Australian Personal Computer

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

JUNE 1985 $2.95

PC TO MAINFRAME LINKS: THE DEFINITIVE SERIES

AUSTRALIA'S TOP SE PUTER MAGAZINE

UNCLE JACK WANTS YOU!

Exclusive Benchtest: Tramiel's challenge to the Apple Macintosh

NOW WITH 80286 & Z80H

Universe

Security and speed
Software compatibility, and

Forget conventional networking. Universe provides superior speed and security necessary in multiuser applications. Running the widest range of 8 and 16 bit software, it has the ability to network IBM PCs and workalikes in the fastest multiuser/ networking microcomputer system in the world.

Multiuser – efficiency
A single Universe runs up to 25 workstations, each with any combination of 8 and 16 bit programs. Advanced AED network technology allows expansion to 100's of users.

Multitasking – productivity
Each operator can use any four 8 and 16 bit programs at the same time. Switching screens takes only a single keystroke.

Networking – flexibility
Up to 255 MS-DOS machines.
IBM PCs and workalikes can be linked into the Universe system using a high speed DR Net local area network.

IBM PCs and workalikes can run applications written for Concurrent PC DOS, CP/M-86 and PC-DOS, while having access to all the benefits of the network. PC users share files, records, printers and other network resources.

Software – compatibility
Dual processor design, provides access to the world's largest software base via CP/M, MP/M and MS-DOS. With AED's new Concurrent DOS you have the best of all worlds.

DMA hard discs and the new high-speed 80286/Z80H dual processor CPU furnish performance necessary to handle multiple 8 and 16 bit programs.

Tough
The Universe is built on a strong square tube frame.

Stays Cool
No fancy operating environment needed. Every Universe is tested at 42 degrees C.

Flexible
Universe accepts an extensive range of terminals, printers, modems, even electronic telex.

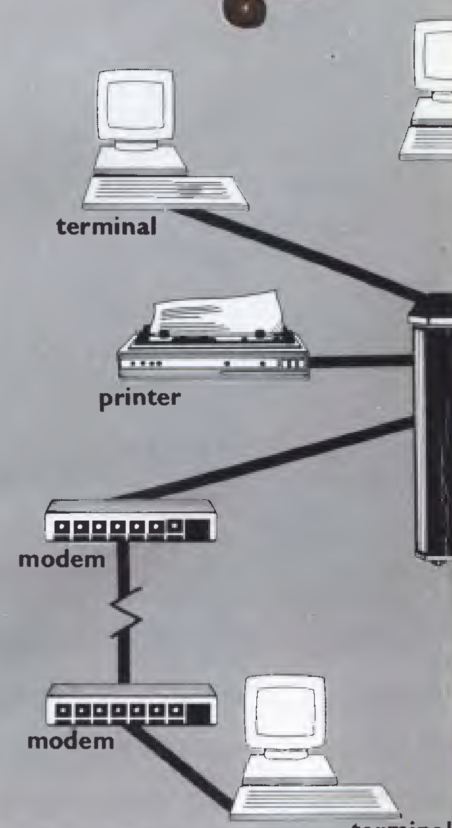
Expandable
20 slot shielded S100 buss. Obsolescence proof using IEEE 696 S100 cards.

Speed and Security – essential to your business
Most networks are slow and insecure. Universe shines here, with full multilevel security enhancements normally found on well engineered minicomputers. Universe is engineered from the ground up to provide facilities essential for the smooth running of a large multiuser system.

Important Security features
Encrypted login passwords.
Users are restricted to specific terminals, directory areas, programs and nodes on the network.

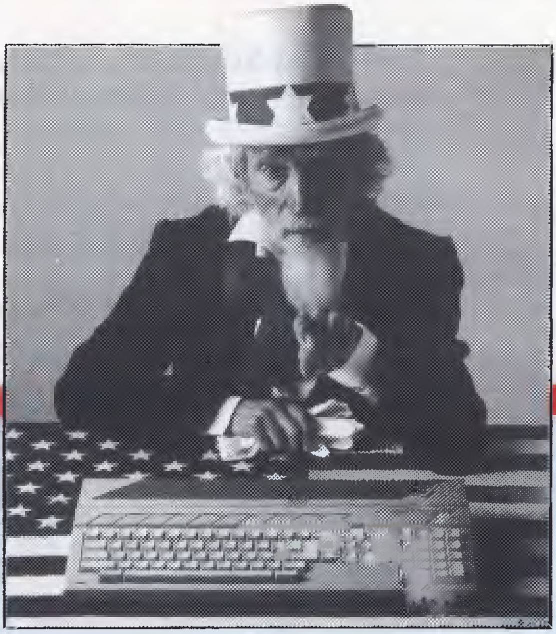
File passwords. File and record lockout and a full password hierarchy. Your System Manager can quickly and easily configure the system so that each terminal only has access to those facilities and data its operator needs. For example:

- ☐ Option to restrict any account to specific programs or workstations



APC

Volume 6, Number 6, June 1985



46

REGULARS

5 NEWSPRINT
Guy Kewney and other *APC* reporters bring you the latest news from around the micro world.

18 YANKEE DOODLES
The dreams and schemes of the American industry explored by our US correspondent.

68 BANKS' STATEMENT
Will the Unix boom ever arrive? Martin Banks ponders the fate of the operating system dubbed the programmers' friend and the users' foe.

117 TJ'S WORKSHOP
Hints and tips to help you make the most of your micro.

145 PROGRAMS
The pick of the programs — including utilities for Commodore 64 disk owners, logic puzzles for the Spectrum, an Amstrad disassembler, darts simulation for the Tandy Color and more.

165 BACK ISSUES
Trackdown the one that got away.

167 SUBSET
Machine code made easy.

170 LETTERS
Readers reveal the true history of bugs — plus our own contribution in Bludners.

195 NETWORKS
Peter Tootill and Steve Withers bring you the latest communications news plus what's happening on Microtex 666.

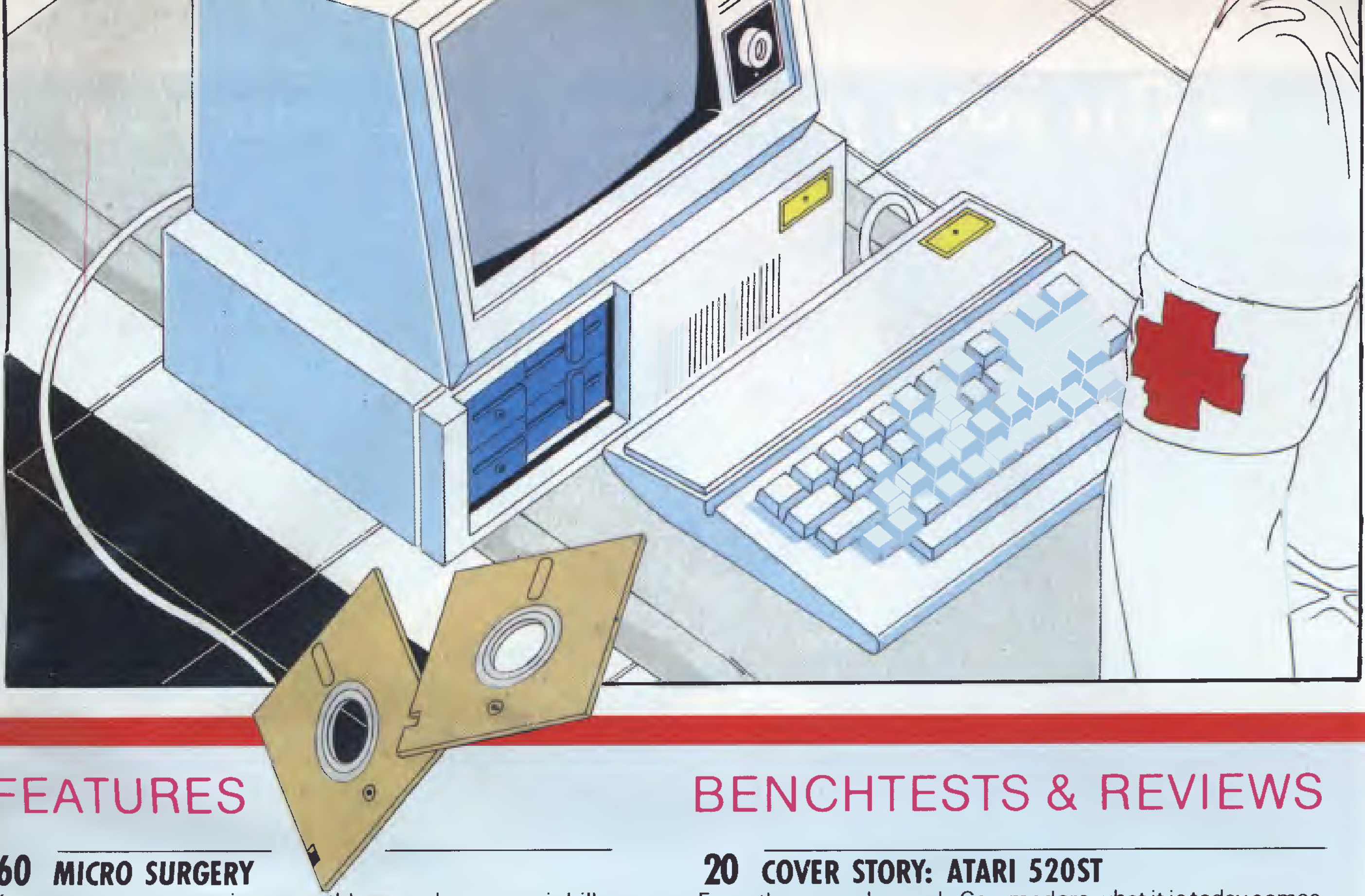
197 SUBSCRIPTIONS
Take *APC* for a three-month test drive.

201 END ZONE
Touchdown for Diary Data, Benchmarks, User Groups Index, Computer Chess, Numbers Count and Lazing Around.

207 CHIPCHAT
Micro magazines making readers confused, our antique-style cover explained and an exclusive review of the Bushlitt — where else could you read such amazing trivia?

208 ADVERTISERS' INDEX
Who's where in this issue.

Managing Editor Sean Howard; *Assistant Editor* Maria Bokic; *Consultant Editors* Steve Withers, Ian Davies; *Advertising Manager* Mark Reiss; *Sales Executive (NSW and Qld)* Simon Tindall (02) 264 1266; *Sales Executive (Vic, SA, WA and Tas)* Sean Carnell (03) 531 8411; *Graphics and Typesetting* Graphic Heart Pty Ltd; *Subscriptions Manager* Valerie Meagher; Subscription rates: Australia $35.00 per annum, Overseas A$47.00 (surface), A$130 (airmail). Printed by Quadricolor Industries Pty Ltd. Published by Computer Publications Pty Ltd, a subsidiary of Consolidated Press (Holdings) Ltd, 77 Glenhuntly Road, Elwood, Victoria 3184. Telephone (Editorial and Publishing): (03) 531 8411. Telex: AA 30333 AMJ. *Distributors* Network Distribution Co. 54 Park Street, Sydney 2000.



FEATURES

60 MICRO SURGERY
Keep your system running smoothly — and your repair bills in check.

84 PC-MAINFRAME LINKS
This in-depth series continues with details of managing the PC to mainframe connection including an overview of available products.

106 WIRED FOR SOUND
Take time out to explore the musical possibilities opened up by MIDI interfaces.

123 TURBO PASCAL
Warmly welcomed when it first appeared, Tanj Bennett finds out if this software package lives up to its reputation.

132 THE C LIBRARY
The functions essential to any good C library.

174 DATAGRAMMING
Spreadsheet techniques which the manuals forgot to mention.

181 EXPERT KNOWLEDGE
Start incorporating artificial intelligence into your own programs.

189 ON THE AIR
Brainwave or communications chaos? Adrian Morant and Steve Withers examine the possibilities for data transmission via cellular radio.

BENCHTESTS & REVIEWS

20 COVER STORY: ATARI 520ST
From the man who made Commodore what it is today comes the Atari 520ST. Peter Bright is first to find out if the machine matches its high specifications.

37 BROTHER WP600
Is it a typewriter, text editor, printer or terminal? How many identities can one peripheral have? Simon Craven takes a closer look.

42 MICROSCRIPT
Word processors for home machines are becoming increasingly sophisticated. This package for AWA's Amstrad is one of the latest attempts to combine power with ease of use.

46 TEXAS INSTRUMENTS PROLITE
In the lap of the god? Yes, it's IBM again, this time being hotly pursued by TI's portable following the path beaten by the PC. Colin Baker joins the chase.

55 GOOD LOOKING
Digital Research's Presentation Master tries to combine film and PC to produce high quality graphics at low cost. Geoff Wheelwright tests its success.

73 MICROSOFT WORD
First Mac question: is the Macintosh a real business machine? Kathy Lang finds out by assessing its merits as a word processor in this evaluation of Word.
For an answer to the second Mac question (How much easier does it make business?) look out for next month's test of Microsoft File.

On your personal computer..

is Framework child's play?

Remember when you were young, the games you used to play, the expanding creativity, the freedom?

And today? Do you still practice creative problem solving. Are you looking for a computer solution that multiplies thought as easily as it multiplies numbers.

Then consider Framework. Framework's outlining feature lets you develop your ideas in the order you choose. Then lets you move them around to make the strongest possible argument.

Nowadays, words often seem to need "financial justifications." So your spreadsheet is completely integrated with the word processor.

Two other features help you make your point even stronger. An integrated database to hold all the facts, and a graphing function to illustrate your message.

Framework from Ashton-Tate is completely integrated software — outlining, word processing, spreadsheet, graphing, database, and data communications.

The Australian Software Guide said —

"... Framework is an impressive product, and pleasing to use. It allows you to create, modify, organize and analyse information, and to present it in a professional way. Its strengths lie in its text management, its ease of use, and the way it allows you to organise interrelated information".

So, is Framework child's play?

Find out for yourself by asking for a demonstration at your local computer dealer. Or complete the coupon below and we'll send you the demonstration disks.

A special offer.

If you buy Framework between May 31 and June 29, 1985 we'll send you a complete Framework reference library of Ashton-Tate books worth more than $200. We've listed the titles below:

Framework: An Introduction

Framework: On-The-Job Applications

Framework: A Programmer's Reference

Framework: A Developer's Handbook.

Framework: An Introduction to Programming

ARCOM PACIFIC ■ 252 Abbotsford Road ■ Mayne, Brisbane Q 4006 ■ Fax (07) 52 7820 ■ Telex AA 44187 ■ Phone (07) 52 9522 ■
Brisbane ■ Sydney ■ Melbourne ■ New Zealand

I would like to discover more about Framework. Please send me the demo disks.
I understand I need an IBM PC, XT, AT or close compatible with at least 384K RAM to run this software.

Name .... Address ....

Send to Freepost 2 Arcom Pacific P.O. Box 13 Clayfield Qld 4011. (No stamp required)

F/W FPC

but can now be ignored by using ON BREAK CONT. Once activated, there is no way to return to your program other than cold starting and reloading.

Included with the system is a disk containing CP/M and DR Logo. There are some problems with CP/M due to the small size of the transient program area (the memory available for application programs) which prevents certain CP/M applications from running. The second problem is that most CP/M programs are only available on 5in disk and not on the new 3in format. Having said that. there is a lot of software available on disk from Amstrad, and you can probably find a commercial company willing to transfer from 5in to 3in disks. AWA intends to make all Amstrad software available on disk, including small business software such as word processors and stock control systems.

The CPC 664 will run all software designed for the CPC 464. For 464 owners, the addition of an external disk drive makes the machine very nearly a 664: certainly, all commercial software will be written for both machines. A few Basic programs using the new commands will not work, but most commands can be simulated by multiple Basic commands or, at worst, a call to one of the ROM routines. 464 owners who still feel they want to upgrade to the new ROM will be disappointed: Amstrad does not intend to sell an upgrade for the 464.

The CPC 664 system includes a hefty manual which isn't just the 464's manual and the disk drive manual thrown together. Although very similar, Amstrad has taken the opportunity to clarify some points from the old manual, such as the use of multiple colour planes to create sprite-like graphics, as well as explain the use of the new commands. The Amstrad documentation is excellent, although it tends to lean a little towards the reference side and needs to be complemented by a suitable book for out-and-out beginners.

By far the best feature of the Amstrad CPC 664 is its price. With a monochrome monitor the system costs approximately $800, a colour monitor version will set you back approximately $1,000. With built-in disk drive, CP/M and an 80-column screen, it represents excellent value for a small business that requires something cheap, or for serious home use. Compared with another British machine, the QL, at $1,095 with Microdrives and no monitor, it certainly seems good value. Admittedly the technology may not be the latest all-singing all-dancing wonder (a Z80 8-bit processor and CP/M) but it will be available early July, while it's still a case of estimating the time of arrival of other more sophisticated machines.

Guy Kewney

The proof's in the running

Minicomputer-builder DEC has a bone to pick with micromaker Intel: apparently, Intel has been telling fibs about how fast its micros are, compared with DEC's minis.

In 1981, an American magazine *(Byte)* supposedly printed a Benchmark which Intel picked to illustrate the power of its System 86/330. When Intel ran the Benchmarks in 1982, it said that its system was 'clearly superior to the LSI-11 on this Benchmark'. The LSI-11 is DEC's micro version of its PDO mini.

Digital has now run the Benchmark and finds that 'on the contrary, just the opposite is true — both the LSI-11/2 and 11/23 executed the Benchmark faster and required less memory than the System 86/330'.

All good fun. What I found fascinating was the fact that this (Pascal) Benchmark was tried out on 20 different combinations of machine and compiler; the fastest being the PDP-11/70, a powerful mini, with NBS Pascal. That took 2.6 seconds, compared with the Intel system's 9.20 seconds the best 8-bit system was a Z80 with MT + Pascal, taking 19 seconds.

No that isn't the amusing bit. The amusing bit is the time taken by the Apple II with UCSD Pascal. It took 516 seconds . . .

Guy Kewney

Barson Computers has begun receiving international orders for a network interface which enables BBC Electrons to be attached to Econet networks.

Developed in Melbourne, the interface immediately opens up networking capability for many smaller schools around the world. The product prototype was shown to other BBC distributors in Ireland, South Africa, the UK and New Zealand earlier this month by Julian Barson, managing director of Barson Computers.

"In the past week Barson Computers has received orders for more than 200 units a month, with the first production units not required in Australia being shipped to UK, New Zealand and Ireland, said Mr Barson.

"I was slightly amazed at the international reaction to the product, as we had not really considered overseas markets during the product's conception," Mr Barson said.

"I suppose if we had planned it ahead you would call it a classic case of niche marketing", Mr Barson said.

These are some of our compatibles

In fact Open Access is compatible with more than 25 of the worlds most popular microcomputers.

Open Access is the popular integrated do-it-all super program that can perform virtually every task you're ever likely to encounter.

Open Access includes an electronic spreadsheet, 3D business graphics, word processor, appointment scheduler and telecommunications module, all revolving around the powerful information manager.

Open Access operates on the following MS/PC DOS microcomputers: AT&T, Challenger, Chamellion, Columbia, Compaq, Corona, Datamax, Data General, Digital Rainbow, Eagle, Emtek, Hewlett-Packard, Hyperion, IBM, ITT Extra, Monroe, NCR, NEC, Olivetti, Pantek, Philips, President, Sigma, Sperry, Tandy, TeleVideo, Toshiba, Texas Instruments, Wang, Zenith.

O·P·E·N ACCESS

Produced, distributed and supported in Australia by: SOFTWARE SUPPLIERS

7 Avon Road, North Ryde, NSW 2113.
Telephone (02) 888 1955. Telex AA 75364

Enter Sybiz Plus

A sister product to Australia's Sybiz integrated accounting software package for MS-DOS based machines has been launched.

Known as Sybiz Plus, the new package offers new features over the original Sybiz, which will continue to be available.

Sybiz Plus has a general ledger module which adds facilities such as standing journals and monthly budgets. It now has the ability to process transactions and print reports in prior months.

The trading, profit and loss statement and balance sheet show management information, including the comparison of this year's results with the previous year.

A useful addition is the 'open window' mode. It is now possible to search for and look at any account in the data base, at any time, without interrupting the work by using a 'view' function key.

It also allows opening of a window to use a four-function arithmetic calculator, the result being transferred for input into the program.

For the extra Sybiz Plus modules, the cost will be $795 per main module, $100 more than Sybiz. Ancillary modules will cost $345 each.

For further details contact: (02) 957 6838.

Where Unix goes, others follow

It's not a bandwagon I feel like jumping on, but the Unix one certainly looks like it's beginning to roll.

The business industry's big guns, IBM and AT&T, neither of whom get all their shots on target, are both on there — plus a host of other companies including Hewlett-Packard (whose integral PC is reviewed in July's issue) and Commodore, which has been receiving some bad publicity recently.

Systems house Digitus, for one, is convinced that the operating system's time has come. It's published a report on the Unix market which says that 'with sales doubling in 1984 and 180 per cent growth predicted by 1986, it is set to become one of the most dominant multi-user systems'. However the report doesn't expect the shortage of IBM PC/ATs to be corrected until the end of the year, and it's even less optimistic about multi-user Xenix. This AT implementation of the Microsoft Unix lookalike isn't expected until next year.

To make life even more confusing, Commodore's contender is based on another Unix clone, Coherent. The Commodore machine should come in two versions, both called the 900. There's a personal workstation with a very high-resolution (1024x800) bit-mapped monitor coupled with Commodore window manager software and mouse control, or alternatively you can plump for the multi-user option with a 67Mbyte hard disk from which you can hang up to eight terminals. Both versions come with 512k of RAM as standard, and are based on an obscure Zilog 16-bit processor, and Z-8001. Coherent at the moment is compatible with System Seven Unix, but will in time be made compatible with the more widely avail-

Two machines of interest here, from Toshiba. On the left is the T1100 which sells for $2,995; it's based on a CMOS (low battery consumption) 8088 processor, and comes with 256k of RAM and a 720k capacity 3½ inch disk drive. The display is the desirable 25 line by 80 characters. A built-in modem is not included.

On the right is the T1500, an IBM PC workalike again with an 8088 processor and between 128k and 640k of RAM. Options include a 10Mb hard disk (which replaces one of the floppies) and a 25 line, 80 character LCD screen. Though why someone would pay the steep premium for the compact LCD screen (presumably to aid portability), when the rest of the machine is the typical size is beyond us.

able System Five, says Commodore.

The best I could get in the way of a price is 'approximately half the price of its nearest rival' or 'very competitive', which could mean anything between $4,000 and $12,000. Delivery starts towards the end of 1985, although software developers are getting the machines now.

For me, optimism based on Unix is misplaced optimism, but I'm pleased to be able to record it so that we can look back with hindsight in a year's time. AT&T is placing its future trust on its 7300 personal micro, which looks like a Macintosh but runs Unix software.

That machine uses the 68000 processor, a close relative to the one inside the Macintosh, but don't run away with the idea that programs will need relatively little conversion from Mac to Bell.

Most programs written for the Mac don't address the chip but use the high-level routines built into the Macintosh by Apple. Unless AT&T produces an emulator of the window manager, there is going to be a whole host of nothing to run on the 73000 for a long time. Except, of course, for the possibility of converting Xenix applications — and despite the theory that Xenix is very like Unix, that won't be the work of a couple of weeks, either.

Guy Kewney

CD ROM is here!

It looks as though compact disk ROMs are finally arriving with the announcement by Hitachi of the CDR-1502S — a CD-ROM with parallel interface for the IBM PC. Memory is not quite up to the one giga-byte first speculated, with the Hitachi drives storing 522Mbytes which is still about 270,000 A4 pages. Interfaces for other machines are expected later this year. Happily it is using the same mechanism as the music compact disks so we can expect the price to drop as music CDs drop and the disks themselves should be relatively abundant.

The CDR-1502S is expected to sell for just under $1,000 which is twice the price of its audio counterpart, apparently the addition of an extra chip will allow it to be used as an audio CD player which makes it sound more reasonable. A number of computer manufacturers have expressed an interest in it with a view to replacing their bulky instruction manuals with a single disk. In addition it seems as if the race is on between the CD-ROM manufacturers to sign a deal with Encyclopaedia Britannica to be the first to offer this on CD.

Guy Kewney

Creating a precedent

Commodore is offering a $100 trade-in on any old calculator, typewriter or computer to purchasers of the '64 Family Pack or SX-64 during the month of June. But don't think of being nasty and lobbing a ten year old broken home-made micro Commodore's way because the electronic mish-mash that's likely to be created by this offer is to be donated to the Smith Family for underprivileged families throughout Australia.

Managing director of Commodore in Australia, Nigel Shepherd, says "the company has been so successful it can afford to make the offer", though, considering its parent company in the US has just posted a $US20 million loss for the first quarter of '85 it's probably no wonder Nigel continues "but it has to be for a strictly limited period".

No doubt we'll get the regular censure call from Commodore for this little dig, but when you're a tall poppy everyone is going to have a go at you (we know, sigh, after all look at *APC* . . .). Congratulations, though, to Commodore for donating the goods to the Smith Family.

ICL releases "FOLIO" Homebanking package

ICL Australia has released a videotext Homebanking software package called FOLIO. The system was jointly developed by the government-owned Development Finance Corporation of New Zealand (DFC) and ICL.

ICL anticipates that as is the case with electronic funds transfer and point of sale systems, most financial institutions will need to offer Homebanking to their customers in the very near future. "Research in the United States suggests that by the mid 1990s, Homebanking will account for up to 70% of all banking transactions. Homebanking will appeal to a wide cross-section of bank customers, preferring the convenience of managing their personal finances from their home or office, and withdrawing cash from automatic teller machines outside regular banking hours."

One of ICL Australia's marketing strategies is to offer FOLIO as a pilot Homebanking service which requires little investment up front but offers the opportuntity for Banks, Building Societies and other financial institutions to "test the water" before committing to a full Homebanking facility. So expect to see FOLIO on Viatel soon.

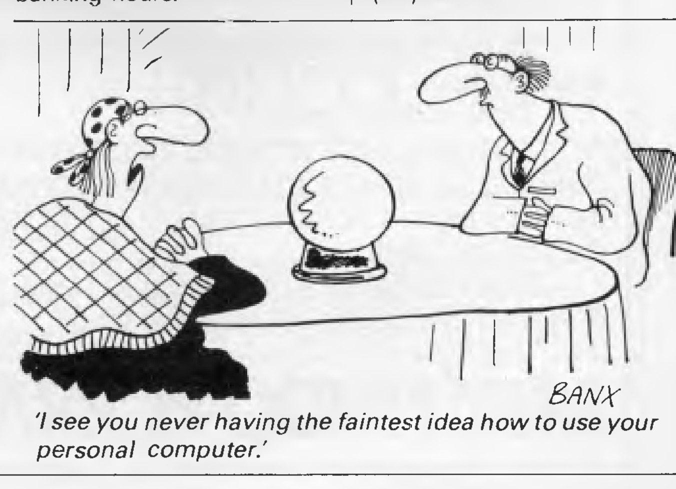
Relational database for Mac

Software Corporation of Australia has signed an exclusive distribution agreement with Blyth Software Inc. of California, for the OMNIS series of relational database software for the Apple II and Macintosh.

"Good data file management software for the Mac has been available for some time, though few programs have the power and flexibility of OMNIS 2," said Arnold Roth of SCA. "But serious users of the Mac have been waiting for a powerful, easy to use applications generator and hierarchical/relational database manager akin to the dBase concept. OMNIS 3 is the answer."

"OMNIS 2 and OMNIS 3 are both designed to take advantage of the power of the Mac's unique user interface.

For further details call (03) 347 7074.



'I see you never having the faintest idea how to use your personal computer.'

It does everything the most pow

The problem with most desktop computers is simply that.

They're desk bound.

And the trouble with most transportable computers is, apart from being under-powered, they look more at home in the boot of your car than on your desk.

Now fortunately, a computer has been designed to fit neatly between the two.

The Apricot.

In a nutshell, it's a 16-bit desktop computer that folds into a briefcase.

So it can take home work when you do.

While at the office, it competes on equal footing with the desktop heavies.

Unbelievably, it's more powerful than the equivalent IBM PC.

It also incorporates a few practical advances in computer technology that no other desktop has caught up with.

For example, a Microscreen on the keyboard which operates as a calculator, digital calendar, time display and as a window on the main screen.

What's more, the Apricot comes ready to go.

*From $4444 with sales tax and $1000 worth of software included in the price. And the software includes Supercalc,

Why Compaq must try harder to stay No. 2

Mark Ivey in Houston in the US finds out . . .

It's too early in the football season to predict if last year's winner will come out on top again. In the personal computer major leagues, however, things are a little different. It's a new season all right, but everybody knows International Business Machines Corp. will take the series. The real contest is the race for No. 2.

The 1984 runner-up, Compaq Computer Corp., finished the year with $329 million in revenues and profits of $12.86 million. What's more, in 1985's generally troubled first quarter its earnings were up 42%. That's impressive performance, considering that Compaq is just two years out of its rookie season. Right now, the Houston company has a market value of $230 million and is the second largest producer of office personal computers. Competitors like American Telephone & Telegraph Co. and ITT Corp. are languishing, and analysts say that only IBM commands a stronger following among corporate buyers.

Building on this success won't be easy. Compaq's future depends on selling look-alike computers that compete with IBM's powerful new PC/AT. On April 29, with a dazzling season opener that included a concert by the Pointer Sisters, Compaq unveiled its PC/AT clones — two portables and two desktop models. The company is betting that this new line will cement its second-place position. But whether Compaq has the financial and marketing muscle needed for this tougher market remains to be seen.

Higher stakes. The biggest change in Compaq's new market is the quality of the competition. When the company seized the lead in the IBM PC-compatible field two years ago, its rivals were mostly inexperienced startups. Certainly there were a few established players, such as Digital Equipment Corp. and Hewlett-Packard Co., but they misjudged the need to be truly IBM-compatible and were eliminated early on.

Things are very different in the AT game. This time the startups have mostly disappeared, and the larger computer makers are coming on strong. The stakes are higher, too, since the AT market is more important to big companies than the PC business. The original PC generally served just one user. The more powerful AT can work as a common processor for as many as three people; it can also be the central data clearinghouse for a network of personal computers. That power is particularly threatening to the minicomputer makers, whose machines have been at the hub of office systems for years.

Those old-line companies are moving to guard their flanks. Texas Instruments Inc. introduced its AT-compatible in March, and NCR Corp. launched its AT clone on April 30. Analysts expect Hewlett-Packard, Wang Laboratories, and AT&T to follow. "By yearend," says Norm DeWitt, analyst with Dataquest Inc., "It could become a very crowded market."

But crowding isn't the only problem. Compaq must hurry to establish its AT knockoffs, because the "window" into the IBM-compatible market is much smaller this time. After introducing its PC in 1981, IBM took nearly three years to catch up with demand, giving Compaq time to build its reputation.

'Closing window'. AT delivery schedules stretched to nine months, because of technical problems on the hard-disk version. But now, IBM is revving up for full production, and dealers expect the shipping delays to disappear as early as July. This would leave Compaq making its entrance just as supplies swell. Says Dave McDonough, a dealer for ComputerLand of San Diego Inc.: "The window is closing fast."

Compaq executives are keenly aware of all this. They know they must move quickly, but they also think they have a unique advantage: dealer loyalty. "We'll see a lot of AT (compatibles) come out," says Rodney Canion, Compaq's 40-year-old president. "But we're the only ones with the shelf space. And that's where the battle will be won."

Those loyal dealers — some 1,700 in the US — are a powerful asset. Compaq has selected them carefully and wooed them with generous discounts and liberal return policies. They respond accordingly. Last summer, when sales were slumping industrywide and sceptics were wondering about Compaq's future, good distribution saved the day. Retailers helped make the company's Deskpro desktop computer — a faster version of the IBM PC — a hit. "We'd have to really make a big mistake to be dislodged (from the retail channel)," Canion maintains.

To keep its dealers happy, Compaq will soon launch a generous advertising rebate program. Unlike standard co-op advertising plans, it will allow dealers to choose where money is spent. "Without a doubt," says Warren Winger, chairman of CompuShop in Dallas, "they have established themselves as the retailers' company."

But some question whether even loyal dealers can make the AT business safe enough for Compaq. Enzo N Torresi, senior vice-president of the Businessland computer retail chain, claims the AT design will make it more difficult to make the old formula work again. Before, Compaq duplicated IBM's PC, added extra features, and then sold its machine at the same price. But this time, "they will have to compete on price," says Torresi. And when that happens, he warns, "IBM always wins."

Canion disagrees: "We just never accepted the folk wisdom that you have to sell below IBM." Compaq, he explains, wants to offer products at close to IBM prices and to give buyers more for their money. One AT-compatible desktop model, for example, sells for $6,254, only about $75 less than a comparable IBM model. But Compaq offers 30Mb of disk storage — 10Mb more than IBM. And the Compaq machine is 30% faster. A similarly equipped AT-compatible portable — something IBM doesn't make —is $6,299.

No strikeouts. With the higher-priced AT-compatibles and a major investment in efficient manufacturing, Compaq hopes to double pretax margins, which were 5% last year. But this may not be possible if as many as 20 AT-clone vendors start driving prices down. InfoCorp predicts that the total market for ATs and AT clones will reach 800,000 units in 1985. It estimates that IBM will win a 70% share, leaving Compaq as the second-largest supplier, with about 5%. That amounts to annual revenues of well over $100 million for Compaq.

Sceptics, however, worry that there aren't enough "power users". These are customers with the need for so much number-crunching capacity that they will spend $6,000 and up to get the storage and speed of an AT. Notes Torresi: "The AT is a product that needs some selling. If you wait passively for a huge demand to build up, you'll be burned."

Even if Compaq's AT does get off to a quick start, the company faces an even tougher battle. Soon, IBM will introduce its PC2, a replacement for the old PC. That hits Compaq where it hurts, making its original portables obsolete. These products still account for the bulk of the company's revenues. To stay competitive, the retail price of Compaq machines would have to drop by 25% to $1,500, says InfoCorp analyst Ralph A Gilman.

Even though that would clobber margins, no one is counting Compaq out. The company's record shows that it can beat the odds, and a recent $75 million debenture offering also gives Canion considerable balancesheet flexibility. "We've distinguished ourselves by not having any strikeouts," he concludes. "And we don't plan on ruining that record." Succeeding in the AT market, however, would amount to more than just a home run. Finishing second to IBM twice would be the computer industry's equivalent of a grand slam.

... SORRY CHARLIE FOR LEAVING YOU OUT!

TAVA PC

Features Include:

- **16 BIT CPU 256K RAM**
- **4.77/8 MHZ SWITCHABLE**
- **DUAL 360K SLIMLINE DISK DRIVES**
- **COLOUR CARD**
- **SERIAL PORT**
- **PARALLEL PORT**
- **10 MEGABYTE HARD DISK (OPTIONAL)**

And More!

TAVA PC — 99.4% Compatible System

(PC WORLD USA APRIL 1984)

The TAVA PC™ is superior to the IBM PC™ at a lower cost. That's why we can give a 12 month warranty on parts, 120 days on Labour. TAVA PC™ comes standard with parallel and serial ports (extra on IBM); is completely Hardware and Software compatible with the IBM PC;™ can easily upgrade with an 8087; Hard disk sub system, multifunction Memory Boards, Multiuser capability; 3270 networking ability and much more.

SOFTWARE COMPATIBILITY

PC World April 1984 ranked TAVA PC™ in the select group rated as 99 44/100% compatible.

Runs LOTUS 123, Symphony, D Base II, D Base III, Wordstar, Spellstar, Mailmerge, Perfect Writer, Speller/Calc/Filer/Link Easywriter, Volkswriter, Visicalc, Multiplan, Super Calc, PFS File/Report, Peachtree and more. You can use operating systems such as PC DOS, MS DOS, CP/M, 86, UCSD, P System.

SUMMARY

Compare TAVA PC™ to any other computer. Compare the design, execution, quality and warranty. Compare versatility and expandibility. Compare price. You'll come to the same conclusion thousands of others have — THE TAVA PC.™

Distributed by:

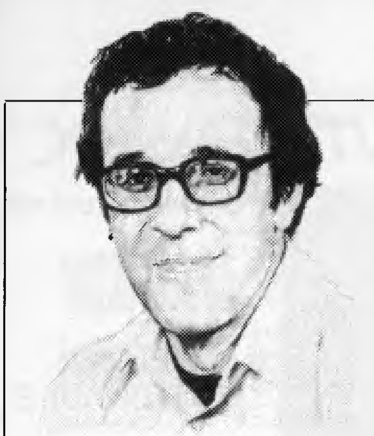
interfaceware PTY. LTD.
P.O. BOX 214, LINDFIELD, NSW 2070
Ph: (02) 46 4374

TAVA PC

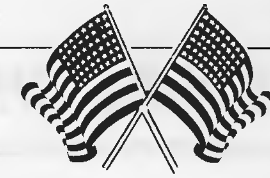
*IBM PC is a registered trademark of IBM Corp.

PLEASE SEND ME THE LATEST INFORMATION ON TAVA PC™

Name ....
Address ....
Post Code ....
Phone ....



YANKEE DOODLES



When you can't see the shelves for the software, there must be another way of selling. Our US correspondent reports from the States on electronic distribution and the latest American news.

Away from home?

Although some industry watchers have interpreted IBM's decision to stop producing the PCjr as the death knell for the home market, I disagree. The PCjr never completely recovered from its initial introduction in the US with a Chiclet-style keyboard and limited expansion capabilities, although the deep price cuts for a bundled system before Christmas gave it a short-term boost.

If anything, IBM's exit provides some opportunities for Apple and Tandy who currently market full-featured systems in the under-$1,000 price range. Later this year when the new Commodore and Atari computers hit the shelves, the competition will be tough but the time window seems to be wide open for Apple and Tandy for the next few months.

Is IBM out of the home market for good? Not likely. It is reportedly still looking at building an MSX machine at a very attractive price, and you can be sure that when IBM sees significant profits in the market, it will jump back in with both feet.

Meanwhile, the bundled PCjr price of late 1984 prompted Apple into responding with a cash rebate program on the Apple II. Although the program ended on 30 April, it still caused a (needless) loss of revenue for Apple.

The hard sell

With over 27,000 personal computer software packages on the market, it's impossible for a computer store to carry more than about 1 per cent of the total offerings. Even the largest software distributors, Softsel and First Software, carry less than 20 per cent. Thus, software manufacturers are turning to a variety of other methods of selling their wares.

Some vendors saw electronic distribution as the answer, but it has not proved commercially successfully. Nevertheless, General Electric, AT&T and several new companies are continuing to experiment with such systems, and are convinced they will be in widespread use by the end of the decade.

An alternative to direct electronic distribution is provided by One Point, a distributor that maintains an electronic software catalogue of some 7,000 titles. A customer can dial up the catalogue from his home computer or one in a retail store, and receive product descriptions and reviews of packages. If he is interested, he can place an order which will be shipped within 48 hours.

Unimart is trying an approach which allows a customer to dial up a software package and try it out with his own data. If he likes it, he can buy it.

Several software vendors have turned to outlets other than the traditional computer or software store. Intuit, for example, sells its home finance package through banks. Several vendors sell packages through trade associations, and at least one vendor sells its package through home builders who include a computer and a home management software system in every new home that they build.

Below target

Kaypro, with its cheap but functional computers and direct distribution to retailers, was riding high a year ago while companies with more advanced products and 'proven' distribution schemes were hurting.

However, an antitrust complaint was filed against Kaypro in March for threatening some of its dealers with termination for not selling at list prices. Kaypro paid $19,500 in civil penalties and court costs to settle the suit and, although the company did not admit or deny the allegations, it would appear that it will now have to give its dealers greater latitude of action in setting prices, selling to non-Kaypro dealers, and advertising mail order sales.

Dealers have also expressed scepticism that Kaypro has the ability to support its recently-announced 286i, a high-end IBM PC/AT clone, or even the K-16, an XT-type entry. The 80286-based 286i with 512k and dual floppy disk drives sells for $4,550, while a 256k, 10Mbyte hard disk version of the K-16 sells for $3,295. All Kaypro systems include bundled software. Although the Kaypro prices are 18-25 per cent under the cost of similar IBM units, the IBM systems are frequently discounted by approximately 20 per cent. Therefore, the only advantage of the Kaypro is the bundled software.

Several dealers to whom I've spoken feel that the extra software — said to be worth $2,000 by a Kaypro spokesman — is not enough. After all, Columbia, Eagle and TeleVideo also had a price advantage and were largely unsuccessful against IBM. Most dealers agree that while many customers are looking for an alternative to IBM, the three magical letters — I, B and M — are still the most sought. One said: 'I hope Kaypro can do in the AT world what Compaq did in the PC world, but they're going to have a tough, tough time.'

Random bits

The Software Publishers Association reports that sales of Macintosh software have jumped from nil to 8 per cent in early 1985 . . . In March, Compaq shipped its 200,000th personal computer . . . IDC predicts that the market for business graphics will grow from $59 million in 1984 to over $1 billion in 1989 . . . Consumer Products, a maverick division of giant AT&T, has released an image capture board (for the IBM PC) which captures a standard composite video image, and allows modification and manipulation of it by the computer. It is made possible by the development of a new design architecture using RARAM memory, a high-density, low cost, two-port dynamic memory with a very fast access time . . . Having discontinued its 16/8 and 820 family of computers in February, Xerox is negotiating with Olivetti to sell the M-24 IBM-compatible unit . . . Lotus and Cullinet have joined forces to develop and market products to connect Lotus 1-2-3 and Symphony to powerful IBM (and compatible) mainframe computers. One catch: while the integrated package, Symphony Link, is expected to cost only $300 to $500, customers will have to buy a communications peripheral for each PC (about $1,100 each) plus Cullinet's mainframe Information Centre Management System package for a cool $150,000 . . . It's no secret that videotex has been a colossal failure in the US. The operators of three experimental systems (Viewtron, Keyfax, and Gateway) spent a total of $90 million in development yet attracted fewer than 5,000 subscribers in total. Nevertheless, even bigger players are planning to enter the business, specifically three joint ventures: one between IBM, CBS and Sears; a second between AT&T, Time Inc, Bank of America and Chemical Bank; and a third between RCA, Citicorp and an unnamed partner.

ACME SOFTWARE FULFILLS ALL YOUR COMMUNICATIONS REQUIREMENTS IN ONE PACKAGE. WITH THE MICROMODEM III AND 64 TALK

MICROMODEM III **only $299**

- 300/300 and 1200/75 Baud
- Connects directly to Commodore 64 (or VIC20) RS232 Adaptor available.
- Latest LSI design for reliable communications.
- Australian designed and manufactured.
- Telecom Approved — Approval No. C85/37/1261
- FREE Trial subscription to MICROTEX 666 and S.T.A.R.S. databases.

64 TALK **$79.95**

- Plug in cartridge for instant and easy use.
- VIATEL and TERMINAL modes.
- Download software (including Microtex 666) to tape or disk and transfer to other users.
- Full control over Terminal characteristics including 40 and 80 column display.
- Save and load screens from tape or disk
- Print screens or workspace. Select high resolution or ASCII printer routines.
- Save workspace and environments.
- And more . . .

ACME SOFTWARE

AVAILABLE FROM . . .

N.S.W. Computer Spot Sydney (02) 235 2455 Parramatta (02) 635 6020
Computer Wave Sydney (02) 29 1631
Vic. High Technology Richmond (03) 429 1966 Brighton (03) 596 6211
A.C.T. Steves Communications Fyshwick (062) 80 4339
S.A. I.D.N. Computer Systems Mile End (08) 352 7573
CYBEX Computing 121 Tynte St Nth Adelaide
W.A. Bruining Headlam Leederville (09) 381 2988 Backgammon Shop City Arcade (09) 321 2932
Garden City (09) 364 9430 Carousel (09) 458 8846 Waraboona (09) 349 9591 Belmont (09) 478 1115
Northern Computers Balcatta (09) 344 7120

ACME SOFTWARE
PO Box 3, Brighton North, Victoria, 3186. Telephone: (03) 596 6732.

Atari 520ST

The Atari 520ST (Jackintosh) is Jack Tramiel's answer to Apple's Macintosh. It is based on the powerful 68000 processor with user-friendly GEM, lots of I/O facilities, large RAM and a low price tag. Peter Bright finds out if this is really the all-singing, all-dancing machine it claims to be.

The 520ST is well-endowed with I/O facilities — including a MIDI synthesiser port

Maybe I'm getting old. Then again maybe the micro world is just becoming a little boring, but there are very few new micros that I get excited about any more. The last one was the Sinclair QL and look what happened to that.

At the beginning of the year I went to the CES show in the US and saw a machine worth getting excited about — the Atari 520ST. At the time the machine was announced there were two models — the 130ST and the 520ST. Both based on the ultra-powerful Motorola 68000 processor, both offering the user friendly GEM user interface in ROM, lots of I/O facilities including both floppy and hard disk interfaces, and either 128k of RAM (130ST) or 512k of RAM (520ST). All this was promised at an unheard of low price level.

We have been trying to get a review machine ever since. Unfortunately even review machines are in short supply and tend to move around a great deal. Just when we thought we had one, we were told 'sorry it's gone to Hanover in Germany'. So if Mohammed won't go to the mountain . . .

