to prove that he has ripped you off. Unless you are writing in Basic, this is unlikely to be a problem. Another easy measure is to arrange for each individual copy of the software to have its own serial number: if pirate copies do appear, you may be able to track down the source from this.

Beyond these measures, the convenience of the user suffers. There are dongles — special purpose hardware which must be present for the software to run. They put the price of the product up and if the user has several dongled packages, the insides of his computer will look like a post-match SCG. There are also techniques for making disks 'uncopiable'. The problem is that the user cannot take back-ups to protect himself from the day when his disk wears out.

The ideal solution will come when computers are released with their own internal serial number (in ROM). Software can then be released in an uncopiable format, which is only unlocked when the serial number has been registered. Thereafter, the software can be freely copied, but will not run unless the correct serial number is present. As most computers are not serialised in this way, I mention this here in the hope that a manufacturer will take the hint and do it. But by the time that happens, software may be so cheap that no-one will be bothered about copying it.

A final point is that it's quite often useful to arrange for a demonstration version of the software to be either given away or sold at cost price - this is usually fairly easy to achieve. The key thing is to keep as many facilities available as possible, but remove one or two essential ones like writing to files, or have greatly reduced capacities, and so on. As regards copying, the opposite is true with demonstration software. Let's face it, you want as many copies as possible to find their way around!

Conclusion

We have touched upon just some of the issues involved in the complex operation of implementing a software product for the mass market.

From the programming point of view, it's different from any other type of project, requiring very high standards of design and reliability, but it also provides an opportunity for some of the most stimulating and rewarding work.

I hope these articles have provoked some of you to have a go.

(E26)

MICRO-EDUCATIONAL PTY LTD

AUSTRALIA'S LARGEST DIRECT MARKETING COMP SUPPLIER

NEW!

DOWN-UNDER CLOCK

for Apple $99!!

Fully PRODOS and THUNDERCLOCK compatible. Battery backed. Grab one NOW!

$65

includes surge suppression!

Clip on fan for your Apple

$349 ex
$399 inc

'AMUST 120'

BOSS-1

$180 ex
$199 inc

Keyboard for Apple II+

The Modem you've been waiting for . . .

HERE NOW!

AUTOMATIC ICE CO. DIRECT CONNECT MODEM FOR APPLE II AND IIE.

1. 300 baud full duplex or 1200/75 & 75/1200 baud with optional automatic line turnaround.

2. Auto answer — auto dial.

3. Fits completely in Apple slot — does not require serial card. Conforms to Apple Firmware' standard — compatible with Pascal and CP/M.

4. Asynchronous or synchronous operation. Supports Bisynch, HDLC, SDLC and X25 at the link level. Flag generation, CRC error code generation and checking are all done by hardware. All registers and operating parameters are fully software programmable.

5. Onboard loudspeaking conference telephone. Software controlled — also audibly indicates call progress.

6. Senses true dial tone, ring tone and busy tone. Generates error messages.

7. Pulse or DTMF (tone) dialling. When used with DTMF telephone exchanges, connection to the called party can be established in less than one second.

8. Has onboard colour Videotext (Prestel) system in firmware. Eventually will also support NAPLPS.

9. Can emulate hayes Micromodem, Hayes Smartmodem, Apple serial card, Prestel terminal as well as its own modes, some of which include automatic error correction based on the Christansen Protocol. A fully transparent mode is available.

10. Default operating parameters are stored in battery backed RAM & can be changed at any time under software control.

11. 2Kx8 CMOS RAM powered by an eight year lithium battery stores default parameters, phone numbers and log on strings, allowing single keystroke call establishment.

12. Provision for EPROM up to 16Kx8.

13. Onboard battery backed clock can time & initiate calls. Also accessible for other uses including PRODOS. Emulates Thunderclock.

14. Optional standard telephone connection on the modem allows easy installation into existing phone plug. When the modem is not engaged the telephone acts in the normal manner.

15. Firmware will eventually contain most communication facilities, making additional communication packages unnecessary. Virtual disk transfer and direct read/write to disk is envisaged.

16. Has socket for installation of DES data cypher chip for security identification and EFT applications. Permission is required from the American State Dept. for purchase of cypher chip.

17. Optional search and setting to incoming baud rate and word format.

18. Can act as an automatic telephone dialler with multiple redial and 'dial alternate number' features.

Note: The hardware is final and is capable of all of the above but the complete firmware may not be available initially. We will exchange the EPROMS when the final firmware is available.

MAIN OFFICE:
40 LAMAN ST
NEWCASTLE
POSTAL ADDRESS: PO 154
CHARLESTOWN 2290
9 to 5 (049) 26 4214

STD FREE ORDER HOTLINE

(008) 02 5229

bankcard MasterCard

Enquiries:
(049) 26 4214
(049) 26 4002
(049) 26 3803

TEACH YOURSELF LISP

Most modern artificial intelligence languages were developed from, and often written in, a dialect of Lisp. In the first of his Teach Yourself Lisp series, Dick Pountain examines the concept of list processing.

The forty year history of digital computing is punctuated by the emergence of powerful ideas, but the most powerful of all may turn out to be that of List Processing (or Lisp!).

In one form or another List Processing has become the dominant programming technique in Artificial Intelligence (AI) research, but it's completely failed to penetrate into the world of commercial computing and is unknown to most microcomputer enthusiasts. An enormous gulf, almost too wide for mutual comprehension, has opened up between those AI programmers who cut their teeth on Lisp (or Prolog or POP-2), and the rest of us who started with Basic and moved on, if at all, to Assembler, Pascal, Forth or C.

In this series we shall try to explain the principles behind Lisp, the mother and father of list processing languages (and many would say still the most elegant).

Lisp is, rather surprisingly, one of the oldest computer languages, having been conceived between 1960 and 1965 by John McCarthy at the Massachusetts Institute of Technology (MIT). Its relative lack of popularity can be put down to the fact that it's not very suitable for routine commercial data processing, being quite inefficient in both speed and memory requirement. All the more modern AI languages such as Logo, Prolog and POP-2 are heavily influenced by Lisp (and were often originally implemented in it).

In recent months implementations of Lisp, Logo and Prolog have become available for machines like the Sinclair Spectrum, Commodore 64 and BBC B (and CP/M versions have been around for a year or two); now is as good a time as any to find out what all the fuss is about.

The available books and manuals on Lisp often underestimate how foreign most of its concepts will be to the Basic trained programmer, and are consequently less useful than they might be. We hope to fill the gap for you in the crucial early learning days.

Concepts

It's tempting with any language tutorial to dive straight in with program examples (even if it's only 'Hello World!'). In this first part though, let's look at the concept of list processing itself, because, until that is understood, the very purpose of Lisp programming will remain a mystery.

In everyday life lists are a common enough device for keeping track of a number of objects or actions; everyone makes shopping lists or lists of things to do. The following list:

Eggs
Milk
Coffee
Cornflakes

requires little explanation; it's a list of food items to buy at the shop (and in this case the order is not important). A list such as this is a natural data structure for representing objects in the real world. The types of operation that we perform on lists like this are: adding a new item (we've just run out of Butter); crossing off an item to show that we've bought it, and searching the list to see if an item is there.

We don't expect there to be any serious limitation to the way these operations are performed (apart from running off the end of the paper). Unfortunately computers, and most conventional computer languages, can't handle such lists in any graceful way.

The computer itself has no notion of any objects apart from binary numbers (whether 8, 16 or however many bits). 'High level' languages like Basic and Pascal can allow us to pretend that the computer understands other objects.

By using the ASCII code we can 'teach' it the letters of the alphabet: in the right circumstances 65 means 'A'. These letters can be put together into strings like "CORNFLAKE". Strings or numbers can be built up into larger units by using the array data structure, so that:

SHOPLIST$ (4) = "Cornflakes"

Are these adequate tools for dealing with lists? Not really. If we represent the shopping list as a string:

"Eggs Milk Coffee Cornflakes"

the computer regards it as an unstructured sequence of characters, since <space> is just another character (ASCII code 32).

If we wanted to search such a string for a given word, we'd have to write a program segment to look for spaces just to extract separate words. Basic's string handling functions, though powerful in their own sphere, also produce notoriously unreadable code such as:

```
IF MID$(SHOPLIST$, SPACES
  (COUNT)+1, SPACES (COUNT+1)))
  = ITEM$
```

merely to extract one item.

An array is a bit better, in that the words are at least stored as separate objects, but it becomes clumsy when you want to add and remove items at will, to an unspecified maximum length, while maintaining an order. Both are unsatisfactory if we want to mix numeric and string data (the numbers will have to be represented as strings and recovered using VAL()).

Basic also has the READ . . . DATA construct which would allow us to write:

```
100  READ EGG$,EGG,MILK$,MILK,
     COFFEE$,COFFEE
200  DATA "Eggs", 12, "Milk", 4,
     "Coffee", 1
```

but this allows us no flexibility at all to modify the list unless we can stomach writing self-modifying code using PEEKs and POKEs!

In any case, we have only just scratched the surface of what lists are all about in our shopping list example. What about a more highly structured list such as:

Monday lunch (trout tripe truffles)


LISP
COFFEE

Monday dinner (oysters pheasant spinach)
Tuesday lunch (spam ham flan (puff-pastry apricots almonds) roquefort)

In this list some of the items are themselves lists, and one of these sublists (flan) has an item which is a sub-sublist. The normal way to deal with this in Basic is to use a three-dimensional string array (dimensions DAY, DISH, INGREDIENTS) and most of its slots would be empty, resulting in a huge waste of memory.

If menus and shopping lists were all that were at stake, then perhaps we wouldn't fuss so much. But it will probably have occurred to you by now that English sentences (including this one) are represented as lists with just this kind of structure, which explains the interest of AI researchers in list processing. Once you begin to manipulate text in sophisticated ways (say parsing sentences into different types of clause), then a language which can manipulate lists directly is a must; the amount of low level string and character twiddling that is required in Basic or Pascal would prevent one ever getting to the heart of a problem.

The menu example above also shows how nested lists can be used to structure concepts; a flan is a dish at one level but a list of ingredients at the next level of detail. By this device, complex tree structures of any depth can be neatly represented by a one-dimensional sequence of symbols, which is quite essential if they are to be fed into most contemporary computers.

The need is for a notation by which computers can be made to recognise lists as a kind of data of a higher level than strings or numbers; strings and numbers (and lists!) will be the building blocks that lists are made of.

In addition, we need to define a few operations that can be performed on lists (just as MID$, RIGHT$ and LEFT$ can be performed on strings). You could produce a list (*sic*) as long as your arm of potential such operations (for example, replace the fifth item in a list with the next-to-the-last item from a second list) but, in a computer language, economy will dictate that a minimum number of primitive operations be chosen, plus a means of bolting these together to do more complex things.

As for the notation, we have most of it already. In the menu example above, I used brackets '('and')' to show sublists in a list and in both examples used spaces to separate items in a list. If we say that anything in brackets is a list, and that items are separated by spaces, then we would rewrite the two examples so:

(Eggs Milk Coffee Cornflakes) and:
(Monday lunch (trout tripe truffles) Monday dinner (oysters pheasant spinach) Tuesday lunch (spam ham flan (puff-pastry apricots almonds) roquefort)

Items can be numbers or words or sublists or a mixture; square or curly brackets could just as well have been chosen, and another separator character such as a comma instead of <space> would work. The advantage of <space>, though, is that it's used in written human languages.

The menu example doesn't quite work as we want (for example, Monday and lunch are separate list items, flan and the list of its ingredients are separate items), but don't worry about that for now. Taking the simpler shopping list example, we can think of the brackets as representing the paper on which the list is written. The advantage is that we can then think of '()' as representing a blank piece of paper, which will turn out to be rather useful. This concrete analogy isn't quite so good with the nested lists in the second example, unless you can imagine them as being on separate scraps of paper stuck on with hinges!

What operations are needed? We'd like to be able to add to the list, remove items from the list, search the list for an item, add lists together, compare lists and more. Mindful of the need for economy, it turns out that everything can be accomplished by combining comparison tests with just three fundamental operations: 'construct a list', 'take the first item from a list' and 'take all but the first item of a list.'

These three simple operations can be depicted as shown in Fig 1.

Notice that 'first' produces an **item**, while 'construct' and 'rest' both produce **lists**. In the shopping list example, the list corresponds to a piece of paper with pencil marks on it, while the items would be 'in your head' waiting to be written down or just having been read. But don't get too hung up on this distinction, because in our menu example an item could itself be a list (for example, the flan ingredients).

It should be clear that if you wanted to extract the third item from a list, you'd need to repeat some combination of 'first' and 'rest' (depending upon how we'd defined the way of combining the operations). For example:
first —> rest —> rest —> (Eggs Milk Coffee Sugar)
would do the trick if the arrows mean 'do it to everything to my right' and they are evaluated from right to left.

To build a list from scratch, you need to 'construct' with an empty list: that is, write on a blank sheet of paper:

Eggs () ———> (Eggs)
'construct'

To remove an item from a list using only these operations, you'll have to 'construct' a new list, lacking the non-required item, extracting the items you *do* want with 'first' and 'rest'.

At this point our concrete analogy is strained to breaking point, for a human being would merely cross the item out. Lisp though, as we shall see, does it precisely this way. As an exercise you might try to write an expression to remove Milk from (Eggs Butter Milk Coffee) using the arrows convention mentioned above; assume construct —> a (b) gives (a b).

The type of comparison tests needed are ones to say whether two items are the same, whether a list is empty, and whether an item is or is not a list.

It would be very handy to have a few other primitive operations (for instance 'last', to take the last item off a list) and some list processing languages do supply them, but they're luxuries in the sense that they can be built up from combinations of these three basics.

There is no good reason why list processing operations can't be added to perfectly conventional languages such as Basic (there are various Pascal-like languages which have them).

Lisp, however, takes an altogether more radical approach. List notation is used throughout the language; Lisp programs are themselves lists and can be fed as data to other Lisp programs (another reason for its attraction to AI researchers who need to create self-modifying programs).

Lisp programming consists almost entirely of defining and evaluating functions, in a manner much closer to mathematics than to the way statements are combined in a Basic program. The familiar control structures IF...THEN, FOR...NEXT and GOTO have no direct equivalents in pure Lisp, in which control is achieved mainly by the list structure itself and by recursion: that is, functions which call themselves.

By the way, just so that you won't feel cheated, the simplest Lisp program to produce Hello World! is:
'(Hello World!)
if you don't mind the brackets. If you do mind the brackets, think hard before proceeding with Lisp!

Butter (Eggs Milk Coffee)	———> 'construct'	(Butter Eggs Milk Coffee)
(Eggs Milk Coffee)	———> 'first'	Eggs
(Eggs Milk Coffee)	———>	(Milk Coffee)

Fig 1

1984 PRODUCT RELEASES VIC-20 C-64

Write Now! PROFESSIONAL WORD PROCESSOR

for the C-64

★ WRITE NOW! comes as a cartridge based unit for fast machine language access and easy loading. ★ The built in 80 Column display allows you to see exactly what you will be printing, including headers, footers, justification, page numbers and page breaks. ★ Directly unlimited use of the previously stored text on tape or disk called directly from within text during the printout so it doesn't use extra memory and allows the creation of unlimited length documents. ★ Can send all special codes to any printer, even in the middle of a line without losing the proper justification. ★ Page numbering in the standard numbers or upper/lower case Roman numerals. ★ Full string search and search/replace functions. ★ No complicated editing modes or commands. ★ Full scrolling up and down through the text. ★ Multiple line headers and footers. ★ Justification and text centering. ★ Block copy and delete commands. ★ Save text to tape or disk. ★ Easy full screen editing. ★ Works with ANY printer ★ 100 page manual. ★ Includes keyboard overlays for computer

Write Now W. P. CARTRIDGE INC MANUAL (C-64) $79.95 R.R.P.

UNIVERSAL PARALLEL PRINTER INTERFACE

The ONLY interface that gives you full COMMODORE Graphics

Now you can use your Vic-20 & C-64 with any standard Parallel input printer. And you don't have to give up the use of your user port (MODEM), or use special printer commands, or load any special software driver programmes to do it. Just plug in our PRINTRON and print away! You can access all of the advanced functions of today's printers.

★ Outputs standard ASCII ★ Plugs in your computer's serial I/O port ★ Understands all Vic Print commands ★ CBM to ASCII Conversion ★ All functions software selectable, (Tab, Upper/Lower Case, Auto Line Feed, all from the keyboard) ★ No special programming is needed ★ Includes all cables for a standard Centronics Parallel hook-up ★ Full Graphics capability ★ Doesn't tie up your cartridge or games ports ★ Includes a 48 p. manual with programming examples.

PRINTRON INTERFACE

R.R.P. $119

COMMODORE COMPATIBLE PRINTERS

GEMINI-2064

At last a version of the acclaimed Gemini-IOX printer is available internally converted to run with your C-64 on Vic-20 computer. Why buy a 1526 when for little more you can purchase a printer with the following features.

★ Plugs in like a Vic Printer ★ 120 C.P.S ★ Sprocket, Friction, Sheet Feeds ★ 9 Dot Head For True Descenders ★ Block, Bit Graphics ★ Compressed, Expanded Italic Super, Subscript and More ★ Special Character Sets and much, much more.

2064
120 CPS
ONLY
$529

TRONIX-LQ1

At last! A true letter quality daisy wheel that runs with all standard software for your Commodore Computer features:

BROTHER HR-15

FEATURES

Bidirectional daisy wheel printing, shadow printing, proportional spacing, automatic underlining, super- and sub-scripting, two colour printing (red and black), carriage skip movement text reprinting, 3K (5K optional) buffer memory.

*** TRACTOR FEED OPTION $95 EXTRA**
*** AUTO SHEET INSERTER OPTION $225 EXTRA**
PRINTER AND INTERFACE $749 INC TAX

* Commodore is a registered trade mark of Commodore Computers

Both the above printers are available with internal or external interfaces.

AVAILABLE THROUGH YOUR LOCAL COMMODORE DEALER OR DIRECT FROM:

DEALER ENQUIRIES WELCOME ON (02) 29 3857

computer technics

5th FL. 383 GEORGE ST., SYDNEY 2000

MAIL ORDER SERVICE
G.P.O. BOX 4936
SYDNEY 2001

Memorex reliability, the inside story.

Vast experience, renowned reliability and technological supremacy of the world's leading computer media manufacturer contribute to the story inside every pack of Memorex flexible discs.

The extensive research, exhaustive testing and stringent quality control procedures which surround every Memorex product ensure that Memorex users have the most reliable flexible disc on the market. Every track of every disc is 100% error free tested.

Use Memorex flexible discs and you'll discover that our "inside story" is not just another "tall story".

Special soft polyester liner constantly cleans disc during use.

Unique burnishing process provides ultra smooth surface for better head to disc contact and reduced head wear.

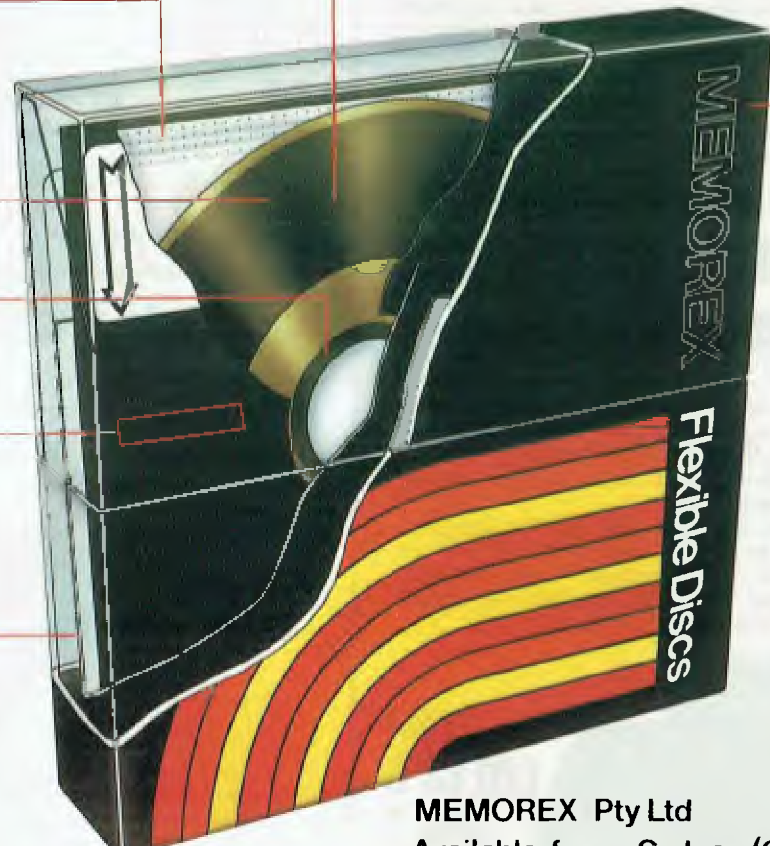
The Memorex name is your assurance of the ultimate in quality and reliability.