When I arrived at the Hanover trade show I was told that Atari had decided to drop the 128k 130ST and concentrate instead on the 512k 520ST. That was fine because that was the machine I wanted to look at anyway. So here I am sitting on the grass next to the fountains at the Hanover show typing my thoughts on the 520ST into a trusty Hewlett Packard 110. It's a hard life!

Hardware

The Atari 520ST is a surprisingly good looking beast. Jack Tramiel's previous machines haven't been renowned for their design content or overwhelming good looks. The 520ST, however, comes in a sleek mid-grey casing with matched mouse, disk drive and a range of monitors.

The keyboard takes up most of the front of the machine with ventilation slots using up the rest of the available top surface space. The design shows neat touches throughout; for example, the way the function keys are sloped to match the ventilation slots. The overall effect is very good. Apart from the keyboard and the ventilation slots the only other objects of note on the top of the unit are a small red LED power indicator and the Atari badge.

If there is one thing the 520ST is not short of, it is I/O ports. They take up the whole of the back of the machine as well as part of each side.

The left side houses a ROM-pack expansion slot which is capable of handling up to 128k of ROM. The right side houses two joystick ports, one of which

The keyboard is well laid out and feels nice. The function keys are styled to blend with the cooling slots

THE MICROTEX 666 V

FOR MORE MICROTEX 666 INFORMATION SEE PAGE 34

CASSETTE OR DISK

TELECOM APPROVED
Microtex 666 modem by Sendata.
Telecom Approval No:
C83/37/1045.

THIS AUSTRALIAN PRODUCED PACKAGE INCLUDES EVERYTHING YOU NEED TO ACCESS VIATEL AND MICROTEX 666.

JUST LOOK!

- TOP QUALITY MODEM UNIT
- COMMUNICATIONS SOFTWARE
- FREE SUBSCRIPTION TO MICROTEX 666
- VIATEL SUBSCRIPTION FORM
- EXCLUSIVE ACCESS TO MICROTEX CLUB 64
- PLUS, A CHANCE TO WIN **$1,000** WORTH OF COMPUTER EQUIPMENT OF YOUR CHOICE.

STOP PRESS

WIN $1000

worth of computer equipment of your choice from OziSoft.

Ask your dealer for a free colour catalogue. And remember, this opportunity is only open until the one thousandth Microtex 666 Viatel package has been sold. So hurry, send the coupon now!

WARNING

This package is supplied by Microtex 666. Beware, some other videotex packages fail to provide a Microtex 666 download facility for programs . . . so you won't be able to download the hundreds of Microtex 666 programs.

SEE PAGE 49 FOR COUPON

is usually used as a mouse port. The back panel houses: power-in, MIDI-in, MIDI-out, RS232 serial-port, Centronics parallel-port, TV video-out, RGB video-out, composite video-out, floppy-disk port and Winchester port. That's quite a long list of I/O ports for what is, after all, a comparatively cheap machine.

Perhaps the most interesting ports on the above list are MIDI-in and MIDI-out. MIDI is a standard interface designed to allow synthesisers and other electronic musical instruments to be hooked together so that they can communicate and control each other.

I'm surprised that the MIDI interface is comparatively rare as a standard fitment on home micros. All it requires in hardware terms is a couple of fast (31,250 bits per second) serial ports, one for input and one for output. Even if you don't want to hang a synthesiser onto your Atari, the two MIDI ports needn't be wasted. As Atari pointed out, the ports could make the basis of a very cheap (if slow) local area network for 520STs.

Getting inside is simply a question of removing the retaining screws and lifting the top off the machine. This reveals the keyboard works and part of the main PCB. To get at the whole PCB, you have to remove the keyboard base plate.

The first thought that struck me when I looked at the PCB was how few chips there were. The electronics in the machine are a work of art. When you consider that it was designed and built in five months, you realise that some good people are working for Atari.

The heart of the 520ST is a Motorola 68000 running at 8MHz. I'm sure most of you are familiar with the 68000. Suffice to say that it is one of the most powerful 16-bit processors around and in many respects it is close to being a 32-bit chip. It is a close relative of the processor in the Macintosh. Atari, like the Macintosh, has also used the full 16-bit version.

Supporting the processor are no less than four custom-designed chips. All four were designed by Atari. Two of the chips are easily recognisable because they are of the new square design with pins down all four sides while the other two are contained on conventional DIL packaging.

Atari is coy about what each chip does, but as far as I can make out one of the square chips controls the RAM and some timing functions. The second square chip is called 'Glue' and does general housekeeping.

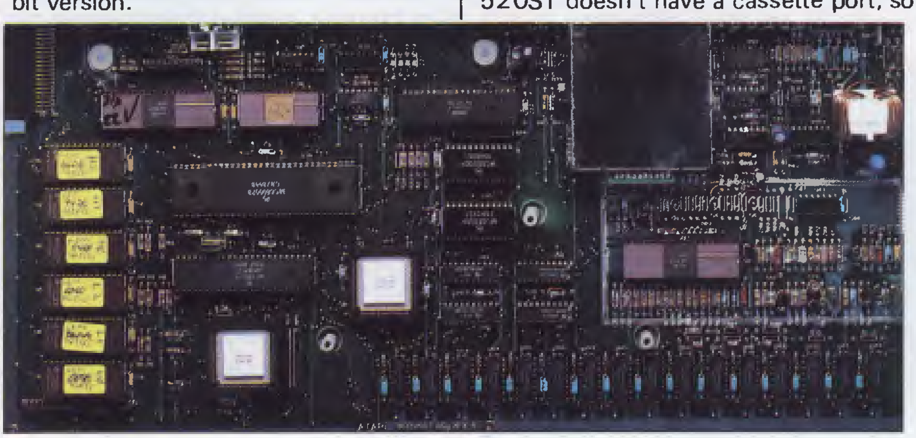
Of the other two custom chips, one acts as a video controller for the different screen modes and the other is a custom DMA controller with a high throughput 32byte FIFO (first-in-first-out) buffer.

The rest of the board is dominated by the RAM and ROM chips. The front right section holds 16 256kbit RAM chips giving a total of 512k of RAM as standard. ROM chips run up the left side of the board giving a total of 192k of ROM. This can be expanded if necessary by adding an external 128k ROM cartridge.

The rest of the PCB is taken up with a few logic chips and standard controllers such as a Western Digital floppy controller and an AY-3-8910 sound chip. Atari says that versions of this machine will be available with more RAM, but unless the company can find cheap one megabit RAM chips, I think it will have to re-engineer the board.

As well as manufacturing the 520ST, Atari is also producing a range of peripherals for the machine. These include 3½in microfloppy disk drives and 5¼in hard disks.

The 520ST comes with a Western Digital floppy disk controller as standard, so adding a floppy disk drive to the system is simply a question of plugging it in. Although floppy disk drives are sold as optional extras to the 520ST, you'd be pretty silly if you didn't buy one. The 520ST doesn't have a cassette port, so disks of some description are necessary if you are going to get anything useful out of the machine.



There are no less than four custom-designed chips on the main board

Floppy disks are available in two versions: either 500k (unformatted) or one megabyte (unformatted). They depart from Atari and Commodore (the company Jack Tramiel previously headed) traditions by using parallel connection rather than the old (and very slow) serial connection. You can daisychain a maximum of two floppy disk drives onto the system.

Hard disks are expected to be available in either 10 or 20 megabyte specifications and to connect via the hard disk interface to the DMA controller. By doing this the system achieves a maximum data transfer rate of 10 megabits per second.

Like most micros of this type, the keyboard is an integral part of the main casing of the 520ST. The keyboard is laid out in a fairly conventional manner with a total of 94 keys in four main areas.

Most of the space is taken up by the main qwerty typing area. Above this are 10 programmable function keys which are shaped to blend in with the cooling slots behind them. To the right of the main typing area is the editing section with 'Help', 'Undo', 'Insert' and 'Clear' keys as well as the four usual cursor-control keys. Finally to the right of the editing keys is a numeric keypad with a the usual mathematical functions and an 'Enter' key.

I liked the keyboard on this machine. The unit is wide enough to allow the different functional areas to be well spaced out, making it easy to find the key you want without risking hitting spurious ones. I also liked the feel of this keyboard. It dispells the myth that because you have a cheap machine, the keyboard must automatically feel cheap.

Both the keyboard and the joystick (mouse) ports are controlled by a dedicated intelligent controller chip mounted underneath the keyboard base plate. The mouse movements simply return control codes into the keyboard buffer. This means that you can mimic the mouse movements from the keyboard if you need to.

The mouse itself is a stylish looking two-button affair. It plugs into one of the joystick ports via a standard 'D' plug. In use the mouse cursor movement was very smooth with none of the jerking that you find on some mouse systems.

The 520ST is very well endowed with display facilities. In fact it boasts virtually every display standard going — TV modulator, RGB and composite video.

Atari has announced a wide range of

monitors which are designed for use with the 520ST. Your choice of monitor governs the display resolution of the 520ST. If you use a domestic television, the resolution is 320 x 200 pixels with up to 16 colours on screen. If you use an RGB monitor, you can choose between low resolution (320 x 200 in 16 colours) or medium resolution (640 x 200 pixels in four colours). Finally if you use a composite video monitor, you can access the highest resolution of the machine (640 x 400 pixels in two colours, black and white).

Whichever monitor you use, the machine will automatically set itself to the correct resolution. I suspect that the majority of people will go for the medium resolution RGB monitor because it gives you access to either low resolution with lots of colour or medium resolution with four colours. Both these modes allow you to produce very impressive graphics although GEM does look a little odd in lo-res.

I think that the hi-res black and white composite video monitor will only appeal to the dedicated Mac imitators and computer-aided design freaks among us. Having said that, I must admit that it was my personal favourite.

In addition to the straight RGB and composite monitors, Atari can also supply a natty little monitor with built-in disk drive. This makes the 520ST system into a very neat desktop package.

System software

The system software in the Atari 520ST works at two levels. On the top is a friendly Mac-like front and underneath is a fairly standard disk-operating system.

Both the front end and the disk operating system are supplied by Digital Research (DR). In fact Atari has licensed the entire range of DR's products from operating systems through utilities to all its languages. So we can expect any advances by DR to be reflected in Atari products. Unlike most other disk-based machines both the DOS and the front end are contained in ROM. This allows the machine to go directly into its operating system on start-up without having to read from disk.

Although it is unlikely that the average user will ever want to descend into the DOS, I think it is a good idea to look at the front end and the DOS separately. The disk operating system on the Atari 520ST has been rather unfortunately named 'TOS'. This apparently stands for 'Tramiel Operating System' although I fail to see why anyone would want to operate a Tramiel.

Although its name is unlike any other operating system, TOS is actually based on Digital Research's CP/M-68k. This is a little-known and not very popular version of CP/M for the Motorola 68000 processor. The main difference between TOS and CP/M-68k lies in the area of file handling. TOS has a hierarchical file structure *a la* MS-DOS whereas standard CP/M-68k does not.

The command language of TOS is a strange hotch potch of CP/M, MS-DOS and Unix-like commands. When you enter the system level you are greeted by a version of the familiar CP/M-DOS-style 'A' prompt. However, unlike CP/M you can enter path-names, play around with sub-directories, and the like.

Most of the old unfriendly CP/M commands are there with the addition of a few commands of uncertain parentage. You can type 'DIR' and get a directory, or you can type the Unix equivalent 'LS' to get the same result.

But as I said earlier, it is unlikely that anyone other than software developers will want or need the command-line functions of TOS. For most peoeple the friendly front end will be the face of the 520ST.

Because it has licensed Digital Research's entire range of software, it was logical for Atari to use DR's GEM friendly front end for the 520ST. I have written quite a bit in past issues about GEM (see *APC*, March 1985), so I'll just do a quick re-cap of the highlights here.

GEM stands for Graphics Environment Manager. It allows companies who use it to give their machines a user friendly graphics interface which closely resembles that of the Macintosh.

This similarity extends to the use of movable re-sizeable windows, ikons to represent objects such as disks and disk files, and the use of pull-down menus and mice. The GEM environment makes it easier for a novice to do all the usual housekeeping chores as well as making it easier to call up programs from disk.

As well as making life as easy as possible for the user, GEM also makes life comparatively easy for the applications programmer who wants to move his program from machine-to-machine. To understand this we need to look at how GEM works.

One of the functions of GEM is to provide a standardised interface to the graphics screen and graphics devices in much the same way as a disk operating system provides a standardised interface to the disks.

By using GEM, a programmer can write to this 'Virtual Device Interface'

Technical specifications

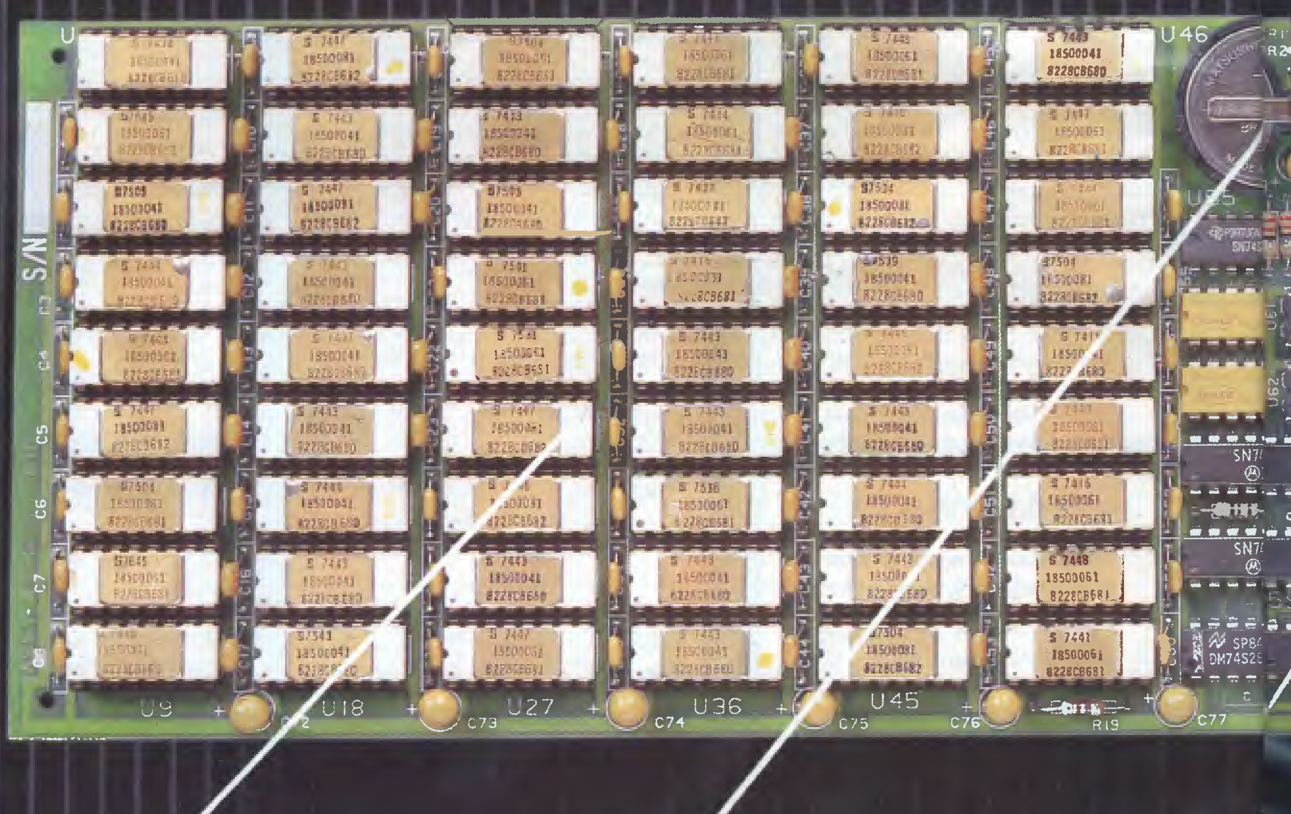
Processor:	Motorola 68000 running at 8MHz
ROM:	192k containing O/S and GEM
RAM:	512k
Mass Storage:	500k or one megabyte (unformatted) 3½in micro-floppy disks, choice of hard disk capacities.
Keyboard:	94 keys including 10 function keys.
I/O:	RS232 serial, Centronics parallel, floppy disk, Winchester, TV video, RGB video, composite video, MIDI-in, MIDI-out, joystick, mouse, ROM cartridge
DOS:	TOS, GEM
Bundled software:	Personal Basic, DR Logo

When they said "Allow 28 days for delivery" I didn't realise they meant the answers to questions.

6PAK PLUS™ QUBIE'

By QUBIE'

9/62 Blackshaw Avenue, Mortdale
N.S.W. 2223 Australia
Telephone: (02) 579 3322
4809 Calle Alto, Camarillo
California 93010 U.S.A.
Tempo House, 15 Falcon Road, London
SW11 2PH, United Kingdom



ACTUAL SIZE

384K MEMORY EXPANSION

Socketed and expandable to 384K. Your 6PakPlus™ comes standard with 64K memory. 384K on the 6PakPlus™ added to 256K on the PC system board provides for the maximum addressable user memory.

CLOCK/CALENDAR

6PakPlus™ eliminates the need to manually input the date on system start-up. The chronograph is fully compatible with all PC-DOS software routines which utilize clock functions. Battery backup power supply keeps the clock running at all times.

PARALLEL PRINTER PORT

Interface the PC to most printers or other parallel devices. The parallel por can be addressed as LPT1 or LPT2. Internal cable and DB25 connector are all included.

Accounting Software

If you already use Lotus 1.2.3.

Presenting C.P.A. +, the small business accounting program for Lotus 1.2.3. Fully featured, fully integrated and all on the one disk! Starting right now, your accounting problems may be solved forever . . . By You! Say hello to C.P.A. +, the most ingenious accounting program ever created for the I.B.M. P.C. For the first time the power of Lotus 1.2.3. is utilized in a full accounting package, one diskette integrates all modules, all files are updated automatically and reports can be custom designed. That's C.P.A. + at $795.00 R.R.P.

And if you don't.

The completely configured, ready-to-run System 4 and System 7 includes a full set of test data files for operator training. These systems offer Accounts Receivable, Accounts Payable, General Ledger, Invoicing or Order Billing, Order Entry, Inventory Control and powerful sales analysis. Totally menu driven with step-by-step documentation. Each module includes a beginner's introduction, and System 4 has a built-in help feature for instant answers to those puzzling questions. Works with all printers. These powerful accountancy packages will give you the information and control that you need to run a better, more profitable business. Manufactured and supported throughout Australia, System 4 and System 7 are competitively priced from $795.00 R.R.P. At Dealers around Australia.

All modified and approved to comply with accountancy procedures in Australia by Busiware

Lotus 1.2.3 is the trademark of the Lotus Development Corporation.

CPA+ is the trademark of G.N.P. Development Corporation.

B·U·S·I·W·A·R·E

A DIVISION OF OZI SOFT PTY. LTD.

For further details complete this coupon and send it to Busiware, Suite 33, 8-24 Kippax Street, Surry Hills, N.S.W. 2010. Telephone (02) 211 1266.

Send me more information on the **SYSTEM 4/SYSTEM 7**________ **C.P.A. +**________

NAME:________________ COMPANY NAME:________________

ADDRESS:________________ PHONE NO.:________________

________________ POST CODE:________________

COMPUTER TYPE:________________

rather than having to go directly to the hardware of the specific machine he is writing for. This makes it much easier for the programmer to move his masterpiece to other machines which use the GEM environment.

Originally, Digital Research demonstrated GEM on the IBM PC, but it also works on compatibles and machines like the Apricot range and the Atari.

GEM Desktop on the Atari looks very similar to GEM on most other machines. You have the disk ikons and trash can at the right of the screen, a menu bar along the top, and windows can be opened, closed and moved in the same way as on other GEM machines. However, as you would expect, because of its fast 68000 processor, GEM moves along with much more of a clip on the 520ST than it does on something like an IBM PC which can sometimes be quite slow — especially when updating more than one window at a time.

Atari has added a couple of desk accessories into the 'Desk' pull down menu. These included a Macintosh-style 'Control Panel' which allows you to play around with some of the system settings such as the colours used to display the GEM Desktop. Atari has also included a simple terminal emulator and a utility to set up the RS232 serial port.

One of the nice features of GEM is that it adapts itself to the different specifications of hardware it is running on. Consequently, as you would expect, there is quite a difference between the way it looks in the hi-res black and white mode to the lo-res colour modes.

Personally I liked the hi-res mode more than the colourful lo-res modes. Other people seem to prefer the colourful modes. In its hi-res black and white mode, GEM makes the 520ST look more like the Macintosh than any other GEM machine I have seen.

One nice feature of GEM on the 520ST is that it allows you to have two logical disk ikons operative, even if you only have one physical disk drive attached to the system. GEM treats them as separate disks and allows you to do all the things you would do on a twin-drive system by prompting you to change disks in the drive when necessary. I found this was extremely useful, especially when I was trying to copy disks.

Overall, the implementation of GEM on the Atari is very good. Having GEM in ROM makes the loading process much faster than the tedious rigmarole you have to go through to get it to load on something like the IBM PC. It also means that if the system bombs out for any reason, it goes to GEM as the base routine rather than some obscure monitor or suchlike.

Although the Atari implementation is good, what you can do with the GEM Desktop is still limited to some extent by the operating system underneath GEM.

Applications software

When the 520ST goes on sale, it will be supplied with GEM, Basic and Logo. At the time of writing there still seems to be some debate within Atari as to whether these will all be in the internal ROM or whether parts will be on disk. It seems likely that early machines will have software on disk for debugging purposes and later machines will have the software in ROM.

Both of the programming languages are again supplied by Digital Research. The Basic is its Personal Basic and the Logo is its DR Logo. Both of these products are widely available on other machines, although they have been

modified somewhat for the Atari machine.

In its usual form, Personal Basic is functionally very similar to standard Microsoft Basic. At the time of writing the Atari modifications were far from finished. The most obvious difference between the standard version and the Atari version is that the user interface has been changed. Whereas before you had a simple command line, there are now pull-down control menus, mouse control, and different windows for editing output and so on.

Eventually your Basic programs should be able to control the whole GEM user interface. So you should be able to write mouse-driven high resolution colour programs with lots of windows, bells and whistles. However, at the time of writing the only way to do this was by using loads of PEEKs, POKEs and machine code CALLs. The people at Atari say that the release version will have extra keywords to control all these features. While I'm happy to believe them, remember that Jack Tramiel's last machine was the Commodore 64! Enough said.

The Atari version of DR Logo was in much better shape. DR Logo is a popular version of the language for CP/M and MS-DOS machines. Like the Basic, the user interface of DR Logo has been changed to take advantage of the user friendly GEM interface.

In the case of Logo there are different windows for commands, editing, output and so on, with many options being available from the pull-down menu bar. So you can watch the turtle rushing around in one window while the debugger works through your program in another. Very pretty.

For the serious software developer types, the most popular language is likely to be C. Most of GEM is written in C and Digital Research provides a full set of C bindings to GEM to make applications easier to develop.

A number of software developers seem to be taking advantage of GEMs Virtual Device Interface and developing software on IBM PCs and then downloading and re-compiling it on the Atari.

As far as finished applications packages are concerned, Atari says that it has sold over 100 development systems to US software houses and that it is beavering away converting its packages to the 520ST.

Only time will tell if this is true, but the Atari software guy did have a lot of interesting looking disks in his bag at Hanover.

Has your love aff
your MICRO go

fair with
ne cold?

See page 22 for the amazing
Microtex 666 Viatel Package

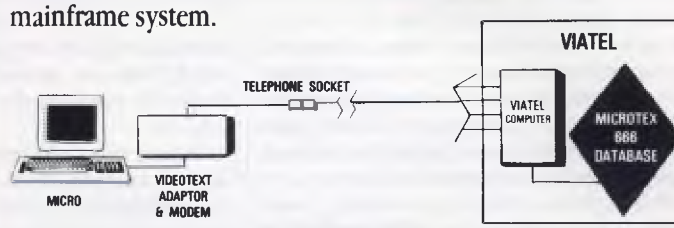
Tap into Microtex 666 and give your computer a new lease of life.

You know what it's like — the love affair that started out by promising so much ... ended as quickly as it began. Leaving you feeling like a jilted lover. And your micro sits there untouched and alone.

But now the promise becomes a reality.

With the introduction of Microtex 666 you now have access to a potentially vast spine tingling menu of services, information and practical applications.

Consider, for the cost of local call and an 8 cent per minute (5 cents off peak) Viatel charge, you can communicate with a network of other micro users, on a nation-wide mainframe system.



Interact with our multi-player games, shop 24 hours a day on our national Trading Post or talk to other subscribers across the nation by utilizing the Microtex 666 electronic mail and telex service.

Feast on the Tips and Hints or toy with Questions and Answers. And then there's our Telesoftware library — download programs, use them, then store them forever on disk or cassette.

Best of all, Microtex 666 is brought to you by Computer Publications, publisher of Australian Personal Computer and Personal Computer Games. So you know you're getting a superior service ... from the experts.

Microtex 666 really is incredible and all it costs is $49.95 to subscribe. The only extra you need to tap into Microtex 666 is a videotex modem unit.

Fill in the coupon for more information and we'll have that jaded love affair blooming again in no time.

MICROTEX 666

Please send me the full facts about MICROTEX 666

Name: ____________________

Make/Model of Micro: ____________________

Address: ____________________

____________________ P'code: ________ Tel: ________

MICROTEX 666: 77 Glenhuntly Road Elwood 3184.

H&L 4056

Documentation

As this was a pre-production machine no documentation was available.

Prices

At the time of going to press, no Australian prices were available. In fact, negotiations for the distributorship were still underway. Nevertheless we are in a position to make an 'informed guess' knowing the projected US prices and the mark-ups from US to Australian prices of other similarly priced machines. It seems likely the 520ST with a single 500k disk drive will sell for around $1,500.

Future products

Atari in general (and Tramiel in particular) seems adamant that the 520ST is just the start of a range of future enhanced machines. The most likely enhancement is more memory (although 512k seems enough for most purposes!) and the introduction of a local area network of some description.

Perhaps the most interesting new product however, is the addition of a CD ROM unit. We ran an article on CD ROMs last month. They use the same kind of optical digital compact disks as the home hi-fi systems except that the computer versions don't need the complex digital-to-analogue converters as in the audio units. Instead of carrying digitised audio signals, computer compact disks store digital computer data.

The main differences between compact disks and magnetic disks is that whereas you can get about 720k on a double-sided magnetic disk, you can get 550*megabytes* of data on one compact disk. The trouble is that whereas you can write to a magnetic disk, you can only read from a laser disk — hence the name CD ROM.

Conclusion

There can be no doubt that the Atari 520ST is a very impressive machine.

The machine seems to have everything going for it — good keyboard, lots of I/O facilities, lots of RAM and what, for many, is the nicest 16-bit processor around.

When you add in the GEM friendly user interface and the availability of cheap disk drives, you end up with a very impressive system indeed. I'm still not sure whether to classify it as a home or business machine. Its abilities are well suited to both. I have a feeling that in the US it will be treated more as a home machine and over here it will appeal to business users.

Having said that it is a good machine, there are still two areas that you should think about before you rush off to your local Atari dealer when the 520ST goes on sale, to part with your hard earned cash.

The first is the availability of good applications software. It's a virtual certainty that Digital Research's word processing and graphics applications programs will be available soon, as will Atari's Infinity integrated package. But I'm not so sure if or when third party applications will begin to appear.

The second point to watch is the price of the system. The original 130ST was set to sell in the US for $399. This set people thinking that the range is cheap. While this is still true when you consider what you get, remember that the 130ST has been dropped and you are looking at about $1,500 for a 520ST and disk drive.

Even so, the bottom line is that when the machine appears in the shops, which will probably be in the third quarter of this year, I'll be at the front of the queue to buy one.

Brother WP600

The Brother WP600 electric typewriter cum printer is in a class of its own. It's not particularly cheap, it's a bit on the slow side but the quality of the dot matrix typeface is excellent. Simon Craven takes a closer look.

The Brother WP600 is an oddity in the personal computing world as the product of an evolutionary — rather than revolutionary process. An early example of Brother's lightweight electronic typewriter/printer was the EP22, which suffered from a poor keyboard and a rather crude typeface (although this was fairly typical of cheap dot matrix printers at the time.)

The EP44 came out about a year later, and offered a relatively classy print style thanks to a high definition 24 x 18 matrix. On this latest model the dreaded button keyboard has been replaced by a conventional typewriter-style unit, and the machine's text editing facilities have been improved.

The WP600 has four distinct personalities, as a typewriter, a simple portable text editor suitable for producing reasonably complex documents up to about 2000 words long, as a near letter quality printer for any computer equipped with an RS232 serial interface, and as a terminal, enabling, for example, programs to be written on the train on the way back from work, then downloaded into a micro and run.

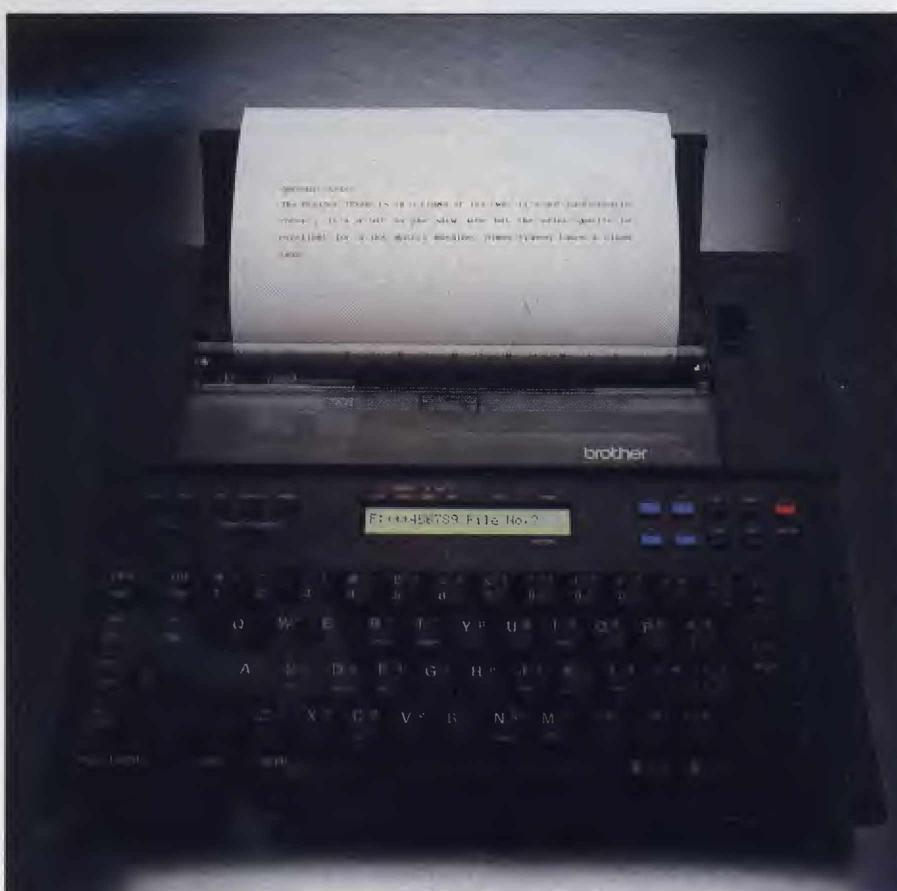
Hardware

Despite its modest size and low weight, it's an imposing beast on first acquaintance, with 59 full-size keys and 19 buttons, switches and other controls to bewilder the first-time user.

The keyboard is excellent, with a light over-centre action and a reasonable layout. The profusion of keytop markings brings the Spectrum irrestibly to mind, but is accounted for by a 'second shift' key. This enables the WP600 user to call up the entire character set directly from the keyboard — a much more convenient arrangement than the slow and error-prone IBM PC solution, which has the user laboriously keying in three digit character codes while keeping the ALT key depressed.

The remaining legends relate to the use of the CODE key, equivalent to the control key on a personal computer.

The display is a simple 24-character LCD screen of one line, which is rather too small for the kind of self-indulgent editing processes most regular word processing users require. As a means of catching spelling errors before they are committed to paper, however, it is excellent. The display is just above the top row of keys, where inexperienced typists can keep an eye on it as they proceed. *Real* typists, of course, are not supposed to look at the keyboard at all — they stare rigidly at the paper or VDU, and catch their infrequent mistypings instantly, but it's nice to see a writing



```
This is a sample of the Brother's
typeface, which is unusually tidy
for a dot-matrix printer. The
dots can be seen, but only on
very close inspection. The line
spacing on this sample is 1 1/2,
and single and double spacing
are also available. The optional
justification is here in automatic
operation, and some underlined
text is also included.
```

Example of Brother WP600's dot matrix typeface

machine which caters equally well for experts and fumblers alike.

Each of the three major modes of operation — normal, word processor and terminal — has a number of sub-modes which let the user configure the WP600 to personal taste. The simplest mode, labelled 'normal', turns the machine into an unusually sophisticated typewriter. Left and right margins can be set anywhere, as long as the typing width is over 25 characters, and one-key operation allows automatic centring and underlining of text. Text can be set with a ragged right margin, fully justified, or right flush with a ragged left-margin.

One often over-looked feature is the provision of temporary left margin settings for hanging indentation — that is, setting a whole paragraph a few characters further in than the rest of the text.

One of the biggest factors increasing the speed of a word processor user over the operator of an electric typewriter is automatic wordwrap, eliminating the need to insert manual carriage returns at the end of each line. Even in normal mode, this feature is available on the WP600.

Text can be printed out as it's typed in, but a better way of working for anyone unsure of his keyboard skills is to have the text printed a line at a time as work progresses. This lets you correct mistakes which are present on the LCD before they are printed.

In word processing mode, the facilities are not vastly superior, but the text is stored in memory instead of being printed. This means that any part of the document can be changed at any time, (the 'normal' mode restricts editing to the current line of text) and also lets you work without the twin distractions of the printer noise and its insatiable appetite for paper.

The editing facilities have been greatly improved over those of the earlier EP44, and include a FIND function and a SEARCH and REPLACE option.

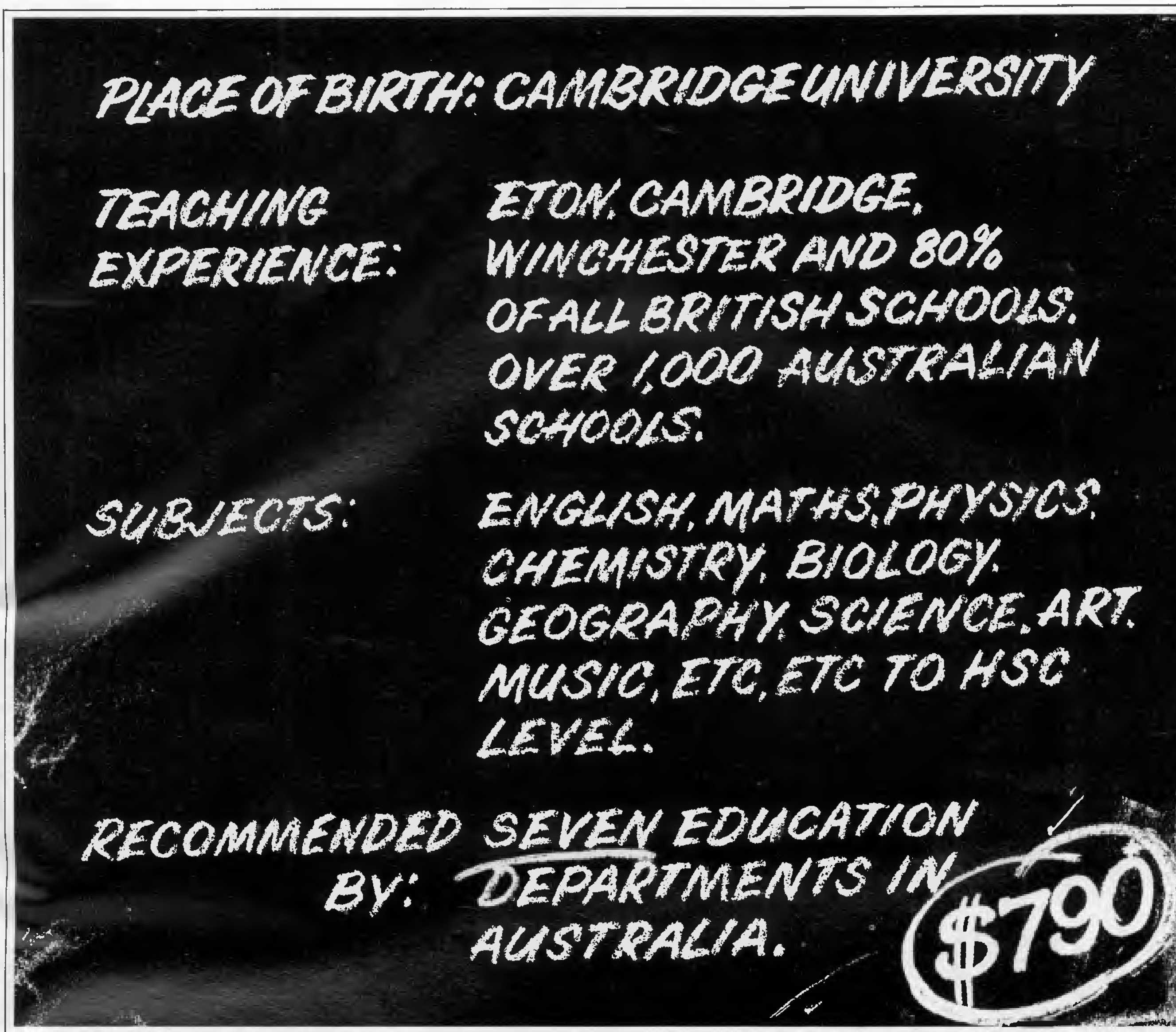
Moving, copying or deleting blocks of text is now possible, but still hard work. Each line of text is numbered, and block operations are carried out by defining the line numbers which make up the block.

Editing operations of this kind are simplified by the logical choice of keys to initiate each task (CODE-D for delete, CODE-C for copy, and so on), and in case even this proves too much to remember, the functions are inscribed in small green letters on the relevant keytops.

Nine separate files can be held within the WP600's memory, but their total combined length may not exceed 14,200 characters — equivalent to a piece of text not much longer than this Checkout. The Brother's architecture is CMOS, and the low power consumption of the chips makes battery back-up for the memory contents a practical proposition. As long as the four hefty D dry cells are in place, or the mains adaptor is connected, switching the Brother off will not make it forget anything. Even changing the batteries need not be hazardous, as deterioration starts about 25 minutes after the batteries are removed or the AC power supply unplugged.

An increasing number of home and professional computers are now appearing equipped with only the RS232 serial interface. This makes printing from these machines rather inconvenient. Most of the low-cost printers which are popular in Australia use the centronics-type parallel interface as standard.

RS232 devices are traditionally strewn with pitfalls for the unwary, as the standard is rather loosely defined and no two serial interfaces ever seem to be



With qualifications like these, shouldn't this teacher be at your school?

Only one teacher in the world comes with credentials such as these.

The teacher is a computer.

The BBC school computer.

You'll understand why it's such a good teacher, when you consider those responsible for its education.

The professors from the Computer Studies faculty at Cambridge, for example.

The panel of experts on educational television from the BBC, to name another.

Along with the hundreds of members of the teaching fraternity, both here and in the U.K., who developed the software.

Without doubt, there is no finer educational tool you can give to the teachers at your school.

And certainly, no other teaching system quite compares as far as the students are concerned.

There is, however, one important consideration we have yet to mention.

The price.

What other teacher offers such impeccable qualifications for just $790?

For more information on the BBC microcomputer and the new compact version, the Electron, send this coupon to the Australian distributor, Barson Computers Pty. Ltd.

The BBC school computer.

Post to: Barson Computers Pty. Ltd., 335 Johnston St., Abbotsford, Victoria 3067, or 7 West St., North Sydney 2060. Or phone Barson Melbourne on (03) 419 3033 or Sydney (02) 957 2588.

Name: ________

School: ________

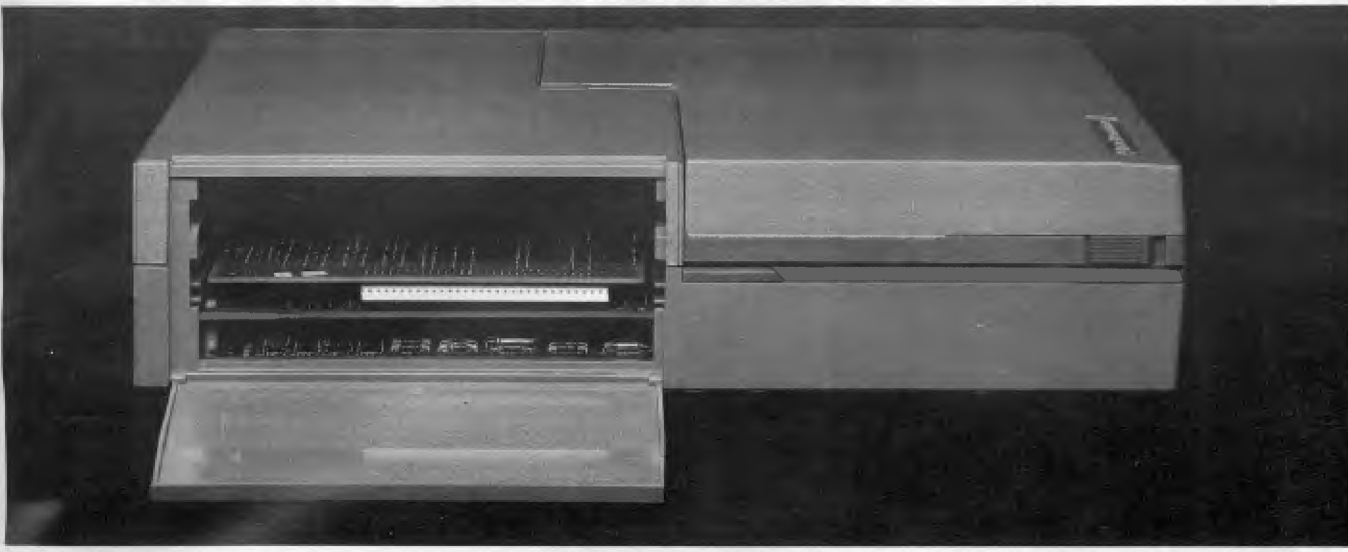
Address: ________

Postcode: ________

BBC is a product of Acorn Computers Limited.

BARSON COMPUTERS

* $790 tax free price to schools.



Two of TI's forthcoming range of option modules can be placed in these spaces

The cursor keys run in a line to the right of the good-sized space bar, and CTRL, ALT and HOME complete the layout. The only IBM PC function not represented on the key tops is SCROLL LOCK.

At the top of the keyboard is a removable plastic panel, under which a legend can be fitted to accompany applications software that use the function keys for specific jobs. The only minus point for the keyboard is the height from the desk, 4.5 cms, which could cause problems for touch typists.

System software

On starting up the system by means of the on/off switch at the front, the ProLite quickly reveals a complete absence of any of the interesting ROM software that you often get with portable systems. No, it says, none of that funny business here, let's have a system disk.

After the system disk is inserted, it is just like any other MS-DOS computer with one exception. TI has come up with Natural Link, which is another 'user-friendly' shell for the 'difficult to learn' MS-DOS operating system. On the face of it, this type of thing is a good idea. However, the opening menu is crammed with information which makes it fairly difficult to follow what to do. This was not helped by the fact that the shell and TI's implementation of MS-DOS was undocumented on the test model.

To check for any hardware problems a diagnostic disk is provided. This takes the user through a menu system and it is possible to test each individual component (screen, keyboard, graphics ROM, RAM, and so on) separately or the whole lot can be done in one go. This takes about 30 minutes, and all the actions and operations are prompted and documented onscreen.

TI plans to make CP/M-86 and the UCSD-p operating systems available for the ProLite, too.

Benchmarks

BM1	1.3
BM2	4.4
BM3	9.4
BM4	9.7
BM5	10.6
BM6	18.6
BM7	29.3
BM8	31.5
Average	14.35

All timings in seconds. For a full listing of the Benchmark programs, see "Endzone".

Technical specifications

Processor:	Intel 80C88
ROM:	16k
RAM:	256k expandable to 512k
Mass storage:	One or two 720k micro floppy disks
Keyboard:	79 keys
Size:	33 cms x 29 cms x 7.6 cms
Weight:	4.75 cms
I/O:	Parallel printer port, two TI proprietary expansion ports, bus connector for battery pack and/or second disk drive
DOS:	MS-DOS version 2.0
Battery:	1.3kg optional battery pack gives two to six hours power

Applications Software

There will be no bundled software with the ProLite but users will, according to TI, be able to choose from the wide range of applications available for the TI Professional. The two systems are compatible, the company says, so software can be easily ported.

The test system was supplied with Multiplan, Easywriter II and Basic. Multiplan and Easywriter appear to work exactly as they do on a standard MS-DOS system with a standard screen. It takes a while to get used to the 'flattened' appearance of the information on the screen due to the screen's rectangular shape, but it was possible to read all the information. Reverse video, as used, say, for highlighting Multiplan prompts, causes the screen no problems although highlighted words are not easy to see.

TI plans to make much of the software available in cartridge form. What these applications will be and how much they will cost have not yet been revealed.

Documentation

The only manual available for the evaluation system was a short and simple set-

Be an investmen

✻ **Save an extra 18% off our low, low prices with t**

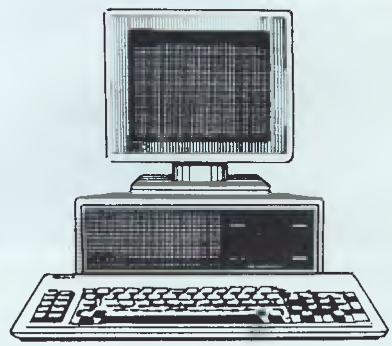
Call us now direct on

(02) 331-3888

(02) 331-6799

Information. Advice. Support. Best Prices

Hardware Specials



OLIVETTI M24 — ONLY $3750.00
Standard System includes Olivetti M24 base unit with 128K RAM, one 360K disk drive, high res. green monitor, IBM style keyboard, full MS DOS operating system with GW Basic.

OLIVETTI M24 — ONLY $4250.00
Expanded System as above with an extra 360K disc drive and a total of 256K RAM.

OSBORNE VIXEN — ONLY $2695
A big full-function system in a small, affordable, *reliable* package.

Printers

Epson FX 100+ .... **call**
The world's best seller

Toshiba 1351 .... **$1550**
The first 3-in-one printer smart enough to change typefaces automatically.
For other leading brands call.

Modems

PC IN/MODEM **$895**
from Data Netcomm Mode 3 Telecom permit No C84/39/1148. Asynchronous and videotex internal modem for IBM PC. Auto dial. Auto answer. Auto disconnect. Fully Hayes compatible.

Graphics

Epson HI 80 ....
Easy-to-use 4 pin 10 colour plotter. HP/ 7470 compatible.
Microsoft Chart ....
AST Preview Graphics Board ......
Mono and colour card — supports Lotu
D.R. Draw ....

Monitors

Kaga Taxan Amber or Green ......
Kaga Taxan Vision III Colour .....

Diskettes

Datalife DSDD ....

Emulation Boards

Irma ....

Multifunction Boards

Persyst ....

Hard Disk Drives

Everex 10mB **$1**
Everex 20mB **$2**
Includes controller, cables, software instruction manual.

Chips

INTEL 8087 ....
High speed co-processor
64K RAM chips ....
256K RAM chips ....

Cables ....

FOR P.C. SOLUTIONS CALL CASH & CARRY COMPUTERS AND SOFTWARE
480 CROWN ST., SURRY HILLS 2010 OR WRITE BOX 696, DARLINGHURST NSW 2020

Wired to the world

The Microbee/Viatel Option
Viatel, Telecom's exciting new interactive teletext system is now in full operation and even more exciting, it's available on your favourite computer, the **microbee.**

The Viatel Option is a hardware/software modification for any **microbee,** that used in conjunction with a 1200/75 BAUD beemodem will bring information, banking, news, weather reports, software and much, much more into your home or business for the cost of a local phone call.

$49.50

Viatel – another good reason to own a microbee

BEEMODEM
Telecom Approved to connect your **microbee** to other computers using the telephone lines. Your BEEMODEM will convert your **microbee** into a complete home terminal that can become your information window to the world!! BEEMODEM operates at 300 or 1200/75 BAUD. **$189.50**

microbee computer

Designed and manufactured in Australia by Applied Technology

PHONE ORDERS bankcard ACCEPTED

microbee technology centres

N.S.W.
1 Pattison Ave, Waitara 2077
Phone (02) 487 2711

VIC.
729 Glenferrie Rd, Hawthorn 3122
Phone (03) 819 5288

W.A.
141 Stirling Highway, Nedlands
Phone (09) 386 8289

S.A.
151 Unley Rd, Unley 5061,
Phone (08) 272 1384

QLD
455 Logan Rd, Stones Corner, 4120
Phone (07) 394 3688

FACTORY
Koala Crescent, West Gosford 2250
Phone (043) 24 2711

up guide in the standard A5 binder. Easy to read and follow, it sets out all the details of plugging in the right bits, adjusting the screen, and so on. An operating system guide will be available with the systems delivered to customers.

Prices

The base system with monochrome graphics, single disk drive, 256k RAM and MS-DOS will cost $6,540 including sales tax. The top system with two drives and 512k RAM costs $8,840. The 45cps thermal printer which can run off the ProLite's batteries will cost $975, and the battery pack itself will cost around $345. The large fake leather carrying case costs an extortionate $230 — I'd forget it and use a large briefcase.

Conclusion

At around $6.500 for a battery-powered, single-disk system, TI is not exactly giving its ProLite away. Price aside, the TI system is good for those who want a genuinely portable computer with more power than the Epson and NEC 'notebook' computers. For those who already own a TI Professional Computer this could be a wise purchase, but for IBM PC users I cannot vouch for the stated ability to upload information from the Pro-Lite. I can't vouch for the battery pack either, and with a TI-declared performance of only two to six hours, depending on disk usage, that does not look like the last word in battery technology.

On the plus side, it is a neat, compact system with some imaginative ideas (the module slots and ability to transfer information to and from PCs or through a modem) thrown in. It's a mixed verdict on this system. It's good in parts but there are better battery-powered portables around.

My main reservation, as with all systems of this type, is the screen.

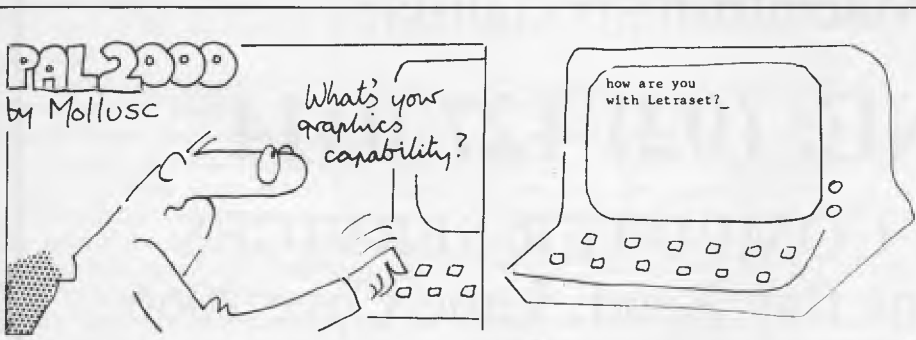
In perspective

The ProLite staggers across a market that at present exists more in the minds of the computer industry marketing men than in any real sense. Potentially, there are many businessmen and women who already use a desk-top machine and want some kind of portable system to carry around between office, home and remote working site. This market has been proven, they say, in the past success of the Osborne and the current success of the Compaq. However, there is another argument that says the sucess of the Osborne and Compaq was based more on the availability of a cheap system running industry-standard software (first CP/M and now PC-DOS).

But the computer industry is sticking to its guns and there is a host of portable systems available and about to be launched, from the notebook computers like the Tandy 100 to the semi-portable IBM PC compatibles. The ProLite is trying to occupy a distinct slot between these extremes.

It is a fully portable system competing with the HP110 and the Data General One. It is a creditable attempt to compete except that it does not have the expensive engineering and ease of use of the 110, has an inferior screen to tha Data General One and is also more expensive than the latter. Like all lapheld portables it's a compromise, and an uneasy one at that, between convenience and usability.

All the battery-powered models have to use the small 3½in disks. Until IBM offers a standard in that format, there is little chance of a genuine IBM PC-compatible lapheld system emerging.



Good looking on PC and film

Geoff Wheelwright tries out Presentation Master, a package which aims to give your PC high-quality graphics.

Only a few years ago, executives were jumping up and down in excitement at the prospect of seeing their spread-sheet neatly and beautifully printed out from VisiCalc. Now graphics output has developed to such an extent that giant screen TV systems plug into your PC. The problem is, though, that the old system looks terrible and the new system is costly and cumbersome. A problem that Digital Research has addressed with its Presentation Master.

Foundation building

Presentation Master tries to bridge the old and new by building on the foundation laid by Lotus 1-2-3. This was the first package to incorporate professional business graphics systems into its spreadsheet. In a similar way Presentation Master uses conventional technology in conjunction with specially designed graphics and machine control software.

Where it wins over Lotus (and the integrated packages like Symphony, Framework or Open Access that followed it) is that it does not rely on printers and plotters for its presentation — the point being that printers and plotters are limited by the medium they work in: paper.

To get material together for a slide presentation of graphic business data, paper-based graphs either have to be processed photographically, or professionals have to be called in to take 'screen shots', which again are often only as good as the monitor on which they are presented. (Low-cost overhead projection systems and giant-screen TV systems were, in fact, projected as the solution to this).

Presentation Master, however, is a more inventive system. It consists of several items: a Minolta 35mm camera with auto-winder, a large white box with several connections coming out of the back, and a large round hole at the front (you could be forgiven for thinking it was used to make fresh pasta), a Polaroid camera back for loading instant print film, a 35mm autoprocessor for developing colour slides in two minutes, and several other odd-looking goodies.

The only familiar looking objects in this strange collection are the large boxed Digital Research manuals with their attendant software.

The image-recorder itself handles the generation of colour for the film so you don't have to worry about using a colour monitor on the PC.

To get them running with your PC, you'll first have to check that you've got at least 256k RAM, an IBM colour/graphics monitor adaptor (or equivalent), a compatible graphics display monitor (colour is not essential), at least two drives, an operating system at least as recent as PC-DOS 2.0 and a serial RS232 communications port.

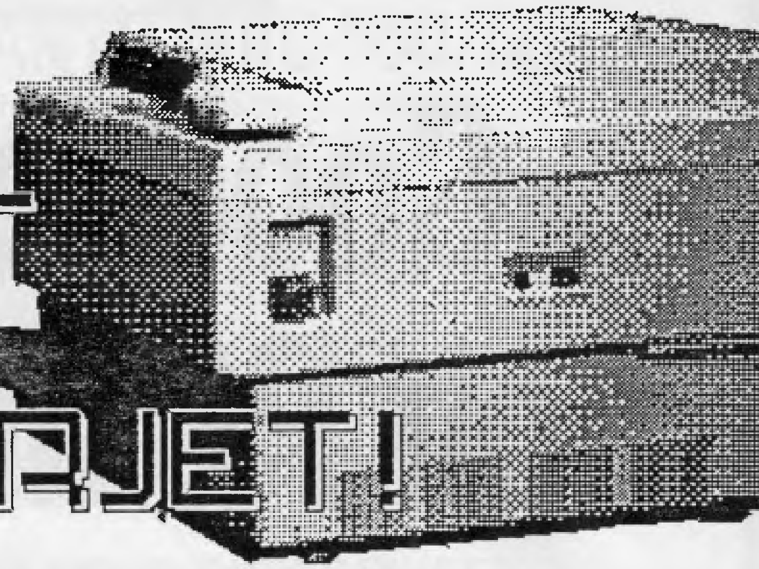
System connection

The system, which was unveiled late last year, costs $3,592 (including sales tax). Connecting the whole system up is pretty easy and doesn't require much of a look at the documentation. The 25-pin DIN lead is connected to the RS232 socket, the video lead is connected to the video out socket on your PC, the camera attaches to the front of the Presentation Master Image Recorder (the item that looks like a pasta-maker) and the image-recorder gets plugged into the mains socket.

The philc sophy behind this seemingly unconnecteo collection of hardware and software is one of flexibility. The camera is a standard 35mm job, so you can choose your own film; the Polaroid section of the package uses standard Polaroid film, so you shouldn't have to run out to computer supply shops to get refills; and the image-recorder itself handles the generation of colour for the film so you don't have to worry about using a colour monitor.

Perhaps the most important consideration is the software flexibility. Although you are likely to have one of the spreadsheet packages supported by the system (they include VisiCalc, Lotus 1-2-3, SuperCalc and Multiplan), Digital Research has included graph-generation software with the system. This software will operate on machines with only 256k, and a revised version of the software (also bundled with the system) will take advantage of the extra memory offered by 320k plus systems. This means that no matter which spreadsheet system you use, you'll still be able to get something out of Presentation Master. It could be argued, of course, that for the $3,600 you pay you should be guaranteed to get something out of the system — but it's nice to know that DR has put enough thought into Presentation Mas-

TAKE CONTROL OF YOUR HP LASERJET!



EVEN 3 VIC 20s CAN MAKE A LASER JET HUM WITH PRINTER OPTIMISER

The Printer Optimizer is the perfect companion for your HP LaserJet. Just look . . .

• The Printer Optimizer combines a printer buffer and multi-printer switchbox in one unit. Up to three printers can be connected to it, and they can be a mixture of serial and parallel printers. The HP LaserJet is serial of course, but using the Printer Optimizer you can run the LaserJet off of any computer's parallel printer port — no adaptors or special software needed. This means that you can have the LaserJet and up to two other printers on line and accessible at the push of a button (or selected by your software if you like).

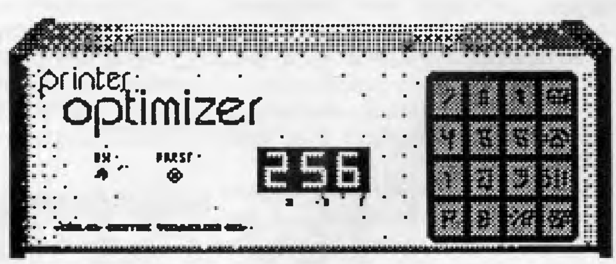
• The Printer Optimizer is capable of adapting the HP LaserJet to a variety of programs that otherwise won't work with it. The Printer Optimizer can literally trick your computer into thinking that the HP LaserJet is another printer.

• The Printer Optimizer gives you total access to all the LaserJet's features. The Printer Optimizer's front-panel keypad lets you instantly change type fonts or lines per inch, set margins, switch to manual paper feed, make copies . . . everything by simply pushing a button!

In addition the Printer Optimizer also lets you access these features from your computer in a document. Change to *italics*, **bold print**, condensed print, **big print**, print two columns or two pages on a single sheet.

4th point. Printer Optimizers can be inter-connected. The laser jet can then be shared by up to 15 computers with the Printer Optimizer queing and spooling. In fact up to 4 parallel or serial plotters, printers and laser jets can be shared in this network.

Prices from $675 incl tax



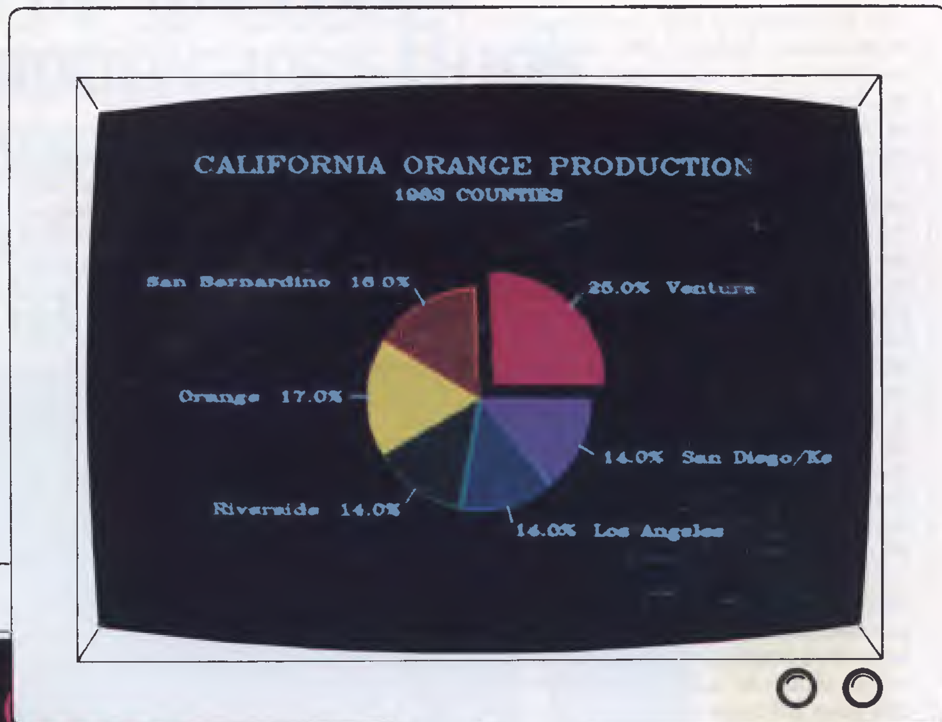
interface technology of Australia

1 CHANDOS STREET • ST LEONARDS • NSW 2065 • AUSTRALIA
TELEPHONE [61] (02) 438 2899 TELEX AA75117
All correspondence to:
P.O. BOX 575 • CROWS NEST • NSW 2065 • AUSTRALIA

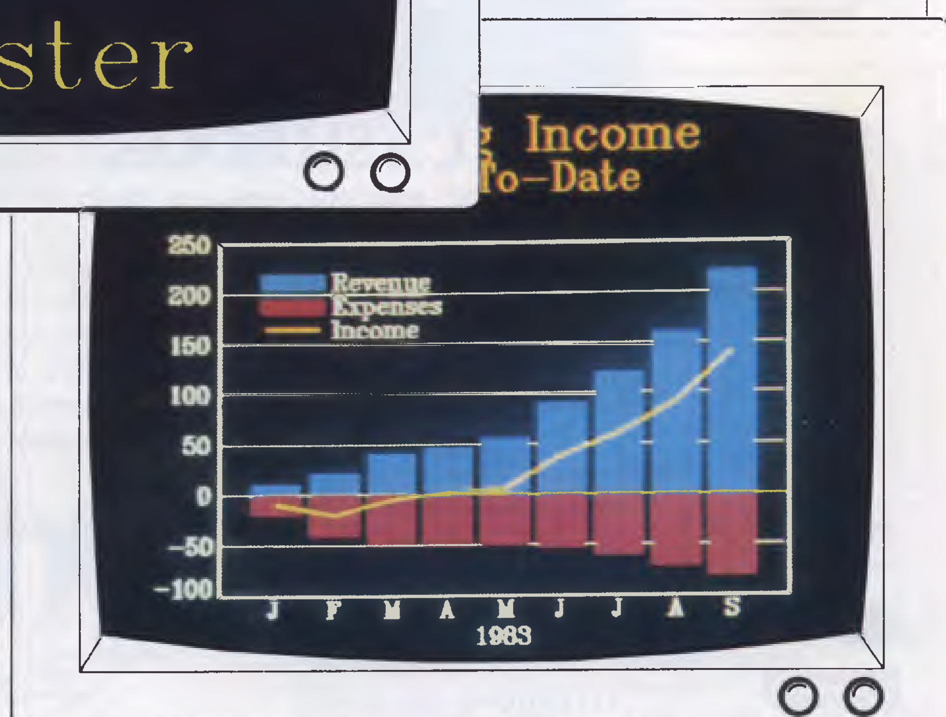
ter to ensure all users will be able to do something with it, even people whose machines provide only 256k of RAM.

Once it's all connected using the instructions laid out in the system documentation, you're ready to load up the DR software that comes with the package and begin generating some graphs.

The software comes on five disks: one Presentation Master tutorial, two versions of the DR Graph business graph preparation software (one for use with the 256k IBM PC system, and one for systems configured with 320k and more), one containing system utilities and the last offering DR Draw 'artist's' package for generating detailed non-graph drawings.



You have got a pretty comprehensive bundle of hardware and software in DR's Presentation Master system — everything from camera to slide mounting frames.



Presentation appeal

The software likely to have the widest appeal in Presentation Master is the DR Graph package. It offers eight formats for graph presentation: line, clustered, stacked, stepped, stick, scattered, pie and textual graphs. The software also allows you to display several graphs on-screen (and thus on-slide at once) as well as to vary the typefaces of the text used in legends, titles and axis labels.

The software is entirely menu-driven and requires little more than a 'fill-in-the-blanks' participation on your part. And if you want to type information into DR Graph, it also accepts spreadsheet files from other applications packages in several formats. The fact is, of course, that like many of the components in Presentation Master, DR Graph was designed as a stand-alone business graphics package and can thus be used as one — it isn't software 'thrown in' to sell the hardware.

The same can be said of the DR Draw package, which is reminiscent of many of the early drawing packages available for the Apple Lisa machine — and which is implemented on DR's OEM Graphic Environment Manager (GEM) systems. Like DR Graph, DR Draw is menu-driven and flexible. Without input from a mouse, joystick or light pen, it's difficult to do any kind of freehand drawing with DR Draw, but all kinds of predefined shapes (such as triangles, squares and circles) are included in the menu-driven command structure, along with extensive colour and fill commands.

DR Draw is designed to accept input from either the keyboard or a mouse, so if you want to free yourself from the cursor keys while creating a masterpiece, then a mouse would be a welcome addition to this suite. DR Draw will accept either the two-button or three-button mouse, where the mouse is used to replace the space bar and return key moves needed to pick and accept menu options.

It should be pointed out here that both DR Draw and DR Graph use the Digital GSX Graphics extension as their starting point, and that you'll see GSX being loaded into your system as either one of the packages boots up. GSX is a machine-code routine which sits on top of MS-DOS and handles communications between the applications software and the operating system. And it's the device-independent nature of the software which allows it to output to printers, plotters, disks, screen and the image-recorder.

There are two routes to provide information flow between the image recorder and your PC — the RS232 serial communication lead and the video lead — and they each serve a different purpose. The video lead sends a high-resolution colour video signal from the PC to the image recorder, where it is displayed internally. The RS232 port sends commands from your PC to the image recorder.

So when you press the button in your DR Draw or DR Graph software telling your PC to send the image to the image recorder, two things happen. First the image-recorder controls the degree of exposure and shutter speed of the camera through the commands sent to it through the RS232 line, and second the video signal from the PC sends the image down the line in its consultant colour components. This way the image can be exposed in each of its major colours and is controlled in a manner that yields the highest quality picture and the best colours.

As with a professional colour separation produced for magazine work, the image recorder exposes each of the colours in the image separately. Not only does this give improved quality to the colours, it also makes a bolder contrast with the background surrounding the main image in the print.

The final photographic element in all this is the Polaroid instant slide file processing machine. It allows you to develop and mount 35mm colour slides taken with the image recorder in about 10 minutes. Because the Minolta camera included with the system can use a variety of other non-instant film types (such as Kodak Ektachrome 64 film, Agfachrome ASA 100 and Fujichrome 100), you don't have to learn about the mounting and developing process if you don't need to. You would then just buy 35mm slide film in the regular way and pay someone else to process the slide film.

That might seem a foolish option, having spent all the money getting the Polaroid part of the system together, but Digital Research has made every effort to ensure this system is flexible.

System manipulation

So you've got a pretty comprehensive collection of hardware and software in the DR Presentation Master system, including everything from camera to slide mounting frames — but what you'll really have to decide is whether or not you have the time, energy and expertise to set the system up to use it.

If you're only doing an occasional slide presentation and you have trouble getting to grips with Lotus 1-2-3, then having to put your files into DIF formats for transfer to Presentation Master, and then having to use DR Graph, the image recorder and the film processor to photograph, then having to process and mount your slides, may all be too much for you.

But if you have either a great interest in photography or a great affinity for business graphics packages, then Presentation Master will be a joy to use. It's powerful, flexible and quite wide-ranging in its scope of abilities. You could find yourself thinking up reports just so you can use the system.

And if you're not overkeen on using the system yourself, you may have someone working for you who could take your spreadsheet information and turn it into slides for you using Presentation Master. The slides would be of high quality and would mean you wouldn't have to contract outside 1-2-3 artwork for presentations.

Presentation Master is well-executed and ties together existing computer hardware and software with off-the-shelf photography equipment in a highly workable business graphics presentation system. It's well worth the price if you do any volume of business with outside commercial artwork firms.

Presentation Master is distributed by Arcom Pacific. Telephone (07) 52 9522.

Micro surgery

When your business micro stops working, do some simple checks before you call the maintenance engineer — it may save you time and money. Jim McCartney shows you how.

To err is human, they say, but to foul things up completely, you need a computer. This is true inasmuch as it is practically always a waste of time to check the computer's arithmetic, but if you do have a breakdown, the system can lose or scramble a large amount of data in a very short time. This is why you should take copies every day, without fail, and keep backup copies of everything. If you don't, it proves that you are so new to the game that you haven't been clobbered yet.

Faults and remedies

Micros are complicated machines and break down in complicated ways. It is not, therefore, possible to give a comprehensive list of faults and remedies, but here is a list of common faults and some remedies which may help you to get out of trouble on your own without waiting for the engineer, or to avoid calling out the engineer on a wild goose chase.

Tests on disk systems also produce a useful printout for the engineer to look at, as faults have a nasty habit of vanishing when the engineer comes. The printouts will show that something really is wrong, and give him an idea of what it is.

Total hardware failure: this is most common in the early weeks, and is not unusual when the machine is just out of the packing case. New users are often convinced they are doing something wrong. If the system won't do anything in the book, call the engineer. This is why you should always insist on an engineer delivering the system, setting it up, and testing it thoroughly before departure.

Static faults: not very common, but if you find that when you walk across the office and touch a radiator, you get a mild shock, then static could cause problems on micros. Symptons: sudden jump in display, system crash or possibly auto-reboot, and always when someone is close to the machine and touches it. If crashed, re-boot — there is rarely any permanent damage.

Cure: antistatic mats around and on the desk. If it continues to give trouble, operators should avoid wearing clothes made from synthetic fabrics.

Power supply glytch: the sensitivity to this fault varies greatly between machines, but the majority will tolerate anything up to a positive supply interruption. Symptoms: as for static, but in the middle of normal operations. Often associated with a quick flicker in fluorescent lights. If someone is working a thermo-statted electric heater on the same socket, try at least plugging the thing in somewhere else. If you suspect this fault, ask your dealer about other users' experience on your type of machine. If you are having awful trouble with a new installation on an industrial site, try getting a good stabiliser on sale or return from your dealer for a week: it will help to establish if this problem really does exist.

The unconnected printer fault: I have frequently been called to inspect 'faulty' machines only to find that the printer has not been turned on, or (more probably) is not online when the machine reaches a printout. This usually results in a hang-up until someone twigs. Good software will warn you to turn on the printer. For your own applications, the following program procedure is useful:

Print message: "TURN THE PRINTER ON, STUPID!"

Line feed to printer (LPRINT)
Clear message
Start printout . . .

If the printer is already on, the message will be an unreadable flash and no action is needed.

A similar hang-up may occur if the interface or cable is damaged. The fault is

then identified by everything else being OK for printing.

If you have several micros, some of which run on serial printers and some on parallel printers, you can also cause this hang-up by using the wrong boot program.

VDU failure: VDUs rarely fail electronically without acting up a good deal beforehand. If your display suddenly blanks out, or fails to come up, the most likely problem is a bad connection. A faulty or disconnected VDU will allow the program to proceed normally, so if the machine boots satisfactorily otherwise and can give a printout, suspect the VDU.

Example: under MS-DOS or PC-DOS, boot a DOS disk, then type ALT-P or CTRL-P to direct output to the printer, and type DIR <return>. A printout of the directory suggests that the rest of the system is OK.

If your VDU is independent from the machine, try another bit of coaxial cable if you have one. If you have another machine, swop VDUs.

RAM failure: most common in early weeks, also frequently occurs after plugging in additional memory. Symptoms: some programs will run, some won't, or will crash in the middle. If you are working with spreadsheet-type applications, a RAM failure has a nasty way of corrupting your data while continuing, apparently, to function normally.

If the RAM fault is in the program/OS area, the chances are that you won't notice it on many programs; it depends what part of the program or DOS is being used. If the application does encounter this type of fault, it will almost certainly cause a complete crash or hang-up. Your display may or may not change, but the machine will not answer to the keyboard.

If the RAM fault is in the data area, it can be hard to pinpoint because it will give rise to inexplicable errors. If you have two machines of the same type, and your software runs on one but not the other, or only occasionally, then there is probably a fault of this type present. Typical symptoms other than a crash are wild and erratic error messages on an application which has hitherto behaved impeccably, but this can also be caused by a corrupted file on a floppy disk. To distinguish between the two, scan or dump the files in use when the fault occurs.

A typical RAM fault in a spreadsheet will cause a hang-up when the cursor, the printout or the recalculate attempts to traverse a faulty cell, but it may simply miscalculate it and proceed onwards. You can use a spreadsheet program to check out the large areas of RAM which a spreadsheet can occupy. A simple method is to set the cells so that each adds 1 to the previous cell until available memory is filled, then recalculate the model as many times as the keyboard buffer will allow.

Example: on VisiCalc, set A1 = 1 + A1, A2=1+A1, and replicate A2 down to A254, relative. Then B1 = 1+A254, B2=1+B1, and replicate to B254. Then replicate B1 to B254 from C1, D1, and so on, until memory is full. Adjust the last column to leave a minimum working space. Set up a window so that the start and end are both in view, and then press the recalculate key (!) and REPEAT until the machine beeps. On Apple VisiCalc this will recalculate 32 times without stopping.

Any RAM faults in the data area will cause aberrations in the finally calculated sum. A single recalculation will not always show up an intermittent fault.

A RAM fault is often no more than a single bit on a chip which is 0 when it should be 1, or *vice versa*. It most often occurs during the first 20 hours of use of new RAM; old machines rarely develop RAM faults if they are in a reasonably clean condition. RAM or other chip faults can occur because of overheating — many business micros now have a fan to keep the insides cool. RAM faults in the early stages can be transient and are notoriously liable to disappear when the engineer comes. You may find heat-sensitive faults which only show after a system has been running for 15 minutes or longer.

Basic afficionados can test RAM by writing a short program to POKE &HFF into all locations from LOMEM to HIMEM (leaving just enough space for the program and variables), and then reading them back to make sure they are still the same. Repeat using &HO, and repeat the lot half-a-dozen times. However, unless you have this type of thing worked out beforehand, the engineer will be with you by the time it's running correctly. This system is useful for home users.

Floppy drive failure: it can be very tricky to distinguish between disk failures and drive failures, so I'll deal with the two together. Firstly, on a busy system, you should clean the heads once a week without fail. Secondly, use a good-quality floppy disk. Thirdly, if the disk system appears to fail in any way, clean the heads and try again.

A physical disk failure (media failure) is usually sudden and permanent, and often occurs when end-of-day copies are being run, with a failure to copy. DOS error messages vary so greatly in style and quality, and depending whether they are encountered in a program or out of it, that it isn't possible to offer any comprehensive guide. One thing which they have in common is incomprehensibility to the average non-technical computer operator; another common feature is a generality so broad as to be almost meaningless.

MS-DOS and its relatives, working from DOS rather than through a language will, however, give you drive, track, sector and error data, which are extremely valuable.

If you get a disk copy failure, try the following procedures — in order of increasing desperation! — to recover the data and diagnose the fault.

Before you start, set the system to print out the DOS operations on listing paper — then you and the engineer will know what happened.

1 Retry the copy.
2 Clean the heads and try again.
3 Try a new or re-formatted copy disk — the error messages will indicate if the copy rather than the original is at fault.
4 Copy in opposite direction, for example from B: to A: instead of A: to B:. If this works, you may have a faulty drive.
5 Try copying on another machine if you have one.
6 If you haven't had a good copy so far, the original disk may be bad.

Prepare a new copy disk, and copy individual files from the suspect disk using, for example, COPY ★.★ B: under MS-DOS. The odds are that most of the files will copy but one or more will fail. If you get a couple of failures, copy any files further down the directory one at a time and isolate the failures. There's a fair chance that the bad files were not in use since the last copy you made; if so, you can get them from there. During this procedure, a failure to format or initialise the new copy disk, or a failure to copy any files at all, suggests a drive failure.
7 If current data files won't copy, but there is no evidence of a drive failure, you have lost the work since the last copy. Produce a new set of master disks from the last copies and start again.

Floppy drive failure should be suspected on new systems or on systems which have been in use for some time, especially those which consistently access the data disks. This simply causes wear and tear on the mechanical systems, and sooner or later they start to behave oddly, usually on an intermittent basis. Look out for the following symptoms during daily running on well-established software:

1 Error messages related to DOS.
2 Error messages indicating that a wrong data type has been read from a file.
3 Garbage in the files — sometimes identifiable in printouts or displays as parts of other files; also numbers in scientific notation with huge indices, for example 7.7384558462 1E-34 (assuming that you don't normally deal with such numbers).
4 Error messages indicating that a record or file cannot be found or does not exist, or that the record number is invalid. This is usually a result of reading false links or pointers.

The cause of these faults may be the disk drive reading from a wrong track, or from a file which has had a wrong sector overwritten on it. If you suspect this, print out a dump of any of the files which might be affected; a length of file which is totally foreign proves the presence of a drive fault, and you need test no further. The printout will also serve to convince the engineer that you really do have a problem.

The program in Fig 1 will read files and print the contents in hex, ASCII characters, or both. It was written in MS-Basic on a Sirius with an Epson printer, but should work with very minor modifications on any hardware using any recent version of Microsoft Basic, on any MS-DOS, PC-DOS or CP/M type of file. It works by pretending that the file, regardless of contents, is a random access file with a record length equal to line length. It will read past the end of sequential and text files, in which the end is otherwise indicated by hex 1A. If it reads a BSAVEd file, for example from an array, the first few bytes may be non-data.

A disk containing corrupt data in this way is to be distinguished from a disk which is physically damaged or worn out. If the corrupt data is caused by the drive, the disk can of course be re-used and it will copy perfectly. The substantial

INTRODUCING IBM
Compatible Hi-Res Monitors
from QUBIE'

The HR 39 and HR 134 Monochrome Monitors are direct replacements for the IBM Model 5151 Monochrome Display. The HR 39 features a GREEN phosphor screen, while the HR 134 an AMBER screen. Both monitors plug into the IBM Monochrome Display and Printer Adaptor (or compatible) card.

The HR31 200 Colour Monitor is a direct replacement for the IBM Colour Display. It plugs into the IBM Colour/Graphics Adaptor (or compatible) card. The HR31 200 features **0.31mm dot pitch and a black matrix picture tube.** This special tube reduces glare and enhances RGB colour to ensure superb picture quality.

All three monitors include a tilt and swivel base. Use your Qubie' monitor as you would the IBM, it will in no way affect the normal use or function of your Personal Computer.

Technical Data - HR31 200

CRT Size:	14" Diagonal (34cm)
Tube:	Black Matrix
Sync-H. Scan Frequency:	15.7kHz
V. Scan Frequency:	60Hz
Signal Type:	RGB I: TTL Level Positive Sync H/V: TTL Level Positive
Display Size (H x V):	245mm x 170mm
Retrace Time (H x V):	0.5Ms x 0.4msec
Resolution:	640 x 200 lines
Input Terminals:	9 pin "D" type connector
Dimensions:	11"(H) x 15"(W) x 13"(D) 266(H) x 367(W) x 318(D)mm
Shipping Weight:	15.9kg

Model HR31 200

Models HR 39 & HR 134

Technical Data - HR 39 & HR 134

CRT Size:	12" Diagonal (29cm)
Phosphor:	HR 39 (Green); HR 134 (Amber)
Sync-H. Scan Frequency:	18.432kHz
V. Scan Frequency:	50/60Hz
Signal Input:	Video - TTL Level Positive Sync. H - TTL Level Positive Sync. V - TTL Level Negative
Video Response:	20MHz
Display Size (H x V):	203mm x 135mm
Display Time (H x V):	44Ms x 18.99msec
Resolution:	Centre 1,000 lines Corner 800 lines
Display Formats:	9 x 14 matrix, 2000 characters in 80 x 25 format
Input Terminals:	9 pin "D" type connector
Dimensions:	10.5"(H) x 15"(W) x 12"(D) 257(H) x 367(W) x 294(D)mm
Shipping Weight:	11.36kg

All models: 240V AC/50Hz

DEALER ENQUIRIES: QUBIE'

9/62 Blackshaw Avenue, Mortdale, 2223 - Phone: (02) 579 3322

America's leading distributor of IBM PC accessories is now supplying your IBM dealer locally. . .

Also available: IBM key boards, multifunction memory boards and modems.

danger is that if it's not dealt with, the corrupt data will be copied to all your current copy disks as well in the course of a few days. Anything that can go wrong sooner or later will, so it's a good idea to archive your master disks monthly in an ongoing system.

Twin floppy failure: the following procedure applies to MS-DOS, and so on; those familiar with the CP/M family can easily translate to a SUBMIT file.

To thoroughly test disks and twin floppy drives, prepare a special system disk with DCOPY on it and otherwise containing as much old data as you can collect, so as to leave it nearly full. If you don't have enough files to fill a disk, copy the same file under several different names. Leave enough space to write the file TEST.BAT under EDLIN:

```
1 DCOPY A: TO B:/E
2 DCOPY B: TO A:/E
3 TEST.BAT
4 Z
```

Running this batch will copy from A: to B:, back again and repeat until someone intervenes. You should keep a master copy and a spare of the disk, as a failure wil probably leave you with a half-copied working disk. To use the disk, boot it in A: and direct the output to the printer, so that you have a record of what happened. The maintenance engineer will appreciate this.

To test the drives, put a good blank or spare used disk in B: and execute TEST.BAT from A: To test a suspect disk, first copy any files you need from it, then use this in B: Half-a-dozen good copies in each direction can be taken as evidence that there is no problem. A bad disk will give errors on one or perhaps two adjacent tracks all the time. A bad drive will generally distribute errors more or less at random over tracks and sectors, but they will be on one drive or the other and you may find them all on one side on double-sided systems.

Neither the disk nor the drive fault will necessarily occur all the time, but a disk fault is the more likely to be consistent. A drive fault may occur only once out of three of four times, or even disappear completely under test.

The erratic nature of this type of fault always makes it seem to vanish when the engineer comes — this is why you need to do several copies in each direction, and to obtain some printout as evidence.

```
100  FILE INSPECTION UTILITY "DATADUMP"
110 ON ERROR GOTO 370
120 PRINT CHR$(27)"E": L=0: BL$="                   "          ESC code clear screen
130 PRINT "FILE CONTENTS PRINTING UTILITY 'DATADUMP'": PRINT
140 FILES: PRINT
150 INPUT "FILE NAME (ALT-C to terminate):  ",FI$
160 INPUT"Line length : ",LIN%
170 INPUT"Start at line : ",START%
180 INPUT"Finish at line: ",FIN%
190 INPUT"Do you want text, hex, or both? (t/h/b):",Q$: -
         IF Q$<>"t" AND Q$<>"h" AND Q$<>"b" THEN 190
200 PRINT:PRINT"PRESS ANY KEY TO STOP...."
210 OPEN "R",#1,FI$,LIN%: FIELD #1, LIN% AS B$
220 LPRINT CHR$(15)        'condensed print for EPSON
230 LPRINT "DATADUMP.BAS utility on file:     "FI$
240 LPRINT "Line (record) length = ";LIN%;"  Starting byte = ";
    (START%-1)*LIN%+1
250 LPRINT
260 FOR J% = START% TO FIN%:
         A$=INKEY$:
         IF A$<>"" THEN 350
270      GET#1,J%:
    IF Q$="t" THEN 300
280      LPRINT LEFT$(STR$(J%)+BL$,7);:
         FOR K%=1 TO LIN%:
         B%=ASC(MID$(B$,K%)):
         HX$=RIGHT$(("0"+HEX$(B%)),2)+" ":
         LPRINT HX$;:
         NEXT: LPRINT
290 IF Q$="h" THEN 320
300      LPRINT LEFT$(STR$(J%)+BL$,7);:
         FOR K%=1 TO LIN%:
         B%=ASC(MID$(B$,K%)):
         IF B%<33 OR B%>126 THEN B%=46
310      LPRINT CHR$(B%)"  ";:
         NEXT: LPRINT
320 IF (J%-START%+1) MOD 24 = 0 AND Q$="b" THEN LPRINT CHR$(12): GOTO 340
330 IF (J%-START%+1) MOD 48 = 0  THEN LPRINT CHR$(12)
340 NEXT J%
350 LPRINT: LPRINT
360 RUN
370 IF ERR=62 THEN B$=CHR$(255): RESUME NEXT
380 PRINT"ERROR: ";ERR;" LINE: ";ERL
390 END
```

Fig 1 File inspection utility

If you get apparent evidence of a bad disk, repeat the same test with a good disk in case the drive is causing a track-related fault — this is unusual but sometimes happens.

If you do find a bad disk, inspect the disk surface to see if you can spot the cause. Look out for a damaged track which could be caused by a faulty head or by trapped grit. You may see crumple marks caused by the user pushing in the disk carelessly and bending it, or other evidence of mishandling, or you may see nothing at all.

If you have evidence of a faulty drive, don't use the machine again until the engineer has attended to it.

Hard disks: one of the main characteristics of hard (winchester-type) disks is their reliability as compared with floppies. A drive which is in any way unserviceable will, in most cases, fail to come up at all. If it does come up, and you find corrupted files or lost data, make sure that the problem is not in any way caused by associated floppies (for example, copying bad files from floppies). Copy any suspect files from the hard disk to a good floppy; a copy failure caused by the hard disk will be reported by the DOS. Likewise, you can use the file dump program in Fig 1 for closer inspection. If you continue to have problems, there is unfortunately nothing you can do except call the engineer.

Restoring or erasing corrupted data: 99 times out of 100 this is a lost cause. It's easiest to go back to the last good copy and re-do the day's work when the system is fixed. To do anything about a lost block requires an intimate knowledge of the DOS and a set of very good DOS utilities, and is really only useful where you can do without the lost block, for example in text files from word processors.

If you have a random access file in which only a short length is corrupted, as revealed by DATADUMP, it is quite easy to write a short basic routine to replace the bad records with blank records, but unless you know the full details of the file structure (that is, as related to other files) this can land you in as much trouble as before. The golden rule here is, unless you know exactly what you are doing, and all its implications, and can do it faster than it would take to re-work it from existing copies, then forget it. The other golden rule is — always make copies.

END

The Computer Trader

Phone Order Specials
(03) 266 1995
Phone now!

SUPER TRADER SAVINGS

IBM-PC'S

Last chance for "TAXMAN" to subsidise your P.C.
Purchase IBM, PC from

$3495

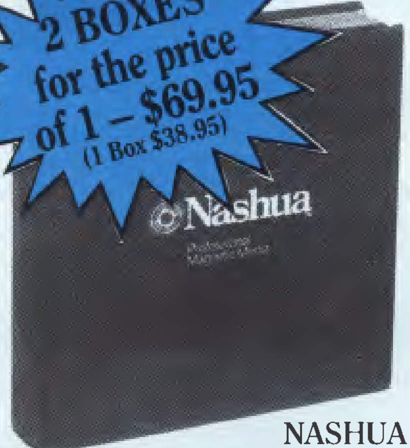
IBM, PC-XT from

$5995

Accounting Systems & Installation can be arranged in Vic. & N.S.W.

All items subject to availability.

2 BOXES for the price of 1 – $69.95
(1 Box $38.95)



NASHUA
Professional Diskettes
5¼" DSDD 48 TPI. 5 year warranty. Unique Quality Circle Control.

1.	LOTUS 123	$485
2.	SYMPHONY	$625
3.	dBASE III	$585
4.	WORDSTAR 2000+	$595
5.	OPEN ACCESS	$495
6.	CROSSTALK	$145
7.	COPYWRITE	$ 95
8.	SIDEKICK	$ 89
9.	TK SOLVER	$495
10.	MULTIMATE	$395
11.	FRAMEWORK	$495
12.	FLIGHT SIMULATOR	$ 65

All items subject to stock availability.

A MUST FOR ALL dBASE II USERS

Increase Development Productivity and Processing throughout with

dTOOLKIT

Containing many dBASE III facilities ...and more

- **dLIBRARY** – OBJECT CODE ROUTINES:
  Arrays, extra 256 Memory Variables, dBASE III Data Facilities, Amount in Words, DOS Type, Soundex Codes, Check Digits, Random Numbers, SQRT, Tolower, Capitalize, Isalpha, Encrypton, compression...and Many More.
  List **$55.00**
- **dBASIC** – FUNCTIONS:
  Process dBASE II Data and Index Files using CB80 or CB86 Compiled BASIC.
  List **$45.00**
- **dFIX** – UTILITY:
  Fix Corrupted Data Files
  List **$40.00**
- **dCRYPT** – UTILITY:
  Encrypt and Decrypt Data Files
  List **$30.00**
- **dMENU** – dBASE RUNTIME MENU SYSTEM:
  Maintenance Routines, Menu Hierarchy, Security.
  List **$40.00**

CP/M-80, CP/M-86, MS-DOS, PC-DOS

TOTAL PACKAGE FOR $99

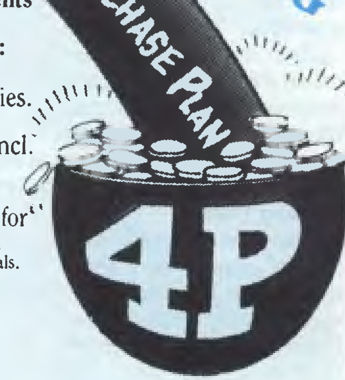
INTRODUCING PREFERENTIAL PRICE PURCHASE PLAN

The pot at the end of the rainbow for our regular clients is our Preferential Price Purchasing Plan (4P). All our serious micro computer clients (not just companies) who participate in 4P are assured of:

1. *Trade discounts on all hardware, software & supplies.
2. Extended warranties.
3. Free door to door delivery (incl. H'ware).
4. Overseas buying service.
5. Inexpensive software rentals for evaluation or occasional use.