Coating uses the latest oxide technology, optimising signal output and recording accuracy.

Precision applied plastic hub ring protects disc edge from damage.

The carefully applied Memorex serial number allows identification of each disc.

Extra stiff glue bonded jacket ensures greater reliability, lower disc torque and easy loading.



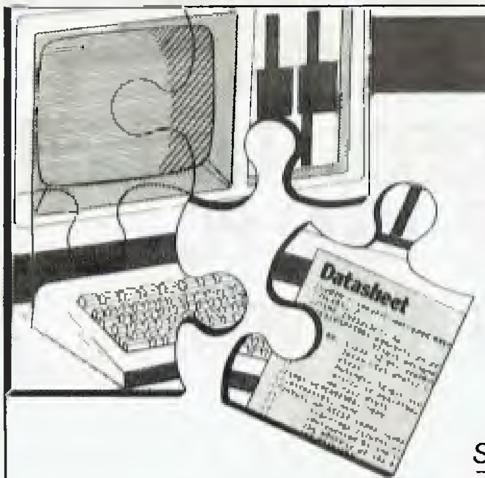
MEMOREX Pty Ltd

Available from: Sydney (02) 908-2211
Melbourne (03) 267-2955

Or a Distributor near you

MEMOREX

A Burroughs Company



SUBSET

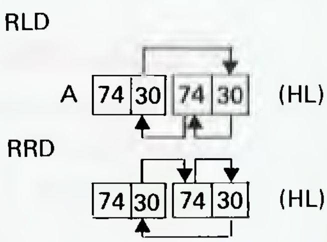
Alan Tootill and David Barrow present more useful assembler language subroutines. This is your chance to build a library of general-purpose routines, documented to the standard we have developed together in this series. You can contribute a Datasheet, improve or develop one already printed or translate the implementation of a good idea from one processor to another. APC will pay for those contributions that achieve Datasheet status. Contributions (for any of the popular processors) should be sent to SUB SET, 77 Glenhuntly Road, Elwood, Vic 3184.

Z80 data conversion

Michael Wilson sends a pair of routines, SQSH and EXPD, for squashing numbers held to 12-bit accuracy in 2-byte words for storing (in his case on disk). They are then expanded for use after retrieval. The required 25% saving in storage space can be cost effective, and other scientific/mathematical users might find this idea useful.

The numbers are held with the least significant byte first. In this example (hexadecimal), the pair of numbers 9C0A AB0F would be compressed to 9CAABF. Since the pointers are advanced in both routines, you can zip through a long block of numbers by putting the routine in a loop which calls it once every three or four bytes, whichever the case may be.

The code makes use of the nibble handling instructions RLD and RRD. Here is a reminder of what they do:



In Michael's comments on the instructions, a prime by a byte number (for example, byte 2') indicates that this is a processed value being placed in the output block.

When compressing data, the input block is corrupted and, in either direction, you may overwrite the input with the output block. The routines are thought to be very fast and reliable (unless you know differently).

DATASHEET

```
;- SQSH - Compress 4 bytes to 3 by truncation
;/ CLASS: 2
;/ TIME CRITICAL ?: No
;/ DESCRIPTION: Removes 4 most significant bits of every other byte
;/              of a 4-byte input block and repacks to a 3-byte
;/              output block. For long data blocks ( consisting of
;/              unsigned 16-bit quantities held to 12-bit accuracy
;/              or less ) set up input and output pointers to the first
;/              byte of the block and CALL once for every 4 bytes in
;/              the data block. The input block will then be replaced by
;/              the output block ( except for the last 25% ).
;/ ACTION:      Takes 4-byte block, processes, places in 3-byte block.
;/ SUBr DEPENDENCE: None
;/ INTERFACES: None
;/ INPUT:  HL points to the start of a 4-byte input block
;/         DE points to the start of a 3-byte output block
;/ OUTPUT: HL advanced by 4 ; DE advanced by 3
;/ REGs USED: AF, DE, HL
;/ STACK USE: None
;/ LENGTH: 22
;/ PROCESSOR: Z80

SQSH:  LD   A,(HL)   ;byte 1                7E
       LD   (DE),A   ;byte 1'               12
       INC  DE       ;bump output ( 1 )     13
       INC  HL       ;bump input  ( 1 )     23
       LD   A,(HL)   ;byte 2                7E
       INC  HL       ;bump input ( 2 )      23
       RLCA          ;                      07
       RLCA          ;                      07
       RLCA          ;swap                  07
       RLCA          ;nibbles               07
       RLD           ;exchange with (HL)    ED 6F
       LD   (DE),A   ;byte 2'               12
       INC  DE       ;bump output (2)       13
       LD   A,(HL)   ;byte 3                7E
       INC  HL       ;bump input (3)        23
       RRD           ;exchange with (HL)    ED 67
       LD   (DE),A   ;byte 3'               12
       INC  DE       ;bump output (3)       13
       INC  HL       ;bump input (4)        23
       RET           ;return                C9
```

```
;- EXPD - Pad out 3 data bytes to 4 ( cf. SQSH )
;/ CLASS: 2
;/ TIME CRITICAL ?: No
;/ DESCRIPTION: Reverses action of SQSH by padding out 2 x 12-bit
;/              values ( 3 bytes ) to 2 x 16-bit values ( 4 bytes )
;/              by adding 4 zero most significant bits to both values.
;/ ACTION: Expansion proceeds from highest to lowest byte. In practice,
;/         start with input pointer at end of squashed data and output
;/         pointer at desired end of reconstituted data.
;/ SUBr DEPENDENCE: None
;/ INTERFACES: None
;/ INPUT:  DE points to END of 3-byte input block
;/         HL points to END of 4-byte output block
;/ OUTPUT: DE, HL decremented by 3 and 4 respectively
;/ REGs USED: AF, DE, HL
;/ STACK USE: None
;/ LENGTH: 40
;/ PROCESSOR: Z80

EXPD:  LD   A,(DE)   ;byte 1                1A
       DEC  DE       ;drop input ( 1 )  .   1B
       LD   (HL),0   ;set a zero            36 00
       RLD           ;byte 1'               ED 6F
       DEC  HL       ;drop output ( 1 )     2B
       RRCA          ;                      0F
       RRCA          ;                      0F
       RRCA          ;swap                  0F
```

Announcing the arrival of the new EME PC 1600 IBM fully compatible. At $3500 (Tax incl.). With every purchase of the EME 1600 PC at this incredible price you also receive a free swivel base monitor

Where can you get a better deal than this??

Features

8088 processor
256k RAM fitted as standard
1 parallel & 2 serial ports
5 IBM PC compatible expansion slots
83 key IBM layout keyboard
13Mbyte hard disk option (XT model), disk controller, video controller, clock & calendar.

EME is a newly developed but fast growing company, offering to Australia's computer buying public the latest in microcomputers, software and associated peripherals.
The EME model III dual processor CP/M and/or Apple compatible has been praised as the first of our range.
Now we introduce to you THE IBM compatible. This outstanding computer boasts the above features. Impressive features indeed, but we don't stop there. With every purchase of the IBM PC or XT (ask about our competitive hard disk price) we also include the following software:

- **PC & MS-Dos 1.10/2.00**
- **lotus 1-2-3**
- **CP/M-86**
- **Wordstar**

As an introductory offer for this month with every purchase of the EME PC or XT we will give you a superb swivel base high resolution green screen monitor free.

For $3500 you simply could not find a better IBM compatible package.
Come in and see our range at 92-94 Bank St. Sth Melbourne

EASTERN MICRO

92-94 BANK· STREET, (Cnr
SOUTH MELBOURNE, VICTORIA 3205 Teleph

EME MODEL III

Special this month

$799 Tax incl.

Features

- ★ 6502 and Z80 Dual Processors
- ★ 64K RAM
- ★ Upper and lower case
- ★ Extendable keyboard
- ★ Built-in cooling fan
- ★ Single or twin Teac slimline disk drives can be installed
- ★ The 10 function keys each have 48 characters of built-in memory
- ★ 188 function keys

MONITOR

$199 (Tax incl.)

TEAC SLIM LINE DISK DRIVES

$299 (incl. Tax) Ideally suited for Model III.

ELECTRONICS

bankcard
welcome here

of Bank Street and Kingsway)

one: (03) 699 3088 Telex: AA 38432 ATT EME

EASTERN MICRO ELECTRONICS
92-94 Bank St., Sth Melbourne, 3205

Please find enclosed cheque for the amount of $ ....

For the following............

....

....

....

Please send me more information

Name ....

Address ....

Suburb ....

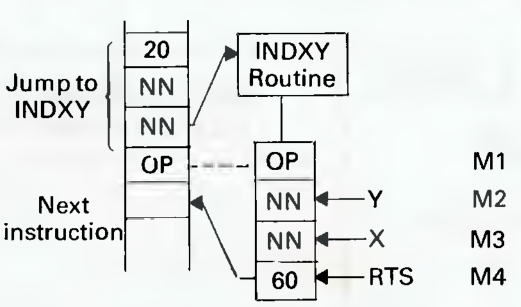
State ....

```
RRCA                ;nibbles                  0F
LD    (HL),0        ;set a zero               36 00
RLD                 ;rotate in                ED 6F
LD    A,(DE)        ;byte 2                   1A
RLCA                ;                         07
RLCA                ;                         07
RLCA                ;swap                     07
RLCA                ;nibbles                  07
AND   0F0H          ;mask low nibble          E6 F0
OR    (HL)          ;get low nibble           B6
LD    (HL),A        ;byte 2'                  77
DEC   HL            ;drop output ( 2 )        2B
LD    A,(DE)        ;byte 3                   1A
DEC   DE            ;drop input ( 2 )         1B
RRCA                ;                         0F
RRCA                ;                         0F
RRCA                ;swap                     0F
RRCA                ;nibbles                  0F
AND   0FH           ;mask output high nibble E6 0F
LD    (HL),A        ;byte 3'                  77
DEC   HL            ;drop output ( 3 )        2B
LD    A,(DE)        ;byte 4                   1A
LD    (HL),A        ;byte 4'                  77
DEC   DE            ;drop input ( 3 )         1B
DEC   HL            ;drop output ( 4 )        2B
RET                 ;return                   C9
```

6502 register indirect

David Heal's XYMOD (Sub Set April 83) providing operations on the address held in the XY registers, set some of you thinking. David's routine is short and simple but depends on modifying the calling code, which we sub-setters are not much taken with.

Cormac Duffin gives us INDXY, which does a similar job without modifying the calling code, by having the indirect instruction executed in page zero memory thus:



Our standards provide for 16 contiguous bytes of page zero memory, designated M0 to MF in the mnemonics and ZZ in the machine code, to be reserved for 6502 routines. INDXY uses eight of these as follows:

M1	OP	Transferred op code
M2	NN	Y register — address low byte
M3	NN	X register — address high byte
M4	RTS	Return from subroutine instruction
M5	ZZ	Store for accumulator
M6	ZZ	Store for processor status
M7	lo	Address of op
M8	hi	code to be transferred

Cormac points out that INDXY, although longer than XYMOD, is called in only four bytes against XYMOD's six.

DATASHEET

```
;- INDXY - To use the address in XY as operand to given op code
;/ CLASS: 1
;/ TIME CRITICAL ?:  No
;/ DESCRIPTION: Executes the op code, embedded in the code following
;/            the jump to subroutine instruction, using the address
;/            in the XY registers as the operand.
;/ ACTION: The op code, address from the XY registers and an RTS
;/         instruction are placed in page zero and executed from there.
;/ SUBr DEPENDENCE: None
;/ INTERFACES:  None
;/ INPUT:  XY holds the 16-bit operand address ( high, low )
;/         The bytes following the jump to subroutine instruction
;/         holds the op code.
;/ OUTPUT: The op code is executed using the address in XY as operand.
;/ REGs USED:  X, Y and M1 to M8
;/ STACK USE:  1 byte ( + 1 byte + any stack used by the subroutine(s),
;/             if the op code is a JSR).
;/ LENGTH:  52
;/ TIME STATES: 94
;/ PROCESSOR:  6502
   INDXY: STA  M5        ;save accumulator.        85 ZZ
          PHP            ;save the                 08
          PLA            ;processor                68
          STA  M6        ;status.                  85 ZZ
          SEC            ;get the                  38
          PLA            ;return                   68
          ADC  #0        ;address off              69 00
          STA  M7        ;the stack                85 ZZ
          PLA            ;and increment            68
          ADC  #0        ;to point to              69 00
          STA  M8        ;op code byte.            85 ZZ
          PHA            ;restore                  48
          LDA  M7         modified                 A5 ZZ
          PHA            ;return address.          48
          STX  M3        ;store X and Y registers  86 ZZ
          STY  M2        ;as operand for transferd 84 ZZ
          LDA  #$60      ;op code anf follow       A9 60
          STA  M4        ;by RTS instructioo.      85 ZZ
          LDY  #0        ;transfer the             A0 00
          LDA  (M7),Y    ;op code                  B1 ZZ
          STA  M1        ;to M1.                   85 ZZ
          AND  #$DF      ;is the op code an        29 DF
          CMP  #$4C      ;absolute or indirect     C9 4C
          BNE  NOJMP     ;jump instruction ?       D0 02
          PLA            ;if so, remove return     68
          PLA            ;address from stack.      68
  NOJMP:  LDA  M6        ;restore                  A5 ZZ
          PHA            ;processor status.        48
          LDA  M5        ;restore accumulator.     A5 ZZ
          PLP            ;                         28
          JMP  $00M1     ;execute instruction.     4C ZZ 00
```

ROD IRVING ELECTRONICS

425 High St., Northcote Vic. 48-50 A'Beckett St., Melb., Vic.
Phone (03) 489 8866, (03) 489 8131
Mail Order Hotline (03) 481 1436
Mail orders to P.O. Box 235 Northcote 3070 Vic.
Minimum P & P S3.00. Errors & omissions excepted.

Please address tax exempt, school, wholesale and dealer enquiries to:

RITRONICS WHOLESALE

1st Floor 425 High St., Northcote 3070. (03) 489 7099
(03) 481 1923 Telex AA 38897

NEW PRODUCT

Excalibur is alive and well kits now available

This stunning design with excellent colour, BASIC in ROM and a host of features is now available as a kit. $595

Z80A CPU 3.5 MHz 64K RAM, 16K ROM RF output, Direct video output and RGB, QWERTY Keyboard, Cassette Interface built in. Serial RS232C Parallel; 8 bit centronics. Color 16 foreground and 8 background colours. 24 lines by 80 characters. 7 x 12 dot matrix (monitor required).

DISK SYSTEM for Microbee 16K and 32K with CP/M like operating system and utilities. Runs CP/M software. 400K drive. Operates form Microworld Basic as well.
Only Mytek could have produced this excellent software.

SINGLE DRIVE SYSTEM	$780.00
DUAL DRIVE SYSTEM	$1100.00
ADD ON DRIVE	$329.00

NEW PRODUCT

MICROBEE ENHANCER 1

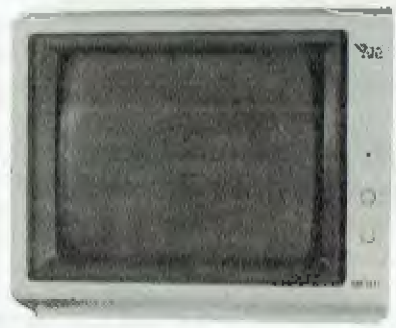
The Enhancer 1 plugs into the parallel port connector on the rear left of the Microbee keyboard. Its many powerful features include:

1) two ATARI type joystick inputs
2) two COMMODORE type joystick inputs
3) two TRS COLOUR COMPUTER type joystick inputs
4) a 4 voice music/sound effects sythesizer
5) a real time clock
6) unlimited vocabulary speech synthesizer (option)
7) a built-in speaker with volume control
8) a comprehensive user manual including listings of all necessary routines for use
9) an impressive demonstration program
10) compatible to all Microbees
11) an 8 channel, 8 bit analog-digital converter
12) an 8 bit analog-digital converter
13) 5 digital input lines
14) 4 digital output lines
15) a 40 pin experimenter socket with all 8 analog inputs, 5 digital inputs, 4 digital outputs, a 3.58 MHz buffered clock, sound output (so that you can play the sound effects through your Hi-Fi), 3 high resolution voltage comparators, DAC output etc.

NEW PRODUCT

Microbee is a Registered Trademark of Applied Technology

RITRON 1
Our most popular Model in a steel cabinet for less interference.
AMBER $199.00 GREEN $189.00



RITRON II
Swivel base monitor in stylish case.
AMBER $199.00 GREEN $179.00

TRY OUR GREAT RANGE OF MONITORS

RITRON III
Great resolution for those who want a little more quality.
AMBER $219.00

6502 ADD ON DRIVES $265.00 (Drive Only) $285.00 (Drive & Controller)

Japanese Mechanism 3 Months Warranty

DP80/CP80 RIBBON RIBBONS	1	10+
	$11.00	$9.50

Special price this month $6.20

Memory
Huge range best prices
Direct Import saves you money

	1	10+
2114	$2.50	$2.00
4116	$2.95	$2.50
4164	$9.00	$8.00
6116	$8.00	$7.50
2716	$5.90	$4.90
2732	$6.90	$5.90
2764	$8.90	$8.50
27128	$39.00	$37.00

Verbatim Discs

525-01	**32.50/box 10**
550-01	**47.00/box 10**
34-4001	**50.00/box 10**

PRINTERS

SX80 — $259 + tax
Amazing offer once only

NP 2200

HAND HELD COMPUTER LBC-1100
WITH PRINTER CASSETTE UNIT CL-1000

New VT100 Emulation terminal (performs 90% emulation – 132 columns not implemented)
$1195.00

LBC-1100 HAND HELD COMPUTER

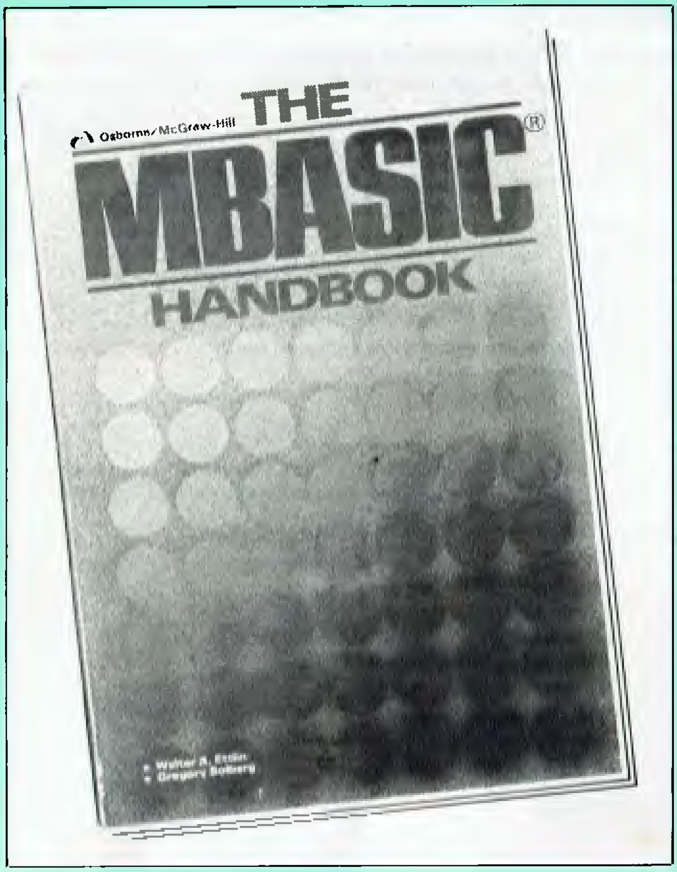
CPU	C-MOS 8-bit microprocessor
Memory	ROM: 20k bytes RAM: 8k bytes expandable to 16k bytes
Program Language	Standard BASIC
Display	Dot matrix LCD 40 characters x 2 lines with cursor
Cassette MT	600 bauds, Kansas-city standard (FSK modulation 1,200/2,400 Hz)

$725 NEW product

Serial impact Dot Matrix letter quality mode 165 cps Centronics.

BIBLIOFILE

This month's selection from the bookshelf, Steve Withers gives his rundown on the MBASIC Handbook and the Personal Computer Handbook.



The MBASIC Handbook

When I picked up this book I half expected it to be one of those that does little more than rehash the material provided in the standard manuals, but I received a pleasant surprise. The MBASIC Handbook revolves around a series of tutorials presenting realistic (if modest) examples of programming, stressing the need to understand the task and to break it into smaller, more manageable pieces.