*Note: Does not apply to Supertrader Specials.

OUR GUARANTEE TO 4P CLIENTS IS TO SERVICE *ALL* YOUR MICRO REQUIREMENTS FOR LESS



The Computer Trader
3 Bowen Crescent, Melbourne. 3004

BANKS' STATEMENT

Under pressure

The old adage 'Nothing lasts forever' bodes ill for Unix: other systems are threatening its position as a 'standard'. Martin Banks observes the moves.

If you look at the way the average company is organised, it's possible to state with a fair degree of theoretical certainty that there ought to be a good future for small multi-user computer systems. After all, the way they can provide small clusters of related work-stations, coupled with the ability to network the clusters to build bigger systems, is a direct analogy of how most companies are departmentalised.

If this is the case, then it would seem reasonable at first glance to state that Unix is the operating system to go for and all else is dross.

However, it doesn't work out that way in practice. There are several different versions of Unix around, with even more Unix lookalike systems of varying degrees of compatibility. Although portability between machines is there in theory, in practice it just isn't that simple.

Despite this, it is still possible to suggest that Unix should become the dominant operating system for small to medium-sized business machines. Virtually any application that requires more than a solitary personal computer has to be a potential target for Unix-based applications.

One of the problems now facing the system is the publicity surrounding it. Having come from the rarefied air of AT&T's Bell Laboratories and the world's universities, it has been ill-equipped for the cut and thrust of the commercial world. In more practical terms, this has also meant that there have been few commercial applications packages available.

This might not have mattered too much if it hadn't been for the IBM, which inconveniently introduced the PC just at the time when lots of micro manufacturers were looking closely at the potential market for supermicro machines running Unix.

The IBM machine, together with its host of clones and close compatibles, is the market to which the majority of the applications software has been written. Packages for Unix have, by comparison, tended to be fairly specialised and therefore expensive. Ironically, the world and its uncle have jumped on the IBM band-wagon; only Apple stands out as a major contender against the stream. Where Unix could have been a rallying point for all manufacturers wanting to get out from under IBM, it has suffered instead from everyone wanting to stay under the big blue umbrella.

The MS-DOS environment is now the *de facto* standard operating system for small micros — stand-alone machines at least. This still leaves some potential in the multi-user area, a potential that is not always satisfied by networking stand-alone machines together. They may appear to work well like that, but for many applications it will not be the best option.

Can Unix move in here? If it's going to then it has two years at most in which to achieve a worthwhile market-share. In that time, the fragmentation of the market for Unix systems must be cleared up. For example, although AT&T now claims that Unix System V is *the* standard, there is still a lot of System III around. At the same time, Microsoft's Xenix is claiming the lion's share of the low-end Unix market-place.

In this context it's not surprising then that Microsoft and AT&T have recently signed a deal to engineer a unified front, bringing Xenix and Unix into line with each other. Digital Research (DR), which recently completed a port of Unix onto the 80286 processor, has developed a library of 15 or more applications programs to run under the system. It has decided to drop the idea of marketing them however, primarily because the fragmented state of the market makes the library's commercial viability suspect.

There will be some scope for more applications coming through as the 286 device becomes more prevalent. This may be helped by such things as language support for both Unix and the more prevalent IBM environment on the 286 processor. DR, for example, has a range of language products available for both the 286 and the Motorola 68000 that are source code-compatible with both Unix and its own Concurrent DOS. This, the company claims, will provide a funnel for software developers to go from one environment to the other.

But what is most likely to deal Unix its severest blow is the Apple Macintosh. This machine, with its fancy graphics, user interfacing, mouse *et al* has demonstrated that there is an alternative to the old character-based user interfaces typified by the dreaded A> prompt that we all know and love. Here is a machine that the non-expert can easily use because he can 'see' what he wants to do from the icons shown onscreen.

Couple this approach with the move towards high-resolution colour graphics, and you have a powerful tool for software developers to make their products really user-friendly for the first time. This is, however, a software technology that demands a considerable amount of local processing power to make it work properly. Unix, being dedicated to multi-user operations, will not always be available on machines that pack enough power to provide this for a large number of users, especially at the low end of the market where competition from machines with such capabilities will be found.

Such facilities are already available for the IBM PC environment. DR has already announced its Graphics Environment Manager (GEM), while Microsoft has Windows due this winter. GEM is interesting in multi-user terms as well: it will be part of DR's Concurrent DOS 286 for the 80286 device, which is itself a multi-user operating system.

The key to developments, as usual, will be what IBM does. With Top View it has shown a direct interest in the systems software market; it is unlikely to have missed the point of the Macintosh, GEM or the upcoming Windows. If it does produce something graphical by the end of the year, it will not only be the leader, but will completely crush the opposition. If it doesn't, then GEM or Windows could become market leader.

Either way, the place of Unix at the small systems end (especially with the emergence of Concurrent DOS 286 with GEM an integral part) looks to be increasingly insecure. Despite all the fine words said about it, everyone is hedging their bets like mad on the subject of Unix.

Illustration by Bill Sanderson

How woul
five times the no
and two free Mic
with your new
Macintosh?

We thought we would make the Apple Macintosh even more irresistible.

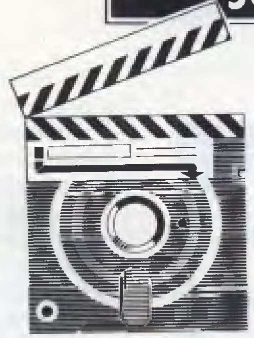
Of course, offers this generous could only come from Robs Computer Centre.

Phone them now on (03) 791 2900. You will find their prices just as attractive.

robs
COMPUTER CENTER
295 Thomas Street, Dandenong 3175
Phone (03) 7912900

ld you like
rmal warranty,
rosoft packages

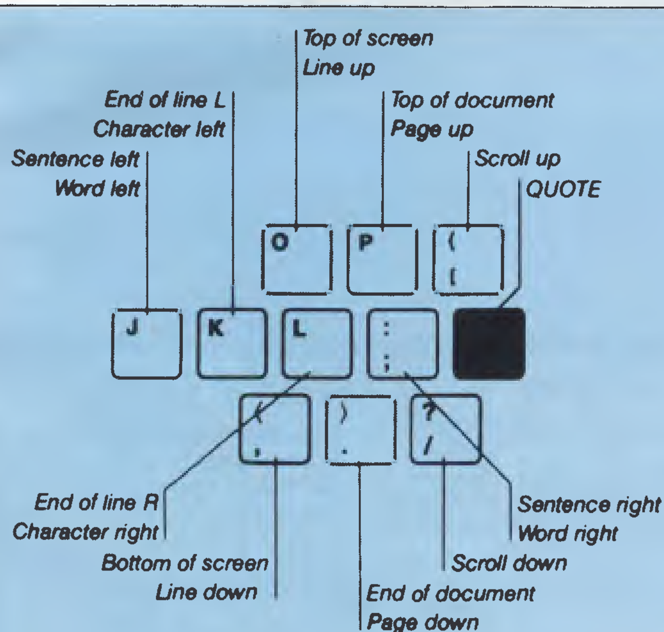
IK 84



Microsoft Word

Word, Microsoft's word processor, has at last found a machine to complement its very individual features — the Macintosh. But is it a perfect marriage? Kathy Lang finds out.

Among the packages available on the Mac, Microsoft's Word is rare in having been available for quite a while, rather than being specially written. Yet, in some respects, Word could be said to have been waiting for just such a machine to come along — many of its features are beyond the capacity of most hardware to provide a structure which will make them as appealing as they deserve. For example, Word provides the ability to use a wide variety of fonts and font sizes, and to display these on the screen, but few systems have the ability to match this functionality.



- Hold down Option-Command and press the appropriate key for the insertion point or scrolling movement shown in black.
- Hold down Option-Command, press the QUOTE key, then press the appropriate key for the movement shown in red.
- Hold down Option-Command-Shift and press the appropriate keys to select text from the beginning insertion point to the end point.

Fig 1 Keypad control keys

In other ways, the Mac might have some drawbacks as a word processing system: the keyboard has a very hard feel, more like an old-fashioned terminal; the screen is rather small for comfortable word processing; and there are no special function keys other than the mouse.

Editing

When you first load a document into Word, the display shows the main Word menu names across the top, and a conventional Mac window across the screen showing horizontal and vertical scroll bars plus the name of the document being edited. No ruler is displayed unless you request it, nor are there any indicators of horizontal or vertical position such as line number. The page number is shown, but may be inaccurate unless you have recently revised the pagination.

Unlike the majority of systems on which Word is available, the Mac has only the mouse with which to move the cursor — there is no conventional arrow key pad with the system. This could mean that professional touch-typists would spend a lot of time moving their hands from keyboard to mouse and back. To get round this, Word allows you to use a system of key combinations based on the Command Option keys; Fig 1 shows the table from the Word command key summary. Despite my years of acquaintance with the alleged horrors of WordStar control keys, I found it easier

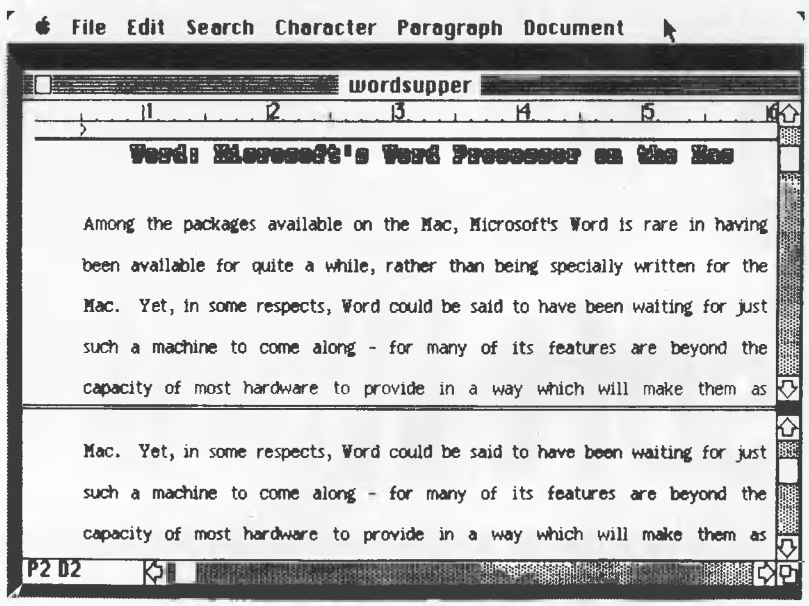


Fig 2 Display of two parts of same document, using split window

and quicker to use the mouse (except for scrolling long distances), but with practice the key layout should become familiar. No stickers are provided to aid memory. With either system, it is reasonably quick and easy to move the cursor (which Word calls the insertion position because it dictates where text will be inserted). If you prefer to overtype, you must first select the words to be overtyped, and then enter the replacement text.

This approach to text selection is used extensively to control the action of a variety of tasks — deletion, emphasis, copy, movement to another part of the document via the Clipboard, and so on; this can be done either with the command key combinations I have mentioned, or with the mouse. While I dislike the use of mice in a context where you are doing a lot of keyboard entry, it does provide a direct approach to movement, comparable with cursor controls in other environments, and it proved easier than I expected to locate the cursor exactly. It also means that you can use one technique across the whole range of Mac packages.

The one thing which did annoy me about editing with Word keys — or rather with Mac keys — was the position of the SHIFT key. Haven't we suffered long enough from IBM's idiocy in placing it in defiance of the typewriter conventions? It's not Microsoft's fault — but it is something you should bear in mind when considering word processing on the Mac.

Word provides other editing features in a sensible, easily-used way. You can search for specified words or phrases, and replace them if necessary; you can choose to ignore case, to look for complete words only, and to replace just one or all of the matching strings of characters. To copy information between documents, one approach is to use the Clipboard, which is preserved between applications and between editing sessions, and can hold a lot of information — how much depends on the amount of space on your system disk. An alternative is to set up two windows, one containing the source document and the other the target, and copy between them, again via the Clipboard, but without the need to close your current editing session between marking and copying.

Word is reasonably forgiving of errors: for example, the most recent action can be reversed with the undo key, and a back-up copy of each file will be kept automatically on request. Text which is cut (either for deletion or for copying elsewhere) goes into the Clipboard, from whence it can be retrieved if you find you have made a mistake, provided you have not subsequently done another cut, since Clipboard can hold only one.

Word provides a glossary facility to store frequently-used text, so that it can be recalled using an abbreviated name. The procedure for recall is to type the name of the abbreviation and press COMMAND and BACKSPACE, so even with single-letter names it's not a procedure you would use for very short words. Nor did there seem to be a way to include Word command sequences in the glossary — it is strictly a text storage medium. On the other hand, display attributes, including font and emphasis, are not only stored with the abbreviation, as they should be, but also remain in force when recalled, until the previous setting is restored. I was doubtful as to whether that would be my preferred option. Probably the best approach would have been to allow a choice when the abbreviation is first entered into the glossary.

Text formatting

Word provides a good range of text formatting options, either by using a ruler or specifying margins and indents through 'dialogue boxes' (question-and-answer displays). If you use the dialogue route, you can specify formats in a variety of units — the default is inches and decimal portions thereof, but you can choose centimetres, point, or 10 or 12 pitch if you prefer. Formats can be preset

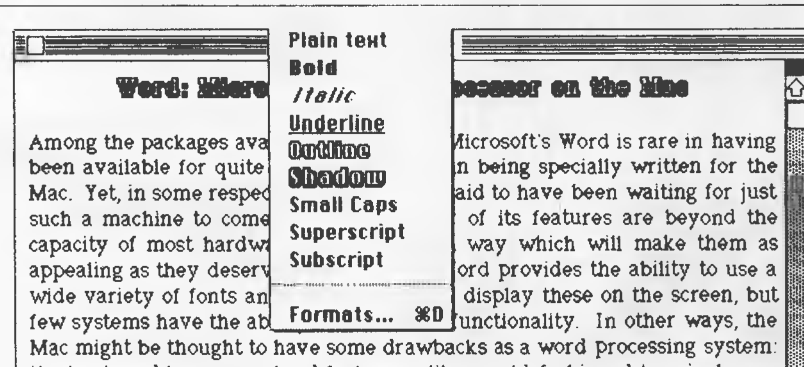


Fig 3 Pull-down menu of character styles

HITACHI

HITACHI PERSONAL COMPUTER MODEL S1, MB-S1/10

- HIGH SPEED C.P.U., 68B09E
- VERSATILE COLOUR GRAPHICS FUNCTION
- 1M BYTE ADDRESSABLE MEMORY AREA
- NEW HITACHI LSI TECHNOLOGY
- HIGH RELIABILITY
- PRICES FROM $995.00

Dist by ILC Systems
Ph (02) 920 1361

Dealers

BENSON Computers Pty. Ltd.

169 Victoria Parade, Fitzroy, Vic 3065

Enquiries (03) 417 6999. Order Toll Free (008) 33 4854

MAIL ORDER • PHONE ORDER • CALL-IN

Skai-64 Disk Drive

For

Commodore 64

1. Over 2000 programs successfully tested
2. Over 5000 drives already sold
3. Money back guarantee
4. Metal case, cool operation
5. Metal backstop for maximum reliability

$349 inc.

THIS IS A GREAT PRODUCT!

We are getting rave reviews from all of our Skai-64 customers, and they are referring their friends . . .

WIN A PRINTER

5 Admate DP100 with Xetec I/F's to be won.
FIVE winners will be drawn from the first 300 retail sales of Skai-64 after release of June APC.
Results will be published in APC.

NAKAJIMA NP 2200
OLYMPIA 165 CPS

$449 inc

Epson compatible, tractor/friction feed, all electronic touch button, heavy duty 165 cps in draft, 90 cps in NLQ (Near Letter Quality).
Use inexpensive Epson fabric ribbon.

NLQ — Near Letter Quality from a dot matrix printer!

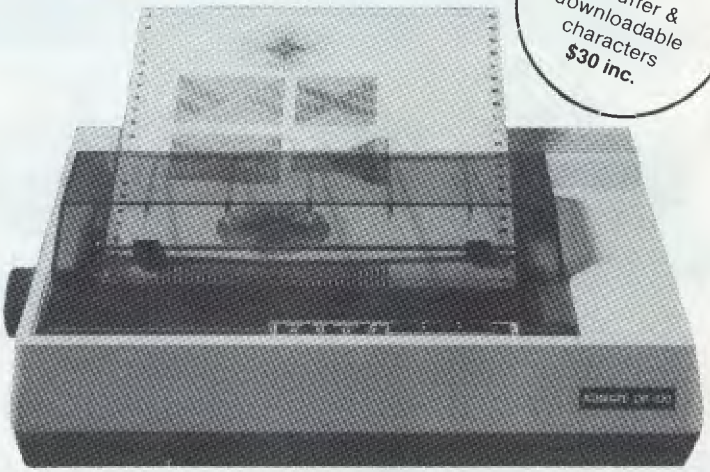
100DP — FEATURES

Printing format	Alpha-numeric — 7 x 8in 8 x 9 dot matrix field. Semi-graphic (character graphic) — 8 x 8 dot matrix. Bit image graphic — Vertical 8 or 9 dots parallel. Horizontal 640 dots serial/line.
Character size	Normal size — 2.22(W)x2.8(H)mm=0.087"x0.11" Condensed size — 1.11(W)x2.8(H)mm=0.044"x0.11" Enlarged size — 4.44(W)x2.8(H)mm=0.175"x0.11" ELITE size — 1.8 (W)x2.8(H)mm=0.07" x0.11" Enlarged ELITE — 3.6(W)x2.8(H)mm=0.14"x0.11" Super/subscript — 2.22(W)x1.4(H)mm=0.087" x 0.0055" Semi-graphic unit — 2.54(W)x2.8(H)mm=0.1" x0.11".
Character sets	ASCII characters — 192 (96 normals and 06 italis). JIS characters — 160 (64 katakanas and 96 alpha-numerics). Semi-graphic units — 103. International Specials — 2 for U.S.A., 1 for UK, 8 for German, 8 for French, 4 for Swedish. 2 for Italian, 6 for Spanish, 48 for Greek, 6 for Danish and 2 for Japan.
Printing Speed	100 C.P.S. for normal size print. 200ms for line feed.
Columns/line	Normal — 80 columns. Condensed — 142 columns. Enlarged — 40 columns. Condensed/double width — 71 columns. 'ELITE' — 96 columns. Enlarged 'ELITE' — 48 columns.

ADMATE DP-100

Discounts available for bulk buys (5+)

SPECIAL FEATURE! 4K buffer & downloadable characters **$30 inc.**



THE ADMATE 100DP

10 pitch, 12 pitch, proportional
100 cps, 8 character sets, graphics

$369 inc.

Also:
Epson printers
15 inch printers
Monitors
Logitec PC/XT

6 MONTHS WARRANTY ON ALL PRODUCTS

Please also enquire about the Logitec PC or XT which runs IBM PC software, plus Videotext, networks, multi-user, accounts etc.

FREE FREIGHT AUSTRALIA WIDE

before you start to type, or they can be changed later by selecting the text to be affected. Indents can be absolute, or relative to the most recent indent; you can indent just the first line of a paragraph, or have hanging indents with all but the first line indented. If you insert text into existing material, the paragraph will be realigned as you type. This is good for authors when creating text but not so good if you are editing from a printed draft, since the reformatting throws away the visual match between printed and displayed versions.

Rulers and formats are stored with the document, and are automatically activated as you move between different parts of the document. It isn't usually possible to tell in advance what changes will take place, though, since Word doesn't automatically display rulers or show when they change, nor are other markers such as carriage returns at the ends of paragraphs automatically displayed. All these markers can be displayed if you wish — I left the ruler displayed so that I could anticipate changes of layout — but you can't really leave the display of carriage returns set, as with it comes the display of spaces that you've typed. You can inspect line endings as they are displayed; Word doesn't offer any help with hyphenation, but you can inset 'soft' hyphens to split over-long words.

Tables can be implemented using tab markers set in the ruler; tab characters are actually inserted into the text, so if you want to change the format of a table, all you need to do is select it, change the tab positions, and Word will do the rest. Tabbed fields can be left or right aligned, centred, or aligned on the decimal point for numeric fields. You would be most likely to feel the drawback of the narrow screen when typing tables: you can display six inches worth of text, so for most ordinary documents the whole line would be displayed. But for wide tables, especially those to be printed on paper inserted sideways, part of the table would extend well beyond the width of the screen.

One of Word's strongest features is its ability to display and print text in a wide variety of character sets, with several different fonts for each set. These are displayed on the screen as they will print, as are emphasis (which includes bold, underlining, two forms of shadow print, and small capitals), justification, centring, and the layout features I've already mentioned. Indeed, Word is about as WYSIWYG (what you see on the screen is what you get on the printer) as it is possible to get. With such a wealth of goodies, it seems churlish to argue that you can take WYSIWYG too far, but it's true.

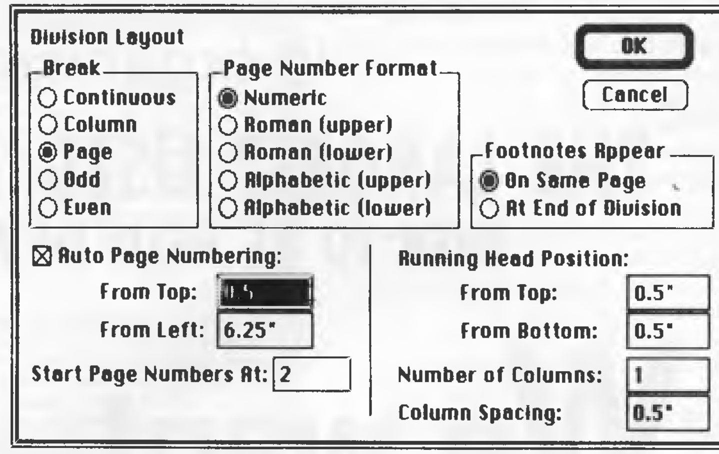


Fig 4 Dialogue box for layout of headers, footers and page numbering

The trap into which Word falls is to include line spacing in its WYSIWYG approach — if you request double-spaced lines, then that is what you get on the screen. Given the limitations of a screen display, there is very little text shown. I got round this by using single-spacing until my document was complete, then changing to double-spacing before paginating, but that doesn't help much if you subsequently need to make further changes.

The line-spacing features themselves are also rather patchy. If you want either single or double-spaced text, then whatever font you use, Word will adjust the line heights correctly. If you need other spacings, you are on your own — you have to adjust the line height to an appropriate unit, taking into account the font(s) you were using. It's good that this capability exists — too many packages are getting 'bossy' in this area — but the provision of one-and-a-half spacing as a third option would save a lot of grief.

Pagination may be left until print time, or explicitly requested so that you can see where page breaks will occur. Word attempts to avoid widows and orphans by ensuring that you will not have just one line of a paragraph printed alone on a page, but this method cannot prevent headings being separated from text where a blank like intervenes. To avoid this, you can insert manadatory page breaks, or use instructions to 'keep the next n lines together'. Page-breaks are shown on the screen with remarkably discreet equal signs in the left margin — I found them quite hard to spot at first.

You can set up running headers and footers in your text, and change them at will; these may include page numbers which are updated automatically. Word copes with changes of page formats by separating your document into 'divisions', each with its own page formatting. You can have one for the whole document, or several within a document. Provisions for formatting within a division are very flexible. They include the ability to have true footnotes, which may be printed either on the same page as the

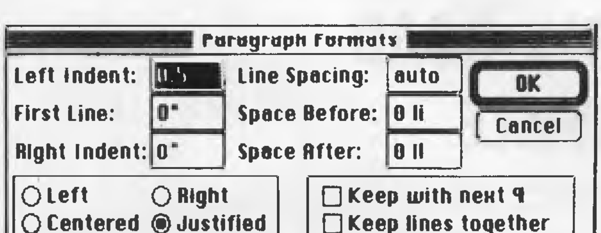


Fig 5 Paragraph formats dialogue box

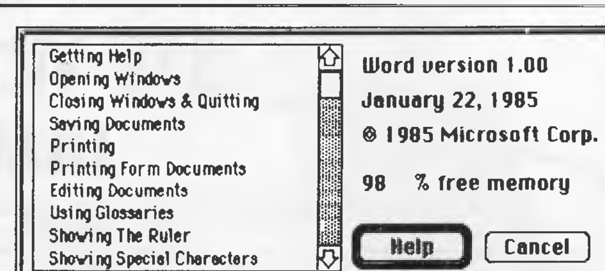
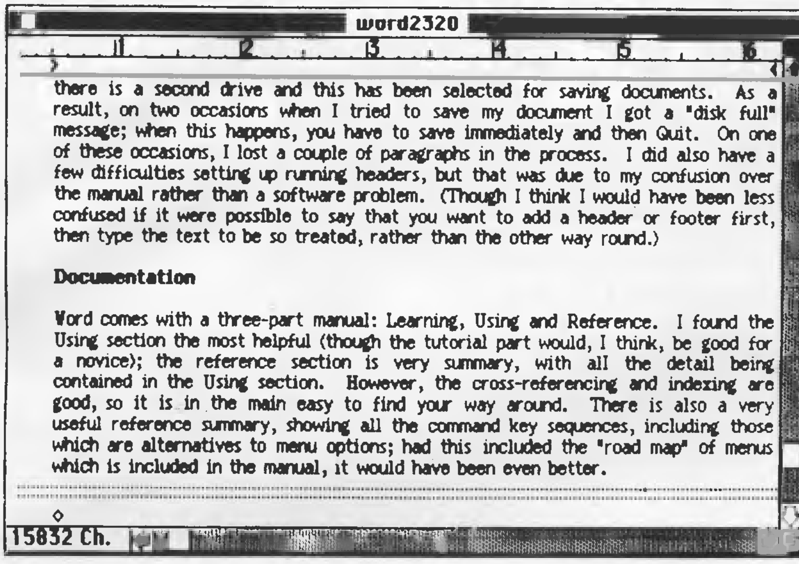


Fig 6 Part of the help menu


word2320

there is a second drive and this has been selected for saving documents. As a result, on two occasions when I tried to save my document I got a "disk full" message; when this happens, you have to save immediately and then Quit. On one of these occasions, I lost a couple of paragraphs in the process. I did also have a few difficulties setting up running headers, but that was due to my confusion over the manual rather than a software problem. (Though I think I would have been less confused if it were possible to say that you want to add a header or footer first, then type the text to be so treated, rather than the other way round.)

Documentation

Word comes with a three-part manual: Learning, Using and Reference. I found the Using section the most helpful (though the tutorial part would, I think, be good for a novice); the reference section is very summary, with all the detail being contained in the Using section. However, the cross-referencing and indexing are good, so it is in the main easy to find your way around. There is also a very useful reference summary, showing all the command key sequences, including those which are alternatives to menu options; had this included the "road map" of menus which is included in the manual, it would have been even better.

◇

15832 Ch.

Fig 7 Text displayed, the diamond indicates end of document

reference or at the end of the division.

Repeated text

The glossary features for handling abbreviations within documents should be adequate for most needs, where it is necessary to directly include the text in the document. For example, this feature could be used for standard contract work, such as conveyancing or personnel records. The ability to store such abbreviations, either for the duration of the session or permanently on disk, and to have a number of separate glossaries on disk, gives a reasonable degree of flexibility.

For repeated text of a different kind, Word has a mailmerge feature. This allows you to set up a standard document such as a letter, specify the points within it which are to vary according to the recipient, and merge this letter with a set of variables for each recipient. The information to be merged can be created within Word, or it can be brought in from outside, for instance from a file written by the File data management system. You can include the ability to request information, such as a date, at the time the letter and data documents are merged, and you can specify the inclusion of other documents within the template. You can also include tests to give optional inclusion of parts of the template: for example, you might have varying degrees of severity in a letter requesting payment of overdue accounts, choosing the paragraph to use with a test for the length of time the debt was owing.

Goodies

The basic features Word provides are quite powerful, but there isn't a spelling checker, nor are there other, more esoteric, extras such as indexing. On the other hand, you can import pictures and tables from Chart and Plan, and reports and sets of records from File. You can export text such as tables to other packages, and in File these can be incorporated as valid File records if you use the correct Word format.

In use

Word uses a combination of pull-down menus in the standard Mac mode, mouse movements, and command key sequences. In many cases, command key sequences can be used instead of menu options or mouse movements, so that touch-typists with good memories can keep their hands on the keyboard most of the time.

The marriage of hardware and software in handling character and font display is excellent. Indeed, in almost every respect except the lack of an index, Word with Mac would be an excellent combination with which to produce a book — and with a good printer, camera-ready copy could be of a very high standard. The Apple Imagewriter has three modes, fast draft, medium, and high-quality, and the high-quality output would probably be adequate for most requirements. It is also possible to attach a variety of daisywheel printers to Word on the Mac, including a diablo model and Apple's own, but not, as yet, the IBM Quietwriter. The drawback with a conventional daisywheel would, of course, be that you would spend a lot of time changing daisywheels if you wanted to take full advantage of Word's ability to handle many different character sets and fonts.

The only area in which I had potentially serious problems was in the handling of work file space. The Word disk as distributed is very full; you can remove a few things like extra printer drivers, but

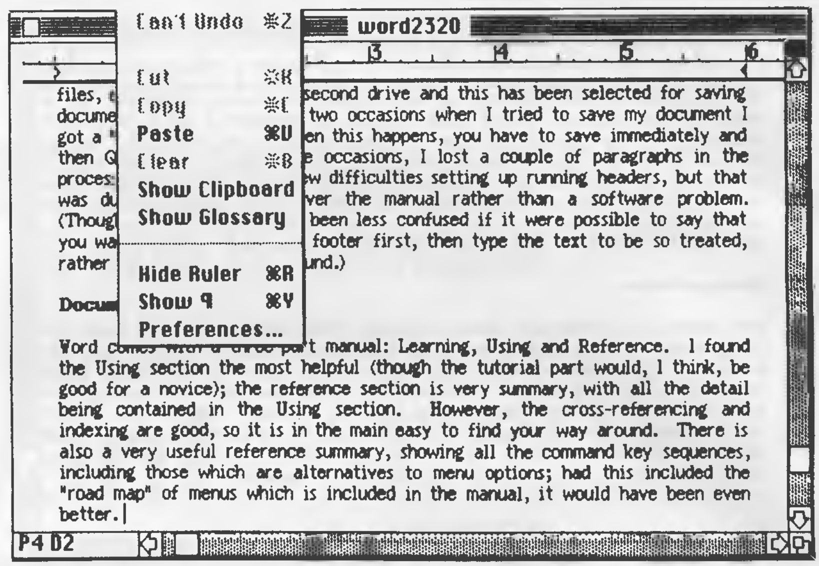


Fig 8 Document screen with current ruler plus pull-down edit menu

unless you remove the Mac Finder with its attendant useful facilities, you can't get much free space on the Word disk. On a two-drive system this shouldn't matter, but Word uses the system disk for work files, even where there is a second drive and this has been selected for saving documents. As a result, on two occasions when I tried to save my document I got a 'disk full' message; when this happens, you have to save immediately and then quit On one of these occasions, I lost a couple of paragraphs in the process. I also had a few difficulties setting up running headers, but that was due to my confusion over the manual rather than a software problem.

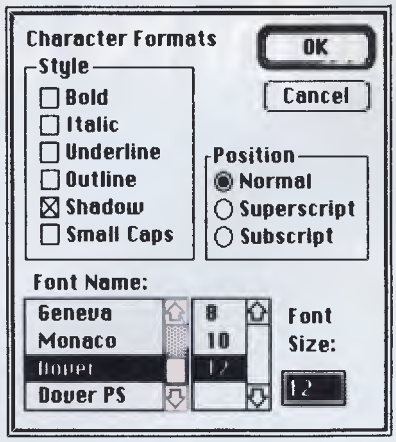


Fig 9 Character formats dialogue box

Documentation

Word comes with a three-part manual: Learning, Using and Reference. I found the Using section the most helpful (although the tutorial part would, I think, be good for a novice). The reference section is very summary, with all the detail being contained in Using. However, the cross-referencing and indexing are good, so it is, in the main, easy to find your way around. There is also a very useful reference summary, showing all the command key sequences, including those which are alternatives to menu options; had this included the 'road map' of menus which is included in the manual, it would have been even better. The onsreen help is good, and is provided in the usual Mac fashion of pull-down windows which overlay the current text. It is possible to move the help into a part of the screen where you can see it while you carry out the task — other suppliers please copy!

Conclusion

For professional word processing — by which I mean the production of long documents, tables and personalised letters, in contrast to occasional use for short letters — Word has a lot of advantages and a few draw backs. The chief of these is the lack of a spelling checker (you may also dislike the fact that it is copy-protected). For documents with a variety of character formats and fonts in particular, Word could be invaluable if your printer could cope, too. On the Mac, you would have to balance the excellent match of hardware and software as far as screen display, the neatness of the menu system and the direct pointing of the mouse is concerned, against the hard feel of the keyboard and the absence of function keys.

Microsoft Word retails for $325.

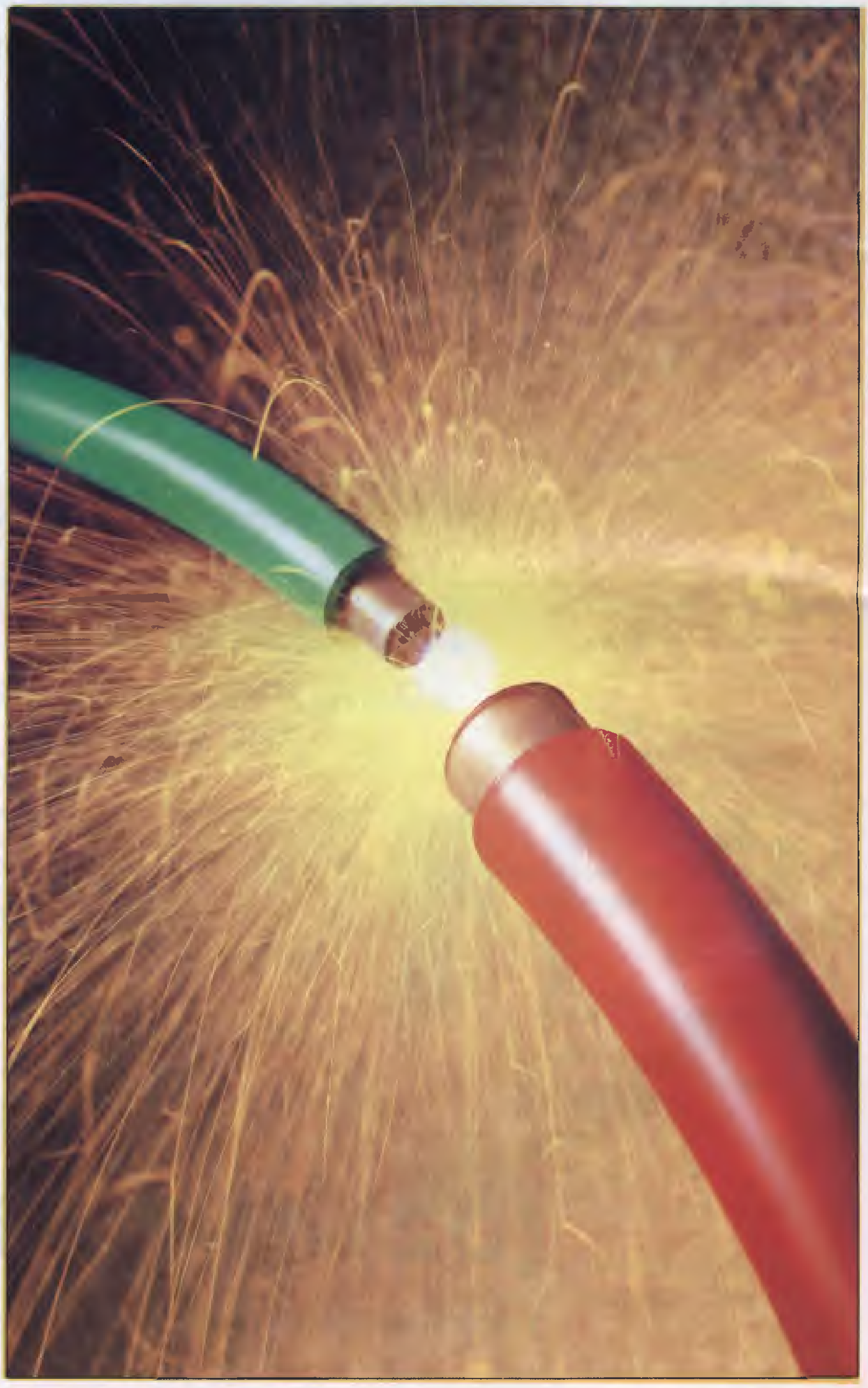
END

SPECIAL REPORT

Micro to mainframe li

This is the second part of Michael Shain's feature on PC to mainframe links. Last month he looked at planning strategy and basic products. This month he begins with advanced applications and there is also a detailed table of products. Next month he'll cover what you should look for in link products and will visit some companies to see those products in use. The final section builds into it Shain's conclusion of how links must be part of a wider DP plan, and the tools, cost and control of the job. At the end of this month's section is a glossary of terms.

Part 2



link-up

Any PC to mainframe link-up, as I wrote last month, needs very careful planning. Otherwise, a confused introduction of products, lack of security and unforeseen drain on mainframe power will cause chaos. But there is a need to link — if your company has centrally stored data, PC users will want, and will need access to it. Now, while you may think this is an obvious statement to make, for the user, access is easier said than done.

The products considered in last month's feature offer terminal emulation and file transfer, but file transfer by itself involves complicated user procedures because of the many steps involved in pulling a file down from the mainframe and having it processed by a package in the PC. The micro-mainframe link becomes more transparent to the user if the mainframe can 'prepare' data for the PC in a format it can immediately process.

A great many advanced micro-mainframe links were announced during 1983 and 1984 but few have been delivered.

The objective of implementing a link is to allow the end user to do some data processing, but the problem is — should it be a simple task on the PC or a more complex one with a more complicated user interface?

There are various ways by which more sophisticated links can be realised. A common approach is to have a package on the mainframe interfacing with the data-base which structures files for specific PC purposes. The file search parameters are developed on the PC and uploaded to the mainframe.

A micro-mainframe link that gives access to any mainframe, enables the user to query any information under security control, is independent of the mainframe package and enables information to be downloaded to any PC package, would be ideal. But the reality is there are different mainframe conventions, different DBMS conventions and different PC formats. (It may be worth mentioning here that a glossary of relevant terms is included at the end of this month's article.) The leading products are therefore limited to IBM mainframes and PCs, and are usually restricted to one set of mainframe software and to one PC software family format.

Password security

At the micro end, the PC software leads the user through a menu-based series of questions and builds up a set of data retrieval commands. Packages usually have password security, which is better than nothing, and some have passwords on the datasets, as well as PC and mainframe passwords.

A great many advanced micro-mainframe links were announced during 1983 and 1984 but few have been delivered (see table 1). They are expensive by PC standards, and they differ widely in their ease of use and speed of operation. It is for the moment a very immature market.

The limitations of personal computers are those that are addressed by mainframes, and the limitations of mainframes are easily addressed by the personal computer acting as an intelligent workstation. Ideally, we would like to have an environment that gives us the advantages of both. The key challenge here is that the workstation user should not be required to learn three command languages: that of the local workstation to which the user has already adapted (PC-DOS for instance), the network command language and the host command language. In general, the user should be able to use the 'language' with which he or she is already familiar. Not surprisingly, this goal is still some way off.

In the discussions which follow we will be drawing heavily upon IBM system software terminology. The main function of the various elements that go to make up a host operating environment are considered separately later.

Advanced micro to mainframe links categorised

The products in this section may be arranged into four categories, each of which will be reviewed prior to considering information download — the most popular class of product — in more detail.

The four categories of micro-mainframe linking are: information download; information access; identical operating systems; and identical applications.

The first, *information download*, represents by far the largest category of advanced link product. Access to mainframe data is required for spreadsheets, and because it is on the mainframe it is correct, up-to-date, consistent and controlled. The main security issues are whether or not extensive live access to production data is allowed, or whether it is restricted for extracts and, conversely, whether or not files downloaded to the PC are themselves just extracts or whole transfers. The degree to which mainframe software has to be modified is also an important factor in selecting a suitable link product.

The next level of functionality is *information access*. PCs can access public database services such as Dow Jones, Insearch/Dialog, BRS and Ausinet. Professionals wanting access to these services previously had to use special terminals with difficult search procedures but now, mainframe data is extracted for the user and downloaded to the PC running special software to interpret it. In all cases the PC understands the host interface and the services are usually dial-up asynchronous. It is a rapid growth area, particularly as modems become faster and the services are easy to use since routine operations are automated.

The third category, *identical operating systems* which run on both the mainframe and PC, is an alternative approach. Data General offers its Desktop Generation, DEC its Professional, Honeywell its Microsystem 6/20 and IBM its PC XT/370, PC-IX (Unix System III) and the PC AT (Xenix).

IBM's approach to co-operative processing is embodied in its concept of a virtual service interface which enables the user to see all host files as being locally available on the workstation. The upshot of this is that the micro should be able to run host system software and this is what the PC XT/370 does — to this end commands must be available to the user so that data can be moved from one area to the other.

Explicit commands are provided on the PC XT/370 to import and export files between the local system/370 environment, and a co-resident PC-DOS file environment. With these, a user can transform an existing PC file into a VM file and vice versa, even though the file

'85PC

the 'Oscars' of Austr

On the 17th July, 1985, on the first
Australian Personal Computer Sho
Awards for microcomputer develop
Originated by Australian Personal
and by far the most widely read ma
'85 PC Awards are designed to rewa
microcomputer development today.

8 Award Cat

Not just a single award from a lone s
annual 'across the board' group of aw
most highly regarded business and co
ations *and* the premier computer exhi
Presented for excellence, from both
technological and practical viewpoints
are eight awards in all. Seven will be fi
prizes for the product judged best by p
skilled and experienced independent j
The eighth award is a Reader's Award
Innovation.

AWARDS

alia's Micro Industry

Supported by:

THE BULLETIN

Australian BUSINESS

Australian Personal Computer

PERSONAL Computer GAMES

PC 85 THE FIFTH AUSTRALIAN PERSONAL COMPUTER SHOW

The sponsors believe that while the 16-bit desktop, perhaps justifiably to date, has received the limelight, it is time to reward excellence and effort from producers of manuals, exceptional software creators and, of course, suppliers to the all-important educational field.

"And the Categories are . . ."

Hardware, Software, Business, Home: the '85 PC Awards are the 'Oscars' of Australia's Microcomputer Industry.

1) Personal Computer – Business
2) Personal Computer – Executive, lapheld
3) Personal Computer – Games
4) Personal Computer Software – Business
5) Personal Computer Software – Educational
6) Personal Computer Software – Games
7) Personal Computer – Documentation

and

8) Reader's Award: Innovative Computer Product of the Year.

The Sponsors thank readers for their superb response. Entries are in. Judging has begun.

?

WHO WILL WIN THE

'85 PC AWARDS

Presenting a marvellous case for packing-up all your accounting software cares and woes.

Attache Accounting Software has been flawlessly designed to provide you with effective financial solutions.

Thoroughly Tested

Developed for small to large companies, Attache Accounting Software has been extensively tested in the United States and Australia and is acknowledged to be bullet-proof. Yet, for all its power and versatility, it is easy to install and easy to use.

Programmed For Growth

Amazingly flexible, it is sensibly designed with "turn-on/turn off" options to help meet your company's expanding needs.

Attache modules can be purchased separately or as an integrated system. Modules include: Accounts Receivable,

Invoicing/Sales Analysis, Inventory, Accounts Payable, General Ledger, and Payroll.

Software To Go

Everything you require to run your accounting system is packaged inside our attractive Attache Case.

CPA Support

Recognised by leading microcomputer manufacturers as the ultimate in accounting software, Attache has also earned the recommendation and support of major CPA firms.

End User Training

"Hands-on" instruction is available from a nationwide network of community colleges.

For more information contact your nearest Attache Software Dealer or Attache direct, 10th Floor, 8 West Street, North Sydney, NSW, 2060. Phone (02) 929 8700.

Attaché

Attaché Software Australia Pty Ltd

Accounting software that carries you far into the future.

systems are different. Further, the virtual service interface mentioned above enables the user to see all host files as being locally available on the workstation, assuming proper security authorisation. So, the user can view the host printers as being locally attached and can copy a file from one environment to another by invoking the local COPY command. With the COPY, IMPORT and EXPORT commands the XT/370 user (in fact, it would be a programmer) can access any TSO files as though they were CMS files without having to use TSO/CMS commands, and can continue to access these files even though they are remote. With SNA, they could well be in another continent!

The XT/370 is really a programmer's tool but it does satisfy some of the objectives of co-operative processing. It is only part of the answer because it does not allow a host application to access the workstation's services and data.

Identical applications

The final category in this section deals with *identical applications* on both the microcomputer and mainframe, and implies, as in the previous case of identical operating systems, a distributed environment. Fourth generation DBMS languages such as Focus from Information Builders and Ramis from Mathematica are typical products. Of the two, PC/Focus is the more powerful, running as it does on the IBM PC with a minimum of 512k of RAM; Ramis, however, requires a PC XT/370 so its use is fairly restricted, although it is much cheaper.