Like most books of this type, it starts with the mechanics of using Basic — how to load the interpreter and how to enter, run, save, and edit programs. This material is nicely presented in a clear, informal style. There's plenty of detail and carefully worded explanations, but the authors never become verbose or cute.

Their programming style sets a good example with neatly indented loops, sensible variable names (eg TOTAL.WAGES), and a fair sprinkling of comments. There are some places where the remarks are few and far between, especially in the answers to the exercises set at the end of each chapter. Yes, unlike some authors, Ettlin and Solberg answer their own questions.

There is so much information in this book's 450-odd pages that it's difficult to isolate particular areas without giving the impression that other aspects are neglected. I'll describe the chapter on random access files anyway, as it is fairly typical. It starts with a description of such files, explaining their advantages and disadvantages, and then discusses the mechanics of using them (opening, reading, writing, and closing). Users of Microsoft Basic will know that integers are stored in two bytes, real variables in four, and double precision variables in eight bytes, and that they may be written to a random access file in this internal form. The MBASIC Handbook points out that a useful saving can be made by using the CHR$ function to convert integers in the range 0-255 into a single byte instead of the usual two bytes. Of course, the real benefit occurs when you have a lot of variables that may be stored in this way, or when there are lots of records.

This is typical of the hints in this book — taken individually they are useful, but not awe-inspiring. Collected together, they are much more impressive. The only problem is that it seems as if the authors intended the reader to work steadily through the book instead of dipping into it at random. However, if you are planning a major programming project using Microsoft Basic such a study could save time in the long run.

A payroll program acts as a continuing theme throughout the book. It first appears in an extremely simple form (three workers with different hourly rates working an eight hour day: what is the total wage bill?), but it gradually becomes more complicated as various programming concepts are introduced. By the end of the book it becomes a suite of programs linked by a menu system and incorporating data entry forms, extensive file handling routines, and cheque printing facilities. In case you find this application boring, many other examples are used to illustrate the material discussed in this book. There are also several useful programs and routines — the ever-popular bubble sort and quicksort, searching arrays, ASCII/hex memory jump, a program compactor, a mailing list system, and a set of routines for file access by key.

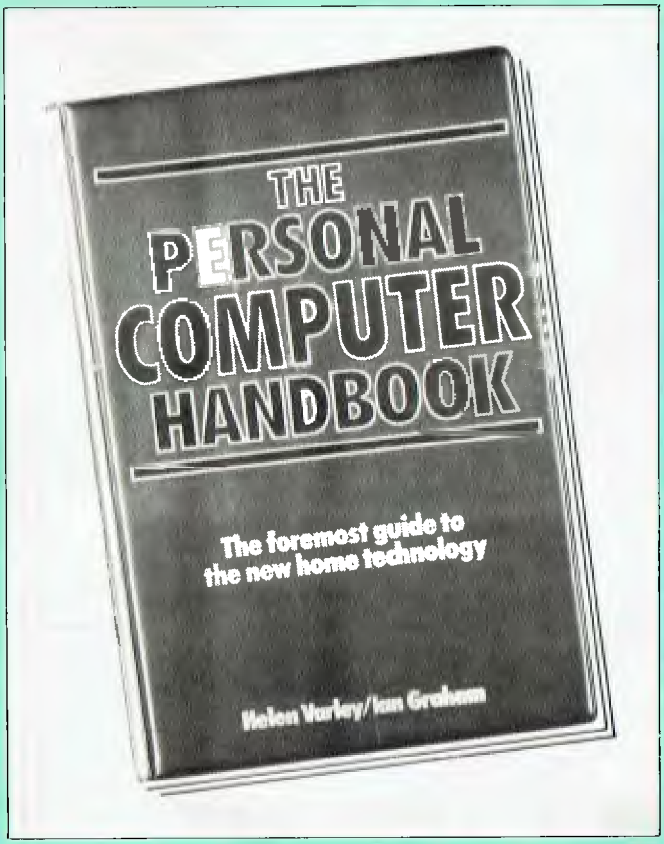
I've been using Microsoft Basic on and off for five or six years, and I would say that The MBASIC Handbook is worth at least a couple of years' experience — it certainly taught me some new tricks. In my opinion it is *the* book to complement Microsoft's reference manual.

The MBASIC Handbook

Authors: Walter A Ettlin and Gregory Solberg
Publisher: Osborne/McGraw-Hill
Price: $30.55

The Personal Computer Handbook

If The Personal Computer Handbook were physically larger it would be tempting to describe it as a coffee-table book, although it bears a stronger resemblance to one of the many introductory volumes on photography — lots of pictures, plenty of colour, and frequent sidebars containing glossaries



and checklists. It makes a nice change to see the authors of a book on computing adopt a well-tested format.

Not only is the material well-presented, but it consists of clear descriptions and sound advice. The chapter entitled 'Living With Computers' includes information about designing a work area that will be suitable for you as well as your computer (a topic that most books on home computing ignore). It even describes some of the problems faced by interior designers confronted by a micro in the living room, and suggests some possible answers.

There is the occasional piece of theory, but the bulk of the book is taken up by ready-to-use information illustrated by real products. When interfacing is being discussed the connectors on the Commodore 64 are shown, while a diagram of the IBM PC keyboard accompanies that section. The range of equipment illustrated in this book is so wide that few manufacturers will have cause for complaint.

'The Electronic Home' gets a chapter to itself, although the authors point out that the possibilities are more potential than real, except for the most enthusiastic and skilled people: "it is neither easy nor cheap to centralise control of the many electrical machines and systems in your home . . . [but] by the end of this decade, home control systems and interfaces for microcomputers may be as cheap and readily available as micros are today". For the time being, entertainment and education are suggested as the main applications in the home, but a much wider range of office applications is suggested.

As you might expect, the Handbook includes a buyer's guide. While the general information is useful, some of the machines listed have already disappeared from the market (the book was first published in 1983), and others were never sold here. Newcomers to personal computing will benefit from the section that describes how to read a microcomputer's specification.

The front cover describes the Handbook as 'the foremost guide to the new home technology' and the content comes very close to living up to that billing — it's just a shame there isn't a new Australian edition to provide those useful names and addresses and details of recent developments.

The Personal Computer Handbook

Authors: Helen Varley and Ian Graham
Publisher: Pan
Price: $16.95

MULTITECH COMPUTERS

THE SOURCE

Serial Impact dot matrix
100 C.P.S. for normal size print.
200 ms for line feed.
Normal — 80 columns.
Condensed — 142 columns.
Bidirectional.
"Centronics" parallel interface.
RS232C.

SUPER 5 SP-110 Printer Buffer
64K Memory Standard (Expandable)
3 Computer to 1 Printer
Copy/Pause/Reset Switches
4800 Char/SEC Data Transfer
Centronics Standard Interface

PERFECT

Brand Floppy Disk
Soft Sector 100% Certified
MD-1D S/Side Double Density
DM-2D D/Side Double Density

Complete system includes:

- CPU with 256K RAM 16-bit 8088
- 2 x 360K slimline drives
- 5 expansion slots
- Real time clock
- Colour graphics card
- 2 x RS-232 serial ports
- 1 x Centronics parallel port

SUPER 5 EN-P1091 Dot Matrix Printer
120 CPS Tractor/Friction Feed
80/136 Col/Line (Normal/Compressed)
Bit Images: 576 × 8 Standard, 1152 × 8 D. Density
Centronic Interface Standard

Dealer Enquiries Welcome

SUPER 5 CP-80 Dot Matrix Printer
80 CPS Tractor/Friction Feed
80/142 Col/Line (Normal/Compressed)
Bit Image: 640 × 8 Standard
Centronic Interface Standard

SUPER 5 T-40 Apple Compatible Drive
Quality TEAC 54A Mechanism
40 Tracks 163KB Capacity

TEAC FD-55 Disk Drives
FD-55B 40 Tracks DSDD (500KB)
FD-55F 80 Tracks DSDD (1.0MB)
FD-55G 77 Tracks DSDD (1.6MB)

MULTITECH COMPUTERS

202 High St. Northcote 3070
Phone (03) 489 0430. Telex AA38581 Polyon

COMMUNICATIONS

APC welcomes correspondence from its readers but we must warn that it tends to be one way! Please be as brief as possible and add 'not for publication' if your letter is to be kept private. Address letters to: 'Communications', APC, 77 Glenhuntly Road, Elwood, Victoria 3184.

Ethics or otherwise

Your columns have been buzzing recently with debate about the ethics or otherwise of copying software.

On one side is the software producer who can see his revenue disappearing into other people's pockets; on the other is the software user who wants to make a back-up copy or study the listing of the program.

However, the facts are too stark for blathering about the 'rights of the computer owner' to inspect other people's programs. When was the last time you heard someone who owns a hi-fi unit saying that it was practically illegal that he wasn't able to read the music on the record? For a best-selling program, the average is *one* legal copy for every ten in use. Contrary to popular belief this flood of copies is not due to some evil baron in a castle tower, fingers flashing over commercial copying machines, but is due to the social exchange of programs — you give me Pacman version no 10027453 and I'll give you Centipede version no 343176.

The economics of the copying boom are simple. The more copies being exchanged, the fewer people who buy the original tapes. This leads to a drop in revenue for software companies, who are then unable to spend as much money on development and whose programs deteriorate as a result. The thing spirals until everyone is copying worthless programs. Copying will kill software and wipe out a brand new industry.

In an effort to put off those with less will power, software houses have been devising complex locks which prevent everything but copying the tape bit by bit. The naive reserve their venom for these devices, calling them everything from 'inconvenient' to 'an infringement of civil liberties'; what they should be doing is bemoaning the fact that to prevent the further spread of illegal duplication, these routines should have to be devised.

The sooner home computer owners realise that software copying is not only illegal but counter-productive the better. Meanwhile, perhaps the government will reinstitute the punishment of boiling in oil for large scale offenders.

Justim Holmes

(Isn't is a bit 'naive' to expect micro users young and old not to swap and share programs out of an altruistic concern for software houses? — Ed)

The case for a European keyboard

I was very interested in the article 'The History of the Keyboard' by Conall Boyle in the June issue of *APC*. Although I do not claim to be an expert in this field, I have become aware in the course of my work as an employee of an international non-governmental organisation that it's certainly untrue that 'qwerty is the standard layout throughout the known universe'. Indeed, there are so many variants from it that, for example, IBM offers no less than *nineteen* different keyboard arrangements with its text processing system in Western Europe and North America alone. Of course, some of those differ only in the characters additional to the Latin alphabet, but there are three main families in which the qwerty layout is not followed.

As well as the German layout qwertz, there is the Italian qzerty and the French azerty, which are also used in certain other countries. But within these families, there are several variants depending on the arrangements for typing accented characters, or additional letters. This is not merely an academic point, as it is vital in the design of text processing systems to ensure that the keyboard can generate the messages that the program and the printer are expecting. In particular, accented letters can either be coded as a single byte, or as a sequence of two bytes, the accent (non-spacing) preceding the unaccented character with which it is to be displayed. Furthermore, in some cases it may be possible to use a 'backspace' operation to generate special characters, such as the (Danish) å, with a keyboard that is not basically equipped for them.

English and Latin are the only languages for which the 2 x 26 characters on the standard qwerty keyboard are sufficient, and so none of the numerous microcomputers on the market are directly suitable for use as the basis of text processing installations intended for other European languages. Of course, until a microcomputer with a keyboard appropriate for the local language becomes available at a reasonable price, some people will make do with a model having only the ASCII character set in a qwerty layout, especially for applications where full literary facilities are unnecessary. But there must be many others, like myself, who are still waiting for manufacturers to produce a variant of their ASCII machines that conforms to the standards of at least one continental European language. Ideally, of course, a 'polyglot' keyboard suitable for several different languages would be preferable; it might possibly be based on the standard for typewriters in Switzerland.

Alan F Reekie

This was typed on an Olivetti Praxis 35 electronic typewriter with azerty keyboard, which can cope with French, German and Italian as well as English, thanks, in part to 11 keys with alternative characters depending on the setting of the KBI/KBII switch.

Captured by goblins

While thoroughly enjoying 'The Hobbit', I have reached an insurmountable problem which I have tried to crack for weeks with no success.

I keep getting captured by goblins and thrown in jail. How do I get out? Thorin is with me, but he's no help because all he does is sing about gold.

T Bailey

In order that I don't give the game away to those who don't want to know the

apricot

The 4th generation personal computer

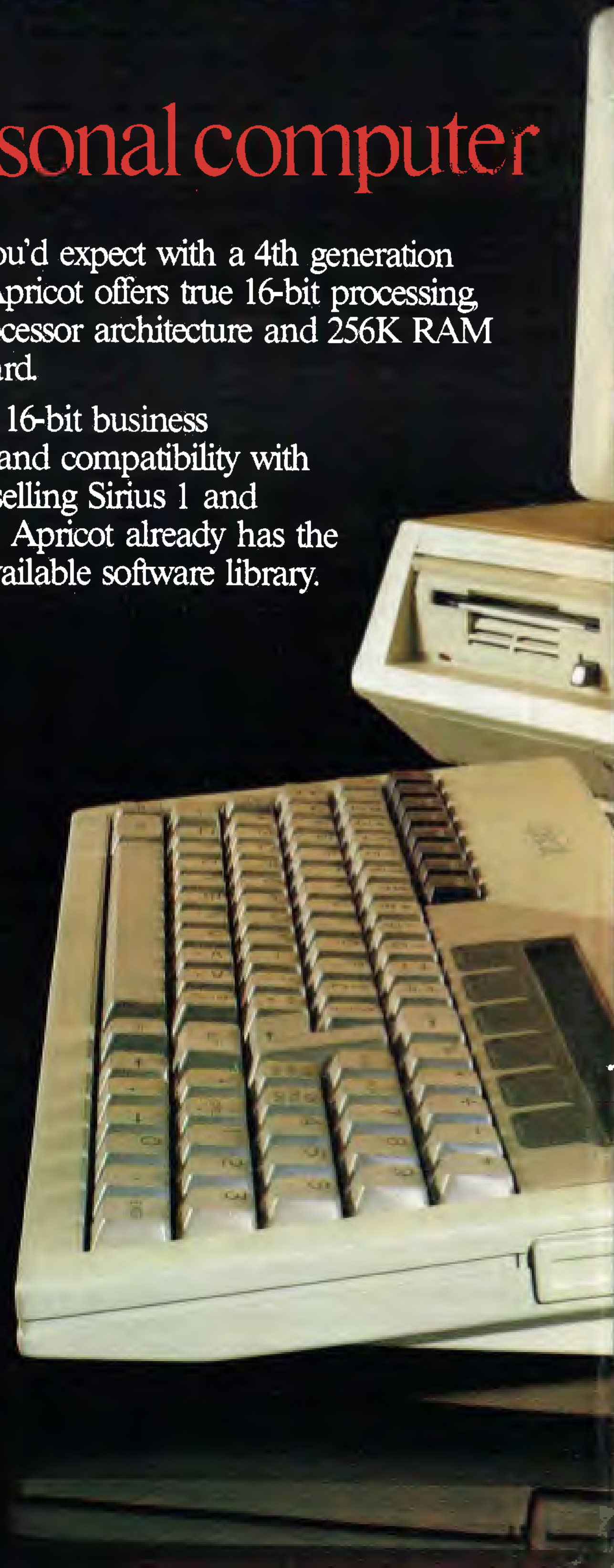
A computer that, for the first time, works with you rather than just for you. Hardware and Software co-ordinating perfectly, Apricot arrives complete and ready to work.

The MicroScreen™ on the keyboard, unique to Apricot, performs a variety of practical functions—as a calculator, calendar, time display, and as a window on the screen. Most significantly the MicroScreen™ prompts and assists the user by presenting simple options whilst running applications.

Incorporating the very latest advances in computer technology, Apricot uses 3½" pocket-size diskettes. Secure and compact, yet offering increased storage capacity.

As you'd expect with a 4th generation system, Apricot offers true 16-bit processing, multi-processor architecture and 256K RAM as standard.

With 16-bit business software and compatibility with the best selling Sirius 1 and IBM PC, Apricot already has the largest available software library.



answer. I'll print it backwards and hope there aren't any typing errors: wodniw eht hguorht uoy yrrac ot niroht ksa.

Tony Hetherington

Video sprites

Thanks for a continually interesting magazine.

In reply to Geoffrey Sivyer — yes it is being done, at least by one other person. I'm using a TI 99/4A to make titles for videoS and have enclosed a listing of a simple but effective sprite routine for this purpose. Recording direct to video gives excellent quality though the effects possible are limited by what you can program unless you have access to video editing equipment. (More sophisticated animation may also be achieved using a super eight camera. There was an interesting article on this in Byte October 1983.)

C Deeble

```
100 REM SPRITE TITLES
110 REM for TI 99/4A
120 REM Extended basic program by Chris Deeble
130 REM Maximum line length 28 characters (incl. spaces)
140 REM Currently for 3 lines; if more required
150 REM (Up to 12), insert line4$ etc. after 230
160 REM Set lines accordingly at 200, and
170 REM Add extra assignments after 510, with
180 REM Appropriate line numbers at 480
190 DEF KAR(N) = ASC(SEG$(A$,N,1))! Function to extract character from string
200 Lines = 3
210 Line1$ = "The tygers of wrath" •
220 Line2$ = "Are wiser than the"
230 Line3$ = "Horses of Instruction"
240 REM
300 REM Clear screen, set screen color black, &
310 REM Character color white on transparent
320 Call clear::call screen (2)::for I = 5 to 8::call color (1,16,1)::next 1
330 REM Set moving lettera to double size and initialise
340 REM Call magnify (2)::A$ = Line 1$::line = 1
350 REM
400 REM Create line of moving letters (each line moves faster than last)
410 For I = I to Len (A$)::
        Call sprite (#1,Kar(I),I/2+2.1,10,10,2*(line + 2),(I-1)*line)::
    Next I ! Note 2.1 in color setting avoids black first character
420 For delay = 1 to 1000/line::Next delay
440 REM stop moving letters & print single-sized station by characters
450 For I = 1 to Len (A$)::
        Call motion (#I,0,0)::
        Call Hchar(24-lines*2+line,1+2, KAR(I))::
        Call delsprite (#I)::
    Next I
460 Line = line + 2
470 REM Select next line (or 700 to exit)
480 On line/2 - .5 goto 500,510,700
490 REM
500 A$ = line 2$::goto 400
510 A$ = line 3$::goto 400
520 REM
700 REM Pause then finish
710 For delay = 1 to 2000::next delay
32767 END
```

Silly processor debates

I would like to propose a method for measuring the 'size' of a processor, which might stop the silly debates about how 'big' certain processors are.

Simply add together the sizes of the data bus, the address bus and the accumulator (or the size of the 'general purpose' registers) and divide the result by four.

Here are some results:

6800, 8080, 6502, Z80 6809	8
8088	11
68008	12
8086	13
68000, Z8000	14

Of course, this says nothing about the 'power' of a processor, which in any case depends on the application.

However, I would say that certain advertisements are guilty of exaggeration!

Stephen Burt

To be or not . . .

Having just read 'Communications' in your July issue, I find it necessary to make an unconditional jump to the defence of D Nowlan (Communications, May '84), and to refute the statement by W White (Communications, July '84) that the year 2000 will not be a leap year.

The Gregorian calendar that we use today was introduced in 1582 (later in some countries) by Papal Decree, as it was found that by using the then current Julian Calendar, religious festivals were getting out of step with the seasons on which they were based.

The reason for this is that the average Tropical year (time the Earth takes to return to an exact point on its orbit) is 365.2422 days, whereas the Julian system gave an average year of 365.25 days, as leap years were added every four years. By 1582, this difference resulted in the calendar being out of step with the seasons (especially the Vernal Equinox on which Easter is based) by eleven days.

The Gregorian reform dealt with this problem by adding a leap day to years ending with "00" only if the year were exactly divisible by 400, ie, every fourth century. This gives a calendar year an average interval of 365.2425, a good approximation of the Tropical year.

The slight difference between the Tropical and Gregorian years will result in the year being about one day out in 4000 years — leading to the suggestion that the calendar should be further modified so that if the year is exactly divisible by 4000, the leap day should be left out. Since the Tropical year is itself changing, at the moment nearly at the same rate as the error, it would seem an unnecessary refinement. In any case, I'm sure neither Messrs Nowlan and White nor the editors of *APC* will be too concerned about it when the time comes.

W White is not the only one in error in this exchange! D Nowlan states that Easter is set according to the stars. In fact, Easter is usually the first Sunday after the fourteenth day after the first new moon after the March Equinox. In addition, I was not aware that Christmas moved around according to the day of the week!

Perhaps in conclusion, an algorithm for W White:

```
10 LY$="LEAP YEAR"
20 IF YEAR/4 <> INT(YEAR/4) THEN LY$="NON LEAP YEAR":GOTO 40
30 IF YEAR/100=INT (YEAR/100) AND YEAR/400 <> INT (YEAR/400) THEN LY$="NON LEAP YEAR"
```

RL Symes

The controversy continues

Although I do not wish to extend a battle between the BBC versus Spectravideo, I feel obliged to correct Mr Myer on his reply to Mr Hyslop (July, *APC*). Although MR Hyslop appears to have some incorrect information, he is closer than Mr Myer on a number of points:

1. Mr Hyslop is perfectly correct in stating that the BBC ROM is expandable to 16 x 16k. The machine only comes with four ROM sockets, however the operating system software is configured to handle 16, and the modification required for it to do so is inexpensive, and can be performed by the user, with no soldering.
2. There are a number of RAM expansion packages available in England for the BBC, using two basic techniques. The first of these is to use the sideways ROM banks as 16k sideways RAM banks. These admittedly are not available for Basic program or variable storage directly, but can easily be used via the operating system, or user machine code to store and retrieve data, or can be filled with machine code utilities, loaded off tape or disk as required. The second techni-

que is to provide separate RAM for the screen display, which produces an extra 1 to 20k of RAM, depending on the screen mode being used. Using these two methods the memory capacity can be raised to well over 100k, and such things as silicon disks can be implemented.

3. Given the fact that the break key and the editing keys can be programmed with extra functions, the BBC has 16 user definable function keys.

4. It is true that some companies claim to provide 'sprites' on a graphics ROM, however these are not true sprites, generated by hardware. They are more 'player missile characters', run under interrupts, and appear to the user to operate like sprites using simple commands, similar to those probably supplied on the Spectravideo.

5. Through another specialised ROM, the BBC is capable of becoming a wordprocessor, with the special functions being available from the red function keys, 3 to a key. These are labelled by a cardboard strip placed under the perspex above the keys.

One final point I should make is that although many of these items are only available in England, most companies seem to make an effort for us down under if you write to them. I have so far received prompt service for any I have purchased from, and in one case actually paid a lower price than British users would have, including postage to Australia by air mail!

Mark Summerfield

Micro to video

Regarding Geoffrey Sivyers letter on PCs and VCRs (Communications, June 1984).

I had a similar problem two years ago, as I could only manage a stable display on my colour TV by passing the signal through my VCR.

This has some advantages: by recording on the VCR while setting up a program, it is possible to check back after altering a program line to view changes in program lines, and the differences caused by these changes.

Also, when starting, I developed a family tree program, which, while primitive in construction and design, displayed all known data. This display is now in printed form on a video cassette, and can be viewed and accessed in the usual manner of video recording (and the information stored on the tape greatly exceeds the memory capacity of my computer!)

As with Geoffrey Sivyer, I have known of no others using this system.

FH Boyes

Commodore flight simulator

Could you tell me the name of any company that produces a flight simulator for the Commodore 64? I haven't seen one in computer shops, or being advertised.

Tony Constantinou

A flight simulator has been written for the Commodore 64 by the same person who wrote the infamous Microsoft version that's used as a test for IBM compatibility. Consequently, this Commodore version, published by Sublogic, is quite similar. It's to be distributed by Imagineering and should be available in most computer shops.

The disk version will cost about $60 but you get a lot for your money. Supporting the disk will be two manuals which take you through basic flying principles to aerobatics.

Tony Hetherington

NETWORK NEWS

Steve Withers and Peter Tootill return with more news of foreign and domestic bulletin boards.

New boards

Another CBBS system has started in Adelaide. Operated by Daniel Schumacher for Computer Ventures, it's available 24 hours a day on (08) 255 9146.

Still in Adelaide, the AMUG BBS now has significantly longer operating hours: 10am–10pm weekends, 9am–9pm weekdays.

If you are a BBC Micro user with plenty of 20 cent pieces, you might like to call Microweb TBBS system on 0011 4461 456 4157 (UK), or BUG on 0011 468 463 528 (Sweden).

MICOM CBBS

Having recorded over 17,000 calls in 20 months, MICOM (The Microcomputer Club of Melbourne) reckons that its bulletin board is the most popular and longest continuously running system in Australia. A lot of development work has been going on, and Peter Jetson has offered to tell us about the new system as soon as it is up and running. System operators go largely unacknowledged, so next time you use your favourite board, why not thank the operator for his or her efforts?

Wanted — terminal software

We would like to publish a list of terminal software for various microcomputers. If you sell such a package (or if you are a satisfied user) please send details to Steve Withers, C/- *APC*, 77 Glenhuntly Road, Elwood 3184.

Australian systems

Micro Design Lab RCPM

Telephone: (02) 663 0150. Hours: 5pm–7am weekdays. 24 hours weekends.

MI Computer Club BBS

Telephone: (02) 662 1686. Program downloading. Hours: 24 hours daily.

Sydney Public Access RCPM

Telephone: (02) 808 3536. System Operators: Barrie Hull and David Simpson. Hours: 24 hours daily.

Software Tools RCPM

Telephone: (07) 378 9530. Hours: 24 hours daily.

MICOM CBBS

Telephone: (03) 762 5088. System Operator: Peter Jetson. Hours: 24 hours daily.

Gippsland RCPM

Telephone: (051) 34 1563. System Operator: Bob Sherlock. Hours: 24 hours daily.

Sorcerer Computer Users Association CBBS

Telephone: (03) 836 4616. System Operator: Bruce Alexander. Program downloading for SCUA members. Hours: 24 hours daily.

Perth RMPM

Telephone: (09) 367 6068. Hours: 6pm–9pm WST.

Adelaide Micro User Group BBS

Telephone: (08) 271 2043. Hours: 10am–10pm, weekends and public holidays. 9am–9pm weekdays.

Computer Ventures CBBS

Telephone: (08) 255 9146. System Operator: Daniel Schumacher. Hours: 24 hours daily.

New Zealand Systems

NZ Micro Club RBBS

Telephone: 0011 64 9 762 309. System Operator: Chris Cotton. Hours: 24 hous daily. Software up/downloading.

This information is correct and current to the best of our knowledge. Please send corrections and updates to: Steve Withers, C/- *Australian Personal Computer*, 77 Glenhuntly Road, Elwood, Vic 3184.

Overseas systems

North America

SYSTEM	*NUMBER*	*NOTES*
Forum 80	0011 1 816 861 7040	
CBBS	0011 1 312 545 8066	
FBBS	0011 1 312 677 8514	
ABBS	0011 1 703 255 2192	
ABBS Ottowa	0011 1 613 725 2243	
MABBS Fort Walton Beach	0011 1 904 862 1072	
Bull-80 Alabama	0011 1 205 492 0373	
SPACE Citadel	0011 1 206 839 4759	
Ckcms Citadel	0011 1 206 329 0436	
Eskimo North Minibin	0011 1 206 527 7638	
Conn-80	0011 1 212 441 3755	TRS-80 Color Computer

Europe

ELFA ABC-MONITOR, Sweden	0011 468 730 0706	Half duplex
ABC-Banken, Sweden	0011 463 511 0771	
ABC-MONITOR, Sweden	0011 468 801 523	Password required
CBBSD Gothenburg	0011 463 129 2160	75/1200 baud
CBBS Sweden*	0011 463 169 0754	
BUG, Sweden	0011 468 463 528	BBC Micro
XD-BBS Helsinki	0011 358 072 2272	
Commodore BBS, Finland	0011 358 116 223	
Tedas, Munich	0011 49 89 596 422	
Decates, Germany	0011 49 66 154 51433	

UK

CBBS South West	0011 44 626 890 014	
Forum-80 Hull	0011 44 482 859 169	
Liverpool Mailbox	0011 44 51 428 8924	
BASUG	0011 44 742 667 983	
Computer Answers	0011 44 1 631 3076	
CBBS Surrey	0011 44 4862 25174	
Blandford Board	0011 44 258 54494	
Microweb TBBS	0011 44 61 456 4157	BBC Micro

Africa

Connection 80, Cape Town	0011 27 21 457 750	
TRShop, Cape Town	0011 27 21 5367	
Clan Computers, Durban	0011 27 31 66356	
Peters Computers, Johannesburg	0011 27 11 834 5134	
Peters Computers, Johannesburg	0011 27 11 834 5135	
War Games, Johannesburg	0011 27 11 642 3722	

* After receiving the tone and connecting your modem, either type <C/R> or <COM C/R>. The system then asks for a password which is 'cbbs' in *lower-case* letters. If you only get a '>' from the system, it needs resetting, so type <I> C/R.

USER GROUPS INDEX

Below is a list of alterations and additions to the list of user groups published in the April issue. The next full listing will appear in the September issue of APC.

NSW

The Central Coast Apple Users Group meets on the first Tuesday of each month at the Niagara Park Public School from 7.30pm. All Apple and Apple compatible users are welcome. For further details telephone (043) 84 3419 or write to 662 The Entrance Road, Wamberal 2260.

A change of address for the Compucolor Users Group of New South Wales. The new address is The President, NSW CCII Users Group, Tony Lee, 52 Cowan Road, St Ives, NSW 2075, telephone (02) 449 8824, or, The Secretary, NSW CCII Users Group, Ian Woodburn, 4 Chisholm Street, Turramurra, NSW 2074, telephone (02) 44 7175.

VICTORIA

The Eastern Suburbs Eighty Users Group (ESEUG) caters for users of System-80 and TRS-80 Models. The club meets on the fourth Wednesday of each month at the Junior Science Lab, Kingswood College, 355 Station Street, Box Hill, Victoria. For further information contact Cameron McKern, ESEUG, 8 Chestnut Street, Surrey Hills, Victoria 3127. Telephone: (03) 288 1713 (AH).

ACT

The Atari Users Group of Canberra meets the first Monday of each month at 8pm. Meetings are held on the first floor, Building A, Canberra TAFE College, Reid. For further information write to ACTARI, PO Box E112, Old Canberra, ACT 2600 or contact C McEwan on (062) 88 7861 or (062) 54 2961.

MICSIG User Group caters for all microcomputer systems. Meetings are held on the second Tuesday of each month at the Oliphant Building, Australian National University at 7.45pm. For further information write to MICSIG, PO Box E237, Old Canberra, ACT 2600.

Meetings for the NEC Users Group of Canberra are held on the first Tuesday of each month at the Main Conference Room, CSIRO Headquarters, Limestone Avenue, Canberra at 7.30pm. For further information contact Mal Smith, PO Box 173, Belconnen, ACT 2616, or telephone (062) 541 614.

QUEENSLAND

The Australian Computer Education Association specialises in communication by modems and setting up of mini clubs in schools. For further information contact Guy Coppens, ACEA, PO Box 194, Corinda 4075. Telephone (07) 379 9365.

Changes to the Southport Commodore Computer Users Group (SCCUG) committee are as follows:— President, Bill Fitzpatrick (075) 32 0061; Treasurer, John Smith (075) 582 9209; Secretary, Merv McFarlane.

The Sinclair (Spectrum) Computer Club meets every third Sunday of each month at 2.00pm. Meetings are held at Everton Park State High School, Brisbane. The aim of this club is to create a greater Sinclair computer awareness. New members are always welcome. For further details contact the President, Norman Lloyd (07) 355 7809 or the Secretary Mrs V Lewis, 37 Samford Road, Leichhardt, Ipswich, Queensland 4305.

A new adventure club is currently being formed. Anyone interested in finding out more information, contact Ms Christine Ogden, 37 Samford Road, Leichhardt, Ipswich, Queensland 4305.

The Brisbane Spectravideo Users Group meets on the third Wednesday of each month. Meetings are held at the Logan City Education Centre, Block G, Woodridge Primary School, Wembley Road, Woodridge. For further information contact the club president Peter Daunton of 'Computer Connection', Shop 10, Logan City Centre, Pacific Highway, Underwood 4119. Telephone (07) 341 3466 or (07) 209 8686 (AH).

A change of address for the TRS-80 System 80 Computer Group. Meetings are held the first Sunday of each month at Lindum Hall, Lindum Road, Lindum at 2pm. For further information contact WJ Allen (Sec.), 16 Laver Street, Macgregor 4109. Telephone (07) 343 5771.

SOUTH AUSTRALIA

SA Peach User Group is a special interest group attached to the SA Microprocessor Group. Meetings are held separately. For further information write to SA Peach User Group, 27 Creslin Terrace, Camden Park, SA 5038, or contact Geoff Drury (08) 352 2555 (BH), or (08) 295 2778 (AH).

WESTERN AUSTRALIA

The Osborne Users Group of WA, Oswest, caters generally for Osborne users, however, other machines running CP/M are also represented. The club meets on the first and third Wednesday of each month. The Palmyra Recreation centre is the venue for the first Wednesday and the Subiaco Exhibition Hall for the third Wednesday. Meetings begin at 7.30pm. For further information write to: The Secretary, Oswest, PO Box 199, Mundaring 6554.

NORTHERN TERRITORY

The CBM/VIC Users Group of NT has now been changed to NT Computer Club. The club meets at Wulagi Primary School on the first and third Thursday of each month at 7.30pm. Users of all machines and other interested parties are welcome. For further details contact Ian Diss, NT Computer Club, 349 McMillan's Road, Anula, NT 5793. Telephone: (089) 27 9208.

PC GAMES

Australia's guide to micro entertainment

$2.40 August 1984

COMMODORE 64
DIY Basic enhancements
VIC 20 to 64 program conversion
plus listings: type in Robotank and roll into action

ADVENTURE COMPETITION
Win a printer!

HOW TO BUY THE RIGHT MICRO
major new series

PROGRAMMING
How to write your first games

UNCH ISSUE! LAUNCH ISSUE! LAUNCH ISSUE! LAUNCH IS

SCREEN TEST
30 new games reviewed for MicroBee • Commodore 64 • VIC 20
Spectrum • VZ 200 • Atari . . .

BENCHMARKS

A list of Benchmarks used when evaluating micros is given below. An explanation can be found in the February '84 issue.

```
100 REM Benchmark 1
110 PRINT "S"
120 FOR K=1 TO 1000
130 NEXT K
140 PRINT "E"
150 END
```

```
100 REM Benchmark 2
110 PRINT "S"
120 K=0
130 K=K+1
140 IF K<1000 THEN 130
150 PRINT "E"
160 END
```

```
100 REM Benchmark 3
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/K*K+K-K
150 IF K<1000 THEN 130
160 PRINT "E"
170 END
```

```
100 REM Benchmark 4
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 K<1000 THEN 130
160 PRINT "E"
170 END
```

```
100 REM Benchmark 5
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K/2*3+4-5
150 GOSUB 190
160 IF K<1000 THEN 130
170 PRINT "E"
180 END
190 RETURN
```

```
100 REM Benchmark 6
110 PRINT "S"
120 K=0
130 DIM M(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB220
170 FORL=1 TO 5
180 NEXTL
190 IF K<1000 THEN 140
200 PRINT "E"
210 END
220 RETURN
```

```
100 REM Benchmark 7
110 PRINT "S"
120 K=0
130 DIM M(5)
140 K=K+1
150 A=K/2*3+4-5
160 GOSUB 230
170 FOR L=1 TO 5
180 M(L)=A
190 NEXTL
200 If K<1000 THEN 140
210 PRINT "E"
220 END
230 RETURN
```

```
100 REM Benchmark 8
110 PRINT "S"
120 K=0
130 K=K+1
140 A=K^2
150 B=LOG(K)
160 C=SIN(K)
170 IF K<1000 THEN 130
180 PRINT "E"
190 END
```

DIARY DATA

Readers are strongly advised to check details with exhibition organisers before making travel arrangements to avoid wasted journeys due to cancellations, printer's errors, etc.

Melbourne	Ausgraph '84 Contact: Australasian Computer Graphics Association Tel: (03) 341 6944	September 18–21, 1984
Hong Kong	SEARCC '84 Contact: John Lyons Tel: (062) 72 2514	September 24–28, 1984
Dallas, USA	PC World Expo Contact: Conference Management Group Tel: US (617) 879 0700	October 3–5, 1984
Berlin, Germany	Mikro Shop '84 Contact: AMK Berlin Austellungs-Messe-Kongress-GmbH Tel: (030) 3038 1	October 9–12, 1984
Melbourne	EPOS '84 Contact: Retail Management Development Program Tel: (03) 536 2386	October 15–18, 1984
Sydney	ACC '84 Contact: Beverley Parrot Tel: (02) 241 1478	November 5–9, 1984
Brisbane	Computer Expo '84 Contact: Robert Woodland Exhibitions Tel: (07) 372 3380	November 8–11, 1984

PROGRAMS

APC is interested in programs written in any of the major programming languages for all home and small business micros. When submitting programs to APC please include the following:
(a) A cassette or disk of the program.
(b) A listing on plain white paper (typewritten if no printer available).
(c) Comprehensive but brief documentation.
(d) A suitable SAE if you would like your materials to be returned after use.
Please mark (a), (b) and (c) with your name, address, program title, machine (state minimum RAM where appropriate) and — if possible — a daytime number.
All programs are paid for at the rate of $20 per page of published listing. Send contributions to:
APC programs, 77 Glenhuntly Road, Elwood, Vic 3184.

This month's assortment of programs includes 'Connect-Four' for the Commodore 64. A nice way to endure a miserable afternoon. 'The Life Game', a simulation game for the VIC 20 resembles the reproductive behaviour of living organisms. For the VZ-200, an information storage and retrieval program, 'Database VZ-200'. Those with more of an artistic inclination, throw away your palette and start dusting your TRS-80 keyboard.'Grafx Editor', (32k RAM min) allows you to create pictures on your colour computer. For the Atari 800, 'Basic System Reset' enables Basic programmers to handle System resets in a straight-forward way. Eggs and telepathic monsters... strange combination . . . are the vital elements in 'VIC Hatchery' for the unexpanded VIC 20. A bizarre but fun game. 'RAM Editor' is a utility for the BBC which compliments last month's BBC 'Sected'. 'RAM Editor' allows you to examine and modify RAM in much the same way as 'Sected' does a disk.

Games
Scientific/mathematic
Business
Toolkit/utilities
Educational/Computer Aided Learning

VIC Hatchery
by Bryn Phillips

The scenario is as follows. As chief designer of a new micro which only works with a ROM hanging onto the back of the cartridge port, you have been transferred to other duties on a little-known planet inhabited by giant hens (don't ask, just don't ask . . .).

Scientists have built giant hatcheries in an attempt to hatch some eggs, and you have been assigned to guarding them. All seems well until you receive a message that a hatchery has been overrun by telepathic monsters (I told you not to ask). Your instructions are clear: destroy the remaining eggs before they, too, can hatch into something large, telepathic and unfriendly.

Only . . . you hear a cracking sound. One of the eggs is hatching! And that's where the game begins. You have to smash the eggs. As you do so, however, fragments of the shell recombine to form new eggs which in turn start to hatch. The monsters are lethal if you come into contact with them, and more than ten of them will destroy you with their combined telepathic powers. (Well, I suppose Pacman is pretty silly if you actually think about it.)

The game has nine levels of difficulty ranging from 1 (simple) to 9 (forget it). You move along the levels using '.' (full stop) for left and '/' for right. You can also use 'S' and 'X' to move up and down ladders. Alternatively, you can use a joystick. To smash an egg (where have I heard that before?), simply run into it — not subtle, but effective.

You have three lives. Uncracked eggs are worth one point each, cracked eggs two points and open eggs three points. At the end of the game, press the space bar to play again.

The listing is in two parts; the first should be saved directly before the second. When you run program one, the second program will be loaded automatically.