The real potential of packages such as Focus is that they support the rapid development of new applications through first prototyping on the PC and then running trials on the mainframe. This ability to shorten project lead times and thus reduce development costs, is now seen as an important development in giving a company a competitive edge. New information systems can be quickly tailored for individual customers — which is particularly relevent in the case of banks and insurance companies.

As a stand-alone system, or as part of a distributed processing environment, PC/Focus enables the user to construct sophisticated databases and perform *ad hoc* queries, generate reports, financial models, graphics and statistical analyses, all within the same integrated language. With PC/Focus users may also receive data, extract it from mainframe files and create local reports. It can also be used as a data entry tool to send information straight to the mainframe, or as a distributed workstation to transmit transactions to the host.

PC/Focus can be used with either the IRMA, Forte or PCOX boards for emulation; Information Builders has rewritten the native IRMA software to make it run faster, although still not as fast as the Forte board. There are about 40 mainframe installations of Focus in Australia, around 200 PC/Focus packages have been sold to date and the number of sales is increasing steadily.

Product comparisons

The packages listed in Table 1 all work in conjunction with established database management systems or relational database systems, and enable the user to pull extracted information down into the PC. They all follow a common procedure: first the database queries are built on the PC, loaded into a job file (after the user has signed onto the mainframe) and then they are transmitted.

Alternatively, a query may be entered directly to the mainframe but then the advantage of the user-friendly PC inter-

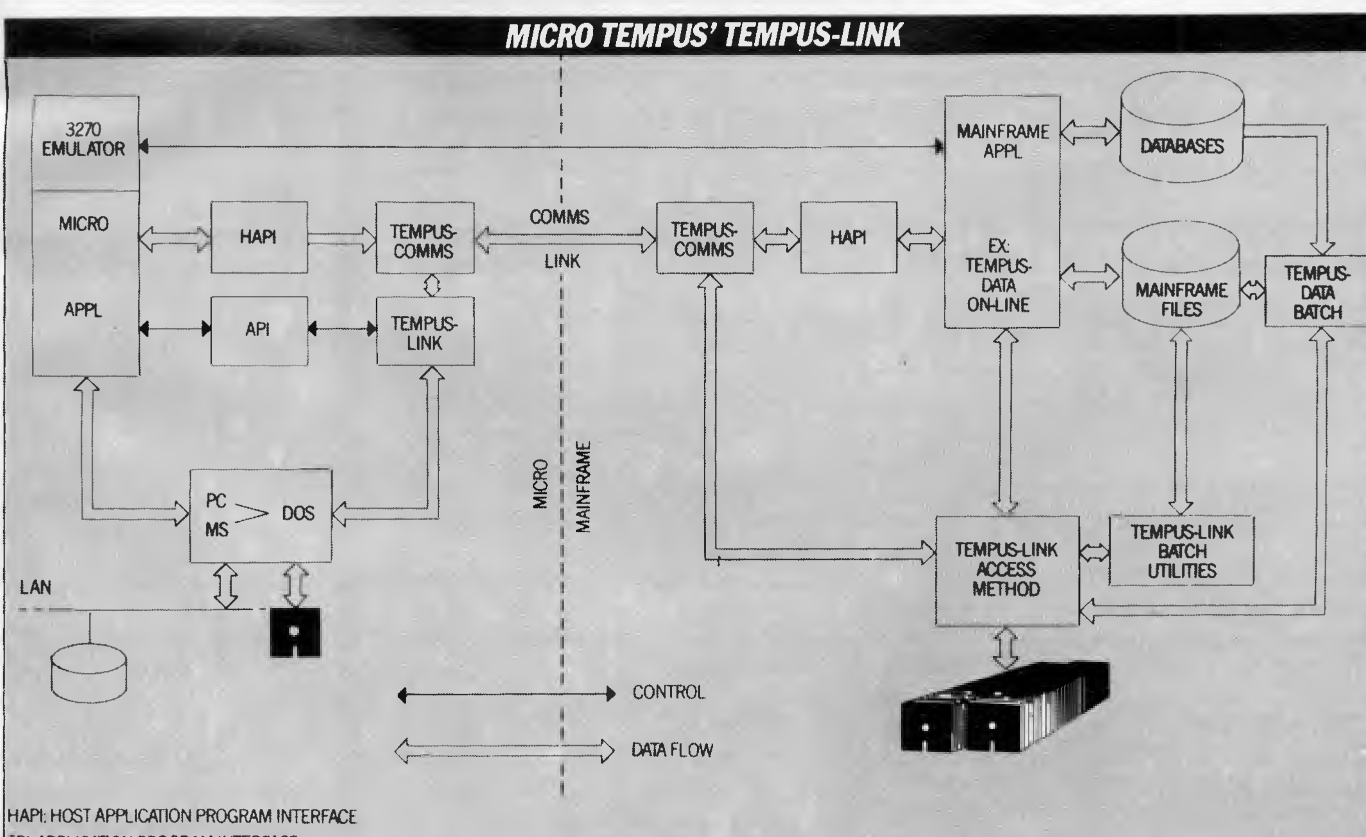


Figure 1: Tempus-Link is provided in four modules. (See Table 1 for details)

PRODUCT/AUSTRALIAN CONTACT	OPERATING SYSTEM	TP SYSTEM	DATA INTERFACE
ADR PC/DATACOM Progeni Tel (02) 419 6300	MVS DOS	CICS	ADR's Data Series
ANSWER/DB SERIES Datec Tel (02) 241 1601	MVS VM VSE	IMS/DC CICS	IMS, IDMS, Total, Adabas, VSAM etc.
CA-LINK Computer Associates Tel (02) 929 4929	MVS VM VSE	CICS CMS TSO	CA-Universe Recommend, VSAM
GOLDENGATE Cullinet Tel (02) 957 3266	MVS VM VSE	CICS IDMS-DC/SHADOW II TSO CMS	IDMS/R, IMS or DL/1
EXPRESS MATE/LINK Management Decision Systems Tel (02) 411 2941	MVS VM	CMS	Express DSS
INTERACTIVE PC LINK McCormack & Dodge Tel (02) 922 3999	MVS VSE	CICS IMS/DC	Millenium, VSAM, DL/2, SQL/DB2
PC CONTACT Cincom Systems Tel (02) 436 0100	MVS VM VSE	CICS CMS IMS/DC	Mantis, Total, TIS, DL/1, VSAM
PC-204 Computer Power Tel (02) 29 2211	MVS VM VSE	CICS	Model 204
PC/FOCUS Computer Power Tel (02) 29 2211	MVS VM	CMS TSO	Focus, VSAM, DL/1, SQL/DB2, IDMS,Total etc.
RAMLINK Mathematica Products Group Tel (02) 923 1677	MVS VM VSE	CMS TSO	Ramis II, IMS, DL/1, IDMS/R, Total
TEMPUS-LINK Distributed Data Processing Tel (03) 62 4698	MVS VM VSE	CICS CMS TSO	IMS/DB, VSAM, Total

Table 1

COMMS. PROTOCOL	PC DATA FORMAT	COMMENTS
Async, BSC, SDLC	DIF, CSV, 1-2-3	Menu-driven at the PC but rather limited software. First installed January 1984. Price $15,900 plus maintenance $2,225 (not applicable for first year). $580 per unit plus maintenance $120 per year.
Async, BSC	Selective data down-load into VisiCalc; dBase II, Friday and Framework, 1-2-3 and Symphony	Comprehensive security features but very complicated. Announced April 1983. Price $49,000 plus $700 per PC.
Async, BSC, SDLC	ASCII, DIF, Binary, Basic, CA programs	First installed in July 1984. Price $1,700 per PC.
Async, BSC	Goldengate Series	Cullinet provides a complete set of integrated PC software. First installed January 1984. Price $75,000 to $125,000 plus $1,090 per PC plus $100 import tax.
Async, BSC	DIF, 1-2-3, VisiCalc, Multiplan	A simple package that only interfaces to Express DSS on the mainframe. First installed March 1984. Price $675 per PC.
Async, BSC, SDLC	VisiCalc, 1-2-3 and other micro applications that accept ASCII	A very successful financial system. Market oriented to users who know what they want. First installed November 1983. Price $35,000 plus $1,500 (without Irma and Lotus 1-2-3) or $3,000 (including Irma and Lotus 1-2-3).
BSC, SDLC	DIF, ASCII, Series 1 Plus (a format of Cincom)	A fairly ambitious link. First launched in September 1984. Price $23,000 to $34.500 plus $575 per PC.
Async, BSC	DIF, 1-2-3	PC/204 interfaces with CCA's Model 204DBMS. Works with a production database. Easy interface mimics spreadsheet. Price $10,000 plus $750 per PC (minimum of 10 PCs).
Async, BSC, SDLC	DIF, 1-2-3, PC/Focus	PC/Focus is a fourth generation language on a microcomputer with similar facilities to Focus on the mainframe. Very popular and sophisticated. Launched 1983. Price $85,000 plus $1,975 per PC.
Async, BSC	DIF, CSV, ASCII, IT Software	Mathematica also offers its mainframe Ramis II on the PC XT/370. Launched in 1984, price $4,500 to $9,000 plus $185 per PC.
Async, BSC, SDLC	DIF, CSV and 1-2-3	A popular package but rather slow. First installed in 1983. Price $10,660 for five PCs communicating concurrently.

face is lost and some mainframe CPU power is absorbed. The host then takes the query, performs the actual processing and places the results in a file for subsequent downloading into the PC. Once the file has been physically communicated, software in the PC will convert the data to a form that is readily usable by intended application — for instance, WKS for 1-2-3, Data Interchange Format (DIF) for VisiCalc and many other packages, and the Comma Separated Variable format (CSV), which is now not very common. For example, although Lotus has its own format (WKS), it does have a facility for reading DIF files.

These procedures involving the use of a database manipulation package may appear to be cumbersome, but the software producers have made every effort to ensure the packages are easy to use, particularly through the use of menu selection facilities on the PC. Indeed, the process is much quicker than with simple file transfer packages which call for very specific operator procedures: log-on, switch to the proper directory, open data file, extract appropriate record(s), put it into a file, translate to PC format, send it to the PC, close the file and sign off.

These packages come in two halves: one for the mainframe and one for the PC. The table gives details of the host environment needed to run each of them, as well as the formatting capability at the microcomputer. Apart from PC/Focus, which is an example of identical applications run in the PC and host, the other products belong to 'information download' category. Most of the leaders in the field are IDMS suppliers and have written software for PCs to access their host software. Some of these firms have produced their own comprehensive range of microcomputer software, although this tends not to be so easy to use as the industry standard packages.

Cullinet's Goldengate Information Database provides a complete set of integrated PC software products including relational database/spreadsheet/text editor/colour graphics and electronic mail. It is a summarised database, not a production one, so all on-line uploading from the micro goes into a summary database on the host. The user either can build up queries at the PC via a series of prompts, menus and colours, or else can interact directly with the host. As can be imagined, this is a very expensive product.

Integrated software

In contrast, Informatics offers a more general purpose approach than does Cullinet. The company's Answer/DB series is totally menu driven at the PC, and enables a naive user to make intelligent selection from data held on the mainframe and convert it for processing in either Lotus 1-2-3, VisiCalc or dBase II. However, a different package running in the PC is required for each of these programs, which makes it tedious, and with dBase II in particular, the user has to go through a sequence of nine menus.

Management Science America (MSA) offers Expert Link which, perhaps more than most other products in this field, offers a consistent data transfer interface from host applications to the PC.

Unlike Cullinet, MSA does not supply database system software, but a whole series of integrated mainframe applications in areas such as payroll, personnel and manufacturing. These are high paramatised packages and can be tailored to fit the needs of a range of users. All the company's mainframe applications contain hooks for the micro-mainframe links and no additional software at the host is required for this purpose.

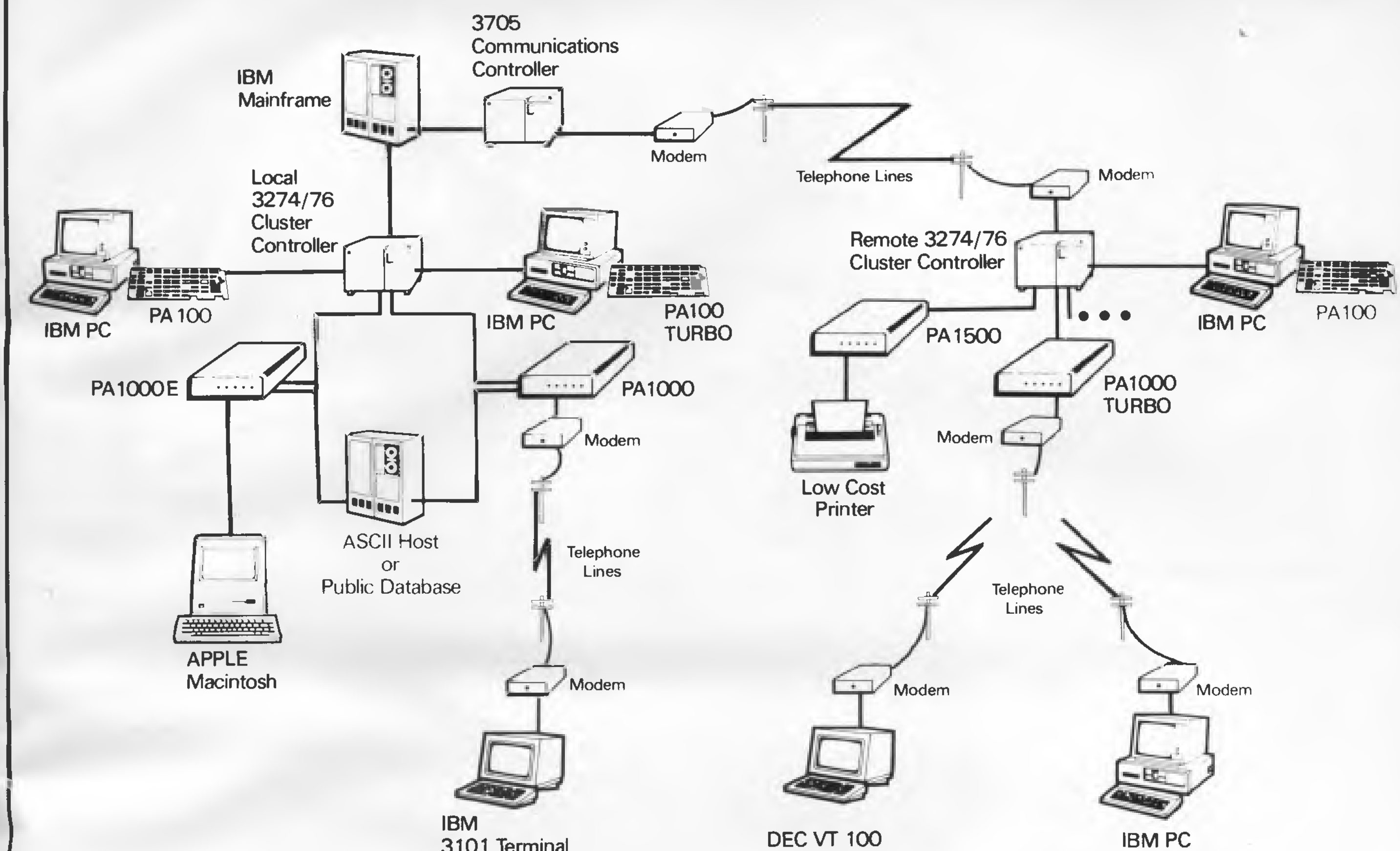
MSA's product provides an easy link to the mainframe in the format and with the tools that its users want. Extracted data is always taken directly from production files and hence the information is always the most up-to-date available. Control is maintained at all stages to ensure that full security is applied to all transfer for requests. Distributed data entry from the micro to mainframe applications is also possible, again with full security features, plus the ability to inhibit display.

Micro Tempus is one of the new breed of start-up companies specialising in micro-mainframe links. Its Tempus-Link is an attractive and popular product and is provided in four major modules (see Figure 1).

- The mainframe virtual disk system, which manages a limitless number of micro-compatible disks of variable size from 32k to 15 megabytes arranged in 'boxes' of up to 255 disks.
- The mainframe access method, which provides access from mainframe programs to the virtual disks with a variety of conversion options.
- The communication module, which makes the entire system independent of the teleprocessing monitor and communication device used by providing a transport system on both sides.
- The micro portion, which provides the micro with concurrent access to four mainframe virtual drives in addition to its physical disks. Any requests from program or from PC-DOS commands are automatically routed to the mainframe where they are processed by the virtual disk manager.

Conceptually, Tempus-Link allows the user to create boxes of disks on the mainframe, each box containing up to 255 diskettes. It also uses virtual disks which provide comprehensive system security among other benefits.

Some Of The Ways AVATAR Protocol Converters Can Provide The Micro-to-Mainframe Links You Need.



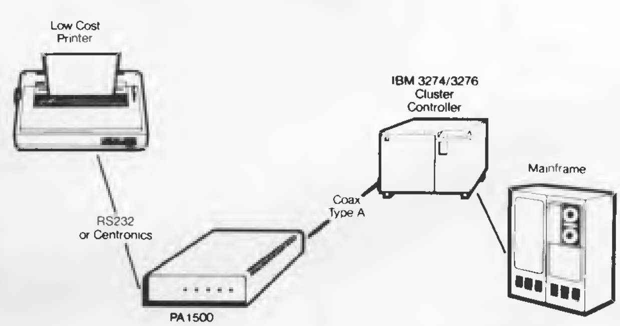
AVATAR

The Link That Thinks

DATATRAN (SALES) P/L

1/54 Malvern Street, Bayswater 3153

TELEPHONE: (03) 729 2844



Break the "printer bottleneck" and expand your total printing resources with the AVATAR PA1500. By emulating the IBM 3287 printer, the PA1500 connects low cost, high performance ASCII printers to the 3270 network using coax cable. The PA1500 supports both parallel and serial printers, and is compatible with BSC and SNA environments.

Herbie Briggs has just destroyed the myth that all floppy discs are created equal.

They seem equal. Until you look at the seams.

That's where equality ends.

Most companies seal their discs with a spot here, a spot there. Leaving most of each seam not sealed at all.

Sooner or later, the seams might do what comes naturally: they bulge. Warp. Pucker. Open up.

Pens, pencils, fingernails—even a four-year-old's, like Herbie—can catch and snag in those wide open spaces.

That's sloppy. And dangerous. Because if you put a sloppy floppy into your disc drive, it can jam your drive. Ruin your drive head. Lose your data.

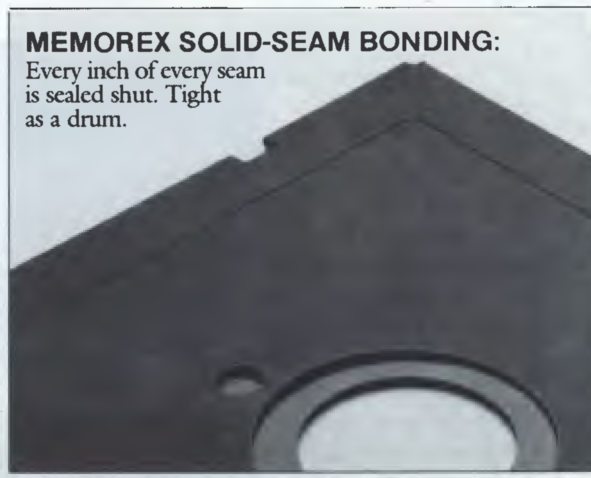
So much for their seams. Ours are different.

THE SLOPPY FLOPPY: Sealed with a spot here, a spot there. Leaving unsealed seams everywhere.

Memorex uses a process we developed, called Solid-Seam Bonding.

Solid-Seam Bonding seals shut every inch of every seam of every Memorex® floppy disc. Tight as a drum. That makes the Memorex floppy stiffer. Stronger. And your data safer.

To resist bulging, warping, puckering, or opening up.



MEMOREX SOLID-SEAM BONDING: Every inch of every seam is sealed shut. Tight as a drum.

To resist all the things that can jam your drive, ruin your drive head, or lose your data.

Which proves that a Memorex floppy disc isn't equal to all the others. It's better.

Solid-Seam Bonding is just one example of the extra care that goes into every Memorex floppy disc. Be it 8," 5¼" or the new 3½." Extra care that lets us guarantee every Memorex disc to be 100% error-free.

The next time you're buying a floppy disc—or a few hundred of them—just remember this:

It's always better to be safe than sloppy.

For more information on the full line of Memorex quality computer media products, including computer tape, call Sydney (02)908 2211 or Melbourne (03) 267 2955

Your Data. When it matters, make it Memorex.™

MEMOREX

Take advantage of the 18% investment allowance Look at these bargains from NEC.

BUSINESSPACK 1.

HARDWARE
- ★ NEC APC III Computer
- ★ 2 x 720KB Floppy disk drives
- ★ 1 x monitor
- ★ 1 x P2 near letter quality printer
- ★ 1 x printer cable
- ★ 1 x box blank diskettes
- ★ 1 x diskette case

SOFTWARE
- ★ IMS System 1 Integrated Debtors, Creditors & General Ledger. New Release.

$4995
Includes Sales Tax

WORDPACK 1

HARDWARE
- ★ NEC APC III Computer
- ★ 2 x 720 KB Floppy Disk Drives
- ★ 1 x monitor
- ★ 1 x 3500 letter quality printer
- ★ 1 x printer cable
- ★ 1 x box blank diskettes
- ★ 1 x diskette case

SOFTWARE
- ★ WordStar Professional including - WordStar, CorrectStar, SpellStar, MailMerge, Star Index

$4500
Includes Sales Tax

COMMPACK 1

Talk to Telecom's Viatel, or any Database around the world
- ★ NEC APC III Computer
- ★ 2 x 720KB Floppy Disk Drives
- ★ 1 x monitor
- ★ 1 x P2 near letter quality printer
- ★ 1 x printer cable
- ★ 1 x box blank diskettes
- ★ 1 x diskette case
- ★ 1 x Avtek MultiModem

SOFTWARE
- ★ Acculink (works with MultiModem)
- ★ WordStar

$4995
Includes Sales Tax

The New Printers are in!

- ★Colour Screen Option: add $700.00 to above prices
- ★10MB Hard Disk Option: add $1500.00 to above prices
- ★Supercalc 3 spreadsheet software: $450.00
- ★Call us first for Integrity (IMS) and Attache accounting packages

We can arrange competitive leasing

NEC

Natwick Management Suite 2, 25 Burns Bay Road
Lane Cove NSW, Phone (02) 428 1666

APC III
ADVANCED PERSONAL COMPUTER III

A glossary of terms

Michael Shain has compiled a glossary of terms and abbreviations which relate to his feature on PC to mainframe links.

Many of the following terms are taken from the second edition of the Dictionary of Information Technology, *by Dennis Longley and Michael Shain, published by Macmillan, whose permission to reprint is gratefully acknowledged.*

Abend In data processing, an ABnormal END of task prior to its completion on a mainframe bacause of an error. A microcomputer in emulation mode should be programmed to handle this. See emulator.

Access method In data processing, a technique for moving data between main storage and an input/output (I/O) device. See I/O.

APA All points addressable. Synonymous with pixel.

Application interface In distributed processing, a routine which enables a program in a microcomputer to communicate with the microprocessor on an emulator board. This routine could be a sequence of operator commands for automating file uploading to the host. See emulator, uploading.

Application mode The operating mode of the IBM 3270-PC keyboard when it is logically attached to one of the following: a host computer session, the personal computer session, or a notepad session. See notepad session.

Architecture The specification of the relationships between the parts of a computer system or network.

ASCII American Standard Code for Information Interchange (pronounced ASKEE). A standard data transmission code that was introduced to achieve compatibility between data devices. It consists of seven information bits and one parity bit for error checking purposes and is the character code used in microcomputers. It must be converted to 8-bit — EBCDIC characters for uploading to an IBM host. See EBCDIC.

Asynchronous transmission Transmission in which each information character is individually synchronised by the use of start and stop elements as the interval of time between characters varies. Compare synchronous transmission.

Autokeying The ability to record frequently used groups of keystrokes and to play them back at designated locations on the screen.

Baud In communications, a measure of the signalling speed in a digital communications circuit, equal to the number of discrete conditions or signal events per second.

Binary synchronous communications In data communications, transmissions in which synchronisation is established between the sending and receiving station before the message is sent. See SNA, synchronous transmission and compare asynchronous transmission.

Bisync BInary SySYNchronous Communication; in data communications an IBM protocol for the synchronous transmission of binary-coded data between stations. It is one of the standard ways in which IBM computers in the 370 and 40XX series communicate with remote terminals. See binary synchronous communications, protocol and compare SDLC.

Box In co-operative processing, an area on mainframe backing store that acts as a virtual box of floppy disks to a remote micro. See co-operative processing.

Bus A common group of hardware lines used to transmit information at high speed between digital devices. A terminal emulator board interfaces with the system bus in a microcomputer.

BSC Binary Synchronous Communication. Synonymous with bisync.

Character based interface The prompts and commands used by an operating system. With bit-mapped graphics, a more flexible and user friendly interface is possible through the use of icons and pointing devices. See bit-mapped graphics, GEM, icon.

Character mapping Synonymous with code conversion.

Code conversion A process for changing the bit grouping of one character in a code into the corresponding grouping for a character in a second code, for example, from the ASCII code used by PCs to EBCDIC used by IBM mainframes. See ASCII, EBCDIC.

CICS Customer Information Control System, an IBM program product that enables terminal transactions to be processed concurrently by user-written application programs. It also includes facilities for building, using and maintaining databases. A CICS file can be downloaded into a PC under host program control. See file transfer. Compare CMS, TSO.

Cluster control unit In communications, a device such as the IBM 3274 control unit that can control the input/output operations of more than one terminal, such as a group of 3278 display stations. See I/O. Synonymous

IBM wants to buy you lunch. In Paris.

Or New York. Or London.
Or Rio. Or anywhere you like in the world.

To be eligible, all you have to do is participate in a demonstration of an IBM Personal Computer before June 30 from an authorised IBM PC dealer.

Then your name will be entered into a draw.

There's never been a better time to buy an IBM PC.

Despite devaluation, IBM has not raised its PC prices.

And if you buy one before June 30 you may also be eligible to take advantage of the 18% Investment Allowance.

Two good reasons to take a close look at the IBM PC now.

These reasons, combined with the chance to enjoy a fabulous dining experience anywhere in the world, make May and June the best time ever to buy an IBM PC.

One of Australia's most successful business PCs.

The IBM PC is made in Australia and has become an acknowledged leader in its field. Talk to a dealer, let him show you how an IBM PC can help you with your business needs.

No matter what your industry or profession, the IBM dealer network has a solution for your business.

How to arrange your demonstration.

Simply call IBM Direct for the name of your nearest authorised IBM PC dealer.

Sydney 2 0531, Melbourne 614 1511, Brisbane 229 6711 or Adelaide 212 6899.

Canberra, Darwin, Hobart, Perth and all other areas 008 222 224. (For all numbers ask for Ext. 23.) Your IBM dealer will give you full contest details.

But hurry. If you attend an IBM PC demonstration before June 30, you'll also be eligible for the special prize.

Lunch for two at the restaurant of your choice anywhere in the world.

IBM

IBM Australia Limited. (Inc. in NSW)

with cluster controller, terminal controller.
CMS Conversational Monitor System, an IBM virtual machine operating system that provides time sharing and program development capabilities and operates only under IBM's VM/370 control program.
Coaxial cable A low loss cable, used for high frequencies, which consists of a conductor within, and insulated from, a tube of braided copper. Used to connect a terminal to an IBM 3274 cluster control unit where high transmission rates are required.
CODASYL In databases, COnference for DAta SYstem Languages. A group created by the US Department of Defence including users and manufacturers, for developing COBOL and hardware independent software for database management. See IDMS.
Concurrency See multi-programming
Co-operative processing In distributed processing, the environment which enables the services and data of either a microcomputer or a host to be transparent to an application program running in the microcomputer. See virtual service interface workstation.
CSV A Comma Separated Value, a free format file standard in which a comma is used for separating fields and quotation marks surround text. See data format.
Customising procedure In distributed processing, the initialisation process, performed at the IBM 3274 cluster control unit, to support the attached terminals. See terminal
Database A collection of interrelated data stored so that it may be accessed by authorised users with simple user friendly dialogues. The database structure is independent of the programs that use the data and a common controlled approach is employed in adding, deleting or modifying the data.
Data format The arrangement of data on a data medium, for example, a floppy disk. When an application program in a microcomputer processes data downloaded from a host it must be in data format it can recognise, such as DIF or CSV. See CSV, DIF, file transfer.
Data stream In data communications, a continuous stream of serial data being transmitted in character or binary digit form through a channel. The data stream may contain control and format information.
Dataset (a) In databases, a named collection of data items bearing a logical relationship to each other and ordered in a prescribed manner, it may contain data for accessing the data, as with indices. (b) In data communications, a modem.
DB/DC Database/Data Communications, an on-line database system accessed by remote terminals. See database, terminal.
DBMS Database Management System, a set of programs which facilitate the creation and maintenance of a database and the execution of programs using the database. See database, IDMS.
DCA Document Content Architecture, an IBM set of architectures and related products for the creation, distribution and display of information. See DIA.
Development centre In data processing, a department which commits the necessary hardware and software to application development.
DIA Document Interchange Architecture, IBM's set of rules and data structures that establish the discipline for predictable information interchange between devices. It prescribes the exchange of information at the DCA layer. See DCA.
DIF Data Interchange Format, developed originally by Software Arts in the US to standardise ASCII data exchange. It is now a *de facto* standard and most spreadsheet and several database-management programs can read DIF files. See data format.
DISOSS Distributed Office Support System, IBM's system software for filing, retrieving and sending internal mail consisting of text and image documents. See OIA.
Distributed function In data communications, the use of programmable terminals and other devices to carry out operations that were previously performed by the central processing unit, such as network management.
Distributed function terminal mode In data communciations, a mode of operation of the IBM 3270-PC which allows up to four concurrent logical terminal sessions. See session.
Distributed data processing In computing and data communications, the distribution of processing functions and data throughout an organisation to the locations where they are needed. Synonymous with distributed processing. See end user computing and Systems Network Architecture.
Distributed processing Synonymous with distributed data processing.
Download The process of loading programs or data files into a computer from a remote location using a communications channel. Synonymous with down line load. Compare unload.
EBCDIC In data communications and computing, Extended Binary Coded decimal Interchange Code, one of two codes used in IBM equipment. This 8-bit code gives 256 combinations but there are international differences, for example, £ and $. Compare ASCII, ISO 7-bit code.
Emulator Special purpose hardware or software which enables one machine to act as if it were another, for example, an add-in board to convert a PC to a 3270 terminal. See terminal.
End-user The source or destination of information flowing through a system.
End-user computing In data processing, personal computing carried out by an end-user, including program development, via a personal computer or terminal linked to a mainframe. See information centre, fourth generation language.
Entry assist In distributed processing, a function that allows a display terminal to operate like a typewriter, providing facilities such as margins, tabbing, a bell to signal the end of the line and word deletion.
Extract In distributed processing, to remove specific items from mainframe file for downloading into a PC. See downloading.
File A collection of records which are logically related to each other and handled as a single unit, for example, by giving them a single name. See record.
File transfer In distributed processing, the downloading of a file from a host to a PC, or vice versa. This may be effected by a PC resident program which enables the emulator board to communicate with the host, for example, by requesting a file to be edited and then requesting each page of the file to be displayed so that it may be read from the emulator buffer and written to disk. This process is repeated until the whole file has been transferred.

Another approach is to use a file transfer software package at both the host and microcomputer. This allows for greater security and flexibility since the pacing of data transfer to the PC, error detection and correction, and code conversion are possible. See code conversion, downloading, emulator.
Fourth generation languages In computer programming, a family of programming languages designed for end-users rather than professional programmers. They are designed to speed up the process of application building, reduce maintenance costs, minimise debugging problems and make languages user friendly so that end-users can solve their own problems. See end-user.
Gateway In telecommunications, equipment used to interface networks so that a terminal on one network can communicate with a terminal or computer on another.
GDDM In computer graphics, Graphical Data Display Manager, an IBM program product that uses the programmed symbols feature of the IBM 3279 colour display terminal to construct and display

graphics pictures. A special graphics board is normally required in the PC if it is to display GDDM mainframe graphics to supplement the 3270 emulator board.
GEM In computer graphics, Graphics Environment Manager, a more advanced version of GSX from Digital Research, which contains additional services to give it an Apple Macintosh type of user interface.
GSX Graphics System EXtension, Digital Research's implementation of the ANSI VDI standard, See VDI.
Handshaking In data communications, the exchange of predetermined signals when a connection is first made across an interface in order to confirm it is working satisfactorily and to prevent data loss. See interface.
Host computer In data processing, a large, central processor that provides services such as computation, database access or special programs, or programming languages. See database, fourth generation languages.
Hot key In distributed processing, a feature of an emulation package which enables a user to press a single key to automatically switch from a session with the local microcomputer's operating system to a session with the host. See session.
IDMS In databases, Integrated Data Management System, a proprietary system specified by CODASYL, which provides facilities for structuring and using large databases. See CODASYL.
Information centre In data processing, a service strategy as well as an organisation within the data processing department that provides a direct interface to, and supports services for, end-user computing. See development centre, end-user computing.
Information systems director In computing and communications, the expanded role of the data processing manager to reflect the strategic importance of information systems in the corporate environment.
Interface In data communciations, a shared boundary between two related devices or components defined for the purpose of specifying the type and form of signals passing between them.
I/O Input/Output, a general term for the equipment used to communicate with a computer and the data involved in the communication.
ISAM In databases, Indexed Sequential Access Method, a method of file access in which a stored index contains the address of a group of records and the highest key value of that group. See key and also record.
ISO 7-bit code In data transmission, one of the two international data codes, The US version of this code is ASCII. See ASCII and compare EBCDIC.
Job In data processing, a full description of a unit of work for a computer. A job will normally include all the necessary application programs, files and instructions to the operating system. See RJE.
Job input file In data processing, a file consisting of a series of job definitions and accompanying data.
Key One or more characters used for identifying a set of data.
Keyboard In data processing, a device for the encoding of data by key depression, which causes the generation of the correct coding sequence.
Keyboard remapping In distributed processing, a feature of a terminal emulator package which redefines the keys on a microcomputer keyboard to imitate the keyboard on a mainframe terminal;, such as the IBM 3278. Because the latter has more keys than the microcomputer, some emulated 3278 keys will be represented by a sequence of key depressions on the microcomputer keyboard.
LAN In data communications, Local Area Network, a high bandwidth bidirectional communications network which operates over a limited geographic area. LANs provide the computer user with the opportunity to communicate with other workers, to supply and access data and share expensive peripherals.
Link In communications, a transmission path of specified characteristics between two points, for example, a coaxial cable or a telephone wire. As well as the physical aspect of transmission, a link includes a protocol and associated devices and software, this means it can also be logical.
Local In distributed processing; (a) a terminal connected to a cluster control unit; (b) the mode of operation of a PC or terminal when processing or input/output is carried out locally rather than at the host.
Macro In a programming language, an operator that, when used in an instruction, causes the generation of a sequence of instructions to accomplish a particular task.
MCU In distributed processing, Multiple Concurrent Users, the numbers of users who are allowed under the licence terms of a mainframe software package to use it at any one time.
Menu selection A technique in which a user is provided with a list of options and details of the keys to be depressed for the selection of each option.
MIS Management Information System, a system designed to provide management with required data that is accurate, relevant and timely, possibly on a realtime basis.
MODEM In data communications, MOdulator-DEModulator, a device that modulates the transmitted signal and demodulates the received signal at a data station. In synchronous communication, the line modems generate timing signals to keep the sending and receiving stations in step. See synchronous transmission.
Multi-programming In operating systems, a multi-user operating system which may also run background programs independent of a specific user. Compare multi-user operating system, next.
Multi-user operating system A system that can service the requests of more than one user or programmer at a time. The users are not handled simultaneously, each is serviced within a short time slice, at high speed, giving the impression to the user of simultaneous operation. Compare multi-programming.
Multi-system networking The distribution of the data processing function between two or more host processors in one or several locations using communications facilities. See Systems Network Architecture.
MVS Multiple Virtual Storage, an alternative name for IBM's OS/VS2, a virtual storage operating system that supports multi-programming with a variable number of tasks.
Notepad In graphics, a program which reserves a portion of main memory as an *aide-memoire* for the user in a windowing environment. See window.
Overlap In graphics, the ability of two or more windows to be overlaid without any loss of data. Compare tile.
Personal computer (PC) A term generally applied to more powerful microcomputers intended for business. See end-user.
Picture element In graphics, the smallest element of a display space that can be addressed. A picture element will have one or more attributes of colour, intensity and flashing.
Pixel Synonymous with picture element.
Port In computing and communications, a functional unit of a node through which data can leave or enter a data network or a computer.
Processor interconnection The distribution of data processing functions between two or more host processors in a single location without the use of communications facilities. Compare distributed data processing, multi-system networking.
Program function key In computing and data communications, a key on a microcomputer keyboard or computer terminal; that (a) invokes a utility to perform scrolling, for example; (b) can be

programmed by the user — for example, to log on to a mainframe.

Protocol In data communications, a formally specified set of conventions governing the format and control of inputs and outputs between two communicating systems.

Protocol converter In data communications, a device that interfaces two communicating systems working to different protocols. For example, a protocol converter can interface to a microcomputer, via the asynchronous RS232 serial port, and to an IBM SNA/SDLC synchronous network. All EBCDIC/ASCII code conversions are carried out, as well as cursor positioning, screen buffering and error handling.

Protocol transfer In data communications, a transfer of data between a microcomputer or terminal and a host using an agreed error checking procedure. The rules for error detection and retransmission form an integral part of most synchronous protocols, for example, IBM's bisync, but there is no common standard for asynchronous transmission. See asynchronous transmission, file transfer, synchronous transmission.

Reconfiguration To add or remove components of a system or to change their interconnections. In the case of a microcomputer emulating a terminal, the microcomputer keyboard will need to be redefined, and this should be catered for in the emulator software. See emulator.

Receive A single PC-DOS session command on the IBM PC 3270 that causes a host file under CMS, TSO or CICS to be downloaded to disk. See CMS, TSO, CICS and compare send.

Record In data processing, a collection of related data treated as a unit.

Referral centre In distributed processing, an independent body outside the dp department set up to help co-ordinate and implement micro-mainframe links. Compare information centre.

Remote In data communications, a cluster control unit that is connected to a mainframe via a modem,. as opposed to a local controller which is directly wired.

RJE In data processing, Remote Job Entry, the submission of a job to a peripheral that is connected to the processor via a data link. Emulator boards are available to turn the PC into an RJE terminal.

Script A prepared set of commands on a microcomputer that automatically initiates log on and password input to a host.

SDLC Synchronous Data Link Control.

Send A single PC-DOS session command on the IBM PC 3270 that causes a file on the microcomnputer to be uploaded to the host. See session, upload and compare receive.

Session The period of time during which a user of a terminal can communicate with an interactive system — usually the elapsed time between log on and log off.

SNA Systems Network Architecture.

Software protocol converter In distributed processing, the software emulation of a protocol convertor. A PC linked asynchronously to a host can behave as a synchronous IBM 3270 terminal because the appropriate protocol conversions are carried out by software in the host and PC. See asynchronous transmission, protocol, synchronous transmission. Compare protocol converter.

Synchronous Data Link Control In data communications, IBM's data link control protococl for SNA. It is a discipline for managing synchronous, code-transparent, serial-by-bit information transfer over a link. SDLC conforms to subsets of the Advanced Data Communication Control Procedures (ADCCP) of ANSI and the High level Data Link Control (HDLC) of the International Standards Organisation. See SNA.

Synchronous transmission In data communications, a transmission method, in which each bit is transmitted according to a given time sequence. It can provide a higher bit rate than asynchronous transmission but requires that the receiver and transmitter maintain exact synchronisation over an extended period. Compare asynchronous transmission.

Systems Network Architecture In computing and data communications, IBM's network architecture for distributed data processing. It allows any distributed system to access any host processor in the network through multi-systems network facilities. It also provides the architecture for distribution between multiple processors, as well as between one or more processors and remote intelligent communication systems. See distributed data processing, multi-systems networking. Compare processor interconnection.

Teleprocessing In computing and communications, data processing combined with telecommunications, as in the use of a telephone network to connect a remote terminal to a computer.

Terminal In communications and computing, an input/output device for transmitting and receiving data on a communication line, for example an IBM 3278 display station.

Terminal emulation In distributed processing, the addition of dedicated software and hardware to a PC so that it imitates the essential characteristics of a mainframe terminal. It may be realised by using a special add-in board connected to the microcomputer bus, or by an externally connected protocol converter. See bus, protocol converter, terminal. Compare software protocol converter.

Tile In graphics, a blanked out or tiled frame around a window which will hide data on adjoining windows. Overlapped windows do not suffer from this defect, but they may not respond as quickly to commands. See window. Compare overlap.

Time sharing Enables a CPU to handle simultaneous users and peripherals.

Toggle In distributed processing, to move freely between sessions on a PC by depressing a hot key, for example, from PC-DOS to a host session. See hot key and session.

Token ring In data communications, a bit pattern passed around a local area network for control purposes. Any node upon receiving the control token, may remove it from the ring shaped network, send a message and then pass on the token.

Transparent mode In distributed processing, a mode of operation of a PC linked to a mainframe in which all characters, including control characters, are displayed exactly as received, but no further action is taken, for example, clear screen. This mode is used for testing purposes or when control characters are intended for another terminal.

TSO Time Sharing Option, an optional configuration of an IBM operating system that provides conversational time sharing from remote terminals. See CICS, CMS, time sharing.

Upload In distributed processing, to send a file from a smaller computer or local one, such as a PC, to a larger or remote computer. See file transfer, download.

Vaporware The advertising of a program that is not commercially available.

3VDI In graphics, Virtual Device Interface, an ANSI graphics standard which defines an interface between device dependent and device independent code in graphics environment. Makes device driver identical to the application program.

Virtual In computing and communications, a description of a facility that is offered to a user, or system, as if it were a physical reality. Virtual systems are realised largely through the use of software. See virtual machine, virtual storage.

Virtual machine A simulation of a computer and its associated devices by another computer system.

Virtual service interface In distributed processing, the mapping of files and

PORCHESTER Computers Pty. Ltd.

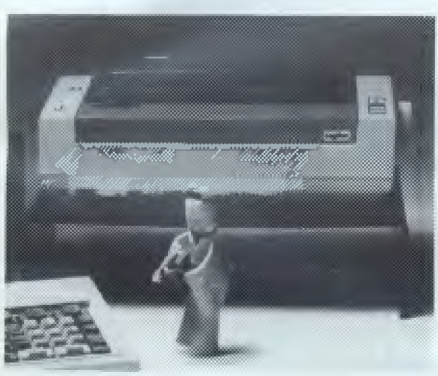
Enquire through Dealers Australia wide and New Zealand

169 Victoria Parade, Fitzroy 3065, Australia, (03) 417 6999

Logitec PC/XT
256K RAM, PC DOS, Forth Dual Floppies (Optional 10.6 MB H/disk) Color graphics, with composite 2 x serial plus printer port.
Logitec systems from $3799 (inc)
Complete Skaipaks from $900 (inc)

Logitec PC and XT can be purchased complete with the **SkaiPak.**
SkaiPak includes the following:
Accounting: fully integrated Debtors, Creditors, Stock, GL.
Electric Desk: Word processing, Database, Spreadsheet, Communications.
256K Upgrade: Upgrades memory to 512K
Monitor: Your chance of Hi-res Color or Monochrome

Logitec FT-5002 NLQ Printer



The Logitec FT-5002 is a high quality dot matrix printer with capability to print Near Letter Quality (NLQ) using special print-method. 120 cps, buffer, graphics, 10/12 pitch. $469 inc tax.

Admate 100DP Printer

The Admate Exec 100DP is a flexible utility printer suited to small business, home and education use. A 4K char. buffer is available for a nominal cost. Commodore interface (Xetec) available. $369 inc tax.

Skai-64 Disk Drive

For the Commodore 64. Designed in Japan, this disk drive is fully compatible with the 1541. It will read and write the same format and runs all Commodore 64 software. $395 inc tax.

Porchester Computers Pty Ltd

Distributors of Logitec, Skai, Admate

NOW EPSON DISTRIBUTORS
In addition to the above Porchester are now approved Epson printer distributors.
Please phone for details or quotes on Epson products.

Phone (03) 417 6999 for more information

Please send information on ....
and also details of your nearest dealer.
Name .... Tel. ....
Company ....
Street ....
Suburb .... P/code ....
Porchester P/L 169 Victoria Pde. Fitzroy 3065

ALL PRICES ARE SUGGESTED RETAIL ONLY **ALL PRODUCTS CARRY 6 MONTHS WARRANTY**

services on a mainframe onto a microcomputer so that the user believes they are available locally and running under the microcomputer's operating system. See co-operative processing.

Virtual storage A large notional main store is made available to the user by mapping the user's virtual address on to real addressess of auxiliary storage. See virtual Synonymous with virtual memory.