PROGRAM 1

```
 10 PRINT"█████████THE HATCHERY"
 20 PRINT"███USE THE JOYSTICK OR   KEYS . / S AND X TO   MOVE MAN AND SMASH
EGGS."
 30 PRINT"██YOU LOSE A LIFE IF    YOU RUN INTO A         MONSTER OR 10"
 40 PRINT"█MONSTERS HATCH"
 50 PRINT"████████PLEASE WAIT"
 100 POKE 52,28:POKE56,28:CLR
 110 FOR I=7168TO7679:POKEI,PEEK(I+25600):NEXT
 120 FORC=7448TO7543:READA:POKEC,A:NEXT
 130 POKE36869,255
 140 DATA255,255,255,255,255,255,255,255,255,129,129,129,255,129,129,1??
 150 DATA60,66,129,129,129,129,66,60,60,90,145,137,133,137,70,60
 160 DATA0,66,195,165,149,137,70,60,90,60,60,219,189,165,153,66
 170 DATA0,0,0,0,8,36,26,169,48,50,38,58,32,80,72,108
 180 DATA176,53,162,48,42,84,73,108,24,24,72,120,72,20,20,50
 190 DATA24,89,138,25,168,85,52,178,255,153,139,189,255,151,161,161
 200 PRINT"████TO LOAD THE NEXT PART PLEASE PRESS RETURN"
 210 PRINT"██████████████LOAD███"
READY.
```

PROGRAM 2

```
5 POKE36869,255
10 DIMJS(2,2):POKE37139,0:DD=37154:PA=37137:PB=37152
20 FORI=0TO2:FORJ=0TO2:READJS(J,I):NEXTJ,I
30 DATA-23,-22,-21,-1,0,1,21,22,23
200 POKE36879,25:PRINT"██████"
210 GOSUB300:GOSUB310:GOSUB300:GOSUB330:GOSUB340:GOSUB350:GOSUB330
220 GOSUB360:GOSUB370:GOSUB330:GOSUB340:GOSUB350:GOSUB330:GOSUB310
230 GOSUB300:GOSUB300:GOSUB380
240 GOTO400
300 PRINT"███$    $    $    $    $ $":RETURN
310 PRINT"███$█####█$█####█$█####█$█####█$█#█$":RETURN
330 PRINT"███$ $ $ $ $ $ $ $ $":RETURN
340 PRINT"███$█#█$█####█$█####█$█####█$█####█$":RETURN
350 PRINT"███$ $    $    $    $    $":RETURN
360 PRINT"███$█####█$█#█$█#█$█#█$█#█$█####█$█#█$":RETURN
370 PRINT"███$    $ $ $ $ $    $ $":RETURN
380 PRINT"███######################":RETURN
400 DIME%(9,5)
410 P1=8143:P2=38863:LV=3
411 PRINT"███DIFFICULTY 1 TO 9"
412 GETD$:D=VAL(D$):IFD<1ORD>9THEN411
413 D=10-D
414 PRINT"█
415 PRINT"███████**"
420 FORC=1TO9:FORA=1TO5
430 POKEP1+(C*2)-22*(3*A-2),37:POKEP2+(C*2)-22*(3*A-2),2
440 NEXTA:NEXTC
450 X=1:Y=1:L=36:M1=42:X1=1:Y1=1:C1=0
455 FORC=1TO9:FORA=1TO5:E%(C,A)=37:NEXT:NEXT
460 POKE8122,42:POKE38842,0:POKE650,128
470 GET A$:GOSUB3000
472 IFN>9THEN2000
475 GOSUB1000
477 IFA$=""THEN470
480 LK=P1+X1-22*Y1
490 IFA$="/"AND(PEEK(LK+23)=35ORPEEK(LK+23)=36)THENX=X+1:GOTO600
500 IFA$="."AND(PEEK(LK+21)=35ORPEEK(LK+21)=36)THENX=X-1:GOTO650
510 IFA$="X"ANDPEEK(LK+22)=36THENY=Y-1:M1=46:GOTO700
520 IFA$="S"ANDPEEK(LK-22)=36THENY=Y+1:M1=46:GOTO700
590 POKE36878,5:GOTO470
600 IFPEEK(LK+1)=32ORPEEK(LK+1)=36THENM1=42:GOTO700
610 IFPEEK(LK+1)=40THEN2000
620 GOSUB1500:E%(X/2,(Y+2)/3)=36:M1=43:L2=41:GOTO700
650 IFPEEK(LK-1)=32ORPEEK(LK-1)=36THENM1=44:GOTO700
660 IFPEEK(LK-1)=40THEN2000
670 GOSUB1500:E%(X/2,(Y+2)/3)=36:M1=45:L2=41
700 POKEP1+X1-22*Y1,L:POKEP2+X1-22*Y1,C1
710 L=PEEK(P1+X-22*Y):C1=PEEK(P2+X-22*Y)
720 IFL2=41THENL=41:L2=0
750 POKEP1+X-22*Y,M1:POKEP2+X-22*Y,0
760 X1=X:Y1=Y
770 GOTO470
1000 R=INT(RND(1)*D):IFR<>0THEN1110
1005 C=INT(RND(1)*9)+1:A=INT(RND(1)*5)+1
1010 IFX1=A*2ANDY1=C*3-2THEN1000
1040 IFE%(C,A)=41THENE%(C,A)=37:GOTO1100
1050 IFE%(C,A)=40THEN1100
1055 IFE%(C,A)=39THENGOSUB2600
1060 E%(C,A)=E%(C,A)+1
1100 POKEP1+(C*2)-22*(3*A-2),E%(C,A):POKEP2+(C*2)-22*(3*A-2),2
1105 IFE%(C,A)=40THENPOKEP2+(C*2)-22*(3*A-2),5
1110 RETURN
1500 IFE%(X/2,(Y+2)/3)=41THEN1520
1510 SC=SC+E%(X/2,(Y+2)/3)-36
1520 PRINT"████SC       █████"SC;TAB(10)"HG "H
```

```
1530 RETURN
2000 POKEP1+X1-22*Y1,L:POKEP2+X1-22*Y1,C1:LV=LV-1:GOSUB2500:N=0
2005 PRINT"SQQQ"TAB(4-LV)" ":IFLV=0THEN2100
2010 GOTO420
2100 PRINT"SQQQQGAME OVER
2110 GETA$:IFA$<>" "THEN2110
2120 IFH<SCTHENH=SC
2130 PRINT"S
2140 PRINT"SQQQQ
2150 SC=0:GOTO410
2500 POKE36878,15
2510 FORI=200TO170STEP-1:POKE36874,I:FORJ=1TO30:NEXT:NEXT
2520 POKE36874,0:POKE36878,0
2530 RETURN
2600 POKE36878,15
2610 FORI=128TO255STEP5:POKE36877,I:FORJ=1TO2:NEXT:NEXT
2620 POKE36878,0:POKE36877,0:N=N+1
2630 RETURN
3000 POKEDD,127:S3=-((PEEK(PB)AND128)=0):POKEDD,255
3010 P=PEEK(PA):S1=-((PAND8)=0):S2=((PAND16)=0):S0=((PAND4)=0)
3020 IFS0<>0THENA$="S"
3030 IFS1<>0THENA$="X"
3040 IFS2<>0THENA$="."
3050 IFS3<>0THENA$="/"
3060 RETURN
```

BBC RAM Editor

by T Allchin

'RAM Editor' is a utility for a BBC or Electron.

Last month, we published BBC 'Sected' — a program allowing users to examine and modify the contents of a disk. 'RAM Editor' performs a similar job for RAM.

The main use of the program is to recover programs which have become partially corrupted — perhaps through saving without verifying. Often, when you get a 'Bad program' message on loading, there is very little you can do about it, except attempt to load the program again.

RAM Editor allows you to search through RAM for the corrupted section and correct it manually, thus allowing the program to be run.

The program displays the contents of a specified memory location, together with the seven following locations. The display is divided into three columns: the address itself (in hex), the contents of the address in hex, and the contents of the address in ASCII form.

The commands available <c>hange a byte, <j>ump to a new location and <e>xit 'RAM Editor'.

To select the precise byte you want to modify, either <j>ump to it or use the cursor keys if the byte is currently displayed on the screen. Before pressing 'c' to <c>hange the byte, use the TAB key to toggle between ASCII and hexadecimal mode.

To change a byte in hexadecimal mode, press 'c' followed by a two-digit hex value. To change a byte in ASCII mode, press 'c' followed by the character to be written to the location.

<J>ump is used to move to a new memory location. Simply enter a decimal number when presented with the 'type new address' prompt.

<E>xit is used to delete 'RAM Editor' from memory, leaving the original (or modified) program intact. The escape key must *not* be used to exit the program as it may corrupt the program.

If you have an Epson printer, one final command not shown on the display is <D>ump. As you would expect, this dumps a screen image to the printer.

The program is 2.3k long, and uses local variables wherever possible to avoid conflict with the program to be modified. As listed below, it runs in mode 6 to allow compatibility with the Electron. I would suggest, however, that BBC users adapt the program to run in mode 7. This uses less memory and thus leaves room for larger programs to be dealt with.

It is obviously necessary to ensure that 'RAM Editor' and the program to be acted on are kept in different areas of memory. The easiest approach is to place 'RAM Editor' above the program you want to play with. To do this, load the program first. Enter 'P'. TOP' as a direct command and note the value. Then set PAGE to a higher value.

If there is insufficient memory on a disk-based system, save both 'RAM Editor' and your own program to tape, and use the disk workspace. To do this, load your own program from tape and then set PAGE to &E00. Then load 'RAM Editor' from tape in the following format: LOAD "EDITOR" E00 (assuming that you saved 'RAM Editor' to tape using the filename EDITOR).

Enter END as a direct command, and then RUN.

```
L.
   10REM MEMORY VIEW
   20REM BY T ALLCHIN
   30REM COPYRIGHT 1984
   40MODE6
   50M=PAGE:C=6:tab%=0:*FX4,1
   60PROCscreen
   70PROCarea
   80PROCmove
   90GOTO60
  100CLS:PRINT" BYE,BYE":*FX4,0
  110END
  120DEFPROCscreen
  130LOCAL X,A
  140IFtab%=0THEN tab$=" ASCII"ELSE tab$="   HEX"
  150PRINTTAB(0,0)"ADDRESS     HEXADECIMAL           ASCII";
  160PRINTTAB(0,1)
  170FORX=M TO M+124 STEP8
  180PRINTTAB(0);~X;TAB(7);
  190D$=""
  200FORY=0TO7
  210A=X?Y
  220IFA<16THENPRINT"0";~A;" ";ELSE PRINT;~A;" ";
  230IFA>31ANDA<128THENB$=CHR$(A) ELSEB$="."
  240D$=D$+B$
  250NEXT
  260PRINTD$;
  270NEXT
  280PRINTTAB(C,10);">";TAB((C+3),10);"<";
  290mem=(M+64+((C/3)-2))
  300PRINTTAB(0,19)"MEMORY ADDRESS  ASCII  MEMORY AREA";;TAB(1)"DEC    HEX":
  320PRINTTAB(0,23);STRING$(34,"=");tab$;
  330PRINTTAB(0,24);"E=END  C=CHANGE  J=JUMP  ";
  340ENDPROC
  350DEFPROCmove
  360LOCAL A,K
  370mem=(M+64+((C/3)-2))
  380PRINTTAB(0,21);mem;TAB(7,21)~mem;
  390A=?mem
  400IFA>31ANDA<128THENPRINTTAB(18,21);CHR$(A)ELSEPRINTTAB(18,21);"."
  410PRINTTAB(24,24);:k=GET
  420IFK=138THEN M=M+8:GOTO60
  430IFK=139THEN M=M-8:GOTO60
  440IFK=136ANDC>6THENPRINTTAB(C,10);" ";TAB((C+3),10);" ";:C=C-3:PRINTTAB(C,10)
;">";TAB((C+3),10);"<";:PROCmove
  450IFK=137ANDC<26THENPRINTTAB(C,10):" ";TAB((C+3),10):" ";:C=C+3:PRINTTAB(C,10
;">";TAB((C+3),10);"<";:PROCmove
  460IFK=67THEN PROCchange
  470IFK=74THEN PROCjump
  480IFK=69THEN100
  490IFK=68THEN PROCdump
  500IFK=9THENtab%=NOT(tab%):GOTO60
  510GOTO80
  520ENDPROC
  530DEFPROCjump
  535PRINTTAB(0,24)STRING$(12," ");
  540PRINTTAB(0,24);M;:INPUTTAB(11,24);"TYPE NEW ADDRESS",M
  550CLS
  560GOTO60
  570ENDPROC
  580DEFPROCarea
  590LOCAL X
  600X=I
  610RESTORE
  620REPEAT
  630READNUM
  640READAREA$
  650IFmem<NUM THENPRINTTAB(24,21)AREA$;:X=2
  660UNTILX=2
  670ENDPROC
  680DEFPROCchange
  690LOCAL NUM,X,CH
  700PRINTTAB(0,24)"   TYPE NEW CHARACTER  ";
  710IFtab%=0THEN CH=GET:PRINTCHR$(CH);:?mem=CH:GOTO60
  720X=0
  730REPEAT
  740CH=GET:PRINTCHR$(CH);:CH=CH-48
  750IFCH>22THENPRINTTAB(0,24)"ERROR PLEASE RE-ENTER ";:GOTO720
  760IFCH>10THENCH=CH-7
  770IFX=0THENNUM=CH*16:GOTO790
  780NUM=NUM+CH
  790X=X+1
  800UNTILX=2
  810?mem=NUM:GOTO60
  820ENDPROC
  830DATA256,"ZERO PAGE      ",512,"6502 STACK     ",786,"OPS WORKSPACE  "
  840DATA1024,"MISC WORKSPACE",2048,"ROM WORKSPACE ",2304,"MISC WORKSPACE"
  850DATA2816,"VAR. BUFFERS  ",3072,"FUN. KEY DEFS.",3328,"CHARACTER DEF."
  860DATA3584,"USER ROUTINES ",32768,"USER RAM AREA ",49152,"LANG. ROM AREA"
  870DATA64512,"OPS SYS ROM   ",65280,"MEMORY MAPPING",65535,"OPS SYS ROM   "
  880DEFPROCdump
  890COLOUR2
  900LOCAL X,Y
  910FORY=0TO24
  920FORX=0TO39
  930VDU2:PRINT FNDUMP(X,Y);
  940NEXT:PRINT:NEXT:VDU3
  950COLOUR3
  960GOTO60
  970ENDPROC
  980DEF FNDUMP(X,Y)
  990LOCALA%,LX,LY,C
 1000LX=POS
 1010LV=VPOS
 1020VDU31,X,Y
 1030A%=135
 1040C=USR(&FFF4)
 1050C=C AND&FFFF
 1060C=C DIV&100
 1070VDU31,LX,LY
 1080=CHR$(C)
```

```
ADDRESS     HEXADECIMAL

1900   0D 00 0A 11 F4 20 4D 45 ..... ME
1908   4D 4F 52 59 20 56 49 45 MORY VIE
1910   57 0D 00 14 12 F4 20 42 W..... B
1918   59 20 54 20 41 4C 4C 43 Y T ALLC
1920   48 49 4E 0D 00 1E 14 F4 HIN.....
1928   20 43 4F 50 59 52 49 47  COPYRIG
1930   48 54 20 31 39 38 34 0D HT 1984.
1938   00 28 06 EB 36 0D 00 32 .(..6..2
1940  >19<4D 3D 90 3A 43 3D 36 .M=.:C=6
1948   3A 74 61 62 25 3D 30 3A :tab%=0:
1950   2A 46 58 34 2C 31 0D 00 *FX4,1..
1958   3C 0B F2 73 63 72 65 65 <..scree
1960   6E 0D 00 46 09 F2 61 72 n..F..ar
1968   65 61 0D 00 50 09 F2 6D ea..P..m
1970   6F 76 65 0D 00 5A 09 E5 ove..Z..
1978   8D 54 7C 40 0D 00 64 19 .T|@..d.

MEMORY ADDRESS  ASCII  MEMORY AREA
 DEC    HEX
6464   1940       .      USER RAM AREA

                                  ASCII
E=END  C=CHANGE  J=JUMP
```

The Life Game

by Kit Durre and Ray Edwards

'The Life Game' is a mathematical modelling game devised by the British mathematician John Horton Conway in 1970. Conway's concept is based on the work of John Von Neumann on mathematical proofs of the possibility of self-replicating robots, which was in turn based on the work of another British mathematician, AM Turing, famous for much of the early work on artificial intelligence and his theoretical 'Universal Calculator', the so-called Turing Machine, capable of performing any desired calculation.

The Life Game is a simulation 'game' in that its rules are set in such a way as to resemble the reproductive behaviour of a population of living organisms. Conway chose his rules very carefully to meet three basic considerations:

1. There should be no initial pattern for which there is a simple proof that the population can grow without limit.
2. There should be initial patterns which apparently grow without limit.
3. There should be simple initial patterns which grow and change for a considerable period of time before coming to an end in three possible ways: (a) Fading away completely, from either overcrowding, (analogous to overpopulation or exhaustion of resources necessary to sustain life) or from becoming too sparse (ie, not having sufficient numbers to maintain a 'birth' rate higher than the death rate). (b) Settling into a stable configuration that remains unchanged thereafter. (c) Entering an oscillating state in which several patterns repeat endlessly over a cycle of two or more generations.

The Life Game takes place on a chessboard-like array consisting of live and dead cells each having eight neighbouring cells. The rules are:

1. Survivals. Every live cell with either two or three nearest neighbours survives into the next generation.
2. Deaths. Every live cell with four or more live neighbours dies from overpopulation. Every live cell with one or no live neighbours dies from isolation.
3. Births. Each empty cell adjacent to exactly three neighbours is a birth cell. It will become live in the next generation.

These rules are by no means the only ones possible. Readers could perhaps experiment with setting up rules of their own, such as introducing another 'species' of cell, obeying slightly different laws of behaviour, or having a fixed lifespan of a set number of generations, or having different birth and death rates.

A few quick runs of the program will show that a bewildering variety of patterns are possible. Some of these are especially interesting due to their ability to roam around the array, 'eat' other groups of cells and even reproduce endless copies of themselves or different patterns. They have been given names describing their individual behaviour. The 'Glider' is a pattern repeating itself over four generations, moving horizontally and vertically one cell in the process. A 'Glider Gun' can emit an endless number of Gliders, which can in turn be 'eaten' by a Pentadecathlon. 'Spaceships' come in various shapes, sizes and speeds. (See diagram.) One such configuration actually has thirteen gliders colliding in such a way that a 'factory' is assembled by the collision mass. This factory then assembles and launches a 'spaceship' every three hundred cycles.

FOR SYSTEM 80/TRS 80

write or call for a free catalogue on the following products:

- Plug in printer interfaces with each unit having decoding for 37E8H and FDH. Programs written for either computer will work without program alteration.

- Four powerful 2K Eprom utilities for the unused 2K block in the System 80/TRS 80 memory map.
  For Disk and Non Disk users
  All include lowercase driver and key debounce.
- Two and Three Eprom plug in P.C. Boards.
  Three Eprom board is designed to run my 3.5K Super or 7K Super Utility Plus using keyboard decoding and bank select.
  Over 50 powerful functions eg: Disassembler, Copier Relocate Object Code, Memory display and edit, Renumberer, etc
  Uses no RAM.
  For Disk and Non Disk users.
- Lowercase with full three dot descenders.
  Involves four wires and plug in board for easy installation.
  Takes one or two character sets.
- Hi-resolution graphics. P.C.G type. Four switching modes Only **$95.00.**
- 32K, 48K and 60K memory upgrades
  No piggybacking.
  Less power drain.
- ROM One replacement.
  Lowercase driver from power up.
  Auto initialisation of Eproms, etc
- Dual Fast Tape ROM board
  First ROM has usual 500 Baud and second 1500 Baud tape routines
  At the flick of a switch everything in ROM or which calls ROM, works at 500 or 1500 Baud.
  Has I.C. timed switching to prevent lockup
  No soldering or track cutting
  Plugs in.
- Repairs. Extra keys. Sound. System 80CHR$(23) as per Tandy. Inverse video, etc

Contact: Geoff Lohrere at

Z80 PROGRAMMING

57A Stanley Ave, Mt Waverley

PH: (03) 543 1485

THE SOFTWARE HOUSE

"HELPING COMPUTERS COME TO TERMS WITH PEOPLE"

QUALITY SOFTWARE

- TEACH/M — CP/M Tutor
- SELECT — Word Processing
- ABSTAT — Statistical Analysis
- GRAPHPLAN — Spreadsheet/graphics
- STATPRO — Stats/Graphics/Dbase
- ASCOM — Communications Facility
- DATAVU — Rel. DB/Report/Data Entry
- RL1 — Rel. DB with COBOL interface
- KNOWLEDGEMAN — Sophisticated Database
- PROJECT SCHEDULER — Project Control/Reporting
- UNIFORM — Disc Conversion Utility
- WRITER'S PACK — WordStar Companion

The Software House is the major Australian & New Zealand distributor for a wide range of quality software products from which we offer full support and back-up. Our aim is to provide computer users with the most efficient, simple and cost effective software products available. We are continually adding new products to our range, so if you have any enquiries or wish to obtain a copy of our catalogue, please do not hesitate to call us.

THE SOFTWARE HOUSE

227 Burwood Road, Hawthorn, Victoria 3122.
Telephone: (03) 819 1488. Telex: 38432 UFIVE

Framework $$795

dBASE III $$795

NEW SOFTWARE

	Product Number	Operating System	List Price	Our Price
ASHTON-TATE				
dBASE II	1028	0 6 M P	620	545
dBASE II with tut.	1030	0 6 M P	669	590
dBASE II tutorial	1031	0 6 M P	75	70
dBASE II update	1032	0 6 M P	105	95
Friday	1033	0 P	360	295
Special just released				
Framework	1066	P	new	795
dBASE III	1067	P	new	795
DATAFLEX				
8 bit single user	5061	0	950	799
8 bit multi user	5062	call us	1200	990
16 bit single user	5063	6 M P	1200	990
16 bit multi user	5064	call us	1800	1475
MICROPRO				
Calcstar	2034	0	220	189
Datastar	2035	0	442	340
Infostar	2036	0 P	666	555
Mailmerge	2037	0 6 M P	377	299
Reportstar	2038	0 P	382	333
Supersort	2039	0	377	299
Spellstar	2040	0 6 M P	377	299
Starburst	2041	0 P	295	249
Star-index	2042	0 P	295	249
Wordmaster	2043	0 6	230	198
Wordstar	2044	0 6 M P	595	495
Wordstar Prof.	2045	0 P	795	685
SORCIM				
Supercalc 80	4048	0 P	320	280
Supercalc 86	4049	6	337	295
Supercalc 2 80	4050	0	395	333
Supercalc 2 86	4051	6 P	395	333
Supercalc 3 (IBM PC)	4052	P	445	363
Super Spellguard	4053	0 6 P	335	295
Superwriter 80	4054	0	486	415
Superwriter 86	4055	6 P	486	415
TYLOG SYSTEMS				
dBASE II Rep. Writ.	3046	0 6 M P	258	229
dBASE II Access Util.	3047	0 6 M P	429	375
BLANK DISKS (BOX of 10)				
Nashua 5 ¼" SS DD	1568		55.70	37.50
Nashua 5 ¼" DS DD	1569		64.10	45.50
Nashua 8" DS DD	1570		71.00	57.50

0 CP/M 80, CP/M 2.2 M MSDOS
6 CP/M 86 P PCDOS

Available for all leading brands.

All prices include sales tax and postage. Allow 5 working days for delivery.

S.F. Programming International

289 Flinders Lane, Melbourne, 3000. Ph. (03) 654 4104

ORDER FORM

Product Number	Computer	Price

TOTAL ....

☐ Cheque/Money Order Enclosed

Charge to: ☐ Bankcard, ☐ Mastercard, ☐ Visa

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Expiry date ....

Name ....

Address ....

.... Ph. ....

Signature .... Date ....

PROGRAMS

The Glider moves at one quarter of the 'Speed of Light', the maximum possible speed of motion on the array. The Speed of Light is one cell per generation, hence the Glider moves at a speed of one cell per four generations. Other mobile configurations can move at their own characteristic velocities. The existence of gliders and guns raises the possibility of simulating the postulated Turing machine, using the gliders as digital pulses for storing and transmitting of information and performing all the requisite logic operations. As you may realise, this is well beyond the capabilities of the authors or the humble VIC 20. There are a vast number of other configurations, too numerous to mention. We suggest you try them out yourself.

The program will run on any size VIC 20, although unexpanded machines will not be able to handle the larger arrays of cells. Unfortunately, the larger and more interesting arrays also run quite slowly. Any suggestions on speeding up the processing time are welcome. One idiosyncracy of our program should be mentioned here also. The top and bottom edges are connected, as are the two sides. This means any pattern going over the top edge will reappear at the bottom, and similarly for the sides. This can of course be changed if so desired. This is to compensate for the VIC's maximum screen display size of 22 x 23 characters. This means that in topological terms, the array is not mapped into a flat plane but onto the surface of a torus.

'Live' cells are inserted onto the array by means of the normal cursor controls. To 'set' a cell as live, Key C is pressed. Pressing C again reverses this. The space bar sets the program running.

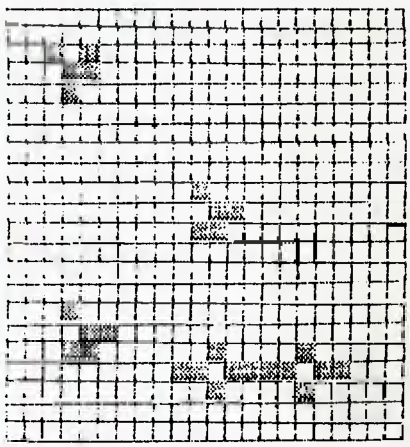
Source material for 'The Life Game' came from *Scientific American* of October 1970 and February 1971.

```
40 INPUT "ARRAY SIZE";D
50 SY=80
60 SZ=2*SY+128
90 IF FRE(X)<2500 THEN CR=38400:FI=7680
   :GOTO 150
91 CR=37888:FI=4096
150 PRINT "♥"
200 DIM A%(D,D),B%(D,D)
210 FOR I=1TOD
220 FOR J=1TOD
225 P=CR+(I-1)+(J)*22
226 POKE P,0
230 A%(I,J)=0:NEXTJ:NEXTI
235 PO=0:GE=0
240 GOSUB 10000
250 I=1:J=1
255 P=FI+(I-1)+(J)*22
260 A$="":GET A$
270 IF A$="" THEN 400
280 IF ASC(A$)=17  THEN 500
290 IF ASC(A$)=145 THEN 600
300 IF ASC(A$)=29  THEN 700
310 IF ASC(A$)=157 THEN 800
320 IF A$=" " THEN 1000
330 IF A$="C" THEN 900
400 FOR K=1TO50:NEXTK
405 POKE P,SZ-PEEK(P)
410 FOR K=1TO10:NEXTK
430 POKE P,SZ-PEEK(P)
440 GOTO 260
500 J=J+1:IF  J>D THEN J=1
510 GOTO 255
600 J=J-1:IF  J<1 THEN J=D
610 GOTO 255
700 I=I+1:IF  I>D THEN I=1
710 GOTO 255
800 I=I-1:IF  I<1 THEN I=D
810 GOTO 255
900 A%(I,J)=1-A%(I,J)
910 POKE P,SZ-PEEK(P)
911 IF A%(I,J)=1 THEN PO=PO+1:GOTO 913
912 PO=PO-1
913 REM
914 GOSUB 20000
915 GOTO 255
1000 PO=0:NS=0
1100 FORI=1TOD
1110 FORJ=1TOD
1111 IP=I+1:IF  IP>D THEN IP=1
1112 IM=I-1:IF  IM<1 THEN IM=D
1113 JP=J+1:IF  JP>D THEN JP=1
1114 JM=J-1:IF  JM<1 THEN JM=D
1115 A=A%(I,J)
1120 S1=A%(IM,JM)+A%(IM,J)+A%(IM,JP)
```

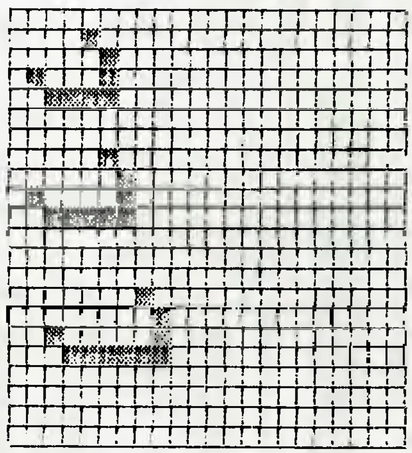
PROGRAMS

```
1130 S2=A%(I,JM)+A%(I,JP)
1140 S3=A%(IP,JM)+A%(IP,J)+A%(IP,JP)
1150 S=S1+S2+S3
1155 B=0
1160 IFA=0ANDS=3THENB=1
1170 IFA=1AND(S=2ORS=3)THENB=1
1180 IF B=1 THEN PO=PO+1
1190 B%(I,J)=B
1195 IF A=B THEN NS=NS+1
1200 NEXTJ:NEXTI
1300 FOR I=1TOD
1400 FOR J=1TOD
1500 A%(I,J)=B%(I,J):NEXTJ:NEXT I:GE=GE+1
1600 GOSUB 10000
1650 A$="":GET A$:IF A$<>"" THEN 250
1700 IF NS<D*DTHEN 1000
1800 PRINT "STABLE
   ":GE=2:GOTO 250
10000 FOR II=1TOD
10010 FOR JJ=1TOD
10020 P=FI+(II-1)+(JJ)*22
10030 POKE P,SY+A%(II,JJ)*128
10035 NEXT JJ:NEXT II
10037 GOSUB 20000
10040 RETURN
20000 PRINT "G=";GE;"P=";PO;
20005 PRINT "";
20010 RETURN

READY.
```



GLIDERS :BOTTOM GLIDER WILL BE EATEN BY PENTADECATHLON.



SPACESHIPS

president

Presents

THE 16-BIT COMPUTER Complete with Colour

THE SOFTWARE *MS DOS 2.11 & Lotus 123*

THE COMPATIBILITY *Takes most available 16-bit software.*

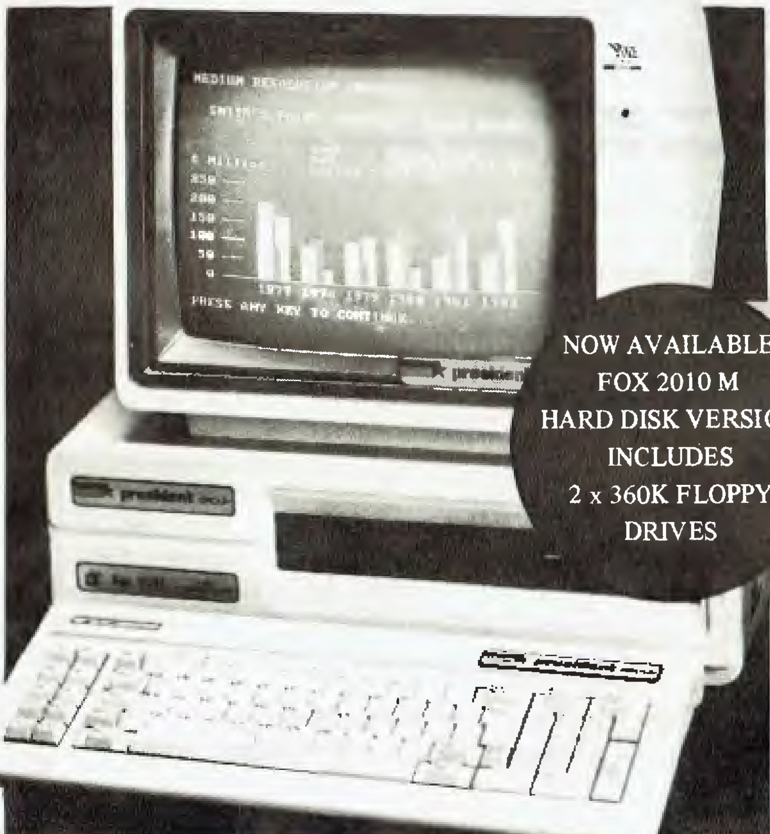
THE PRICE $3800 (ex. tax) *"REMARKABLE"*

Fox 2001 16-BIT MICROCOMPUTER

Complete system includes:

- 14" HI-RES. COLOUR MONITOR
- CPU WITH 256K RAM 16-BIT 8088
- 2 x 360K SLIMLINE DRIVES
- 8 EXPANSION SLOTS
- REAL TIME CLOCK
- COLOUR GRAPHICS CARD WITH SUPER HI-RES MONO CONTROLLER, SOFTWARE SELECTABLE
- 2 x RS-232 SERIAL PORTS
- 1 x CENTRONICS PARALLEL PORT
- MS-DOS 2.0 OPERATING SYSTEM
- Lotus 123
- FAN COOLED 100W POWER SUPPLY

- • OPTIONAL ACCOUNTING SYSTEM
- • WORD PROCESSOR
- • SPELLING DICTIONARY
- • FILER (DATABASE)



NOW AVAILABLE
FOX 2010 M
HARD DISK VERSION
INCLUDES
2 x 360K FLOPPY
DRIVES

Fox 2001 16-BIT MICROCOMPUTER

President Computers NSW Pty Ltd 100 George Street Hornsby 2077 Tel (02) 476 2700
President Computers Vic Pty Ltd Suite 1/609 St Kilda Road Melbourne 3004 Tel (03) 529 1788
President Computers ACT Pty Ltd Shop 2 Warramanga Centre W'manga 2611 Tel (062) 88 2000
President Computers Qld Pty Ltd 220 Boundary Rd Spring Hill (ent Main St) 4000 Tel (07) 229 0600
Micro EDP (Western Australia) 9a/1 Leura Avenue Claremont 6010 Tel (09) 384 5787
Computer Marketing HQ 3 Lindley St Pooraka SA Tel (08) 260 2444
Darwin Business Machines 47 Stuart Highway Darwin 5790 Tel (088) 81 5905
Strawberry Enterprises cnr Main & Kensington Sts Glenorchy 7010 Tel (002) 72 2621

PLEASE SEND ME MORE INFORMATION

Name ....

Address ....

....

Telephone ....

President Computers NSW Pty Ltd
100 George Street Hornsby NSW 2077

Connect-Four

by Robert Jackson

Connect-Four for the Commodore 64 computer is a modified version of 'VIC Connect-Four' by Adrian Millet *(APC January 1983)*.

The objective of the game is to get four of your pieces in a row either vertically, horizontally or diagonally. You can choose to play against the computer or another player. The game is played on a board of 7 columns x 6 rows. All you have to do is press the corresponding column number of where you want your piece to go. The piece will fall all the way to the next vacant bottom position in that column.

The computer will prompt you for your turn, and will accept the first number only. There is a slight pause while it does a few calculations, then it will place your piece.

```
5 REM: GOTO 3000 FOR HELPFUL HINTS
10 REM: TITLE SCREEN
20 POKE53280,0:POKE53281,0:DIMB(8,7),E(2,9),Y(3),X(3)
30 PRINT"                                          "
40 PRINT"                                          "
50 PRINT"                                          "
60 PRINT"                                          "
70 PRINT"                                          "
80 PRINT"                        C64 VERSION BY:"
90 PRINT"                        ROBERT JACKSON."
100 PRINT"                         "
110 PRINT"                                      "
120 PRINT"                                                 "
130 GOSUB2500:IFG=1THEN340
140 PRINT"THIS IS THE BOARD GAME OF CONNECT-FOUR."
150 PRINT"YOUR OBJECTIVE IS TO GET FOUR PIECES IN"
160 PRINT"A ROW, EITHER VERTICALLY, HORIZONTALLY"
170 PRINT"OR DIAGONALLY. YOU MAY CHOOSE TO PLAY"
180 PRINT"AGAINST THE COMPUTER, OR ANOTHER PLAYER.":G=1
190 PRINT" PRESS ANY KEY TO CONTINUE "
200 GETA$:IFA$<>""THEN300
210 FORT=1TO200:NEXT
220 PRINT" PRESS ANY KEY TO CONTINUE "
230 FORT=1TO100:NEXT:GOTO190
299 :
300 REM: SELECT COMPUTER OR TWO PLAYERS
310 C1$="                                        ":REM 40 SPACES
320 PRINT"":FORL=1TO6
330 PRINTC1$:NEXT
340 PRINT""
350 PRINT"     1      - PLAY AGAINST THE COMPUTER."
360 PRINT"     2      - PLAY WITH ANOTHER PLAYER."
370 GETA$:IFA$<>""THEN390
380 FORT=1TO300:NEXT:GOTO300
390 IFA$<>"1"ANDA$<>"2"THEN300
399 :
400 REM: DISPLAY PLAYING BOARD
410 NP=VAL(A$):PRINT""
420 FORL=1TO18:PRINT"                                 ":NEXT
430 FORL=1TO3:PRINT"                                  ":NEXT
440 PRINT"1234567"
450 PRINT"";
460 PRINT"CONNECTFOUR."
```

```
470 PRINT"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■";
480 PRINT"■ R ■ TO■■■■■■■■RESTART"
499 :
500 REM: VARIABLES
510 A1$="■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■"
520 A2$=A1$+"■■■■■■"
530 A3$="■          ■■■■■■■■       ■■■■■■■■       ■■■■■■■■       ■■■■■■■■"
540 A4$="■"+A1$+A3$
550 A5$="■"+A1$+"■■■■■■"+A3$
560 B1$="■■        ■■■■■■■■ WHITE ■■■■■■■■       ■"
570 B2$="■■        ■■■■■■■■ YOUR  ■■■■■■■■       ■"
580 B3$="■■        ■■■■■■■■  YOU  ■■■■■■■■       ■"
590 B4$="■        ■■■■■■■■ BLACK ■■■■■■■■        "
600 B5$="■        ■■■■■■■■  MY   ■■■■■■■■        "
610 B6$="■        ■■■■■■■■   I   ■■■■■■■■        "
620 C1$="■■■■■■■■       ■■■■■■■■ MOVE  ■■■■■■■■       ■"
630 C2$="■■■■■■■■       ■■■■■■■■ WINS  ■■■■■■■■       ■"
640 C3$="■■■■■■■■       ■■■■■■■■  WIN  ■■■■■■■■       ■"
650 Y1$=A1$+B1$+C1$:Y2$=A1$+B1$+C2$:Y3$=A1$+B2$+C1$:Y4$=A1$+B3$+C3$
690 Z1$=A2$+B4$+C1$:Z2$=A2$+B4$+C2$:Z3$=A2$+B5$+C1$:Z4$=A2$+B6$+C3$
799 :
READY.
800 REM: AGAINST COMPUTER
810 IFNP<>1ANDNP<>2THEN300
820 IFNP=2GOTO900
830 IFRND(0)>.5THEN860
840 C=1:GOSUB1000
850 C=2:PRINTZ3$:GOSUB1900:GOTO840
860 PRINT"■■■ I'LL GO FIRST THIS TIME, OK "
870 FORT=1TO2000:NEXT:GOTO 850
899 :
900 REM: TWO PLAYERS
910 C=1:GOSUB1000
920 C=2:GOSUB1000:GOTO910
969 :
970 IFC=1THENPRINTA4$:PRINTA5$:GOTO990
980 IFC=2THENPRINTA4$:PRINTA5$
990 FORT=1TO150:NEXT
999 :
1000 REM: SELECT WHITE OR BLACK MOVE
1010 IFNP=1ANDC=1THENPRINTA5$:PRINTY3$
1020 IFNP=2ANDC=1THENPRINTA5$:PRINTY1$
1030 IFNP=2ANDC=2THENPRINTA4$:PRINTZ1$
1099 :
1100 GETM$:IFM$="R"THEN30
1110 X=VAL(M$):IFX<1ORX>7THENFORT=1TO400:NEXT:GOTO970
1120 GOSUB2050:GOSUB1750
1130 IFY>6THEN1100
1140 GOSUB1500
1150 IFE<500THENRETURN
1199 :
1200 REM: WE HAVE A WINNER
1210 PRINT"■■■  ** ***  GAME OVER  *** **  "
1220 FORL=1TO5
```

```
1230 POKE53280,0:POKE53281,10:FORT=1TO150:NEXT
1240 POKE53280,10:POKE53281,0:FORT=1TO300:NEXT
1250 NEXT
1260 POKE53280,0:POKE53281,0
1270 IFNP=1ANDC=1THENPRINTY4$
1280 IFNP=1ANDC=2THENPRINTZ4$
1290 IFNP=2ANDC=1THENPRINTY2$
1300 IFNP=2ANDC=2THENPRINTZ2$
1310 FORT=1TO2000:NEXT:PRINT"  DO YOU WANT ANOTHER GAME  -  Y/N ?  "
1320 GETA$:IFA$="Y"THEN30
1330 IFA$=""ORA$<>"N"THEN1320
1340 PRINT"":END
1499 :
1500 REM: PLACE PIECES ON BOARD
1510 CL=54272
1520 FORY=1TO6:IFB(X,Y)THENNEXT:RETURN
1530 FORP=X*4+1141TOX*4+1925-Y*120STEP40
1540 POKEP-40,32:POKEP-39,32:POKEP-38,32
1550 IFC=1THEN1600
1560 POKEP,78:POKEP+1,99:POKEP+2,77:POKEP+40,77:POKEP+41,
     100:POKEP+42,78
1570 POKEP+CL,1:POKEP+CL+1,1:POKEP+CL+2,1
1580 POKEP+CL+40,1:POKEP+CL+41,1:POKEP+CL+42,1:GOTO1630
1590 :
1600 POKEP,233:POKEP+1,160:POKEP+2,223:POKEP+40,95:POKEP+41,
     160:POKEP+42,105
1610 POKEP+CL,1:POKEP+CL+1,1:POKEP+CL+2,1
1620 POKEP+CL+40,1:POKEP+CL+41,1:POKEP+CL+42,1
1630 NEXT:POKE198,0
1640 B(X,Y)=C:IFE>500THEN1200
1650 RETURN
1749 :
1800 I=I+Q:J=J+R:N=N+1
1810 IFB(I,J)=C1THEN1800
1820 O=(B(I,J)=0):I=X-Q:J=Y-R
1830 IFB(I,J)<>C1THEN1860
1840 I=I-Q:J=J-R:N=N+1
1850 IFB(I,J)=C1THEN1840
1860 O=-(B(I,J)=0)-O:A=N*3+O:IFA>9THENA=9
1870 C2=C1:IFC=2THENC2=3-C2:E=E+E(C2,A)
1880 E=E+E(C2,A)
1890 NEXT:NEXT:RETURN
1899 :
1900 REM: COMPUTER MOVE
1910 PRINTA4$;"  CHECKING ALL POSSIBLE MOVES "
1920 B=-1:BE=-1E9
1930 FORX=1TO7
1940 GOSUB2050
1950 IFY>6THEN1980
1960 GOSUB1750:PRINT" "TAB(X*4-2)"●"
1970 IFE>BETHENBE=E:B=X
1980 NEXT:X=B:IFX<0THENRETURN
1990 GOSUB1500
2000 FORP=1024TO1140:POKEP,32:NEXT
```

```
2010 IFBE<500THENRETURN
2020 GOTO1200
2049 :
2050 FORY=1TO6
2060 IFB(X,Y)THENNEXT
2070 RETURN
2099 :
2100 DATA0,2,3,0,4,6,1,12,16,1000,0,0,0,0,2,3,0,8,11,120
2110 DATA0,1,1,1,1,1,0,-1,0,0,1,3,1,0,0
2499 :
2500 REM: INITIALISATION
2510 FORX=1TO7:FORY=1TO6:B(X,Y)=0:NEXTY,X
2520 FORX=0TO8:B(X,0)=-1:B(X,7)=-1:NEXT
2530 FORY=0TO7:B(0,Y)=-1:B(8,Y)=-1:NEXT
2540 RESTORE
2550 FORC=1TO2:FORX=0TO9:READE(C,X):NEXTX,C
2560 FORX=0TO3:READX(X):NEXT
2570 FORX=0TO3:READY(X):NEXT
2580 FORX=1TO7:READEC(X):NEXT
2590 RETURN
3000 :
3001 :
3002 :
3003 :
3004 :  HELPFUL HINTS  - DO NOT KEY IN.
3005 : "
3006 : " SHIFT CLR : CLEAR SCREEN        CTRL 2 : WHITE
3007 : " CLR       : CURSOR HOME         CTRL 4 : CYAN
3008 : " CTRL 9    : RVS ON              CTRL 8 : YELLOW
3009 : " CTRL 0    : RVS OFF             COMM 1 : ORANGE
3010 : " CRSR LEFT                       COMM 3 : LIGHT RED
3011 : " CRSR RIGHT                      COMM 5 : GRAY 2
3012 : " CRSR UP                         COMM 7 : LIGHT BLUE
3013 : " CRSR DOWN

READY.
```