VM/370 control program In data processing, that part of IBM's VM/370 operating system that creates one or more virtual machines. This interface enables guest operating systems to believe they are actually running on a real System/370. See virtual machine.

VS See virtual storage.

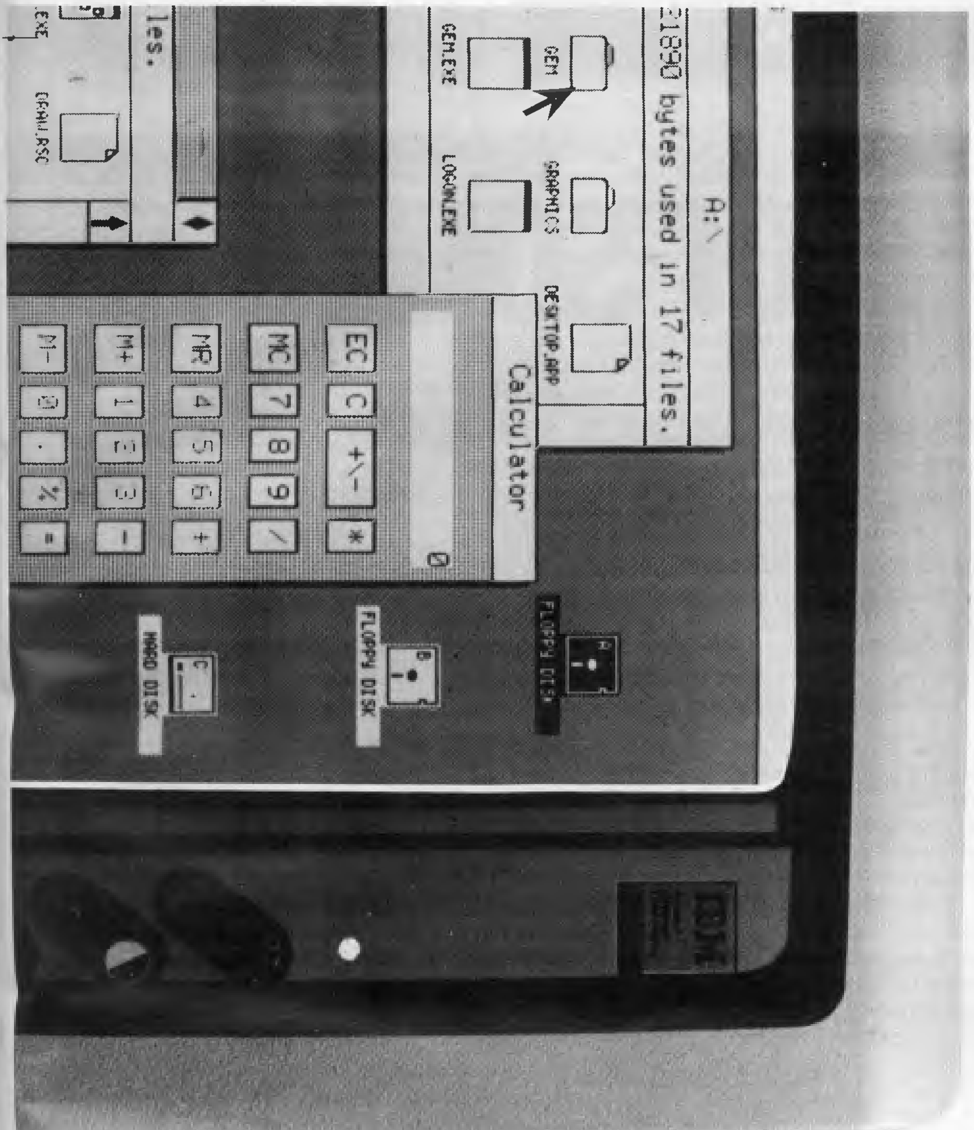
VSAM In data processing, Virtual Storage Access Method, IBM's proprietary software for virtual storage access. See access method, virtual storage.

Window A software technique for dividing a bit-mapped graphics display into a number of independent, rectangular displays, or windows. Each window provides all the necessary functions for a user to interact with an application running in that window. A window can also display the output of a host session when a microcomputer is linked to a mainframe. See zoom.

Wordstation A personal computer either linked to a local area network, or attached to a mainframe terminal network, which the user can call upon for a number of office automation services. see cooperative processing, *LAN*.

X.25 In data communications, interface between data terminal equipment and data circuit terminating equipment for terminals operating in the packet mode on public data networks.

Zoom The expansion or contraction of a window. See window.



GEM'LL BOWL YOU OVER

DIGITAL RESEARCH

GEM™ lets you operate your personal computer* through visual commands and symbols. It's ideal for anyone who is frustrated by computer jargon. The best news is that existing business programs like Supercalc³, dBASE III and Framework can all work without change with GEM.

Whenever your computer is turned on, **GEM Desktop** surrounds you with familiar items found on a desk e.g. a calculator, clock or file folder. You work by pointing the arrow at the appropriate icon, and selecting commands from a menu of choices.

GEM Draw allows you to produce charts, logos and other pictures. The program includes a library of predesigned graphics for you to use.

GEM Graph is designed specifically for you to communicate information about your data. It's not just a data analysis tool. Pre-designed templates can be selected from the gallery of graph-styles provided.

**GEM works on the IBM PC, PC XT, PC AT, and most close compatibles. It is also being ported to many other machines including the Apricot, and Atari ST.*

GEM Wordchart produces high quality text slides, the kind which make up over half of most presentations. The quality rivals those made by commercial artists and at a fraction of the cost and time.

Other GEM products include **GEM Paint,** an on-screen graphics tool, and **GEM Write,** a full function word processor. **GEM Programmers Toolkit** is available for software developers who wish to drive their application software with icons.

GEM Desktop and Programmers Toolkit are available now. GEM Draw will be available at the end of May, other GEM products will be released in the following 3 months. For more information contact your local computer dealer, or complete the coupon below.

arcompacific
▪master distributor▪

------------------------------ CUT ALONG THIS LINE ------------------------------

Please send me more details about GEM Desktop ☐ GEM Draw ☐ GEM Graph ☐
GEM Wordchart ☐ GEM Paint ☐ GEM Write ☐ GEM Progammers Toolkit ☐

Name .... Address ....

Post to: FREEPOST 2, ARCOM PACIFIC, P.O. Box 13, Clayfield, Q. 4011

Wired for sound

MIDI can open up a new area of creative technology with its expanding musical capabilities. David Rosam and Ian Beynon take it from the top.

MIDI has been very much in the spotlight over the last few months, with many pieces of musical technology finding their way into the pages of computer magazines. But for those hoping to find some hard information in the heap of hype being piled up by manufacturers, there has only been disappointment — MIDI is both much misunderstood and underutilised.

Musical Instrument Digital Interface is of great interest to both musicians and computer enthusiasts. For those with suitable musical instruments, MIDI can open up many new possibilities. For the interested computer enthusiast, proper music is, at last, a possibility. Even micros with the most sophisticated sound chips, such as the Commodore 64, cannot approach the capabilities of the simplest synthesiser. But computer enthusiasts are faced with learning the language of musical instruments, and musicians are faced with learning about the mysterious terminology associated with computers. As a result, very few people are getting the most from MIDI.

Musical instruments

The technology behind modern electronic instruments is identical to that used in computers — the microchip is at the heart of both. All the instruments you can connect to MIDI can be termed 'synthesisers'. When synthesisers are mentioned, most people immediately think of the keyboard instruments which are, by far, the most common type of synthesiser.

A synthesiser is really just a box of electronics for producing sounds. It can be triggered by any kind of switch, or an analogue signal which has been digitised. There is no reason, in theory, why any kind of instrument cannot be used for triggering the electronics, but in practice, it's a slightly different story. The keyboard remains the most popular kind of synthesiser instrument as it is a very convenient way of triggering a synthesiser, and traditional keyboard instruments are widely taught. There are some guitar synthesisers on the market which are played exactly like a guitar.

The three programs in this article are in Commodore 64 or Spectrum Basic, but should be fairly simple to convert to other computers if you read your interface manual carefully.

Drum machines also come within the range of synthesisers. Drum machines, until recently, had a very characteristic sound, so any record made with a machine instead of a drummer was instantly recognisable. As technology advances, their sound is becoming more and more realistic. These machines allow anyone to set up even very complex rhythms. Most drum machines have some memory facilities which allow rhythms to be stored, as well as offering preset rhythms.

For the purposes of this article, a synthesiser means a keyboard device, as these are the cheapest kind and are more likely to be bought by a non-musician or beginner in electronic music. Modern synthesisers are extremely sophisticated devices. Instead of the limited number of notes which can be played on a computer, and the limited or non-existent enveloping facilities, typical synthesisers allow you to play chords of up to eight notes, and allow far more flexibility when shaping sounds.

Almost all synthesisers offer an array of preset sounds, so the sounds of a piano or violin, for example, are available at the touch of a button — try doing that on a computer!

Now that both computers and musical intruments are employing the same kind of technology, it's relatively easy to send information from a computer to a musical instrument, or *vice versa*. This is what MIDI is all about.

An interface box will allow you to connect your computer to any piece of MIDI-compatible equipment. MIDI synthesisers, at present, start at about $800 for the Casio CZ 101, but prices are falling all the time. The situation is very like that for other types of computer peripherals — how many of us would have been considering the purchase of printers, colour monitors and disk drives, even two years ago? In the near future, a synthesiser will be comparable in price to these types of peripherals.

'. . . computer enthusiasts are faced with learning the language of musical instruments, and musicians are faced with learning about the mysterious terminology of computers.'

But even before prices fall for the rest of us, current owners of MIDI-compatible musical equipment will find that a home computer and interface is a very attractive proposition. With suitable software a whole range of possibilities is opened up, and any number of dedicated add-ons can be imitated at a fraction of the cost.

Communications

MIDI is a communications standard like Centronics or RS232, although, strictly, it is more akin to RS232 since it is serial and two-way. In this case, though, MIDI is used exclusively in sending musical codes between musical instruments. There have been previous attempts to impose a standard for communication between musical instruments — notably DCB, or Digital Communications Bus — which have failed to become generally accepted.

The story seems destined to be different for MIDI. The standard is set to be universally adopted by all manufacturers of electronic musical equipment and computer equipment which can be used with it — in fact, virtually every piece of serious electronic music-making equipment that has been recently released by every manufacturer is MIDI-compatible.

Yamaha has launched its CX5M MIDI computer, (see Benchtest, May *APC*), an MSX machine that has much circuitry in common with the company's synthesisers — so much so, that all you need add to the machine to obtain a top-flight musical instrument is a piano-type keyboard. But don't be misled, there is no need to sell existing computer equipment in order to use MIDI.

Each piece of MIDI-compatible equipment has three five-pin DIN sockets: these are labelled 'IN', 'OUT' and 'THRU'. IN allows the equipment to receive MIDI signals from another piece of MIDI equipment, OUT is simply the reverse, allowing one piece of MIDI equipment to send out MIDI signals to another piece of MIDI equipment, and THRU sends a direct copy of the incoming information on to another piece of MIDI equipment, as OUT can only drive one IN. In some cases, THRU is not included.

In use

MIDI has been around since 1982, although it has only just been brought to the attention of home computer owners. Musicians have been using MIDI to trigger one instrument from another. For example, two synthesisers can be set to produce different sounds, but can be played simultaneously from one keyboard by connecting the two instruments together via MIDI and playing the keyboard of one of the two instruments.

MIDI also allows a musician to connect a drum machine to a keyboard and synchronise a rhythm track with the melody. Other possibilities include connecting a sequencer: a sequencer is a device which remembers notes, ready for playback at any time. During playback, the music can be changed — tempo can be speeded up or slowed down, as well as the sound characteristics of the notes being changed. As an example, a tune could be played slowly with a sound like a flute. There are two types of sequencer — real time and step time. A real-time sequencer plays back exactly what the musician has played, whereas a step-time sequencer literally steps through the tune with the musician playing each note in turn, filling in individual slices of time until the tune is completed.

Computers

MIDI is much more than just a way of connecting synthesisers and drum machines, and its full potential can only be realised with a computer. The beauty of the system is that any cheap home computer can be used, provided that a MIDI interface is available for it. There is absolutely no advantage in using an expensive business micro, and there is no advantage in using a micro with more sophisticated onboard sound over one with crude onboard sound.

The sound capabilities of the chosen computer are not used at all when connected to MIDI — the sound is always generated by the synthesiser or drum machine — so there isn't any point in purchasing an expensive micro for use specifically with MIDI. Even the extra memory offered by a business machine is largely superfluous as any computer with a memory size of, say, 32k offers far more storage than almost any dedicated sequencer, for example.

Using a computer as a sequencer is one obvious application for a computer connected to MIDI. Using a computer

Commodore 64 programs

```
READY.

10 POKE 56836,3: POKE 56836,86
15 REM *** TRIGGER NOTES ***
20 FOR N=1 TO 12:READ A:POKE 56837,A:NEXT N
30 FOR N=1 TO 400:NEXT N
35 REM *** RELEASE NOTES ***
40 FOR N=1 TO 12:READ A:POKE 56837,A:NEXT N
50 FOR N=1 TO 400:NEXT N
60 GET A$:IF A$<>"" THEN STOP
70 RESTORE: GOTO 20
95 REM *** TRIGGER NOTES DATA ***
100 DATA 144,60,127
110 DATA 144,64,127
120 DATA 144,67,127
130 DATA 144,72,127
135 REM *** RELEASE NOTES DATA ***
140 DATA 128,60,0
150 DATA 128,64,0
160 DATA 128,67,0
170 DATA 128,72,0

READY.
```

Fig 1

```
READY.

10 POKE 56836,3: POKE 56836,86
20 FOR N=36 TD 96
30 POKE 56837,144
40 POKE 56837,N
50 POKE 56837,127
55 GET A$:IF A$="" THEN GOTO 55
60 POKE 56837,128
70 POKE 56837,N
80 POKE 56837,0
90 GET B$:IF B$=" " THEN STOP
100 NEXT N
110 GOTO 20

READY.
```

Fig 2

```
10 POKE 56836,3: POKE 56836,86
50 M$="OFF":S=30
90 PRINT ""
100 PRINT " MIDI STEP TIME SEQUENCER "
110 PRINT "1 :- PLAY SEQUENCE"
120 PRINT "2:- EDIT SEQUENCE"
140 PRINT " REPEAT MODE ";M$;" "
150 PRINT " SPEED=";S;" "
200 BET A$
210 IF A$="1" THEN GOTO 500
220 IF A$="2" THEN PRINT "":LIST 1000-
230 IF A$="R" AND M$="OFF" THEN M$="ON":GOTO 100
240 IF A$="R" THEN M$="OFF":GDTO 100
250 IF A$="F" AND S<60 THEN S=S+1:GOTO 100
260 IF A$="S" AND S>1 THEN S=S-1:GOTO 100
270 GOTO 200
500 PRINT " PRESS ANY KEY TO STOP ":FOR N=1 TO 100:NEXT N
510 RESTORE: READ E
520 FOR N=1 TO E
530 READ A,0,C
540 POKE 56837,144:POKE 56837,A:POKE 56837,B
550 FDR Z=1 TO C*(60-S)+10*C*C
560 NEXT Z
570 POKE 56837,128:POKE 56837,A:POKE 56837,0
580 GET A$:IF A$<>"" THEN GOTO 90
590 NEXT N
600 IF M$="ON" THEN GOTO 510
610 GOTO 90

READY.
```

Fig 3

has advantages over using a dedicated sequencer, such as those offered by the instrument manufacturers. Something like a Commodore 64 and an interface will cost about half the price of the sequencer itself: production on a mass scale and the competition in the computer market-place will probably mean this will remain the case, too.

'The keyboard remains the most popular kind of synthesiser instrument as it is a very convenient way of triggering a synthesiser, and traditional keyboard instruments are widely taught.'

The computer isn't just limited to being used as a sequencer — there are many other possible applications, all hinging on the nature of the software that is being run. For example, a computer could be used to produce a graphics display which could aid in composition. The contents of the computer's memory can be saved to disk — very interesting, when you consider the popularity of digital sound recording when trying to ensure the highest standards of reproduction.

However, the range of software available is still restricted and comparatively high in price. Little or none has found its way into computer shops as yet, but the situation will change as more and more people want to use MIDI.

Nonetheless, even within the restricted software range, there is software which enables sequencers to be duplicated, multitrack music to be composed, and music edited. Although someone with a MIDI set-up may not be able to play a note on any musical instrument, it is perfectly possible to play music by composing on a computer monitor and sending the information to the musical instrument to be played. At the end of this article is some software which will allow you to do just this, although, for reasons of space, it is a little crude.

It has been predicted that some sheet music will be available on MIDI-coded EPROMs, so it will be possible to have either whole pieces of music which can be played back, a little like a record or tape, or to have backing tracks that you can play along with.

Codes

Anyone wishing to write MIDI software will have to know about MIDI codes. These are bytes of information which are sent down the MIDI bus in a standard format that a MIDI-compatible instrument will understand.

A note, for MIDI purposes, consists of two events — triggering and releasing. Triggering is what happens when a key is hit on a keyboard (or a string strummed); and releasing is what happens when the key is allowed to return to its usual position. One thing that always must be borne in mind when writing MIDI software for keyboard synthesisers is that for every triggering, there must be a corresponding release — this is one of the fundamentals of MIDI programming.

An interesting side issue is the question of sequencer specifications. Advertisements usually claim, for example, that a product has a capacity of 10,000 events. Each note consists of two events, so the sequencer will only store 5000 notes.

The number of notes which can be triggered at one time entirely depends on the synthesiser that is being used. Popular synthesisers vary greatly in this respect. For example, the Juno synthesisers are six-voice polyphonic machines, which means that up to six notes can be played simultaneously. On the other hand, the Oscar synthesiser will only allow one note to be played at one time when in monophonic mode. At the very top of the scale, Yamaha DX synthesisers offer 16-note polyphony. If you try to trigger more notes that your synthesiser has voices, the last note will be totally ignored. If you fail to release notes, your synthesiser will simply carry on playing the same note all the time.

In order to trigger the note three consecutive bytes need to be sent to a synthesiser. The three bytes should contain the following information:

(i) A header, or status information, to tell the synthesiser that what follows relates to the triggering of a note.
(ii) A MIDI channel number.
(iii) A pitch or note value.
(iv) A velocity value.

Some of these terms will need an explanation. A MIDI bus can contain information on up to 16 channels (think of them as being like TV channels). A number of instruments can 'tune in' to any particular channel, but there are only 16 programs (in the TV sense) to choose from. You must be sure that your synthesiser is working on the correct channel to receive information that is being sent to it. Some older instruments will only operate on one particular MIDI channel.

'. . . synthesisers offer an array of preset sounds, so the sounds of a piano or a violin, for example, are available at the touch of a button — try doing that on a computer!

Spectrum programs

```
 10 OUT 159,3: OUT 159,86
 15 REM   Trigger notes
 20 FOR n=1 TO 12: READ a: OUT 191,a: NEXT n
 30 FOR n=1 TO 200: NEXT n
 35 REM   Release notes
 40 FOR n=1 TO 12: READ a: OUT 191,a: NEXT n
 50 FOR n=1 TO 200: NEXT n
 60 IF INKEY$<>"" THEN  STOP
 70 RESTORE : GO TO 20
 95 REM  Trigger notes data
100 DATA 144,60,127
110 DATA 144,64,127
120 DATA 144,67,127
130 DATA 144,72,127
135 REM  Release notes data
140 DATA 128,60,0
150 DATA 128,64,0
160 DATA 128,67,0
170 DATA 128,72,0
```

Fig 1

```
 10 OUT 159,3: OUT 159,86
 20 FOR n=36 TO 96
 30 OUT 191,144
 40 OUT 191,n
 50 OUT 191,127
 55 PAUSE 0
 60 OUT 191,128
 70 OUT 191,n
 80 OUT 191,0
 90 IF INKEY$=" " THEN  STOP
100 NEXT n
110 GO TO 20
```

Fig 2

Fig 3

```
  10 OUT 159,3: OUT 159,86
  20 POKE 23658,8
  50 LET M$="OFF": LET S=30
 100 BORDER 7: PAPER 7: INK 0: CLS : BRIGHT 1
 110 PRINT AT 3,2;" MIDI STEP TIME SEQUENCER "
 120 PRINT AT 7,5;" 1:- PLAY SEQUENCE "
 130 PRINT AT 9,5;" 2:- EDIT SEQUENCE "
 140 BRIGHT 1: PRINT AT 14,6;" REPEAT MODE "; INVERSE 1;M$; INVERSE 0;" "; BRIGH
T 0;" "
 150 PRINT AT 16,9;" SPEED ";TAB 19-LEN STR$ S; INVERSE 1;S; INVERSE 0;" "
 180 BRIGHT 0
 200 PAUSE 0: LET A$=INKEY$
 210 IF A$="1" THEN  GO TO 500
 220 IF A$="2" THEN  CLS : LIST 1000: STOP
 230 IF A$="R" AND M$="OFF" THEN  LET M$="ON": GO TO 140
 240 IF A$="R" THEN  LET M$="OFF": GO TO 140
 250 IF A$="F" AND S<60 THEN  LET S=S+1: GO TO 140
 260 IF A$="S" AND S>1 THEN  LET S=S-1: GO TO 140
 270 GO TO 200
 500 PRINT  INVERSE 1; BRIGHT 1;AT 20,3;" PRESS ANY KEY TO STOP ": PAUSE 12
 510 RESTORE : READ E
 520 FOR N=1 TO E
 530 READ A,B,C
 540 OUT 191,144: OUT 191,A: OUT 191,B
 550 FOR Z=1 TO C*(20-S)+C*30
 560 NEXT Z
 570 OUT 191,128: OUT 191,A: OUT 191,0
 580 IF INKEY$<>"" THEN  GO TO 100
 590 NEXT N
 600 IF M$="ON" THEN  GO TO 510
 610 GO TO 100


1000 DATA 48
1010 DATA 53,120,2
1020 DATA 60,110,2
1030 DATA 65,110,2
1040 DATA 60,110,1
1050 DATA 53,115,2
1060 DATA 60,110,1
1070 DATA 65,110,1
1080 DATA 60,110,1
1090 DATA 53,115,1
1100 DATA 60,110,1
1110 DATA 65,110,1
1120 DATA 60,110,1
1130 DATA 49,120,2
1140 DATA 56,110,2
1150 DATA 61,110,2
1160 DATA 56,110,1
1170 DATA 49,115,2
1180 DATA 56,110,1
1190 DATA 61,110,1
1200 DATA 56,110,1
1210 DATA 49,115,1
1220 DATA 56,110,1
1230 DATA 61,110,1
1240 DATA 56,110,1
1250 DATA 51,120,2
1260 DATA 58,110,2
1270 DATA 63,110,2
1280 DATA 58,110,1
1290 DATA 51,115,2
1300 DATA 58,110,1
1310 DATA 63,110,1
1320 DATA 58,110,1
1330 DATA 51,115,1
1340 DATA 58,110,1
1350 DATA 63,110,1
1360 DATA 58,110,1
1370 DATA 48,120,2
1380 DATA 55,110,2
1390 DATA 60,110,2
1400 DATA 55,110,1
1410 DATA 48,115,2
1420 DATA 55,110,1
1430 DATA 60,110,1
1440 DATA 55,110,1
1450 DATA 48,115,1
1460 DATA 55,110,1
1470 DATA 60,110,1
1480 DATA 55,110,1
```

The velocity value information only applies to touch-sensitive instruments — a touch-sensitive keyboard acts in the same way as a piano keyboard. If you hit a key hard, the note sounds loud; if you hit a key soft, the note sounds quiet. This information has to be sent even if your instrument is not touch-sensitive — the instrument will ignore the unwanted information.

Returning to the four pieces of triggering information, the first two pieces make up the first byte as two four-bit codes. The binary number 1001 means 'trigger a note', and constitutes the high end of the byte — bits four to seven; the low bits constitute the channel number. Confusion could creep in here. As far as MIDI is concerned, these channels are numbered 0 to 15. You will find that musicians and instruments refer to them as 1 to 16, so don't forget to make the adjustment.

It will probably become clearer with an example. A byte looks like this:

10010000

means 'trigger a note (1001) on channel zero (0000)'. In decimal, the byte has the value 144. Any synthesiser tuned into channel zero would prepare to receive a note.

There is no need to start off in binary and convert the codes to decimal every time, as there is a general formula which will help you calculate the first byte:

144+required MIDI channel-1

or, more simply:

143+required MIDI channel

10010000 is the binary equivalent of 144, and −1 makes the adjustment from 1 to 16 channel numbers into 0 to 15 channel numbers.

The second byte in the set of three contains a number in the range 1 to 127, and relates to the note you want to trigger. Notes are governed by a simple number system, with middle C at note number 60. Increasing the value by 1 will make the note increase in pitch by a semitone. Likewise, decreasing the value by 1 will lower the note by a semitone.

Notes 16 and 19 represent a minor third (or three semitones). Notes 30 and 42 are an octave apart — 12 semitones. If you have a knowledge of music you will see that this system is far easier to use than the old one volt per octave system, which was analogue as opposed to digital.

Although your instrument may only be able to be played through a keyboard, say, five octaves long, MIDI can trigger notes over more than 10 octaves. One interesting side-effect of this is that you can play music through MIDI that would be impossible to play directly from the keyboard. It depends on the type of synthesiser you have, but it's certainly worth a try once you are happy with the system.

The third and final byte holds the velocity information; it should contain a value between 1 and 127. Most instruments, particularly those at the cheaper end of the market, will not be able to interpret this byte, but it should be there even if the instrument ignores it. A touch-sensitive synthesiser like the Roland JX8P, or the Yamaha DX7, will interpret higher numbers as harder hit keys. Thus, 127 will trigger the note with a maximum velocity, while 20 will trigger a much quieter note. The exact result will also depend on how the synthesiser itself has been programmed.

An example of an event which will trigger a note could look like this:

144,64,100

These three bytes, when sent in succession, will trigger E above middle C on any keyboard 'listening' on channel 1, with almost maximum velocity.

Releasing a note is almost the same as triggering one, except that the code in the high bits of the first byte has to be

AMPEX 230

The Wise Alternative

Introducing the latest Ampex product into Australia, the 230 terminal; setting new standards in features and value-for-money.

The 230 follows on from the incredibly popular Ampex 210. It is a terminal that the wise should consider when evaluating top-of-the-range terminals.

Why?

Because the 230 is super smart, with a 14" amber CRT and selectable 80 or 132 column mode.

It has been specially designed for the modern office, and boasts ergonomic styling, Selectric style adjustable keyboard and tilt-and-swivel display.

Other outstanding features of the 230 include:

- High level emulations including the entire Ampex family, Televideo up to the 950 and the WYSE WY-50. **Note:** DEC emulations are not supported by the 230. These will be available in the 220 terminal, which will be released soon.
- 15 line and 64 block graphic symbols.
- Two standard 80 column display pages, expandable to four.
- 32 programmable function keys with memory expandable to 6,400 bytes.
- Addressable cursor with double high and wide characters and a four speed scroll.
- A primary and auxiliary port with a baud rate to 38.4K.

The Ampex 230 is supported throughout the nation by Datascape, Australia's largest independent peripheral supplier.

If you're still not convinced then check the price. It will prove just how wise you are to select the 230.

Mail coupon to:
Datascape.
Sydney office: 44 Avenue Rd, Mosman. Tel. (02) 969 2699.
Melbourne office: 27 Raglan St, South Melbourne.
Tel. (03) 690 3622.
I would like to know more about the Ampex 230 terminal. Please
☐ send literature
☐ call me

Name ....
Title ....
Company ....
Address ....
....
Phone ....

OPR

DATASCAPE

Freedom for your information

changed. The formula calculating the value of the first byte in the case of releasing a note is:
127+MIDI channel number

The second byte holds the note number and follows the same conventions as for triggering a note.

The third byte has no real role to play in releasing the note, but is sent to preserve symmetry between triggering and releasing. You can send any value you wish, but conventionally, the byte is set to zero.

'MIDI is an expanding system, with more codes being added to control different types of instruments, or features within instruments which are being added as technology advances.'

To release the E above middle C triggered in the above example, these three bytes would have to be transmitted:
128,64,0

Attempting to release an untriggered note will have no effect.

Programs

For those who already have MIDI equipment, here are three MIDI programs for you to try. The program explanations should make MIDI codes clear. Program 1 demonstrates how to trigger a chord of four notes and release them — you will need a polyphonic synthesiser for this one. Line 10 initialises the interface and prepares it to transmit data in MIDI format. Line 20 triggers the four notes by reading 12 bytes from the data statements in lines 100 to 130. These bytes are output consecutively through port 191 on the Spectrum and location 56837 on the Commodore 64. If you understand music you should be able to examine the data and work out that the chord is C major. In any case, you should see that the chord has been triggered with maximum velocity via MIDI channel 1. Line 30 is a delay loop and represents the sustain time — how long the notes are played for — before line 140 to 170, and releases the notes that have been triggered earlier.

Having released the notes, the computer's keyboard is checked for key presses. If no key is being pressed, line 70 restores the data pointer to the start of the data and the program runs again. If any key is being pressed, the program stops.

Running this program may result in a little disappointment! The notes do not trigger anywhere near simultaneously, owing to the time taken by a Basic program to output 12 bytes. It is an unfortunate fact that machine code will have to be used for any polyphonic work, but for clarity's sake the programs in this article have been written in Basic.

Before you finish with the program try altering the values in the data statements, and if you have a touch-sensitive keyboard try altering the velocity byte, currently set at the maximum value of 127.

Program 2 shows how to trigger a chromatic scale over a five-octave range from MIDI. The scale starts from two octaves below middle C — note number 36 — and the program triggers each number in turn. The key numbering system lends itself to using simple FOR/NEXT loops. The program's main loop runs between 36 and 96 using the value assigned to the variable N as the MIDI key number in triggering or releasing. Lines 30 to 50 transmit the three bytes of data needed to trigger a note. Line 55 waits for a key to be pressed — if you press the space bar the program will stop, but any other key will release the note. The releasing is done in lines 60 to 80. Line 90 is another check for pressing the space bar — if a different key is being pressed, the program loops back to play the next note in the scale.

Try substituting these alternative line 20s:

```
20 FOR n=96 TO 36 STEP -1
```

or

```
20 FOR n=36 TO 96 STEP 4
```

The first plays the scale backwards and the second plays every fourth note (intervals of major thirds).

If you want to see how many of the possible 10 1/2 octaves you can play on your synthesiser, try altering line 20 to:

```
20 FOR n=1 to 127
```

In some cases, you will be triggering sounds of just a few hertz (cycles per second), and you may be able to distinguish individual oscillations as a series of clicks.

It is relatively easy to extend the basic MIDI principles in order to write a piece of serious MIDI software. Program 3 is a simple but complete monophonic sequencer. As a monophonic program, it should work with any MIDI-compatible synthesiser.

After initialisation, lines 100 to 200 print the menu. Option 1 plays the stored sequence. Option 2 allows you to edit the sequence — when you have finished altering the sequence, you will need to run the program once more. Pressing R will set the computer in repeat play mode, and pressing R once more will switch off the option. Pressing F will make the sequence play faster, and pressing S will make the sequence play slower.

If you choose '1', the routine from line 500 to 610 is selected. This is the core of the program, triggering and releasing all the notes, and taking care of the timing which forms the rhythm.

The note data is in the lines from 1000 onwards. Some sample data has been given so that you can test the program, but by substituting your own data you will be able to create sequences.

Choosing option 2 will list the data lines; you can now use the Basic editor to edit the sequence. The first data line contains a single figure — this is the number of notes in the sequence. All the subsequent lines contain three pieces of data: the first is the MIDI note code (falling between 1 and 127); the second is the key velocity value (falling between 1 and 127, although if your synthesiser is not touch-sensitive, this value will have no effect); the third is a relative note length (a 4 will last twice as long as a 2). The third piece of data is only relative because F and S adjust the speed of playback.

The program suffers slightly from the limitations imposed by Basic. As you saw earlier, it is not really possible to write polyphonic programs because of the speed limitations of the language. You will also find that, at higher speeds, the timing becomes increasingly inaccurate. Writing the software in machine code will allow you a polyphonic synthesiser with correct timing, but will take considerably more effort to write.

Conclusion

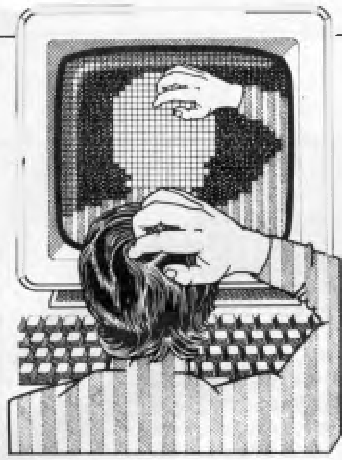
There is much, much more to MIDI than I can cover here. MIDI is an expanding system, with more codes being added to control different types of instruments, or features within instruments which are being added as technology advances.

At present, in addition to the information covered in this article, MIDI is able to transmit data relating to 'performance control'. Pitch bend is one example of this — most synthesisers allow the player to 'bend' the pitch of a note and create some of the effects heard in today's pop music. MIDI also enables the musician to select a different sound from the systhesiser's internal memory (assuming it is programmable). This feature is particularly useful when playing live by keeping both of the musician's hands free to play the instrument.

The other main area of MIDI concerns the so-called 'system exclusive' information; this is the non-standard part of MIDI. Each manufacturer of MIDI equipment has his own ID number, and if this ID byte is transmitted then the following information will only make sense to that particular manufacturer's equipment. The most common application of this feature is software which enables you to create sounds for your instrument from the computer keyboard.

This software was written using the SIEL MIDI computer interface, and will work on any system using port 159 on the Spectrum or location 56836 on the Commodore 64 for control, and port 191 on the Spectrum or location 56837 on the Commodore 64 for transmission.

If your interface does not use these ports or locations, you will have to modify the OUTs in the Spectrum program or the POKEs in the Commodore 64 program. Consult your machine manual for the correct values.



TJ'S WORKSHOP

Our monthly pot-pourri of hardware and software tips for 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–$25 for any tips we publish. APC can accept no responsibility for damage caused by using these tips, and readers should be advised that any hardware modifications may render the maker's guarantee invalid.

SPEEDING UP THE SPECTRUM

The following routines will save processing time on either Spectrum: machine code routines to change the value of PROG in order to speed up GOTO and hence GO SUB, FOR NEXT loops, and so on; and as an example of this, and also of using string arrays instead of string variables, a routine for INPUTing data. It is rather lengthy but more user-friendly than any I have met in a professional Basic program or even most machine code programs.

1) GOTO lines late in a Spectrum Basic program are slow, and make some FOR NEXT and other loops a lengthy process. If you temporarily change the system variable PROG to point to the middle of your program, you can hide the first part of the program and dramatically speed up some jumps.

I use two machine code routines, 'prognext' and 'progback' (Fig 1). Prognext sets PROG to the address in NXTLIN, which is the address of the next program line. Progback resets PROG to its normal value, which it calculates from STREAMS and CHANS. If you have set up a channel of your own, you will get a STATEMENT LOST error report unless you have opened a stream to point to it. Both routines return zero and are relocatable anywhere above RAMTOP. They are normally used in pairs, and there must be no references to earlier lines while PROG is changed.

Fig 2 shows a typical use, setting up data for a new run. At the end of a very long program, the time for a similar loop was cut from 288 frames to 55.

Fig 3 shows a use in which there is more likely to be an error while PROG is changed, and here a safety device is included in case anyone tries to RUN with PROG pointing to line 2015.

2) When a string variable is assigned a new value, it is always moved to the end of the variable list which is why string manipulation can be slow.

It is often worth avoiding the use of a string variable either with, for example, LET k=CODE INKEY $, or by using a string array. Fig 3 inputs a string of up to 30 characters (with CAPS LOCK set) using the main screen. It uses LAST K (23560) to take advantage of the auto repeat, and is much faster than simpler input routines that use string variables.

R Oakeshott

```
3000 LET prognext = USR "a"
3010 LET progback = prognext + 10
3030 RESTORE 3050
3040 FOR 1 = prognext TO prognext + 44: READ a: POKE i,a: NEXT i
3050 DATA 42,85,92,34,83,92,1,0,0,201,33,16.92,175,6,19,190,48,1,
        126,35,35
3060 DATA 16,248,198.4,79,42,79,92,9,126,254,128,32.6,35,126,254,
        40,56,217,195,236,27
```

Fig 1 Prognext and progback

```
9900 REM set up array from data in line 9040 onwards
9920 RESTORE USR prognext: REM computer now thinks prog starts at 9925
9925 DIM a(100)
9930 FOR i = 1 TO 100: READ a(i): NEXT i
9940 DATA
```

Fig 2

Fig 3

```
10 REM Declare variables to get them high on list
20 DIM a$(31): LET key = 0: LET len = 0: LET curs = 0
30 REM
```

```
1000 REM
2000 REM input i$ with up to 30 chars
2002 REM enter at 2010 to edit i$
2005 LET i$ = ""
2010 RANDOMISE USR prognext: GO TO 2020: REM prog points to line 2015
2015 RANDOMISE USR (USR "a"+10): RANDOMISE USR 7148: REM it is an error
     if you get here. Reset PROG & report STATEMENT LOST
2020 LET a$ = i$: LET len = LEN i$: LET curs = len: PRINT£0|: REM
     PRINT£0 clears bit 5 of flags
2025 REM
2030 PRINT AT 0,0; ">";a$(TO curs)|"*"|a$(curs+1 TO Ien+1)
2040 IF PEEK 23611,<224 THEN GOTO 2040
2050 POKE 23611,PEEK 23611-32: LET key = PEEK 23560: BEEP .004,40
2055 REM
2060 IF key >= 32 AND key <= 90 THEN IF len < 30 THEN LET a$(curs+1 TO)
     = CHR$ key + a$(curs+1 TO): LET curs = curs+1: LET len = len+1:
     GOTO 2030
2070 IF key = 12 THEN IF curs THEN LET a$(curs TO) = a$(curs+1 TO):
     LET curs = curs-1: LET |len = Ien-1: GOTO 2030
2080 LET curs=curs-(key=8 AND curs)+(key=9 AND curs<len)
2090 IF key<>13 THEN GOTO 2030
2095 REM
2100 PRINT AT 0,1; a$(TO len+1)
2110 LET i$=a$(TO len)
```

ALPHABETICAL SORT

This routine sorts a list by initials (but can be expanded with a subroutine or multi-dimensioned arrays). The sort is arranged 510-530 and the printed sort at 550-590. This listing is from a Sharp but needs little change for any Basic.

Sort time for 400 entries is 24 seconds; on the Sanyo MBC550 it is around five seconds. The sort program takes 345 bytes plus 26 for the array, and requires four bytes per entry (this can be three if the string$ does not have a space before the number). Resort varies little. If it was frequent, lines 515 and 520 could be moved to 435, thereby creating sort on entry with no noticeable delay and instant print from 550.

The technique is of interest rather than the program, as it can be used to good effect in many areas. Data can be merged from several sources and sorted on entry.

H Parker

```
10 REM ALHABETICAL SORT ROUTINE - H.W.PARKER KNIGHTON  POWYS
100 DIM S$(200):S1=1
110 DIM S2$(25)
200 REM CLS:PRINT ".    MENU":PRINT "    --------"
210 PRINT "1.  ADD TO LIST":PRINT:PRINT"2.  LIST AS ENTERED"
220 PRINT:PRINT "3.  LIST SORTED"
230 PRINT:INPUT X:ON X GOTO 300,400,500,200
300 REM CLS:PRINT"."
310 PRINT "ENTER  .  TO END":PRINT
320 FOR S1=S1 TO 200:PRINT S1+1;" "::INPUT ".",S$(S1)
330 IF S$(SI)="." THEN 200
340 NEXT S1
400 REM CLS ".
410 A=1
```

```
420 FOR S=1 TO S1:PRINT S$(S): A=A+1
430 IF A=20 THEN A=1:INPUT "C = CONTINUE : E = END:";A$:IF A$="E" THEN 450
440 NEXT S
450 INPUT "E = END : R = REPEAT ";A$:IF A$="R" THEN 400
460 GOTO 200
500 REM CLS:PRINT".":FOR I=0 TO 25:S2$(I) ="": NEXT I
510 FOR S = 1 TO S1-1
515 Z = ASC(S$(S))-65
520 Y=LEN(STR$(S)):S2$(Z)=S2$(Z)+SPC(4-Y)+STR$(S)
530 NEXT S
540 INPUT "SORTED L = LIST :";L$
550 A=1:B=0
560 FOR S= 1 TO LEN(S2$(B))+4 STEP 4:W=VAL(MID$(S2$(B),S,4)):IF W=0 THEN 600
570 PRINT S$(W):A=A+1
580 IF A=24 THEN INPUT "C = CONT:";A$ :A=1
590 NEXT S
600 B=B+1:IF B<26 THEN 560
610 INPUT "END OF LIST ";A$:GOTO 200
```

COMMODORE 64 JOYSTICK READER

This routine is a joystick reader. There are many methods of reading the joystick from Basic, but they all involve slow bit manipulation and a whole series of commands to return a simple value. This program is entirely in machine code, so it operates almost instantaneously and returns a three-character string. The string is put into any specified variable; it always contains three characters, and represents the condition of either one of the joysticks. The command takes the form:

10 SYS 49152,1,A$

In this case, the variable A$ would be left containing a string, representing the condition of joystick number one (on port one). It is possible to use any name for the string variable, and either one or two for the port number. The data returned is always one of the following:

—: No direction
U—: Up only
D—: Down only
L—: Left only
R—: Right only
UL: Up & left
UR: Up & right
DL: Down & left
DR: Down & right

'F' indicates that the fire button is being pressed. The string 'ULF' would mean that the joystick was being pushed up and to the left, with the fire button being pressed.

The demonstration program shows it in use. If you miss out one of the commas or don't use the exact syntax required, the program will halt with a syntax error.

S Mehen

```
10 REM ******************************
20 REM *                            *
30 REM *  STEVE'S JOYSTICK READER   *
40 REM *  -----------------------   *
50 REM *    COPYRIGHT JAN., '85     *
60 REM *    -------------------     *
70 REM ******************************
80 :
85 AD=49152:CS=0
90 FOR L=0 TO 21:FOR E=1 TO 7
100 READ DA:CS=CS+DA:POKE AD,DA
105 LS=LS+DA:AD=AD+1:NEXT
110 READ C:CS=CS+C:IF LS<>C THEN 200
120 LS=0:NEXT:IF CS<>30362 THEN 300
130 PRINT:PRINT"ALL DATA COMPLETE."
140 END
150 :
200 PRINT:PRINT"DATA ERROR IN LINE";
210 PRINT 500+L*5:STOP
220 :
300 PRINT:PRINT"THERE ARE AT LEAST TWO"
310 PRINT"ERRORS IN THE DATA, POSSIBLY"
320 PRINT"INCLUDING THE LINE CHECKSUM."
330 STOP
499 :
500 DATA 32,253,174,201,49,240,9,958
505 DATA 201,50,240,5,162,11,108,777
510 DATA 0,3,233,49,73,1,133,492
515 DATA 10,32,72,192,32,82,192,612
520 DATA 32,115,0,32,253,174,32,638
525 DATA 139,176,133,73,132,74,165,892
530 DATA 122,72,165,123,72,169,67,790
535 DATA 133,122,169,192,133,123,32,904
540 DATA 158,173,104,133,123,104,133,928
545 DATA 122,76,44,170,34,46,46,538
550 DATA 46,34,166,10,189,0,220,665
555 DATA 41,31,133,10,96,165,10,486
560 DATA 72,41,15,133,10,169,15,455
565 DATA 229,10,10,170,189,122,192,922
570 DATA 141,68,192,189,123,192,141,1046
575 DATA 69,192,104,41,16,208,6,636
580 DATA 169,70,141,70,192,96,169,907
585 DATA 45,208,248,45,45,85,45,721
590 DATA 68,45,45,45,76,45,85,409
595 DATA 76,68,76,45,45,82,45,437
600 DATA 85,82,68,82,40,67,41,465
605 DATA 83,46,77,69,72,69,87,503

READY.
```

```
10 REM JOYSTICK READER DEMONSTRATION
20 REM READER (C) STEVE MEHEW, 1985
30 :
40 POKE 53281,0:POKE 53280,0:PRINTCHR$(147)
50 PRINT:PRINT"PLUG JOYSTICK INTO PORT 2"
55 PRINT:PRINT
60 SYS 49152,2,A$
70 PRINTSPC(4);A$
80 PRINTCHR$(145);:GOTO 60

READY.
```

AMSTRAD TEXT PROCESSOR

The Amstrad CPC464 has a powerful resident program editor. Here is a short program which uses its facilities to provide a simple text processor.

Each line of input text must be preceded by an apostrophe, and the last line of text must be finished by pressing the ENTER key. This line must not be subsequently edited.