Database VZ-200

by Ted Barker

This is an information storage and retrieval program for the VZ-200 with 16k expansion used in conjunction with a suitable cassette recorder and 80 column printer. The program has been adapted from one written for the Commodore VIC 20/64 by John Stilwell of Madison, WI, USA which was published in the February, 1984 issue of the magazine *RUN*.

When you run the program you will be asked to enter a file name, [RETURN] Without entering a file name will result in a default to the file title 'NO NAME'. Some three seconds later a list of one-letter commands will be displayed. [M] will display a full menu, detailing the meanings of the one-letter commands. (Menu may be called at any time without affecting file entries).

Information is entered into pages, [P], each of which holds 10 line numbers. [E]. The total amount of information which may be filed is determined by the value of 'N' (number of lines) in Line 140. In the listing 'N' has a value of 400 which should allow up to 32 characters of entry per line.

Commands

'C' (Catalogue) will display the file name together with any lines you have designated as catalogue entries. (See Using The Catalogue). 'P' (Page), will ask you to enter a page number, (1 to 40 in the program listed). Enter a page number and press [RETURN] and the page, together with 10 lines will be displayed, ready for reading or making an entry. 'E' (Entry) asks for a line number; enter the required line number and press [RETURN]; enter the information you wish to file and press [RETURN] once more; your entry will then be confirmed on screen. 'I' (Insert) follows the same procedure as Enter. When you enter the desired information, it will be entered at the designated line number. All lines with a higher number will be incremented by one and no information will be lost. 'N' (New File Name) will ask 'Are you sure?'. Entering a new file name will result in the loss of any information stored in the current file. 'S' (Save to Tape) will ask 'Are you sure?'. If your answer is 'Y' just follow the screen prompts. As each entry is saved onto tape, its line number will be displayed at lower left screen. This serves as a check that the saving process is working OK. The word 'COMPLETE' will appear when all of your file is saved to tape.

'L' (Load From Tape) will again ask for confirmation. Load is similar in operation to Save except that you will be asked to enter a file name. During loading, the word 'WAITING' will appear as usual. This will be followed by the word 'FOUND'. The word 'LOADING' does *not* appear. As each item is loaded its line number will be displayed at lower left screen as a check that the loading procedure is going according to plan.

'H' (Hard Copy) will allow you to produce a print-out of your file. The file name will be enhanced, followed by the remainder of your file in unenhanced type. You will have the option of printing the entire file or of specifying a starting and ending line number.

'D' (Delete) follows the same procedure as Enter and Insert. A line number entered after the 'D' command will result in that line being deleted from the file. Higher line numbers will be decreased by one, leaving no gaps in your file. No information will be lost, except for the line you deleted. 'A' (Alphabetize) allows entries to be placed in order after they have been entered. You may place the entire file in alphabetical order, or specify starting and ending line numbers. (Note remarks in Using The Catalogue.) 'M' (Menu) displays menu on screen in case you forget what the single letter commands mean.

Using the catalogue

If you wish to split your files into separate categories you may display category headings in the file catalogue. To do this, the entry is made in the usual way but with an inverse 'C' preceding the entry. Line 170 in the program reads this character, ASC(195), and places those entries in the catalogue, together with the page number on which they appear.

As these entries still appear in their correct position in the body of the file, it may be an advantage to enter the whole of that entry in inverse print, thus making the category headings stand out when going through the file. When using the Alphabetize routine on a file containing inverse 'C' entries, it is essential that the line number after the inverse 'C' entry is used as a starting line and that the ending number should not be greater than the entry containing the next inverse 'C'. If this rule is not observed the inverse 'C' entries will be alphabetized with the rest of the file, thus destroying its usefulness as a category heading.

If you enter a command letter by mistake, just enter another command instead of a line number. Omit all line numbers below 100 as this will allow a little more memory available for your database.

When you have completed all your entries, make one more entry on the next line. This entry should be 'END'. If you do this the alphabetize and catalogue routines will run much more quickly.

```
3 CLS:COLOR2:PRINT@  4,"[illegible]"
4 PRINT@ 36," [illegible]"
5 PRINT@ 68," [illegible]"
6 PRINT@100," [illegible]"
7 PRINT@132," [illegible]"
8 PRINT@164," [illegible]"
9 PRINT@196," [illegible]"
10 PRINT@326,"[illegible]"
11 PRINT@358," [illegible]"
12 PRINT@390,"[illegible]"
13 PRINT@422,"[illegible]"
14 PRINT@454 "[illegible]"
15 PRINT@267,"[illegible]"
16 FORI=1TO5000:NEXT
20 REM +++++++++++++++++++++++++
21 REM +   DATA BASE VZ-200    +
22 REM +++++++++++++++++++++++++
23 REM
24 REM
25 REM +++++++++++++++++++++++++
26 REM +      TED BARKER       +
27 REM +   3 SOUTHWARK WAY,    +
28 REM + MORLEY,W,AUST 6062 +
29 REM +++++++++++++++++++++++++
30 CLS:PRINT@8,"[illegible]"
31 PRINT@66,"THIS PROGRAMME WILL STORE"
32 PRINT@98,"ITEMS OF INFORMATION ON UP"
```

```
33 PRINT@130,"TO 40 PAGES, EACH PAGE"
34 PRINT@162,"CONTAINING 10 LINES."
35 PRINT@226,"YOU MAY ENTER,INSERT,DELETE"
36 PRINT@258,"OR OVERWRITE INFORMATION-"
37 PRINT@290,"ALPHABETIZE OR PRINT ALL"
38 PRINT@322,"OR PART OF YOUR FILES-"
39 PRINT@354,"WHICH CAN THEN BE SAVED"
40 PRINT@386,"AND/OR RETRIEVED FROM TAPE."
41 PRINT@450,"[illegible]";
42 K$=INKEY$:I$=INKEY$:IF I$<>" "THEN 42
43 CLS:PRINT@8,"[illegible]"
44 PRINT@34,"YOU WILL REQUIRE THE 16K"
45 PRINT@66,"EXPANSION WITH YOUR VZ-200,"
46 PRINT@98,"A CASSETTE RECORDER AND"
47 PRINT@130,"SUITABLE PRINTER."
48 PRINT@194,"WHEN RETRIEVING A FILE"
49 PRINT@226,"FROM TAPE, THE WORD"
50 PRINT@258,"[illegible] WILL NOT APPEAR."
51 PRINT@290,"YOU WILL SEE [illegible].
52 PRINT@322,"FOLLOWED BY [illegible].
53 PRINT@354,"THE NUMBER OF EACH FILE"
54 PRINT@386,"ENTRY WILL BE DISPLAYED"
55 PRINT@418,"AS EACH ENTRY IS LOADED."
58 PRINT@451,"[illegible]"
59 K$=INKEY$:I$=INKEY$:IF I$<>" " THEN 59
60 PRINT@482,"[illegible]";
61 K$=INKEY$:I$=INKEY$:IF I$<>" " THEN 61
62 CLS:PRINT@66,"WHEN ENTERING THIS"
63 PRINT@98,"PROGRAMME, YOU WILL HAVE MORE"
64 PRINT@130,"MEMORY FOR YOUR FILES IF"
65 PRINT@162,"YOU OMIT LINES BELOW 100."
66 PRINT@298,"GOOD LUCK!"
68 PRINT@451,"[illegible]"
69 K$=INKEY$:I$=INKEY$:IF I$<>" " THEN 69
100 CLS:PRINT@200,PEEK(30897)+256*PEEK(30898):CLEAR 12000
110 CLS:PRINT"FILE NAME,";:INPUT T$:IF T$="" THEN T$="NO NAME
120 PRINT@134,"[illegible]"
130 T$=LEFT$(T$,14)
140 N=400 :P=1:X=(N+1)/10:DIMS$(N):R$="LINE NUMBER"
145 GOSUB 1110:GOTO 200
150 K=0
160 CLS:PRINT@7,"[illegible]",T$:PRINT:PRINT"PAGE "
165 FOR J=K TO N
170 IFASC(S$(J))=195,PRINTINT(J/10+1);RIGHT$(S$(J),LEN(S$(J))-1)
180 IFS$(J)="END" THEN 200
190 NEXTJ
200 GOSUB 490:IF A$="A"THEN GOTO 200
210 IF A$="C" THEN GOTO 150
220 IF A$="P" THEN GOTO 340
230 IF A$="E" THEN GOTO 410
240 IF A$="I" THEN GOTO 440
250 IF A$="N" THEN GOTO 520
260 IF A$="S" THEN GOTO 640
```

```
270 IF A$="L" THEN GOTO 680
280 IF A$="H" THEN GOTO 950
290 IF A$="D" THEN GOTO 730
300 IF A$="A" THEN GOTO 770
310 IF A$="M" THEN GOTO 560
320 IF J<N+1 THEN K=J:GOTO 160
330 GOTO 150
340 PRINT@384,"ENTER PAGE NUMBER";:INPUT A$:P=VAL(A$)
350 CLS:PRINT"PAGE "P" "T$:PRINT
360 FORI=0TO9:L=(P-1)*10+I:PRINTL;S$(I):NEXT
370 GOSUB 490
380 IF A<>12 THEN210
390 P=P+1:IF P>X THEN P=1
400 GOTO 350
410 A$="-1":PRINT@384,"ENTER "R$;:INPUT A$:J=VAL(A$)
420 GOSUB 500:IF A<>0 THEN GOTO 210
430 INPUT S$(J):GOTO 350
440 A$="-1":PRINT@384,"INSERT "R$;:INPUT A$:J=VAL(A$)
450 GOSUB 500:IF A<>0 THEN GOTO 210
460 INPUT D$:CLS:IF R=N THEN GOTO 350
470 GOSUB 1160:FOR I=KK TO J+1 STEP-1:S$(I)=S$(I-1):NEXT
480 S$(J)=D$:GOTO 350
490 E$="[illegible]":PRINT@490,E$;:GOSUB 890
500      FOR I=1 TO 13:IF MID$(E$,I*2-1)=A$ THEN A=I:I=13
510 RETURN
520 CLS:GOSUB 920:IF A$<>"Y" THEN GOTO 150
530 PRINT@195,"NEW FILE NAME.";:INPUT T$
540 PRINT@259,"[illegible]"
550 GOSUB 1110:GOTO 150
560 CLS:PRINT@12,"[illegible]":PRINT@68,"[illegible]ATALOGUE"
570 PRINT"    [illegible] CALL PAGE":PRINT"    [illegible]NTER":PRINT"    [illegible]NSERT"
580 PRINT"    [illegible]EW FILE"
590 PRINT"    [illegible]AVE ON TAPE":PRINT"    [illegible]OAD FROM TAPE"
600 PRINT"    [illegible]ARD COPY ON PRINTER"
610 PRINT"    [illegible]ELETE":PRINT"    [illegible]LPHABETIZE":GOSUB 490
630 GOSUB 890:GOTO 210
640 CLS:PRINT@41,"[illegible]"
650 GOSUB 920:IF A$<>"Y" THEN GOTO 150
660 CLS:PRINT@132,"PREPARE CASSETTE":PRINT
670 INPUT"    THEN PRESS <RETURN>";X:GOTO 1190
680 CLS:PRINT@35,"[illegible]"
690 GOSUB 920:IF A$<>"Y" THEN GOTO 150
700 INPUT"     FILE NAME":T$
710 CLS:PRINT@132,"PREPARE CASSETTE"
720 INPUT"    THEN PRESS <RETURN>";X:GOTO 1320
730 PRINT@384,"DELETE "R$;:INPUT A$:J=VAL(A$):GOSUB 500
740 FOR I=J TO N-1:IF S$(I)="-" AND S$(I+1)="-" THEN I=N-1
750 S$(I)=S$(I+1)
760 NEXT:S$(N)="-":GOTO 350
770 CLS:PRINT@40,"[illegible] "
775 PRINT@104,"ENTIRE FILE";:INPUT Z$:IF Z$="N" THEN GOTO 1500
780 GOSUB 1120:U=VAL(A$):IF U<0 OR U>N THEN GOTO 200
790 GOSUB 1130:K=0:FOR I=U TO KK
800 NN=I-1:I=KK
```

```
810 NEXT I:IF K=1 THEN GOTO 830
820 NN=KK
830 I=0
840 J=U:IF I=NN-U THEN GOTO 350
850 IF J=NN-I THEN GOTO 880
860 IF S$(J)>S$(J+1) THEN TP$=S$(J):S$(J)=S$(J+1):S$(J+1)=TP$
870 J=J+1:GOTO 850
880 I=I+1:GOTO 840
890 K$=INKEY$:A$=INKEY$:IF A$="" THEN GOTO 890
900 RETURN
910 PRINT@456,;I:RETURN
920 PRINT@131,"ARE YOU SURE,(Y/N)";
930 INPUT A$
940 RETURN
950 CLS:PRINT@41,"[illegible]"
960 PRINT@105,"ENTIRE FILE";:INPUT X$:IF X$="Y" THEN F=0:N=M
970 GOSUB 1120:F=VAL(A$):K=F:IF F<0 OR F>N THEN GOTO 950
980 PRINT@232,"ENDING "R$;:INPUT A$:M=VAL(A$)
1010 A$="":FOR I=1 TO INT(40-LEN(T$))/2:G$=CHR$(8):F$=CHR$(15)
1020 A$=A$+" ":NEXT:LPRINT CHR$(14)A$+T$+F$,CHR$(10),CHR$(10)
1030 LL=4:FOR I=K TO M
1040 IF S$(I)<>"-" THEN LPRINTS$(I)CHR$(10):LL=LL+2
1050 GOTO 1090
1060 A$="":FOR KK=7 TO LEN(S$(I))*6:A$=A$+CHR$(255):NEXT KK
1070 LPRINT"        "+G$+A$:LPRINTF$+"        "+S$(I)+G$
1080 LPRINTF$+"        "+G$+A$+F$,CHR$(10):LL=LL+4
1090 IF LL>59 THEN FOR NL=LL TO 72:LPRINTCHR$(10):NEXT:LPRINT""
1100 LL=3:NEXT:GOTO 150
1110 FOR J=0 TO N:S$(J)="-":NEXT:RETURN
1120 PRINT@168,"STARTING "R$;:INPUT A$:RETURN
1130 FOR I=N TO 0 STEP-1:IF S$(I)<>"-" THEN KK=I:I=0:GOTO 1150
1140 KK=I
1150 NEXT:RETURN
1160 FOR I=J+1 TO N:IF S$(I)="-" THEN KK=I:I=N:GOTO 1150
1170 KK=I
1180 NEXT:RETURN
1190 CLS:PRINT@196,"[illegible]"
1200 FOR I=N TO 0 STEP-1:K=I:IF S$(I)<>"-" THEN I=0
1210 NEXT I:K=K+1
1220 PRINT#"VZ-DATA",K
1230 FOR I=0 TO K
1240 PRINT#"DATA",S$(I)
1250 GOSUB 910
1260 NEXT
1270 CLS:PRINT@194,"[illegible]"
1280 PRINT@260,"PRESS <F> FOR FILE."
1290 K$=INKEY$
1300 I$=INKEY$:IF I$="" THEN GOTO 1300
1310 IF I$="F" THEN GOTO 350
1320 CLS
1330 PRINT@196,"[illegible]"
1340 INPUT#"VZ-DATA",K
1350 FOR I=0 TO K
1360 INPUT#"DATA",S$(I)
```

```
1370 GOSUB 910
1380 NEXT
1390 CLS:PRINT@194,"[illegible]"
1400 PRINT@260,"PRESS <F> FOR FILE."
1410 K$=INKEY$
1420 I$=INKEY$:IF I$="" THEN GOTO 1420
1430 IF I$="F" THEN GOTO 350
1500 PRINT@168,"STARTING "R$;:INPUT A$
1510 PRINT@232,"ENDING   "R$;:INPUT B$
1520 U=VAL(A$):KK=VAL(B$)
1525 FORI=KKTO0STEP-1:IFS$(I)<>"-"THENKK=I:I=0:GOTO 1527
1526 KK=I
1527 NEXT
1530 K=0:FOR I=U TO KK
1540 NN=I-1:I=KK
1550 NEXT I:IF K=1 THEN GOTO 1570
1560 NN=KK
1570 I=0
1580 J=U:IF I=NN-U THEN GOTO 350
1590 IF J=NN-I THEN GOTO 1620
1600 IF S$(J)>S$(J+1) THEN TP$=S$(J):S$(J)=S$(J+1):S$(J+1)=TP$
1610 J=J+1:GOTO 1590
1620 I=I+1:GOTO 1580
```

Grafx Editor

by Nick Cooper

Grafx Editor lets you create pictures on your color computer's screen. The program uses Extended Color Basic and a minimum of 32k RAM. It will not work with 16k because it PCLEARs 8 graphic pages.

When you first RUN the program, the menu will appear with 14 options. These are: circle, color, line, line box, line box fill, paint, Pcls, Pmode, preset, Pset, screen, see picture, save screen to tape, and load screen from tape. The first eleven are extended color Basic commands and function in the same way.

You should first select option 8 (Pmode). This will get you into the desired graphic mode. Next you should choose option 11 (Screen). This will get you into your desired graphic screen. Finally, select option 7 (Pcls). This will prepare you for your next option.

If you wish to stop work on your picture and continue later on, you can save the screen on tape. When you wish to load it back in, first of all, you have to set up the graphic screen. If you want to load the screen back in without using the Grafx Editor, use this program:

```
10 PMODE 4,1:PCLS:SCREEN 1,1
20 CLOADM
30 GOTO 30
```

```
10 '     ***GRAFX EDITOR***
20 '     *                *
30 '     * BY NICK COOPER *
40 '     *                *
50 '     *                *
60 '     *                *
70 '     *                *
80 '     *    COPYRIGHT   *
90 '     ******************
100 GOTO1150
110 CLS:PRINT@6,"*** GRAFX EDITOR ***"
120 PRINT@35,"<1>  CIRCLE"
130 PRINT@67,"<2>  COLOR"
```

```
140 PRINT@99,"<3>  LINE"
150 PRINT@131,"<4>  LINE BOX"
160 PRINT@163,"<5>  LINE BOX FILL"
170 PRINT@195,"<6>  PAINT"
180 PRINT@227,"<7>  PCLS"
190 PRINT@259,"<8>  PMODE"
200 PRINT@291,"<9>  PRESET"
210 PRINT@323,"<10> PSET"
220 PRINT@355,"<11> SCREEN"
230 PRINT@387,"<12> SEE PICTURE"
240 PRINT@419,"<13> SAVE SCREEN TO TAPE"
250 PRINT@451,"<14> LOAD SCREEN FROM TAPE"
260 PRINT@488,;:INPUT"SELECT (1-14)";A
270 IFA<1ORA>14THEN11Ø
280 ONA GOTO29Ø,4ØØ,46Ø,56Ø,65Ø,74Ø,81Ø,87Ø,94Ø,
    1ØØØ,1Ø7Ø,113Ø,116Ø,125Ø
290 CLS:PRINT@9,"*** CIRCLE ***"
300 PRINT@66,"CIRCLE(X,Y),R,C,HW,START,END"
310 PRINT
320 INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN32Ø
330 INPUT"Y (Ø-191) ";Y:IFY<ØORY>191THEN33Ø
340 INPUT"RADIUS (Ø-255) ";R:IFR<ØORR>255THEN34Ø
350 INPUT"COLOR (Ø-8) ";C:IFC<ØORC>8THEN35Ø
360 INPUT"HEIGHT/WIDTH RATIO (Ø-4) ";HW:IFHW<ØORHW>4THEN36Ø
370 INPUT"START (Ø-1) ";S:IFS<ØORS>1THEN37Ø
380 INPUT"END (Ø-1) ";E:IFE<ØORE>1THEN38Ø
390 CIRCLE(X,Y),R,C,HW,S,E:GOTO11Ø
400 CLS:PRINT@9," *** COLOR ***"
410 PRINT@67,"COLOR FOREGROUND,BACKGROUND"
420 PRINT
430 INPUT"FOREGROUND (Ø-8) ";F:IFF<ØORF>8THEN43Ø
440 INPUT"BACKGROUND (Ø-8) ";B:IFB<ØORB>8THEN44Ø
450 COLORF,B:GOTO11Ø
460 CLS:PRINT@1Ø,"*** LINE ***"
470 PRINT@67,"LINE(X,Y)-(H,V),PSET,PRESET"
480 PRINT
490 INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN49Ø
500 INPUT"Y (Ø-255) ";Y:IFY<ØORY>255THEN5ØØ
510 INPUT"H (Ø-255) ";H:IFH<ØORH>255THEN51Ø
520 INPUT"V (Ø-255) ";V:IFV<ØORV>255THEN52Ø
530 INPUT"1 PSET OR 2 PRESET ";P:IFP<1ORP>2THEN53Ø
540 IFP=1THENLINE(X,Y)-(H,V),PSETELSEIFP=2THENLINE(X,Y)-(H,V),PRESET
550 GOTO11Ø
560 CLS:PRINT@8,"*** LINE BOX ***"
570 PRINT@64,"LINE(X,Y)-(H,V),PSET OR PRESET,B"
580 INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN58Ø
590 INPUT"Y (Ø-255) ";Y:IFY<ØORY>255THEN59Ø
600 INPUT"H (Ø-255) ";H:IFH<ØORH>255THEN6ØØ
610 INPUT"V (Ø-255) ";V:IFV<ØORV>255THEN61Ø
620 INPUT"1 PSET OR 2 PRESET ";P:IFP<1ORP>2THEN62Ø
630 IFP=1THENLINE(X,Y)-(H,V),PSET,B:ELSELINE(X,Y)-(H,V),PRESET,B
640 GOTO11Ø
650 CLS:PRINT@6,"*** LINE BOX FILL ***"
660 PRINT@65,"LINE(X,Y)-(H,V),PSET/PRESET,BF":PRINT
```

```
670 INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN67Ø
68Ø INPUT"Y (Ø-255)";Y:IFY<ØORY>255THEN68Ø
69Ø INPUT"H (Ø-255) ";H:IFH<ØORH>255THEN69Ø
7ØØ INPUT"V (Ø-255) ";V:IFV<ØORV>255THEN7ØØ
71Ø INPUT"1 PSET OR 2 PRESET ";P:IFP<1ORP>2THEN71Ø
72Ø IFP=1THENLINE(X,Y)-(H,V),PSET,BF:ELSELINE(X,Y)-(H,V),PRESET,BF
73Ø GOTO11Ø
74Ø CLS:PRINT@1Ø,"*** PAINT ***":
75Ø PRINT@73,"PAINT(X,Y),C,B":PRINT
76Ø INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN76Ø
77Ø INPUT"Y (Ø-255) ";Y:IFY<ØORY>255THEN77Ø
78Ø INPUT"COLOR (Ø-8) ";C:IFC<ØORC>8THEN78Ø
79Ø INPUT"BORDER (Ø-8) ";B:IFB<ØORB>8THEN79Ø
8ØØ PAINT(X,Y),C,B:GOTO11Ø
81Ø CLS:PRINT@1Ø,"*** PCLS ***"
82Ø PRINT@77,"PCLS C"
83Ø PRINT
84Ø INPUT"COLOR (Ø-8) ";C:IFC<ØORC>8THEN84Ø
85Ø PCLSC
86Ø GOTO11Ø
87Ø CLS:PRINT@1Ø,"*** PMODE ***"
88Ø PRINT@7Ø,"PMODE MODE,START PAGE"
89Ø PRINT
9ØØ INPUT"MODE (Ø-4) ";MODE:IFMODE<ØORMODE>4THEN9ØØ
91Ø INPUT"START PAGE (1-8) ";SP:IFSP<1ORSP>8THEN91Ø
92Ø PMODEMODE,SP
93Ø GOTO11Ø
94Ø CLS:PRINT@9,"*** PRESET ***"
95Ø PRINT@74,"PRESET(X,Y)":PRINT
96Ø INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN96Ø
97Ø INPUT"Y (Ø-255) ";Y:IFY<ØORY>255THEN97Ø
98Ø PRESET(X,Y)
99Ø GOTO11Ø
1ØØØ CLS:PRINT@1Ø,"*** PSET ***"
1Ø1Ø PRINT@74,"PSET(X,Y,C)":PRINT
1Ø2Ø INPUT"X (Ø-255) ";X:IFX<ØORX>255THEN1Ø2Ø
1Ø3Ø INPUT"Y (Ø-255) ";Y:IFY<ØORY>255THEN1Ø3Ø
1Ø4Ø INPUT"COLOR (Ø-8) ";C:IFC<ØORC>8THEN1Ø4Ø
1Ø5Ø PSET(X,Y,C)
1Ø6Ø GOTO11Ø
1Ø7Ø CLS:PRINT@9,"*** SCREEN ***"
1Ø8Ø PRINT@66,"SCREEN SCREEN TYPE,COLOR SET":PRINT
1Ø9Ø INPUT"SCREEN TYPE (Ø-1) ";ST:IFST<ØORST>1THEN1Ø9Ø
11ØØ INPUT"COLOR SET (Ø-1) ";CS:IFCS<ØORSC>1THEN11ØØ
111Ø SCREEN ST,CS
112Ø GOTO11Ø
113Ø GOSUB136Ø:PMODE,SP:SCREEN ST,CS
114Ø IFINKEY$=""THEN114ØELSE11Ø
115Ø CLEAR1ØØØ:PCLEAR8:GOTO11Ø
116Ø GOSUB136Ø:CLS:PRINT@2,"*** SAVE SCREEN TO TAPE ***"
117Ø PRINT@66,"CSAVEM"CHR$(34)"FILENAME"CHR$(34)",1536,7679,Ø"
118Ø PRINT:INPUT"FILENAME ";N$
119Ø PRINT:PRINT"PREPARE TAPE AND PRESS <ENTER>"
12ØØ T$=INKEY$:IFT$=""THEN12ØØ
```

```
1210 IFT$<>CHR$(13)THEN1200
1220 PRINT:PRINT"SAVING...";N$
1230 CSAVEM N$,1536,7679,0
1240 GOTO110
1250 GOSUB1360:CLS:PRINT@1," *** LOAD SCREEN FROM TAPE ***"
1260 PRINT@72,"CLOADM"CHR$(34)"FILENAME"CHR$(34)
1270 PRINT:INPUT"FILENAME ";N$
1280 PRINT:PRINT"PREPARE TAPE AND PRESS <ENTER>"
1290 T$=INKEY$:IFT$=""THEN1290
1300 IFT$<>CHR$(13)THEN1290
1310 PRINT:PRINT"LOADING...";N$
1320 FORH=1TO500:NEXTH
1330 PCLS:PMODE MODE,SP:SCREEN ST,CS
1340 CLOADM N$
1350 IFINKEY$=""THEN1350ELSE110
1360 IFSP=0ORST=0ORSC=0THENCLS:PRINT@224,"CHOOSE SCREEN AND
     MODE FIRST !!":FORE=1TO1000:NEXTE:GOTO110ELSERETURN
```

Basic system reset

by FM O'Dwyer

This program for the Atari 800 enables Basic programmers to handle System resets in a straightforward way. It uses machine code and Atari's auto-entry mode to implement a sort of ON RESET GOTO . . . command.

The program is used as follows: key in the data statements and initialisation code. When this is run, pressing System reset will cause a normal Reset, but instead of returning to a 'Ready' prompt, the machine will execute GOTO RESET as if you had typed this as a direct command. You can therefore direct the machine to the Basic line of your choice by assigning Reset to be the line number.

Note that POKE 842,12:GRAPHICS 0 is essential to turn off auto-entry and prevent the machine getting stuck. The net effect is as if the TRAP command worked for RESET, except you cannot test for the line number on which the RESET occurred.

The program can be used in many applications, but it is particularly valuable because it stops inexperienced users from breaking out of programs by inadvertently pressing system reset.

Any program that uses I/O channels such as disk or cassette could also use this program to close all open channels on a system reset.

```
0 POKE 842,12: REM THIS IS FOR SAFETY
10 RESET=20000
20 RESTORE: FOR N=0 TO 63: READ A: POKE 1536+N,A: NEXT N:
X=USR(1536)
30 GRAPHICS 0: PRINT"IF YOU PRESS RESET NOW,"
40 PRINT "THE ROUTINE AT LINE 2000 WILL EXECUTE" :END
1000 DATA 104,169,14,133,2,169
1010 DATA 6,133,3,169,2,133
1020 DATA 9,96,169,5,141,197
1030 DATA 2,162,0,138,72,189
1040 DATA 43,6,32,164,246,104
1050 DATA 170,232,189,43,6,208
1060 DATA 240,169,13,141,74,3
1070 DATA 96,29,29,29,29,71
1080 DATA 79,84,79,32,82,69
1090 DATA 83,69,84,155,28,28
1100 DATA 28,28,28,0
20000 POKE 842,12: GRAPHICS 0
20010 LIST
```

Basic program notes

```
1000       *=$600       ;Set the origin of the code to 1536
1010 ;
1020 CASINI=2           ;Address of RESET vector
1030 EOUTCH=$F6A4       ;Output character routine
1040 COLOR=709          ;Color register
1050 BOOT=9             ;RESET control register
```

```
1060 ;
1070 ;Initialisation routine
1080 ;
1090 INIT      PLA                     ;Strip Basic parameter count
1100           LDA  #RESET&256         ;Low byte of reset address
1110           STA  CASINI
1120           LDA  #RESET/256         ;High byte of reset address
1130           STA  CASINI+1
1140           LDA  #2                 ;Enable reset routine
1150           STA  BOOT
1160           RTS
1170 ;
1180 ;Reset routine.
1190 ;
1200 RESET     LDA#5                   ;Make foreground colour
1210           STA COLOR               ;same as background
1220           LDX#0
1230 NEXT      TXA                     ;Output the string
1240           PHA                     ;"GOTO RESET" one character
1250           LDA STR,X               ;at a time, including the
1260           JSR EOUTCH              ;cursor positioning codes
1270           PLA
1280           TAX
1290           INX
1300           LDA STR,X
1310           BNE NEXT
1320           LDA #13                 ;enable auto-entry
1330           STA 842
1340           RTS
1350 ;
1360 STR       .BYTE"4 cursor downs"
1370           .BYTE"GOTO RESET",155
1380           .BYTE"5 cursor ups"
1390           .BYTE 0
```

Assembly language notes

AUSTRALIA'S NO 1 PRINTER NOW PRINTS 25% FASTER.

THE AMUST 80 DT/2 DOT MATRIX PRINTER.

No wonder it's the number one selling dot matrix printer. Not only is it Australia's lowest cost printer, but now the DT 80/2 prints 100 C.P.S.

- ☐ Manufactured in Japan, to ensure the highest standards of componentry and workmanship.
- ☐ Superior 9 × 11 dot matrix quality and a total of 12 character variations.
- ☐ 11 inch carriage, now with friction feed, will take any software stationery.
- ☐ Oblique run, ribbons have an extra long 2.5 million character life.

Find out how Australia's top selling dot matrix printer is now even better at your Amust Dealer soon.

IN A WORLD OF UNCERTAINTY, THE CORRECT DECISION IS AMUST.

AMust COMPUTER CORP.

For full details contact the Amust Distributor in your state.

VIC. METRO: Compak Computer Shop 350 South Road, Moorabbin. Telephone: (03) 555 9844

VIC. COUNTRY: Data Parts 11 Edward Street, Shepparton. Telephone: (058) 21 7155 (Also at Albury, Bendigo & Ballarat.)

NSW: Andy Stenberg 7 Arcadia Street, Coogee. Telephone: (02) 260 1249

ACT: Computech Shop 7, Benjamin Way, Church Centre, Belconnen. Telephone: (062) 51 2525

S. AUST: Amust Computers (S.A.) 73 Manton Street, Hindmarsh. Telephone: (08) 46 9329

W. AUST: Microbase Computers 422 Newcastle Street, West Perth. Telephone: (09) 328 9544

QLD: Amust Computer (Qld) 2/6 Harvton Street, Stafford. Telephone: (07) 352 6355

RGS 24002B

ADVERTISERS INDEX

CHIP CHAT

If you're a journalist for a computer rag, were in Sydney for the 2nd *APC* Show and spotted lurking in King's Cross, have been to Lorne, flew to Melbourne on the same plane as an ad. rep from a sister publication of ours — we're watching you.

You were spotted at King's Cross during the *APC* Show, entering a theatre (presenting an 'R' rated flick of course) at Lorne and, now, back again at King's Cross. Chip Chat is everywhere. We know who you are and we know what you've done. How do you ever get time to write?

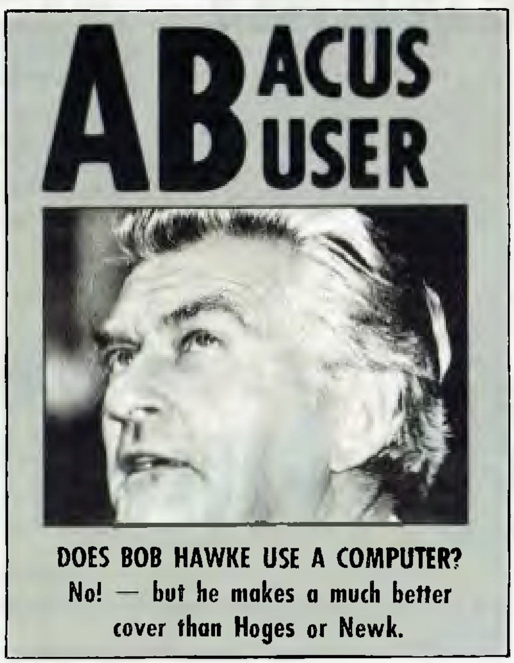
The latest anti-pirate scheme is to sell games software with a dongle. It's a small circuit board encased in resin which must be plugged into one of the ports of a computer before the game will load/operate. Chip Chat can reveal that as the resin (usually) contains aromatic solvents, the dongle isn't really an anti-pirate device but an attempt to make games more addictive.

In old Taiwan, where according to legend the Apple clones are as numerous as fruit on a tree, disaster looms.

The blossoming electronics industry faces an unseasonal frost in the form of the worldwide chip shortage. An Intel 8088 costs twice as much on the spot market than it does in long-term contracts.

The Taiwanese response is typical of the spirit that has made some of its companies such a thorn in the side of western micro makers. The number of thefts is growing. One victim, Multitech, says it lost $1 million's worth of integrated circuits in one fell swoop last month. The thieves even stole a pair of company trucks to get away.

Anybody offered an IBM PC-lookalike with a 20-ton Magirus Deutz truck where the processor ought to be should contact the Taiwanese police urgently.



We don't often give new computer mags a go... but this one deserves a plug for its dig.

IF YOU CAN'T GET AN IBM PC, SETTLE FOR SOMETHING BETTER.

Once you discover the performance and the power of the Corona PC's,™ you'll quickly see the difference in today's standard for micro computing in the IBM-compatible world. Because the Corona PC's go beyond the standard to give you much more of what you're looking for in an IBM-compatible PC.

- ☐ Powerful, low-cost desktop and portable computers.
- ☐ 16-bit, fully compatible with IBM PC standard.
- ☐ Crisper, cleaner display and higher resolution graphics (640x325 pixels).
- ☐ 128K byte memory, room for 512K bytes on main board.
- ☐ Fast-access 320K byte floppy, communication and printer ports and 4 expansion slots are included.
- ☐ RAM-disk software.
- ☐ Includes MS-DOS[1], GW BASIC[2], PC Tutor[3], and MultiMate[4] professional word processing system.
- ☐ Optional hard disk.
- ☐ Fully supported direct by AWA.

Corona PC's are available through AWA's continually expanding dealer network. For further information, ring AWA, or clip the coupon.

NSW (02) 922 3300
VIC (03) 529 4133
WA (09) 321 9334
SA (08) 79 9581
ACT (062) 47 7522
QLD (07) 369 3577
TAS (002) 34 4532
Auckland 76 0129

AWA COMPUTERS

To: Corona PC Co-ordinator, AWA Computers,
132 Arthur St, North Sydney, N.S.W. 2060
Yes! Please send me more information on the powerful Corona PC.
Name: ....
Position: ....
Company: ....
Address: ....
City: .... Postcode: .... Tel: ....

AWA Computers Division. Amalgamated Wireless (Australasia) Limited Incorporated in N.S.W.
IBM is a registered TM of International Business Machines. 1. TM Micropro. 2. TM Digital Research. 3. TM Comprehensive Software Support. 4. TM Software Systems.

ANOTHER HARD ACT TO FOLLOW!

Act's fixed removable disk systems give you MEGABYTES IN THE PALM OF YOUR HAND

COMPATABILITY

No matter which micro you have, you can make it ACT. The ACT fixed removable winchester is compatible with most micros — • APPLE I & IIE • COLUMBIA P.C. • CORONA/WORDPLEX • DEC RAINBOW 100 • HEATH/ZENITH H89/Z89 • HITACHI PEACH • IBM P.C. • KAYPRO • MICROBEE • MORROW MICRO-DECISION • NEC APC • NORTHSTAR ADVANTAGE & HORIZON • OSBORNE I & EXECUTIVE • SANYO MBC-1000 & 555 • SIGMA/OKI MODELS 20 & 30 • SIRIUS • SUPERBRAIN I/II • TANDY II & III • TIME OFFICE • TOSHIBA T-300 • XEROX 820 • PANASONIC JB-3001 • AND OTHERS.

A GREAT STORAGE SYSTEM

Combine the best of two storage systems — the speed and capacity of a hard disk and the mobility of a floppy.
You can backup the days data, remove the disk and store it safely or carry it interstate in your briefcase — a great storage system!

AUSTRALIAN

You don't have to pay more to buy Australian!
ACT is manufactured in Australia and supported by an Australian organisation committed to backup support and service. It's a professional ACT.

*A DIVISION OF AUSTRALIAN PROTECTION INDUSTRIES

act

Australian Computer and Telecommunications
75 Willoughby Road, Crows Nest, NSW 2065.
Telephone (02) 439 6300, Telex AA 24816

DISTRIBUTORS.

Direct Data, 1357 Malvern Road, Malvern, Victoria 3144. Telephone (03) 209 9045.
Computer Mart, 356A King William Street, Adelaide, South Australia 5000. Telephone (08) 51 5363.