H Pitcher

```
10 INPUT"Enter c/d/e/p/s to create, display, edit, print or save the text: ",a$
15 IF a$="c" THEN GOSUB 45 :AUTO 100
20 IF a$="d" THEN stream=0 :GOSUB 50 :GOTO 55
25 IF a$="e" THEN GOSUB 45 :LIST 100-
30 IF a$="p" THEN stream=8 :GOTO 55
35 IF a$="s" THEN INPUT"Enter filename: ",f$ :SAVE f$ :GOTO 80
40 GOTO 10
45 KEY 139,CHR$(206)+CHR$(13)
50 MODE 2 :INK 0,25 :INK 1,0 :RETURN
55 i=900
60 WHILE PEEK(i)<>192 :i=i+1 :WEND
65 i=i+1 :n=PEEK(i)
70 WHILE 31<n AND n<164 :PRINT#stream,CHR$(n); :i=i+1 :n=PEEK(i) :WEND
75 PRINT#stream :IF n<>206 THEN 60 ELSE END
80 KEY 139,CHR$(13) :MODE 1 :INK 0,1 :INK 1,24 :END
```

BBC OS CALLS

These hints and tips have many uses, both from Basic and machine code, and can also be used as a method of program protection.

OSBYTE calls on the BBC:

*FX 200,x where x is
0-normal break & escape
1-escape disenabled
2-break clears memory
3-esc disenabled & break clears memory

*FX 201,1 disenables keyboard.

*FX 201,0 re-enables keyboard.

*FX 138,0,x inserts ASCII value x into character buffer. x=128 gives function key 0 so that:

Mastermodeller. Mainframe modelling power on your desk.

Mastermodeller starts where Lotus ends.

Microcomputer spreadsheet and modelling programs have revolutionised financial modelling. But none could rival the power of a mainframe modelling system.
Until Mastermodeller.

Mastermodeller really exploits the power of modern 16 bit business micros. A complete 5th generation modelling language that gives you complete control over modelling and reporting. Fast. Easily.

- The command structure is like English. There are no obscure command codes to learn.
- Consolidation can be created graphically – you set up the structure, the machine does all the coding. Automatically.
- Report generation is easy and flexible. You define on the screen what the report should look like. Mastermodeller does the coding.
- Mastermodeller can accept data directly from spreadsheet and database files. Including direct access to Lotus, dBase II and Micromodeller.
- Goal seeking or backward iteration capability.
- Powerful NPV, IRR and conditional logic functions.
- Unlimited use of variables and ONERROR.

Sound interesting? To get a hands-on demonstration of Mastermodeller, call Intelligence Australia today. It's a lot cheaper than a mainframe. And takes up a lot less desk space.

Mastermodeller runs on IBM PC, IBM compatible, DEC, Hewlett-Packard, Wang, Sirius and Apricot microcomputers.

Intelligence

Sydney: 60-64 Shepherd Street Chippendale NSW 2008
(02) 699 3877 TLX: 25331
Melbourne: 123 Moray Street South Melbourne VIC 3205
(03) 690 6600 TLX: 38027
Brisbane: (07) 343 9122 Perth: (09) 322 1677

```
10 *KEY 0 *.''MCHAIN''
20 *FX 138,0,128
30 END
```

will catalogue the disk and type chain ready for another program.

*FX 211,channel number — this redefines the channel number for the BELL character (& COPY KEY) :default=3.

*FX 212,amplitude or env. number
— for env:-(ENV-1)*8 AND 255
— for sound:-(VOL-1)*8 AND 255

For example, *FX 212,216 gives softer bell (VOL= −4) :default=144

*FX213,freq. Bell frequency: default=101

*FX 214,duration. Bell duration: default=7.

*FX 219,ASCII Define TAB key: default=9.

*FX 220,ASCII Define ESC key: default=32.

*FX 247,76
*FX 248,lo
*FX 249,hi

These redefine the BREAK key's vector, and can be used to jump to the user's own routine in the computer.

*FX 255,8-15 mode number+8 on break.

Memory locations:

?&355 current mode number.

?& 350,?&351 — start of screen memory as seen by 6845 chip. These vary as the screen is scrolled.

~!&3C2 gives the execution address of a tape program.

~!&3BE gives the load address of a tape program.

Finally, $&3B2 gives the last used file name, again only on tape.

M Warriner

ATARI ASSEMBLER

The Atari assembler-editor lacks an important feature — the ability to direct output to the printer. This feature would be very useful if, for example, you wanted to use the built-in disassembler to disassemble a large block of memory. It would be nice if you could dump the whole disassembly listing to your printer, to be perused at your leisure. Or if you were debugging your program using the TRACE command, to have a printed copy of the trace listing.

Fortunately there is a way of tricking the system to divert the output meant for the screen to your printer instead. While in debug mode type:

C346<A6,EE

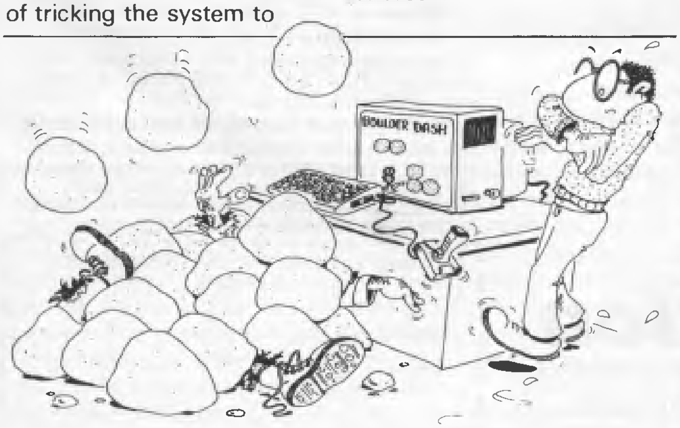
This substitutes the address of the putbyte subroutine of the printer handler for the putbyte subroutine of the display handler. Whenever the assembler-editor tries to print a character on the screen, it will appear on the printer instead.

To redivert output to the screen, type in the following in debug mode:

C346<A3,F6

This puts the address of the putbyte subroutine of the display handler back into its proper place.

H Hsu



Turbo Pascal

Borland International's Turbo Pascal was given a warm reception when it was first announced — if only for its price. Tanj Bennett considers whether its welcome was deserved.

Turbo Pascal (TP) is a compiler designed solely for use with a personal computer. In addition to converting Pascal programs into machine code, it acts as a user interface between edit and compilation, and includes a useful editor which is closely linked to the compiler. Borland sells TP for CP/M, CP/M-86 and MS-DOS.

This review is based on the MS-DOS version, running on a Sirius. If the manual is accurate, the CP/M-86 version is closely related, while the CP/M version lacks a few features but is still fairly complete.

Probably every micro owner has seen or used Basic computer programs. Basic is an 'interpreter', which means that it operates in only two steps: you type in the program, and then the computer runs what you typed. It is called an interpreter because the Basic language you type in is not the same as the machine code language built into the hardware, so an interpretation from Basic to machine code is occurring as the program runs. This principle applies to most computer languages.

Compilers work by planning everything in advance. They must work through the typed program (the 'source'), planning the space for all the data and variables, and preparing the interpretation of the statements as machine code. The result is tied together with a 'run-time library' — machine code and data prepared for such things as reading disk files or calculating exponentials —and the whole result is a program ready to run (the 'object'). This is generally stored on disk and called up to run whenever needed. The compilation activity makes programming a three-stage effort: type the source program, compile source into the object, and run.

The object program can be re-run any number of times, independently of the first two stages, and can be sold as a product separate from the source. The neatness of the source, the length of variables, or the comments, have no effect on the way the data is spaced out or the machine code is prepared, so good style costs nothing when the program is in use.

Starting up

Turbo Pascal comes on a single diskette, with a manual and newsletter sheets. An installation program, Tinst, is provided to tailor the keyboard and screen control for a variety of terminals; the range is wide and the tailoring seems effective. My copy came properly set up for the Sirius. If you need a terminal which is not on Borland's list of more than 20 common types, the program will allow you to specify a new type. A printing program is also on the diskette which will paginate your text, supply line numbers, and underline keywords.

When you have checked that Borland has shipped the proper keyboard and screen definition, you are ready to begin with TP.

TP takes control of the screen and keyboard. It asks if you want to load error messages (so that it can explain problems, instead of just giving an error number.) Unless you are cramped for memory, answer 'y'. Next, the main menu comes up. All TP's menus use single keystrokes, which are brightened, and the menus are generally good but you do need to read the manual before using some of the options. Response is as fast as the screen can be written.

To begin programming you will need a work file — that is, a source program text. You can name this by using the W option. If you type E for edit when you have no work file, TP will first ask you for a work file, then switch to editing.

Borland claims that its editing is based on a subset of WordStar. It is a fairly good full-screen editor but not a word processor, as it lacks any formatting control. The features include searching, replacement, auto-indent, block operations, and reads and writes to outside files. It is a little slow and sometimes clumsy to use. The tab function deserves special mention for peculiarity. Tab stops are defined by the words in the preceding line. This is presumably intended to give an attractive layout, but is likely to be exasperating if you are accustomed to any other editor. It is interesting that none of the example texts provided by Borland are laid out using tab stops like these! The redeeming feature of the editor is the close tie it has to the compiler, and on balance it serves its purpose.

All the editor's text must fit into memory. TP is unusually small, requiring around 50k of RAM for non-text purposes. With a 128k Sirius about 48k is left for text: the editor warns you just before you reach the end of memory. The compiler has several techniques, such as including subordinate files or structuring a program as a set of overlays, which should allow you to reorganise a program to avoid this limit. There is a limit of 64k on the size of your text file irrespective of how much memory your computer may have, and a limit of 64k on the object program which can only be escaped through overlays.

Compiling

When you have prepared a program, you

exit from the editor back to the main menu, and you might wish to save the program before running it. Saving a program takes one keystroke and a normal length of time to write the file to disk. TP does rename the last version to a .BAK extension as a backup file in case you regret a change.

TP can compile a program either to memory or to disk. When TP is started, it takes the available memory and organises it for its own purposes. The editor requires each source file to fit within memory, and so does the compiler — in fact, it seems reasonable to assume that the compiler and editor share the text. If you have enough space left over, the compiler will place the object code in memory as well. This mode of compilation is very quick — on a Sirius the compiler regularly compiles at 2500 lines per minute (around 1000 characters per second).

Of course, you can't hold an arbitrarily large program in memory. On the Sirius, a program of 600 to 700 lines is the largest that fits; to go beyond this, you must compile programs to disk. The Calc example program supplied by Borland runs to more than 1200 lines and compiles at a slower 600 lines per minute. This figure is disk-limited, since a trial with a fast 'fixed' disk ran nearly as fast as compiling to memory.

The option of compiling to disk is manually selected. One neat trick is that if you have successfully compiled a program to memory and then decide to compile it to disk, TP detects that it has the compilation in memory and writes it directly to disk without recompilation.

The slower compilation to disk can be frustrating with longer programs. TP only compiles up to the first error. Each time it compiles, it starts from the beginning. When an error is found, an error message is displayed. You are asked to press a key, and the mode swaps into the editor with the cursor at the position of the error (the message disappears, so make sure you read it). The messages are usually appropriate, although there are times when the real problem is not obvious to the compiler and it stops in confusion, so you may need to hunt backwards to find the true problem. This is not Borland's fault: it occurs with all Pascal compilers.

TP produces only .COM files (.CMD for CP/M-86) which are complete programs. These generally load faster than .EXE programs and are advantageous as long as the compiler can produce them. There is no way of using TP to produce object modules for later linkage into complete programs, and only the most rudimentary of mechanisms for making use of outside objects from TP programs. There is a mechanism for planting machine code in-line within TP programs, but unless you are very fond of hexadecimal numbers this is not a useful method.

Documentation

TP comes with a manual and some examples. The manual is paperback, 300 pages thick, mostly written for version one but with about 40 pages of appendix to describe version two. Each section has a table of contents and an index. The manual is written in more or less the order you need to get started. It includes a description of the editor, the installation program, the Pascal language, and the appendices describe extensions to the language or the system. I have mainly used the book for reference, for which the index proved rather weak, although the required information was usually there somewhere.

The examples consist of one medium-sized program for a toy spreadsheet and a few smaller examples of how to call into the MS-DOS operating system. The spreadsheet is the most important since it's the most interesting to try, and uses the widest variety of TP features. However, the program as supplied supports only 20 rows and six columns, so you can't even squeeze your home finan-

```
s ('move');
FOR k := 1 TO 10000 DO p := q;

s ('sets');
somenumbers := [2,3,5,7,11,13];
FOR k := 1 TO 10000 DO
    BEGIN
    j := k MOD 16;
    IF j IN somenumbers THEN somenumbers := somenumbers - [j]
                        ELSE somenumbers := somenumbers + [j];

    END;

s ('opts');
a := 2; b := 3; c := 5; d := 7;
FOR k := 1 TO 10000 DO
    BEGIN
    j := k MOD 10;
    l := k DIV 10;
    matrix [j] := (a * b - c) - (a * b + d);
    l := matrix [j];
    END;

s('....');

END.
```

Figure 1 Three new Benchmark tests

The real alternative from Dick Smith Electronics!

OLIVETTI/AT&T M24 PC

Yes! The finest alternative yet. Born of the most important partnership in the history of computing.
A fast, flexible, reliable personal computer that is much more than just a superior piece of hardware.
Behind it is a commitment to a better way of managing information and communications. One that can pay off handsomely in increased productivity and profits: Because it is designed to be flexible enough to meet all your business needs - today and in the future.
Whether you use it as a stand-alone work station or in a fully integrated system, you'll find the M24 a high performance machine. And a hard working addition to your office.

Its faster processing and high resolution graphics will help make any computer task a computing pleasure. In addition, you can expect more standard features. More expansion slots. More options for future growth.
As you'd expect, the M24 is also totally compatible with the MS-DOS operating system. It runs the most popular off-the-shelf software (including Symphony, Lotus 1-2-3, Framework and Flight Simulator) with no modification.
Financial analysis. Forecasting. Budgeting. Word processing. Inventory. The Olivetti -AT&T M24PC does it all.
And its flexibility means that when it's time to expand, the M24 will actually make your computer growing pains painless.
Now the Olivetti-AT&T PC is ready for business.

Standard System ONLY $3995

Includes Olivetti M24 base unit with 128K RAM, one 360K disk drive, high res. green screen monitor, IBM-style keyboard, full MS DOS Operating system with GW Basic.

Expanded system $4495

As above with an extra 360K disk drive and a total of 256K RAM.

MS-DOS and Flight Simulator are trade marks of Microsoft Corporation. Lotus 1-2-3 and Symphony are trade marks of Lotus Corporation. Framework is a trade mark of Ashton Taite.

Specifications

Basic Unit:
- ☐ 16-bit CPU (8Mhz)
- ☐ Socket for optional 8087 numerical processor.
- ☐ 16KB ROM for bootstrapping, power-on diagnostics and BIOS.
- ☐ Alphanumeric and graphic monochrome and colour display controller with 4 simultaneous "shades of grey" or colours from a palette of 16.
- ☐ Asynchronous serial port (RS232)
- ☐ Parallel port (Centronics)
- ☐ Sound circuitry
- ☐ Calendar/Clock with battery
- ☐ One expansion slot (7-slot bus optional)

Display:
- ☐ 80 x 25, 40 x 25 (col. x lines) characters.
- ☐ 640 x 400 pixel resolution

Dick Smith Electronics Pty Ltd

COMPUTERSTOP™

Your one stop computer shop at your nearest Dick Smith Electronics centre.

bankcard welcome here · MasterCard · Terms available to approved personal applicants through Custom Credit · VISA

A 927

```
program reference;
var j,k:integer;
procedure refer5 (var i:integer);
begin
  j := 1
end;
procedure refer4 (var i:integer);
begin
  refer5 (i)
end;
procedure refer3 (var i:integer);
begin
  refer4 (i)
end;
procedure refer2 (var i:integer);
begin
  refer3 (i)
end;
procedure refer1 (var i:integer);
begin
  refer2 (i)
end;
begin
  writeln ('s');
  j := 0;
  for k := 1 to 10000 do
    refer1 (j);
  writeln ('e')
end.
```

```
program literalassign;
yar j,k,l:integer;
begin
  writeln ('s');
  for k := 1 to 10000 do
    for j := 1 to 10 do l := 0;
  writeln ('e')
end.
```

```
program equalif;
var j,k,l:integer;
begin
  writeln ('s');
  for k := 1 to 10000 do
    for j := 1 to 10 do
      if j < 6 then l := 1
        else l := 0;
  writeln ('e')
end.
```

```
program unequalif;
var j,k,l:integer;
begin
  writeln ('s');
  for k := 1 to 10000 do
    for j := 1 to 10 do
      if j < 2 then l := 1
        else l := 0;
  writeln ('e')
end.
```

```
program whileloop;
var j,k:integer;
begin
  writeln ('s');
  for k := 1 to 10000 do
  begin
    j := 1;
    while j <= 10 do j := j + 1
  end;
  writeln ('e')
end.
```

```
program repeatloop;
var j,k:integer;
begin
  writeln ('s');
  for k := 1 to 10000 do
  begin
    j := 1;
    repeat
      j := j + 1
    until j > 10;
  end;
  writeln ('e')
end.
```

```
program maths;
var k:integer;
  x,y:real;
begin
  writeln ('s');
  fork: = 1 to 1000 do
  begin
    x := sin (k);
    y := exp (x)
  end;
writeln ('e')
end.
```

```
program vector;
var j,k:integer;
  matrix:array[0..10] of integer;
begin
  writeln ('s');
  matrix[0] := 0;
  for k := 1 to 10000 do
    for j := 1 to 10 do
      matrix[j] := matrix[j-1];
  writeln ('e')
end.
```

```
program realalgebra;
var k:integer;
  x:real;
begin
  writeln ('s');
  for k := 1 to 10000 do
    x := k/k*k+k-k;
  writeln ('e')
end.
```

```
program value;
var j,k:integer;
procedure value5(i:integer);
begin
  j := 1
end;
procedure value4 (i:integer);
begin
  value5 (i)
end;
procedure value3 (i:integer);
begin
  value4 (i)
end;
procedure value2 (i:integer);
begin
  value3 (i)
end;
procedure value1 (i:integer);
begin
  value2 (i)
end;
begin
  writeln ('s');
  j := 0;
  for k := 1 to 10000 do
  value1 (j);
  writeln ('e')
end.
```

```
program noparameters;
var j,k:integer;
procedure none5;
begin
  j := 1
end;
procedure none4;
begin
  none5
end;
procedure none3;
begin
  none4
end;
procedure none2;
begin
  none3
end;
procedure none1;
begin
  none2
end;
begin
  writeln ('s');
  j := 0;
  for k := 1 to 10000 do
    none1;
  writeln ('e')
end.
```

Figure 2 A complete listing of the APC Pascal Benchmarks suite devised by Chris Sadler

```
program memoryaccess;
var j,k,l:integer;
begin
   writeln('s');
   for k := 1 to 10000 do
      for j := 1 to 10 do l := j;
   writeln('e')
end.
```

```
program forloop;
var j,k: integer;
begin
   writeln('s');
   for k := 1 to 10000 do
      for j := 1 to 10 do;
   writeln('e')
end.
```

```
program realarithmetic;
var k:integer;
   x:real;
begin
   writeln('s');
   for k := 1 to 10000 do
      x := k/2★3+4−5;
   writeln('e')
end.
```

```
program magnifier;
var k: integer;
begin
   writeln('s');
   for k := 1 to 10000 do;
   writeln('e')
end.
```

ces onto it. If you are already familiar with Pascal, you might consider it to be a useful starting point for creating a better spreadsheet. The one thing it does immediately demonstrate is the nice screen control you can achieve with TP built-in functions. All the examples worked first time, although the Calc program does contain a number of bugs.

Performance

The real test of a compiler is the quality of programs that can be developed with it. I compared it with the Microsoft Pascal version 3.11 (last year's version) which is fairly widely used and several times more expensive. Turbo Pascal begins with a head start — its run-time library. The library helps with filing, floating-point arithmetic, screen handling, starting and stopping the program, and so on.

TP's library is written in assembler and occupies around 9.5k of space. This 9.5k is apparently an identical copy of the first 9.5k of the TP program itself, which becomes the first part of any .COM generated by TP. For MS Pascal, the library is approximately 26k of code which is taken from a separate disk file. With that extra 16k Microsoft provides more filing facilities, powerful floating point arithmetic compatible with the 8087, and automatic assignment of program parameters.

Three new tests (fig 1) have been added to the existing *APC* Benchmarks (fig 2) and merged into one program for convenience. Two of these tests check important features of Pascal not found in the original Benchmarks. Structured variables are records which may contain a mixture of characters, integers, reals, or other variables. The 'move' test checks the assignment of one such structure to another. Sets are bit maps which can be used to model classic set operations such as union intersection and disjunction. The 'sets' test gives a work-out to three of the more important set operations. The third new test, 'opts', is to find out whether the compiler is clever enough to do any optimisations. This test contains a variety of common opportunities for a clever compiler to reduce the amount of work needed in a program, mainly by recognising expressions and variables which can be used twice or held in registers.

The Benchmarks were run with equivalent run-time checking. The compiler switches were the default (no switches chosen) for Turbo Pascal, which corresponds to the selection of a stack overflow check for MS Pascal. With the exception of the sets test, TP trailed in all sections. In all tests which do not use the run-time library, TP was about 30 to 40 per cent slower. In floating point, the ratio was similar, even though MS Pascal calculates to a higher precision and takes on overheads to ensure compatibility with the 8087 standard. Both compilers provide binary arithmetic, not decimal.

Only in the sets test was TP (very much) the faster. The code size for the test itself (subtracting the run-time library) was 2512 bytes for Turbo Pascal and 1716 for Microsoft Pascal. An inspection of the object code under debug revealed that TP does not perform any optimisation.

Overall, the .COM file produced by TP was 11,838 bytes long and ran the suite of tests in a total of 304.5 seconds, excluding load time. The .EXE file produced by MS Pascal was 35,414 bytes long and ran in a total of 263.7 seconds, excluding load time. Unless you are planning to write programs of 5000 lines or more, the finished Turbo program is likely to be smaller. If your program has a lot of calculations, in Turbo Pascal it may be slower and less accurate. For small programs TP is a clear winner, with its convenient development environment, compact overall programs and useful speed.

Both Benchmarks were run with stack check enabled. TP also provides range checking, plus a selection of switches for the control of I/O error handling and operator interrupts from the keyboard; these worked effectively. When errors are detected, the program prints an error type number and the program counter at which the failure occurred. If you are running from memory under TP control, TP immediately starts searching your source for the equivalent position. If you were running a .COM file, you could invoke TP, select the source as a work file, and ask TP to find the program counter. You must look up the error type number in the manual for an explanation.

Standardisation

There must be something missing, you're thinking, so maybe it isn't real Pascal? Yes, in the sense that it is not quite standard (neither Jenson and Wirth 6192 nor BSO versions) but no, in the sense that this is an implementation of a powerful member of the Pascal family with many extensions. Let's look at the standard features which are missing:

Get and Put Borland has not provided accessible file buffers, so Get, Put, and file pointer types are not available. The explanation given is that you can do the same things more conveniently with Read and Write, which is true: it does wonders for keeping the run-time library small (these can be expensive to implement). This has affected about half the programs which I tried to port to Turbo from other Pascals.

No functional or procedural parameters This is one of the more tricky and rarely used parts of Pascal, and many compilers do not implement it. You can stimulate these with a CASE switch to select which procedure or function to call.

String width on writing in standard Pascal, you can use the record type

```
lstring = RECORD
   len  :  0 . . stringTop;
   s    :  PACKED ARRAY [1 . .
              string Top] OF char;
   END;
```

to do string processing. It is not quite as powerful as the string facilities built into Turbo or other Pascals, but it does have the advantage of portability.

A key part of this technique is the use of the Write statement

```
Write ( sString.s : aString.len);
```

to write a string neatly trimmed to its proper length. Unfortunately this does not work correctly with TP, which always prints the whole string.

No program parameters You should be able to declare file parameters in the program statement. TP allows you to put whatever you like in the program parameters, but ignores it. You are

(02) 212 1622 **POSTWARE** (02) 212 1592

SOFTWARE AND HARDWARE DISCOUNT WAREHOUSE

Super Specials for this month only:

IBM: Harvard Total Project Manager ... $490.00
Apple: Zardax II ... $199.00
Mac: Multiplan ... $240.00
Ramchips 64kb ... $39.00 set

"Our prices speak for themselves."

IBM S/Ware

	R.R.P.	Our Price
Electric Desk		499.00
Knowledgeman	945.00	590.00
Sidekick	99.00	89.00
Friday	425.00	325.00
Dbase II		455.00
Dbase III	976.00	735.00
Microsoft + Spell word	595.00	495.00
Milestone	425.00	329.00
Lotus 1-2-3	835.00	589.00
Multimate 3.3 incl Brit Dict.	695.00	454.00
Open Access	995.00	669.00
Word Perfect	625.00	425.00
Wordstar	415.00	349.00
Wordstar 2000	645.00	469.00
Wordstar Prof	645.00	469.00
Polywindows	125.00	107.00
80-Mate/Term	295.00	257.00
Turbo Pascal + Turbo TB		199.00
Typequick	79.00	72.00
Symphony	1095.00	739.00
Crosstalk	245.00	165.00
Framework	1095.00	812.00
Supercalc 3	496.00	415.00
Dollars + Sense	99.00	79.00
Clout	300.00	229.00
RBase	630.00	519.00
Infostar	595.00	355.00
Quickcode III	405.00	329.00
PFS File Graph Report	195.00	165.00
Sideways	95.00	79.00
Concurrent cp/m	315.00	270.00
Jazz	896.00	675.00
Flight Simulator	99.00	79.00

IBM Hardware

	R.R.P.	Our Price
Paradise 64k/384k 5 pack card	615.00	425.00
Paradise Serial printer port	184.00	159.00
Paradise clock/256k	578.00	478.00
Paradise Modular graphics card	797.00	627.00
Paradise Mouse	250.00	220.00
Hercules Graphics card	445.00	375.00
Qubie 10mb hard D.D.	2250.00	1500.00
Qubie 20mb hard D.D.	2750.00	2250.00
Ast Megapak 256k	919.00	742.00
Qubie Monitors Gr+Amb	295.00	

CP/M Software

	R.R.P.	Our Price
Turbo Toolbox + Pascal		199.00
Nevada Languages	65.00	53.00
Access Manager	595.00	442.00
Supercalc	320.00	255.00
Infostar	595.00	410.00
Display Manager	750.00	588.00
CP/M + (3) unconfigured	488.00	436.00
Concurrent cp/m	465.00	399.00

MACINTOSH SOFTWARE

	R.R.P.	Our Price
Filevision	275.00	170.00
Dollars and Sense	175.00	148.00
Keystroke	859.00	699.00
DB Master	360.00	280.00
Think Tank	210.00	159.00
MacVision	435.00	385.00
Word (Microsoft)	325.00	285.00
Multiplan	325.00	285.00
File	325.00	285.00
Excell	750.00	595.00
Entrepeneur	145.00	85.00
Logo	235.00	199.00
For all available games		Call

Apple

	R.R.P.	Our Price
Printshop	74.95	64.95
Bank St. Speller	105.00	89.00
Bank St. Writer	104.95	89.00
Megaworks	179.00	155.00
Wordstar	395.00	320.00
Insofar	595.00	395.00
Milestone	295.00	240.00
Friday	400.00	298.00
PFS File/Write/Report/Graph	195.00	165.00
PFS School Record	195.00	155.00
Dollars + Sense	164.00	138.00
Zardax II Pro	300.00	245.00
DBase II cp/m	599.00	475.00
Sensible Speller	213.00	185.00
Zork I/II	@ 54.95	49.95
Turbo Pascal + Toolbox	219.00	199.00
Sam Speech Synthesiser	210.00	65.00
Wizardry	79.95	67.95
Ultima II	79.95	67.95
Sorcerer	74.95	64.95
Sargon III		
Auto Ice Printer Interface Card	130.00	85.00
RS232 Serial Card	99.00	79.00
Microsoft 280 Card	769.00	569.00
Superserial Card	249.00	198.00
For rest of available hardware and software		Call

Printers

	R.R.P.	Our Price
Epson LX80	492.00	Call
RX80 F-T+	594.00	Call
RX100	840.00	Call
FX100 +	1108.00	Call
LQ1500	2040.00	1790.00
Star Radix 10 200cps	999.00	799.00
Star Delta 10 160cps	575.00	520.00
Star Delta 15 160cps	850.00	675.00
Silver Reed 500	787.00	559.00
Silver Reed 550	1281.00	894.00
Silver Reed	770.00	1349.00
Silver Reed EB50	430.00	340.00
NEC P3/P2	1450.00	Call
64kb Buffers	399.00	280.00

DISCS

	R.R.P.	Our Price
Nashua Professional SSDD	49.95	37.00
Nashua Professional DSDD	79.95	44.50
Nashua 8" DSDD	92.00	59.60
Memorex SSDD	49.95	37.00
Memorex DSDD	80.00	44.99
3½" Microfloppies SSDD	95.00	65.00
Xidex	49.95	Call
Sony 3.5"		Call

Modems

	R.R.P.	Our Price
Sendata 300AA	410.00	299.00
Sendata 700AC	319.00	165.00
Apple Auto Ice modem	399.00	Call
For the best price on any modem in Australia		Call

Commodore

Call for our best price on the entire range of Commodore software.

Hardware

Apricot Transportable 4th generation microcomputers	FROM $3135.00
President 16PC IMB Compatible 512KB standard, colour graphed, 2 x 360KB drivers etc. etc. Plus heaps of software	$3900.00
Walkabout Portable PC with all above specs and software except with built-in 9" hires monitor runs all IBM software	$3899.00

All products carry full warranty. We support what we sell. All products are latest release versions. Software from Arcom Pacific, Imagineering, SCA, Microsoft, Digital Research, Sourceware, Software Source, MicroPro, Ashton-Tate, Sorcim and others. Prices have risen due to fall in A$ exchange rate. Should they be reduced since the printing of this ad, we'll pass it on. Please do not hesitate to call us to confirm the latest prices, and we will do our best to beat any genuine price quoted by our competitors.

For all products not listed call (02) 212 1622 or 212 1592.

All prices are subject to change without notice.

Call for specials and sales-tax exempted prices: (02) 212 1622
Order by phone, or send cheque or money order to:
POSTWARE, 3rd Floor, 405-411 Sussex Street, Sydney 2000. Tlx: 70208 CAMPEX
Please note: We are still operating from the same address

VISA

expected to control your own file assignments, though there are built-in connections to the console and printer which do not require any effort.

Errors not caught in the WITH and FOR statements WITH was designed as a form of optimisation hint but TP uses it only as a convenient shorthand. Therefore, TP does not complain if you change the location of the WITH structure within the WITH statement, as is gratuitously done in the Calc examples from Borland. The Calc program also shows FOR loops with non-local control variables. In Pascal and some other languages the value of the control variable is undefined at the finish of the loop, so the control variable must be local to limit the potential for mistakes.

Keyword ELSE is used where the standard will use OTHERWISE for the branch on the CASE statement which catches all non-specific cases.

Borland's attitude to standards seems to be 'use the standard except where we don't like it.' I disagree. The usefulness of a Pascal standard lies in the transfer of programs between machines, and as a common ground for teaching or sharing programs. Unfortunately Borland has clipped a few important corners off the standard, just enough to prevent practical use of standard programs. You will need to change programs you find in textbooks, and may have trouble with programs you give to other people.

Beyond the standard

Borland has provided a large and useful set of extensions; the outstanding extensions are in the declaration area. Constant, Type and Variable sections can be in mixed order, and constants can be assigned to be of any type. This has been elegantly done. You should have no trouble creating tables or lists of constants, which is a major problem with standard Pascal.

This is a form of string handling similar to UCSD Pascal. There are operators for searching, comparing, extracting and concatenating strings. String input and output facilities are provided.

The filing facilities include random access to files. The Reset operation opens a file for Write as well as Read (without deleting the file, as ReWrite would) so that files can be edited. There is an important limit to random access, the seek pointer. You may Assign new files and Close old files, so your programs can work with many files and change among them.

Various 'type escapes' are provided. One of the features of Pascal is that every variable has a type classification, and like apples and oranges they are kept distinct. The purpose of this is to help prevent logical errors (the compiler will helpfully prevent you from adding apples into the oranges) and does annoy some people by forcing them to plan ahead. However, it cannot be denied that sometimes, as you plan to make fruit salad, the rules must be relaxed. TP will allow you to specify that apples be considered as oranges, or to pass both as parameters to a procedure expecting fruit.

The screen control facilities include cursor positioning, insert line, delete line, clear screen, and normal/bright intensity control. The screen controls for your programs are the same as those used by TP's editor, and are changed using the Tinst program. These are fast and effective.

Extensions for machine control provide access to I/O ports, and a memory array allows you to access any byte in memory. TP uses 32-bit pointers so you can use all the memory in your machine. There is a mechanism provided for using software interrupts; these allow you to get direct access to the operating system, BIOS, and OS extensions such as graphics. You can embed machine code within your programs, but this is minimally supported and practical only for very short sequences.

For structuring large programs you can include standard procedures or declarations which you need to use in many programs. An overlay mechanism is provided to allow you to create a program with several phases, where the code for only one phase is in RAM at any time and each is pulled from disk as required. All overlaid code çan share a common base of data and subroutines.

Calling outside assembler programs looked even more primitive than the call facility in Basic. In practice I found that I didn't need assembler, especially since direct software interrupts can be done within TP.

Problems

TP is a robust program, but it does have errors and annoyances: for example, the cancellation of type-ahead on the console input file. TP offers either line-edited input (allowing you to erase and re-type before the carriage return) with echo, or direct input without echo. For most casual input the line-edited input would be the obvious choice — but for some strange reason, TP wipes out all type-ahead for this mode. If you work with a computer with a type-ahead buffer this becomes aggravating, as it is second nature to type ahead when using familiar programs.

TP provides only 16-bit signed integers and so the seek pointer in theory stops at 32k records. Some types of file have many records so the 32k limit may be awkward. The solution is to use software interrupts direct to the OS if you need advanced filing.

It should be a simple matter for Borland to close open files at the end of a program, which most Pascals do.

If you use _ (underscore) to separate words within identifiers, the word search feature in the editor (which should only match complete words to the pattern you

Benchmarks

	Turbo Pascal 2.0	*MS-Pascal 3.11*
Magnifier	.2	.2
FOR loop	2.5	1.9
WHILE loop	3.3	2.5
REPEAT loop	2.9	2.1
Literal	3.2	2.7
Memory	3.4	2.9
Real arithmetic	87 (11-digit)	64 (15-digit)
Real algebra	77 (precision)	59 (precision)
Vector access	5.9	3.9
IF, balanced	4.8	3.9
Procedure calls	4.0	3.1
IF, unbalanced	4.7	3.8
Call by value	4.5	3.6
Call by reference	4.8 (32-bit pointer)	3.6 (16-bit pointer)
Transcendentals	80	58
Structure move	1.1	.6
Set operations	12.8	46
Optimisables	2.5	1.8

All results are in seconds

provide) sometimes treats it as a space. Therefore, a word search for 'two' will match at 'two_words', although a word search for 'two_words' will not match 'two words'.

The ideal

Apart from fixing the bugs, what more could you expect from Turbo Pascal? Here are a few things to yearn for.

A fast syntax check for programs too big to compile to memory.

Unattended operation. At the moment, TP is entirely interactive. Suppose you have a standard procedure to ask the date, and you change it to accept US format as well as Australian. If you have 10 programs which include that procedure, you can't create a batch file to recompile them all. You have to do them one by one, interactively.

A debugging switch to catch references to heap records which have been deleted.

DOS-2 filing. TP at the moment uses only DOS 1.25 features. If you wish to use files in a DOS 2.0 or 2.11 system, you must either code your own interfaces to DOS and ignore TP's filing, or ensure that all the files you need are in the current directory.

Learning

TP must be one of the best compilers available for people learning their first compiled language. Results are immediate and TP can protect itself against mistakes. However, there are some cautions. All Pascals are non-standard to some extent, but TP is one of the few I have seen which leaves out parts of the standard important for practical programs. Hence, you will come to depend on features which might not be available on other compilers or machines.

The second problem with using TP is that the manual is not sufficient for beginners. It is a good idea to work with a book designed for teaching rather than reference, and keep in mind that examples in standard Pascal may need changes for TP.

There is also a more expensive version ($149) which makes use of the 8087 chip for arithmetic, which is better and faster than the (incompatible) software arithmetic in ordinary TP.

Conclusion

TP is a complete and practical compiler for small to medium projects. It scores poorly on points such as code optimisation and standardisation, but scores so strongly on convenience and the compact run-time library that you might revise your thinking on what features a compiler should have. The built-in editor is good but slow; the error detection and handing is effective; the extensions are useful. Errors are present in proportion to the price; the value-for-money is outstanding.

It may permanently cure you of Basic, although I do leave you with one final point to ponder — Borland apparently writes its products in assembler!

Turbo Pascal is available from several suppliers in Australia including Archives Computer Services, Mailware and Cash and Carry for around $80–$90.

Plus/4

Introducing Software That Comes With Its Own Built-In Computer

Word processing, electronic spreadsheet, graphic plotting and file management certainly need no introduction.

But a computer that has them all built-in certainly does.

Especially for under $600*.

You could easily say the built-in software alone is worth a lot more than the price.

But then again you could easily say the Commodore PLUS/4 computer alone is worth a lot more than the price. With 64K of memory, 60K fully usable, full size typewriter keyboard, four separate cursor keys, high resolution colour graphics, extended BASIC, split screen and windowing capabilities.

Turn on the power and judge for yourself. Write and edit letters, reports, papers and novels with the built-in wordprocessor. Do the books, the budgets, the projections, profit/loss statements with the built-in spreadsheet. Draw circles, boxes, complex shapes, images and combine them with text with the graphics program. Collect, store, organize and retrieve information such as mailing lists, inventories, personal files, business files and recipes with the file management program.

Go from one program to another at the touch of a key. For not only are they built into the computer, they're built into each other. So, for example, in one document, you can go from words to numbers to graphs to names and addresses. In just a matter of moments.

The Commodore PLUS/4. The first new computer that doesn't have one new thing about it. It has four.

For further information contact your nearest Commodore Dealer. *Recommended retail price

commodore COMPUTER

Keeping up with you.

Commodore Business Machines Pty. Ltd. New South Wales: (02) 427 4888
Victoria/Tasmania/South Australia: (03) 429 9855 Western Australia: (09) 389 1266 Queensland/Northern Territory: (07) 393 0300

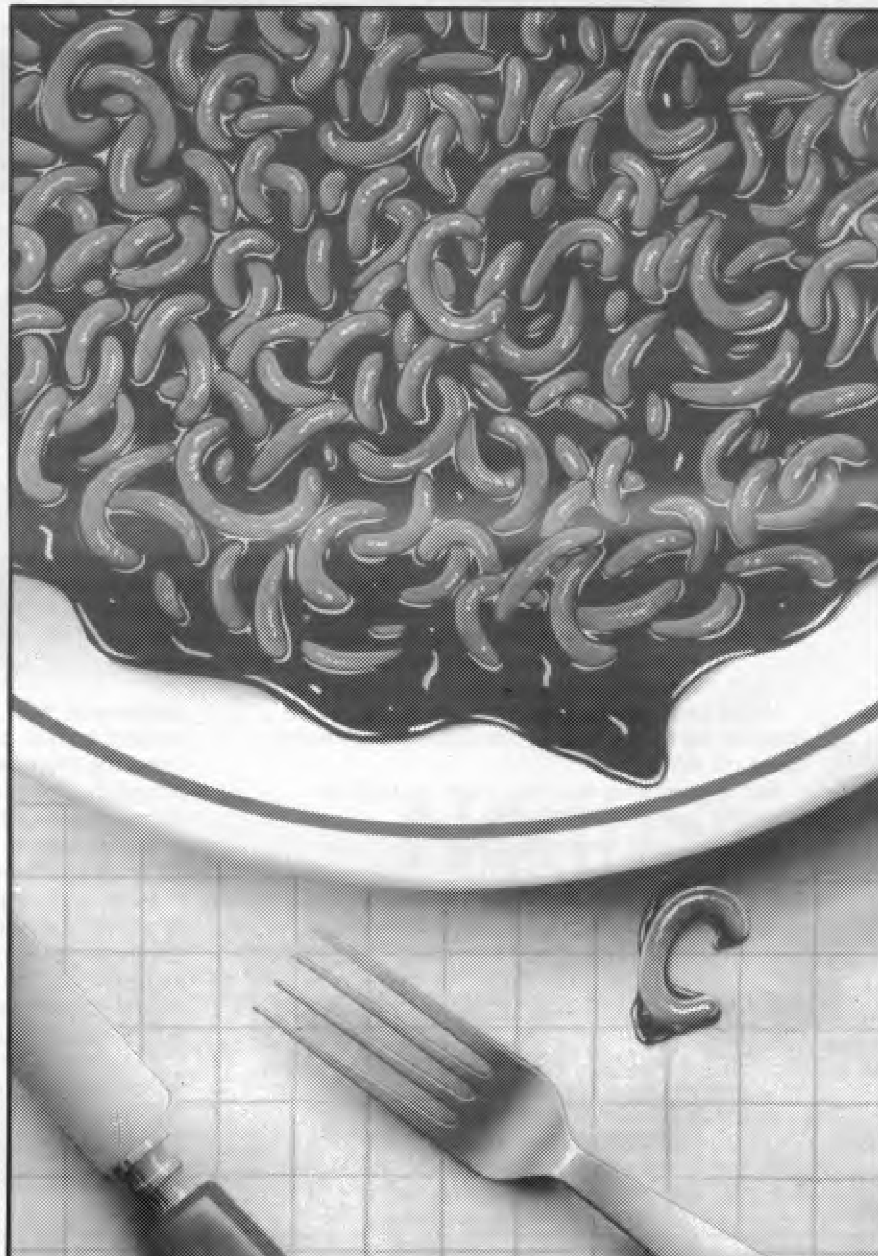
BMS/CC365

The C library

In the last of the series, Les Hampson examines library functions and gives solutions to the more common mistakes that newcomers will make.

The C programs you write will combine your own functions with those in the library supplied with the compiler. Some of the standard functions are so widely used that they tend to be thought of as part of the language, but not even the most frequently-used functions are built in. What is actually provided and how it works depends on the originator. Most versions have functions similar to those provided under the Unix operating system, but there are inevitably some compromises which have to be made to match the operation of CP/M, MS-DOS, and so on. This article considers the essentials of a typical library, and concludes the series by looking at some common errors and sources of information.

The library functions for a particular version of C will be described in the manual. Apart from information on what a function does, this describes the arguments expected, the value returned, data types, and how errors and exceptions are handled. You will probably have functions for console and file input and output, string manipulation, memory management and maths.



Input and output

Access to the display, keyboard and disk files is by means of function calls. These operations are hardware-dependent and require a low-level link to the operating system. Any library should have suitable functions as they are required by everyone and few users will want to write their own. All input and output in C is a 'file' operation with access to devices treated as a special case. This means that you are provided with a file for the display so characters written to it appear on the screen, and another for the keyboard which can be read.

A disk file is simply treated as a sequence of characters. This can be read or written sequentially with a position marker moving along at each access, or the user can set the position giving random access to the information. Two modes of file access using separate sets of functions are provided under Unix, and for most other systems the libraries duly emulate these. The 'low-level' functions provide direct access to the operating system for reading or writing a number of bytes, but give few other services. In many programs you want to read or write only

```
#include "stdio.h"     /*defines type FILE and stdin etc*/
#define ERR -1

long fseek();          /*returns present position as a long*/
FILE *fp;              /*declare a file identifier*/
int i;
char c,array[100];

main()
    {
    fp=fopen("myfile","w");                      /*create myfile*/
    if(fp==0) exit();                            /*could not create*/
    for(i=0;i<500;i++)                           /*write 500 chars*/
            {
            if(fputc(i%127,fp)==ERR)
                    {
                    puts("write error");
                    exit();
                    }
            }
    if(fclose(fp)==ERR) puts("close error");
    else
        {
        fp=fopen("myfile","r");                 /*open for reading*/
        if(fp==0) exit();                       /*could not open*/
        if(fseek(fp,200L,0)==200L)              /*position 200 chars from sta
                fread(array,1,100,fp);          /*put 100 x 1 byte in array*/
        fclose(fp);
        }
```

Fig 1 File access

a few characters, often just one, at a time, but this cannot be efficiently done with this approach. So 'high-level' or 'stream-style' functions provide file access, using a buffer in memory which is automatically replenished or written to disk as required. These also provide comprehensive services, including formatted output and input.

For most purposes you can keep to the high-level functions, which are meant to meet the needs of the user rather than the operating system, but generally it's not a good idea to mix the two methods in a program. The usual stream functions are:

- fopen: open existing file or create new file
- fputc: write a character
- fgetc: read a character
- fseek: change position in a file
- fread: read a number of bytes
- fwrite: write bytes
- fclose: close file
- fprintf: formatted output
- fscanf: formatted input
- fputs: string output
- fgets: string input

Before you can use a disk file it has to be opened using the fopen function. You can specify that read, write or append operations are going to be used. After negotiation with the operating system this returns an identifier for use in subsequent operations. By convention, this is a pointer to a block of data which holds essential information, including the present position and the buffer address. The various input, output and seek functions can then be used, and finally the file is closed so that the buffer is written to disk and everything is tidied away.

Some basic file operations are illustrated in Fig 1. Note that the function fseek, which changes the position for the next access, has a long value passed to it and returns the new position or -1L on error. All the file functions return special values if anything goes wrong and at end-of-file these can be tested and action taken.

For access to the console, three standard file identifiers are provided — stdin, stdout, and stderr. These can be used with the file functions to read or write characters although some operations, like calling fseek, obviously have no meaning. It is usually possible to access other devices like printers and serial ports in a similar way. Communication with the console is so common that a special set of functions is provided for convenience. Getchar gets a character from the keyboard and is equivalent to fgetc(stdin), whereas putchar is the same as fputc to stdout. Similarly, gets and puts will probably be provided for string operations.

Consequently, access to stdin and stdout can be redirected to disk files. For example, you might write a utility to display a file directory and, when required, send the output to disk. However, stderr cannot be redirected, and so is useful for ensuring that error messages appear on the display. Since redirection is not an intrinsic feature of many operating systems, including CP/M, it has to be provided through code in the executable files.

Input and output can be formatted in any way required by using functions which accept a string indicating how the other arguments are to be handled. Formatted output to the display uses the printf function (Fig 2). The first argument is the control string and the others are the variables to display. You can probably work out that the % symbol starts a control sequence which includes a letter indicating what type of argument to expect (c/character, s/string, d/decimal, x/hexadecimal, f/float). A width and precision can also be specified. Any other characters are printed literally, so this function can be used just to display a message.

There are two features of microcomputer operating systems which affect disk access. CP/M keeps account of its files in units of 128 bytes, and pads out the last unit with an end-of-file marker (0×1A, ASCII 26) as required. This means that the reading functions must check for the marker, and care is required in functions which change the position in the file since the end of data is not directly known. Some versions have a different mode for text (any 0×1A is taken to be the end) and binary files. Even using version 2 of MS-DOS, which provides Unix-like file access, the problem has to be faced as many programs first developed for CP/M, such as Microsoft Basic and WordStar, pad out the files they write.

The second problem is that both CP/M and MS-DOS use a linefeed/return pair to initiate a new line, whereas Unix uses only a linefeed (this is the newline character '/n'). Different implementations try to achieve compatibility in different ways but there is not much consistency. In some versions special access modes are used to control the expansion, although an argument can be made for leaving it all to a simple filter routine. The library manual will explain how these problems are tackled.

String manipulation

Although C treats strings simply as arrays of characters, library functions are available to deal with them as units. These will add one string to the end of another, copy and compare strings, check for the occurrence of a character, determine the length, and convert strings to numbers. All the string routines are simple and any variations required can be written, given some basic knowledge of pointers.

The flexibility of C and the library functions allow you to use any type of

WORKING WITH THE OSBORNE EXECUTIVE

YOU GET THE LOT

COMPUTER—SOFTWARE—PRINTER

- ★ *WORD PROCESSING: WORDSTAR 3.3*
  The biggest seller.
- ★ *FINANCIAL ANALYSIS: SuperCalc*
- ★ *ACCOUNTING: MULTILEDGER*
  - *— Debtors*
  - *— General Ledger*
  - *— Creditors*
- ★ *DATABASE: Personal Pearl*
- ★ *PROGRAMMING: M-basic*
  C-basic
  Pascal

COMPUTER

2 Double Density Disk Drives
Large 80 col Amber
Screen-inbuilt.
128K RAM CP/M Plus
IBM Selectric Style Keyboard
Telecommunications Port

PRINTER

Executive DT-100
Wordprocessing Quality

WITH NO HIDDEN EXTRAS

OVER $2000 WORTH OF SOFTWARE FREE

$2995.00

LIMITED OFFER TO JULY 31 OR UNTIL STOCKS SOLD OUT!

THE BEST INVESTMENT YOU'Il EVER MAKE

Full 6 Month Warranty

AVAILABLE AUSTRALIA-WIDE

CALL (02) 290-1122 FOR YOUR NEAREST DEALER

OSBORNE

Eco-C Compiler now for MS-DOS

The fastest C compiler for 8 bit CPUs is now available for MS-DOS (version 2.00 or later).

String Manipulation Functions

atof()	_atoi()	_atol()
atol()	decimal()	_ftoa()
ftoa()	_hex()	hex()
index()	itao()	lmod()
ltoa()	_octal()	octal()
strcat()	strcmp()	strcpy()
strlen()	strncat	strncpy()

Transcendental Functions

acos()	asin()	atan()
atan2()	cos()	cosh()
cotan()	exp()	log()
ln()	power()	sin()
sinh()	sqrt()	tan()
tanh()		

- Perfect for beginner and experienced professional.
- Supports all data types (including longs, floats and doubles) and operators (except bit fields)
- Over 100 library functions, including transcendentals. These conform to UNIX v7 specifications. MS-DOS version has file compatibility with UNIX v7 specs. Programs written with Eco-C can be run under the UNIX C Compiler with few (if any) changes.
- Generates Z80 assembler output (Zilog mnemonics) for use with Microsoft's MACRO 80 assembler (M80) to produce REL files for the linker (L80). Uses true LL(1) grammar with predictive parsing.
- Error messages are in English.
- Floating point is fully 8087 compatible; same floating point answers regardless 8087 being present (as emulator).
- No more third party royalty fees for application software written in 'C'.
- Over 60% Australian content. Fully supported by University graduate and experienced 'C' programmers. We use it for our own software development.
- As users of 'C', we have found this compiler to be quite superior in speed, function and accuracy to a well known Australian 'C' compiler.

Eco-C vs Hitech C

BENCHMARKS (TI PROFESSIONAL NO 8087)
All times in seconds

SEIVE	EXECUTION	COMPILATION
ECO-C	13	68
HITECH	17	42
FLOAT		
ECO-C	5.5	79
HITECH	35	47
FIBONACCI		
ECO-C	43	50
HITECH	72	37

Microstat

New release – version 4.1

Microstat© Rel 4.1 has all the powerful features that have made Microstat the most popular interactive statistics package for the past several years.
It now:

- Interfaces with ASCII files and most external database/spreadsheet programs:
These include Dbase II, Open Access, Multiplan, Lotus.
- Output redirection to text files for editing and report generation.
- Missing data handling
- Improved Recode/Select options
- Handles large data sets. File size is limited only by disk capacity for most procedures.
- Data Management Subsystem (DMS) for file creation plus the ability to edit, list, destroy delete cases, augment, sort, rank order, lag, move, merge and transform data.

Data transformations include: add, subtract, multiply, divide, reciprocal, log, natural log, natural antilog, exponential, linear transformation, adding any number of variables to create a new variable, plus others.

- The DMS places you in complete control of the data files and allows you to create new variables from existing variables via the transformations.

Other features include:

- Descriptive statistics
- Hypothesis tests
- Analysis of Variance
- Scatterplot
- Correlation analysis
- Simple, Multiple and Stepwise regression analysis
- Time series analysis
- Non paramatric analysis
- Crosstabs
- Factorials, permutations, combinations and probability distributions

Prices:
Complete version: $590.00
Upgrade for existing users: $90.00
Demonstration version: $90.00

Microstat 4.1 runs on MS-DOS/PC-DOS, CP/M-86 and CP/M-80. It requires 128K of memory (64K for CP/M- 80).

Please send me more information on:
☐ Eco-C Compiler
☐ Microstat

Name ____________
Address ____________
Postcode ____________
Telephone ____________
My current computer is ____________
Current software ____________

Contact:
Cerebral Solutions
Suite 1, 7th Floor, 8 West Street,
Crows Nest NSW 2065
Phone (02) 923 2288

```
char cdog[]="sailor";
int var=128;
float num=123.6789;
main()
    {
    printf("abcdefgh\n");
    printf("Hello %s\n",cdog);
    printf("%d %x\n",var,var);
    printf("The number=%7.2f",num);
    }

/*
OUTPUT is

abcdefgh
Hello sailor
128 80
The number=123.68
*/
```

Fig 2 Formatted display output

```
int getnum()
    {
    int num=0;
    int c;

    while((c=getchar())!='\n')    /*get input until RETURN pressed*/
        {
        if(c>='0' && c<='9') num=num*10 + (c-'0');
        else break;              /*stop if non-digit*/
        }
    return num;
    }
```

Fig 3 Function for number entry

```
char *ptr;
char *malloc();          /*returns pointer to char*/

ptr=malloc(1024);        /*try and allocate 1024 bytes*/
if(ptr==0) puts("memory not available");
else
    {
    ptr[215]=128;        /*can now use as array*/
    free(ptr);           /*release memory when finished*/
    }
```

Fig 4 Dynamically-sized array

```
#define MAXCOLS 80
#define TRUE 1
#define BEGIN {          *for Pascal and typing enthusiasts*
#define END }
#define beep() putchar(7)
#define max(x,y) (x>y? x:y)
```

Fig 5 Substitution directives

input: for example, a Basic-like routine to display a prompt and input a string could be written. A common need is to enter a number at the keyboard but there is no standard function to do this. One approach is to combine various library functions to do the job: for example, gets to read in a string and then an atoi (ASCII to integer) or asscanf (formatted input from a string) function to do the conversion. Alternatively, a simple function could be written (Fig 3); you could develop this to allow a leading sign, error checking, and so on.

The getchar function is not suitable for many uses because it echoes the character to the screen, pressing <control> C can abort the program, and it often uses a buffer which only makes the characters available after the RETURN key has been pressed. Most libraries have an alternative function which simply returns the value of a keyed character.

Memory management

When a program is loaded into memory, some is used for the code, some for permanent data, and some is reserved for the stack which grows and shrinks for function calls and local data. There may well be some extra accessible memory which can be used if suitable management functions are provided. This is likely to involve a function, malloc, to allocate memory and another, free, to release it. These can be used to provide a variable amount of space for an array (Fig 4).

The library functions are by nature general, and some, printf for example, involve calling in a lot of code. If this matters, write your own function to give just the properties you need. If all you want to do is display strings and integers, it is not difficult to avoid using printf. There is nothing to stop you writing all the functions you need and not use any from the supplied library. The other things that affect the size of your programs are the overhead (how big is a 'do nothing' program) and the 'granularity' of the library. Some versions link your program to only the functions you call, others call in 'modules' containing groups of functions, and some even add the whole standard library. There is a wide difference between implementations, and a basic program might vary from 2-20k for these reasons.

The preprocessor

The C preprocessor is the first stage of converting a written program into machine instructions. It reacts to simple directions in the source file which indicate that substitutions should be made, additional files read in, or part of the source code ignored. This is all done before any compilation and makes programs easier to read and to change, and more flexible. The basic use is substitution, which is requested by directives as shown in Fig 5.

It is much better to use symbols like MAXCOLS instead of numerical constants because they are clearer in the middle of a large program and simplify changes. You can also substitute for C keywords, functions calls, or use arguments in the form of 'macro' definitions. The preprocessor will correctly expand a statement like y=max(z,8), but as it knows nothing whatever about C and just blindly substitutes one thing for another, such macros should be simple. You can see that they can save a function call, but perhaps with some increase in program size.

The preprocessor can be directed to read in another file before compilation,

PRINTERS (ALL BRANDS)

$499 (inc) **OLYMPIA NP165** **$425** (exc)

Epson Compatible, tractor/friction feed, all electronic touch button. Heavy duty 165 cps in draft, 90 cps in NLQ (Near Letter Quality). This one uses inexpensive Epson fabric ribbon.

• BMC CP1300 (15")	**$599**
• BMC CP80 90 cps	**$299**
• Super 5 (P88/2)	**$439**
• Epson RX100 (demo)	**$499**

• **NEW** LX 80 Epson with T/F, 100 cps and NLQ — introductory price **$449** (inc) or with Commodore interface **$499** (inc)

TAXAN PRINTER BUFFER

$369 (inc)

This super buffer remembers it all. Standard with 64 K memory and expandable to 256 K. Multi dual printer function switch, copy, duplicate, store, etc.

NEW COPY HOLDER SET

Features include: **$32.50**
- Copy holder set
- Adjustable arm set, makes copy holder moveable and easily adjusts to the position and angle for you.

COMMODORE 64 MODEM

1200 baud, phone included, comms software, ready for Viatel. **$298** (inc)

IBM

"type cards" inc. tax

• Memory 512 K cards	**$298**
• Multifunction, 384 K, batt-packed real-time clock/calendar, par/serial parts	**$495**
• Comms/serial card	**$98**
• Colour cards	**$298**
• IBM type mouse	**$99**

APPLE

"type cards" inc. tax

• Pal colour cards	**$59**
• Speech	**$49**
• Z-80/CP/M cards	**$59**
• ICE Parallel printer	**$95**
• Super serial async card	**$145**
• Drive card	**$58**
• I.C. tester card	**$98**
• Apple type mouse	**$99**
• RGB card (color)	**$99**

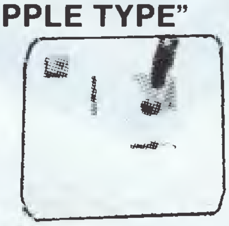
GRAF PAD FOR APPLE

$79 inc

WITH JOYSTICK

Joystick, paddle and pad controller in one. create charts, graphic tables etc.

QUALITY "APPLE TYPE" JOYSTICK



$39 (inc)
- self centring
- our best quality ever

DISK DRIVES

		inc
• Qume	DSDD slimline	**$199**
• Mitsibishi	DSDD slimline	**$219**
• Chinon	SSDD slimline	**$149**
• Commodore	compatible	**$349**
• Atlas	Apple compatible	**$239**

COMPONENTS

		(exc)
4864/4164	RAM	**$2.95**
6116	RAM	**$4.95**
4116	RAM	**$1.45**
2764	EPROM	**$5.95**
27256	EPROM	**$14.95**

Phone for Quantity Prices

DEMO MODEL "BUSINESS" COMPUTERS

	(inc)
• Osborne 1 (Fully optional)	**$1695**
• Osborne 16 bit 256k complete	**$2995**
• IBM 256 K	**$3450**
• Morrow MD3 with terminal (with bundled software)	**$2250**

BBC "NEW PRODUCTS"

Solidisk expand your RAM

• SWR 16K	$90 exc
• SWR 32K	$139 exc
• SWR 128K	$299 exc
• SWR 256K	$599 exc

GRAFPAD IS HERE:

forget lightpens, pads, rodents.

A4 Grafpad	$299 exc
A3	$599 exc

FREE

$19.90 VALUE

delivery on this one anywhere in Australia

PAPER

Box 750 sheets
Quality fine perforation
for word processing etc

DISKS AND DISK STORAGE

NEW Double sided Data Parts **$2995** for 10 or **$248** in 100 lots

Data Parts Diskettes By Verbatim
$24.95 (inc)
Box of ten, or $198 for 100. (Or $168 exc)
- CDC diskettes $19.00 (inc) Box of 10
- DATALIFE

	inc	exc
• SSDD 525.01	**$29.95**	$26.95
• DSDD 550.01	**$36.95**	$31.95

or buy 100 and deduct a further 5%

Freight $6 anywhere in Australia. Prices are subject to change without notice. This ad must be quoted to qualify for special price. Or $10 overnight most places (Comet). If product is not in stock please accept "Data Parts Raincheck".

TRADE ENQUIRIES WELCOME   CASH/CHEQUE, BANKCARD, VISA, MASTER WELCOME HERE

DATA PARTS

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

- ALBURY 518 Macauley Street Telephone: (060) 21 8080.
- BENDIGO 419 Hargreaves Street Telephone: (054) 43 4866.
- SHEPPARTON (as above) Telephone: (058) 21 7155
- WAGGA 33 Thompson Street Telephone: (069) 21 6466
- BALLARAT 307 Mair Street Telephone: (053) 31 3399
- CAMBERWELL 519 Burke Road Telephone: (03) 20 7895

NOW MOHAMMED NEED NOT GO TO THE MOUNTAIN...

Imagine! A computer that's powerful enough to run your business . . . yet small enough to take home with you. That's the Bondwell 14 Portable.

It has all you need to run a business – yet costs a fraction of the price of other computers.

Cat X-9000

That's why it's the success story of 1985!

In just a few short months, the Bondwell 14 portable has become our largest selling business computer. It offers performance and reliability far exceeding its low price tag. Even first glance will tell you that the Bondwell 14 portable is no ordinary computer!

You'll see:

- **128K RAM on board!**
- **Twin 360K disk drives!**
- **220mm amber-screen monitor!**
- **Twin RS-232 serial ports!**
- **Centronics parallel port!**
- **External video monitor port!**
- **Ergonomically designed adjustable keyboard!**
- **PLUS a package of famous Micropro 'Star' series software valued at over $1200:** Wordstar word processing, Calcstar spreadsheet, Datastar information handling and Reportstar report preparation. AND if that's not enough, a host of CP/M utilities including a speech synthesiser! Yes, this is all included FREE with your Bondwell 14 portable.

And just in case your business needs other specialised software, you have one of the biggest 'libraries' in the world at your disposal: the Bondwell 14 portable operates under the world-standard CP/M system. (And it's the latest, most powerful version – Vs 3.0.)

With the huge number of programs written for this system, you'll find the one you require. And the Bondwell 14 portable will read disk files from many other computers – including the IBM PC (& compatibles), Osborne, Kaypro and Spectravideo!

Weighing in at just 12kg and measuring just a little larger than an electric typewriter, the Bondwell 14 is truly portable. Take your Bondwell 14 to where the job is: Mohammed need no longer go to the mountain.

Your business survival depends on your making the correct decisions. Decide on the Bondwell 14 portable. And survive!

Want to know more? Call in to your nearest Dick Smith Electronics Computerstop for a test drive. Ask for a copy of the Bondwell 14 brochure. And then get your own Bondwell 14!

Bondwell Computers: exclusive to Dick Smith Electronics.

ALL THIS FOR ONLY **$2495**

Or $247 deposit and $19.85 weekly over 48 months to approved purchasers.

A964/A966RT585

Dick Smith Electronics Pty Ltd

COMPUTERSTOP™

Your one stop computer shop at your nearest Dick Smith Electronics centre.

bankcard welcome here | MasterCard | VISA | Terms available to approved personal applicants through Custom Credit

perhaps because the same information is needed in each module of a program. One use is for global data declarations, for example:
#include "global.dat"

Most implementations of C require a file called 'studio.h' to be included in every program as it contains essential information, for example the definition of the type FILE, stdin, stdout and stderr. The actual contents vary from version to version to bring their approaches in line with expected behaviour.

Directives are very useful for selecting parts of the code for particular purposes. This is invaluable if you are writing a program for a number of machines and need different sections of code to account for all their quirks. Rather than have several source files which all have to be kept up-to-date, you can just have one with sections. For example:

```
#ifdef IBMPC
/*machine-specific code, data, definitions, and so on, go here*/
#endif
```

and select the ones you want with a single #define directive at the top of the file. A similar use is to remove or add sections of code during development and debugging.

Common mistakes

No-one should expect to learn C without making a lot of mistakes. Syntax errors, like omitting a bracket, will produce an error message from the compiler. If you see a huge list of messages don't panic — a mistake has confused the compiler so that it doesn't know good code from bad. Just correct the first few problems and try again. When they are sorted out there may still be other errors which mean the program won't do what was intended. Some of the most common are:

Misplaced—
for(i=0;i<10;i++); <-- error
arr[i]=i;.

Confusing = with ==
if(x=45) probably isn't what is meant. If you find this confusing use '#define EQ ==' in every file and then write if(x EQ 45).

Missing type of returned value for function.
Unless a function returns an int, the type should be declared before it is used as well as in the function definition.

Providing arguments of the wrong type —
int x=5
y=sqrt(x); sqrt needs a double argument

Providing the wrong number of arguments —
printf("%d %d",23); the second number displayed will be whatever happens to be at the expected place in memory.

Confused operator precedence —
while(c=fgetc(fp)!=EOF) should be while ((C=fgetc(fp))!=EOF)

Confusing a character constant and a string —
'A' is not the same as "A"

Wrong array bounds —
int array[7];
array[7]=45; array[6] is the last element

Not providing space in a character string for the NULL at the end.

Omitting the break statement in switch cases so control unintentionally falls through to the next case.

Declaring a pointer but not making it point at anything before use. Pointer errors can cause mysteriously varying results, depending on whether memory in use is overwritten or not.

When you get rid of all such errors, the program will not do what you want if the logic is wrong. This is the time to use printf statements to display the values in variables at critical points, or to use a debugging aid.

Some newcomers to C complain that they have written a program which is no faster than the same thing in interpreted Basic (this could be doing something like displaying the square root of numbers from 1 to 1000). The complaint is based on a misconception of what is going on. The time taken will be dominated by calculating the root and displaying the number, and even in Basic these will use an efficient machine code routine. The slow, interpreted portion controlling the loop will have an insignificant effect on the total time — using a compiled language like C can only speed up your contribution to the program. Conversely, do not worry that a function call has to be made in C for even simple operations like comparing strings — the same process is used in Basic if you say 'if a$=b$', but this is hidden from the user and will be less efficient. Writers of C compilers make sure that calling a function and returning is very fast.

Conclusion

The standard reference book on C is *The C Programming Language* by Kernighan and Ritchie (Prentice Hall 1978): this is essential for any serious C user. A useful additional reference text which is less terse is *Programming in C* by SG

Kochan (Hayden 1984). Introductory texts include *The C Primer* by Hancock and Krieger (McGraw Hill 1982), and the *C Programming Guide* by Jack Purdum (Que Corp 1983).

The C Puzzle Book by Alan Feuer (Prentice Hall) will appeal to some people because it presents fragments of code and asks what they will do.

Looking at other peoples' source code can be a useful way of learning how to best use (and not to use) the language. Something of a cult has built up around using C on micros and provides a useful supply of programs to study. You can get a lot of public domain (free) software in source form from the US. It includes the code for editors, games and all kinds of utilities. The best starting point is the C Users' Group (PO Box 97, 415 Euclid, McPherson, Kansas 67460) which lists what's available in periodic bulletins.

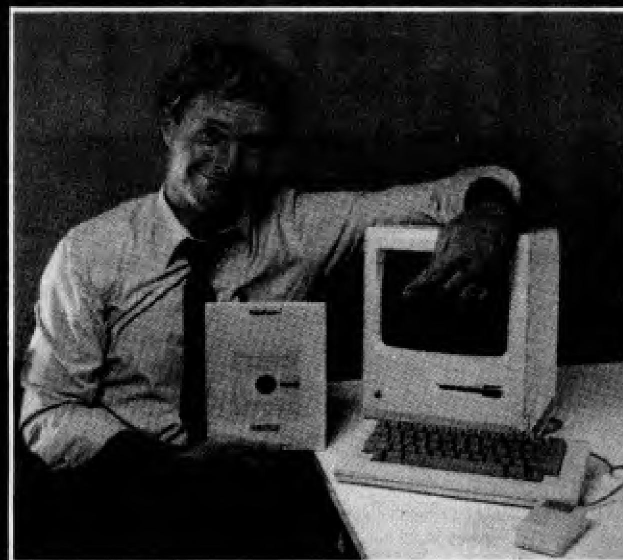
Some companies sell libraries of specialised functions for things like statistics, communications, display windows, graphics and database manipulation which you can use in your programs.

Some C implementations provide the source code for the library and even for the compiler itself. If you really want to see how C makes everything happen then these will be of interest. The best-known compiler in source form is Small C, which gives a useful subset of the language for Z80, 8080 and 8086 processors.

COMPANY	MACHINE/OPERATING SYSTEM	PACKAGE	PRICE
Arcom Pacific, 252 Abbotsford Rd Mayne, QLD 4006	CP/M-80 CP/M-86, PC-DOS	BDS C Digital Research C	238.00 466.00
Ardent Information Services, 19 Waranga St, Dandenong, VIC 3175	CP/M-86, MS-DOS	De Smet C	220.00
Blue Sky Industries 2A Blakesly St Chatswood, NSW 2067	CP/M-80 CP/M-86, MS-DOS, PC-DOS Apple Macintosh (cross compilers to Commodore 64 and others also available)	Aztec CII Aztec C86 Aztec C65 Aztec Mac C	291.18 364.35 245.00 245.00
Fawnray Prance Ltd. 9th Floor, 1 Rosebery Av, Rosebery, NSW 2018	8080/Z80 8086 68000 (PDP-11, VAX, IBM 370 and cross compilers also available)	Whitesmith's C Whitesmith's C Whitesmith's C	769.00 948.00 948.00
Hi-Tech Software P.O. Box 103 Adderley, QLD 4051	CP/M-80 CP/M-86, MS-DOS, PC-DOS	Hi-Tech C Hi-Tech C	250.00 300.00
Software City 1/27 Forge St, BLACKTOWN, NSW 2148	CP/M-80	Software Toolworks C/80	79.95
Software Source P.O. Box 311 BONDI JUNCTION, NSW 2022	CP/M-86, MS-DOS	Computer Innovations C	530.00

Commercially available C packages.

MegaForm is here!



Do you fancy yourself as a designer?

With MegaForm working hand-in-hand with your Macintosh, you'll be able to design any kind of a form that fits your needs...**easily**!

MegaForm is a totally new kind of program. It's like a spreadsheet, graphics package, database, forms generator, and report generator...all rolled up into one!

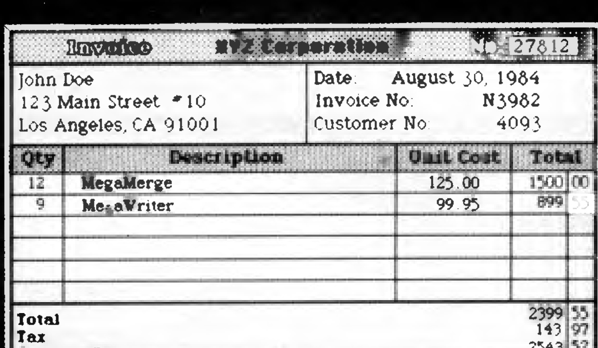
MegaForm lets you design and create any kind of form you like, such as: ● **Invoice Forms.** ● **Order Forms.** ● **Special Report Forms** and more! These forms can also include any kind of graphic image you like, even pictures you've created with MacPaint and MacDraw!

Once you've designed the form, you can tell MegaForm to automatically fill out the form using information stored in your MegaFiler database.

MegaForm is powerful enough to make MegaFiler a relational database!

Plus, MegaForm gives you all of these other advantages:

- Full use of the Macintosh environment, including windows, fonts, icons, desktop accessories.
- Sequentially numbers your forms and stores them for future reference.
- Automatically calculates information like a spreadsheet.



Invoice | XYZ Corporation | 27812

John Doe 123 Main Street #10 Los Angeles, CA 91001	Date: August 30, 1984 Invoice No: N3982 Customer No: 4093

Qty	Description	Unit Cost	Total
12	MegaMerge	125.00	1500 00
9	MegaWriter	99.95	899 55
Total			2399 55
Tax			143 97
Amount Due			2543 52

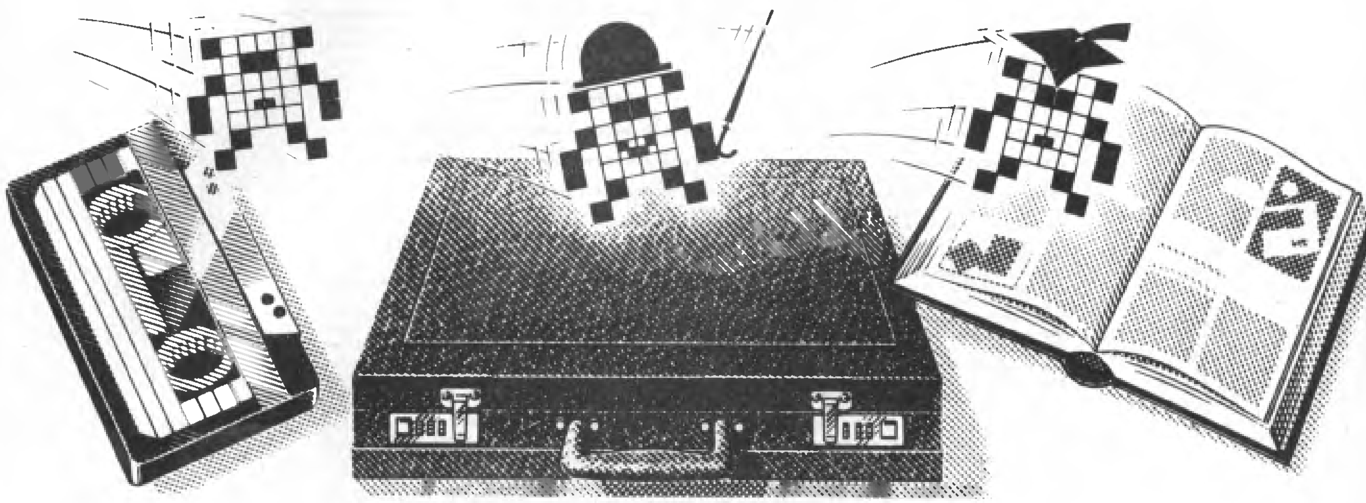
- Prints multiple copies of your forms — each with a different title.
- Gives you several ready-made forms that you can use right away.
- Gives you a selection of business graphic designs to include in your forms.

If you're planning to purchase, or if you already own a Macintosh, you'll be amazed at what it will do with MegaForm!

See your nearest dealer for a complete demonstration.

Software Source Pty Ltd

P.O. Box 311, Bondi Junction N.S.W 2022, Phone: (02) 389 6388



Nick Walker selects the best of readers' programs — for details on submitting your own, see the end of this section.

Two months ago I published a very popular program for the Commodore 64 that enhanced its Basic by adding extra commands to drive sprites, operating the sound channels, and providing numerous other useful commands. This month it's the turn of the 1541 Commodore disk drive to be enhanced. Disk Utility provides a friendly menu-driven interface to this drive instead of the usual jungle of OPENs, CLOSEs and disk strings. In addition, there are extra operations not normally available from 1541 DOS; one in particular, RECOVER, is well worth typing in. It allows you to save files that have been accidentally erased, and I guarantee that if you don't type it in there will be a time in the future when you will wish you had.

For BBC owners there is a short program which initialises disks into a common format that can be read by both 40 and 80-track drives. If you've a 40-track drive and you want to give a program to a friend with an 80-track drive, or *vice versa*, this is the program to do it.

Apologies to Atari owners for not printing the second listing of MultiMode Text in the April issue. I've made up for it this month by not only publishing the second listing, but also a program which generates proper error messages instead of the usual error numbers found on Ataris.

Spectrum owners who are bored with zapping aliens may like to try Triplets, a collection of logic puzzles all with an interesting link.

Tandy Color owners will be pleased to see a listing for their machines in the form of a darts simulator and Amstrad owners will find the disassemble promised last month also in this issue.

- *Games*
- *Scientific/mathematic*
- *Business*
- *Toolkit/utilities*
- *Educational/Computer Aided Learning*

Commodore 64 Disk Utility

by Steve Mehew

I think even the most fervent Commodore fan will agree that the 1541 disk drive is awful, not only because it's very slow but also due to the convoluted commands needed to drive it. I've yet to see software that makes it operate faster, but this program certainly makes it a lot friendlier to use and also adds some extra commands.

Disk Utility for the Commodore 64 allows you to easily perform all the necessary housekeeping on your disks but without the trouble of opening, closing and playing about with disk strings that normally accompany disk work. All the disk operations are easy to use from a menu so I'll say no more about the normal ones such as DELETE, RENAME and CATALOG. However, there are a couple of extra commands.

It's possible to lock a file so that it can't be erased or saved over. This can normally only be done by altering the file entry data on the disk for which there isn't a command. This program will allow it to be done by simply entering the name of the file. It will, of course, also unlock a file.

Probably the most useful facility is the Recover option. This allows a file that has been accidentally erased or saved over to be recovered, provided that no further disk operations have occurred. There is no command for this in the normal 1541 DOS system, but this program will do it easily.

The menu option that provides a disk directory is written in machine code. This section of the program can be lifted out and used in other programs, but don't forget it is copyright material. The lines you'll need are 1890 (excluding the reference to location 2), 1900, and lines 1930 to 2250 inclusive.

```
100 rem 1541 disk utility.
110 rem (c) steve mehew, october 1984.
120 :
130 poke 2,0
140 poke53280,0:poke53281,0:printchr$(147)chr$(14)chr$(8);:poke646,5:clr
150 dimns(35):forr=1to35:readns(r):next
160 data20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,18,18,18,18,18,18,18
170 data17,17,17,17,17,17,16,16,16,16,16
180 forr=0to39:a$=a$+chr$(192):next:rn$=chr$(18):rf$=chr$(146):poke650,128:i=-1
190 forr=0to39:spo$=spo$+chr$(32):next:y=1
200 printchr$(17)rn$a$spo$a$rf$:printchr$(145)chr$(145);chr$(145);
210 t$="Steve's Amazing 1541 Disk Utility"
220 t=20-len(t$)/2:printrn$left$(spo$,t)t$:print
230 ifsrthenreturn
240 gosub 1860
250 printchr$(28)chr$(13)"   Press number of choice:":print:poke646,5
260 data"LOCK    a file.","UNLOCK  a file.","DELETE  a file.","RECOVER a file."
270 data"CATALOG a disk.","RENAME  a file.","FORMAT  a disk."
280 data"Examine disk for errors.","Read disk status.","VALIDATE a disk.","*"
290 readpo$:ifpo$="*"then320
300 i=i+1:printspc(5);:i$=str$(i):j=len(i$):ifi=10thenprintchr$(157);
310 printleft$(i$,j)"]  "po$:goto290
320 print:print:printspc(9)chr$(28)"Choice:   ";:poke198,0:poke646,5
330 ifpeek(2)<>192thengosub1890
340 getpo$:ifpo$=""thenpo=po+1:ifpo<>20then340
350 ifpo=20thenprintchr$(157);chr$(35+3*(y=1));:y=-y:po=0:goto340
360 ifpo$<"0"orpo$>"9"thenpoke53280,1:forr=0to10:next:poke53280,0:goto340
370 on(val(po$)+1)goto390,630,850,930,1220,1260,1400,1510,1760,1820
380 :
390 sr=-1:printchr$(147):gosub200
400 print:print"  Enter Fileneme to be Locked.":print:print:poke19,1:poke198,2
410 poke 631,46:poke 632,157
420 input"       NAME:   ";f$:poke19,0:print
430 iff$="."then 140
440 iflen(f$)=16then460
450 forr=len(f$)+1to16:f$=f$+chr$(160):next
460 dt=18:t$=chr$(dt):s$=chr$(1):open1,8,15:open 2,8,2,"#"
470 print#1,"u1:2,0,"+str$(dt)","+str$(asc(s$))
480 print#1,"b-p:2,0"
490 gosub600:nt$=i$:gosub600:ns$=i$
500 fori=0to7:gosub 600:ft$=i$:gosub600:gosub600
510 na$="":forsm=0to15:gosub600:na$=na$+i$:next:forsm=0to10:gosub600:next
520 ifi<>7thengosub600:gosub600
530 iff$=na$then570
540 next
550 t$=nt$:s$=ns$:if nt$<>chr$(0)then470
560 close1:close2:goto140
570 po=i*32+2:print#1,"b-p:2,"+str$(po):v=asc(ft$)or64:print#2,chr$(v);
580 print#1,"u2:2,0,"+str$(dt)","+str$(asc(s$)):print:print:printf$;" now LOCKED
."
590 close1:close2:forr=1to2000:next:goto140
600 get#2,i$:ifi$=""theni$=chr$(0)
610 return
620 :
630 sr=-1:printchr$(147):gosub200
640 print:print" Enter Fileneme to be Unlocked.":print:print:poke19,1:poke198,2
650 poke 631,46:poke 632,157
660 input"       NAME:   ";f$:poke19,0:print
670 iff$="."then 140
680 iflen(f$)=16then700
690 forr=len(f$)+1to16:f$=f$+chr$(160):next
700 dt=18:t$=chr$(dt):s$=chr$(1):open1,8,15:open 2,8,2,"#"
710 print#1,"u1:2,0,"+str$(dt)","+str$(asc(s$))
720 print#1,"b-p:2,0"
730 gosub600:nt$=i$:gosub600:ns$=i$
740 fori=0to7:gosub 600:ft$=i$:gosub600:gosub600
750 na$="":forms=0to15:gosub600:na$=na$+i$:next:forms=0to10:gosub600:next
760 ifi<>7thengosub600:gosub600
770 iff$=na$then810
780 next
790 t$=nt$:s$=ns$:if nt$<>chr$(0)then710
800 close1:close2:goto140
810 po=i*32+2:print#1,"b-p:2,"+str$(po):v=asc(ft$)and191:print#2,chr$(v);
820 print#1,"u2:2,0,"+str$(dt)","+str$(asc(s$)):print:print:printf$;" now Unlock
ed."
830 close1:close2:forr=1to1000:next:goto140
840 :
850 sr=-1:printchr$(147):gosub200
860 print:print" Enter Fileneme to be Deleted.":print:print:poke19,1:poke198,2
870 poke 631,46:poke 632,157
880 input"       NAME:   ";f$:poke19,0:print
890 iff$="."then 140
900 open1,8,15,"s0:'+f$:close1
910 printf$;" has now been deleted.":forr=1to1000:next:goto140
920 :
930 sr=-1:printchr$(147):gosub200
940 print:print"Enter Fileneme to be Recovered.":print:print:poke19,1:poke198,2
950 poke 631,46:poke 632,157
960 input"       NAME:   ";f$:poke19,0:print
970 iff$="."then 140
980 iflen(f$)=16then1000
990 forr=len(f$)+1to16:f$=f$+chr$(160):next
1000 print:print:print"Enter file type  1] SEQ":printspc(17);"2] PRG"
1010 printspc(17);"3] USR":printspc(17);"4] REL":print:poke19,1:poke 198,2
1020 poke 631,46:poke 632,157
1030 input"    Enter number:  ";yt$:poke19,0:if yt$="."thengoto140
1040 ifyt$<"0"oryt$>"4"thenpoke53280,1:poke53280,0:goto1020
1050 yt=val(yt$)+128
1060 d=0:dt=18:t$=chr$(dt):s$=chr$(1):open1,8,15:open 10,8,10,"#"
1070 print#1,"u1:10,0,"+str$(dt)","+str$(asc(s$))
1080 print#1,"b-p:10,0"
1090 gosub600:nt$=i$:gosub600:ns$=i$
1100 fori=0to7:gosub 600:ft$=i$:gosub600:gosub600
1110 na$="":forna=0to15:gosub600:na$=na$+i$:next:forna=0to10:gosub600:next
1120 ifi<>7thengosub600:gosub600
1130 iff$=na$then1170
1140 next
1150 t$=nt$:s$=ns$:if nt$<>chr$(0)then1070
1160 close1:close10:goto140
1170 po=i*32+2:print#1,"b-p:10,"+str$(po):v=asc(ft$)oryt:print#10,chr$(v);
1180 print#1,"u2:10,0,"+str$(dt)","+str$(asc(s$)):print:print"Now recovered "f$
1190 close1:open1,8,15:print#1,"v":close1:open1,8,15:print#1,"i"
1200 close1:close10:forr=1to1000:next:goto140
1210 :
```

```
1220 sr=-1:printchr$(147):gosub200
1230 print:sys52736:print
1240 goto1800
1250 :
1260 sr=-1:printchr$(147):gosub200
1270 print:print"Enter name of file to be renamed."
1280 poke 631,46:poke 632,157
1290 print:print:poke19,1:poke198,2:input"       NAME:    ";f$:poke19,0:print
1300 iff$="."then140
1310 iflen(f$)>16thenprint:print"This name is too long...":forr=0to4000:next:got
o1260
1320 print:print:print"Enter new name for file.":print:print:poke19,1:poke198,2
1330 poke 631,46:poke 632,157
1340 input"      NEW NAME:    ";n$:poke19,0:print
1350 ifn$="."then140
1360 iflen(n$)>16thenprint:print"New name if too long...":forr=0to4000:next:goto
1260
1370 open1,8,15,"r0:"+n$+"="+f$:sys65511
1380 goto1770
1390 :
1400 sr=-1:printchr$(1471:gosub200
1410 print:print"Enter name for disk.":print:print:poke19,1:poke198,2
1420 poke 631,46:poke 632,157
1430 input"     NAME:   ";n$:poke19,0:print:ifn$="."then140
1440 iflen(n$)>16thenprint:print"Sorry, that is too long...":forr=0to5000:next:g
oto1400
1450 print:print"   Enter disk ID.":print:poke19,1:poke198,2
1460 poke 631,46:poke 632,157
1470 input"     ID:    ";id$:print:poke19,0:ifid$="."then140
1480 iflen(id$)>2thenprint:print"Sorry, that is too long...":forr=0to4000:next:g
oto1400
1490 open1,8,15:print#1,"n0:"+n$+","+id$:sys65511:goto1770
1500 :
1510 sr=-1:printchr$(147):gosub200
1520 print:print"Do you want a fast check or a full       sector by sector check
1530 print:poke19,1:input"    [F]ull or [Q]uick  ";a$:poke19,0:print
1540 ifa$<>"f"anda$<>"q"thenpoke53280,1:poke53280,0:print:goto1530
1550 print:print:print"Check in progress.":print:print
1560 if a$="q"then1680
1570 open10,8,10,"#":open1,8,15
1580 for t=1 to 35:for s=0 to ns(t)
1590 print chr$(145)"t=";t;" s=";s;chr$(157);" ";
1600 print#1,"u1";10;0;t;s
1610 input#1,a:ifa=0then 1630
1620 input#1,b$,c$,d$:printa;",";b$;",";c$;",";d$
1630 print:next:next
1640 print:print"Press "chr$(18)" SPACE "chr$(146)' to continue...";:poke198,0
1650 geta$:ifa$=""then1650
1660 ifa$<>" " then poke53280,1:poke53280,0:goto1650
1670 goto140
1680 open10,8,10,"#":open1,8,15:fort=ito35
1690 printchr$(145)"t=";t;" s=";s;
1700 print#1,"u1";10;0;t;s
1710 input#1,a:ifa=0then 1730
1720 input#1,b$,c$,d$:printa;",";b$;",";c$;",";d$
1730 print:next
1740 goto1640
1750 :
1760 sr=-1:printchr$(147):gosub200
1770 print:print" Disk status is as follows.":print:print
1780 open1,8,15:input#1,a$,b$,c$,d$:close1
1790 printspc(4)a$","b$","c$","d$:print:print
1800 goto1640
1810 :
1820 sr=-1:printchr$(147):gosub200
1830 gosub 1860
1840 open1,8,15,"v":sys65511:print:goto1770
1850 :
1860 open1,8,15,"i0":input#1,a:ifa<>0thenclose1:printchr$(147)"DISK ERROR":stop
1870 sys65511:return
1880 :
1890 poke2,192:forr=52736to52965:reada:poker,a:cs=cs+a:next
1900 ifcs<>30910thenprint"data error.":stop
1910 return
1920 :
1930 data169,1,160,206,162,190,32
1940 data189,255,169,1,162,8,160
1950 data0,32,186,255,32,192,255
1960 data162,1,32,198,255,32,183
1970 data255,32,205,206,32,205,206
1980 data169,13,32,210,255,32,205
1990 data206,32,205,206,32,205,206
2000 data133,251,32,205,206,133,252
2010 data166,251,165,252,32,205,189
2020 data56,32,240,255,160,4,24
2030 data32,240,255,169,1,133,212
2040 data32,205,206,133,252,201,34
2050 data208,247,32,205,206,201,34
2060 data240,6,32,210,255,76,86
2070 data206,32,205,206,201,32,240
2080 data249,133,252,169,0,133,212
2090 data56,32,240,255,160,22,24
2100 data32,240,255,162,0,165,252
2110 data157,167,2,169,0,32,205
2120 data206,201,0,240,7,232,157
2130 data167,2,76,129,206,162,0
2140 data189,167,2,32,210,255,232
2150 data224,5,208,245,169,13,32
2160 data210,255,76,40,206,162,0
2170 data189,191,206,240,6,32,210
2180 data255,232,208,245,169,1,32
2190 data195,255,162,0,32,198,255
2200 data96,36,66,76,79,67,75
2210 data83,32,70,82,69,69,46
2220 data13,0,32,183,255,41,64
2230 data208,4,32,207,255,96,104
2240 data104,76,166,206,255,77,69
2250 data72,69,87,40,67,41,96

ready.
```

Spectrum Triplets

by Nick Phillips

Spectrum triplets for the 48k Sinclair Spectrum is a trio of logic puzzles of the kind that give you the feeling you're very close to solving them even if you're hours away. While struggling to solve them, see if you can see a rather curious relationship between the three puzzles. Unless you can figure it out, this relationship will only be revealed upon solving all three puzzles.

Very little is needed in the way of instructions as these are included within the program. Your aim in each game should be either, if the computer has first go, to draw with it, or, if you have first go, to beat it.

```
   0>REM TRIPLETS                    NICK PHILLIPS 1985
   1 REM Nick Phillips,


  10 INK 0: PAPER 7: BORDER 7: FLASH 0: BRIGHT 0
  20 DIM r(4)
  30 DEF FN p(a$)=INT ((32-LEN a$)/2): REM CENTRE PRINTING
  40 POKE 23658,8
  50 GO SUB 100: REM INTRODUCT'N
  60 GO SUB 200: REM MENU
  70 GO SUB 300: REM PLAY GAME
  80 GO TO 60
  90 REM
 100 REM INTRODUCTION
 110 CLS : LET t$="TRIPLETS": GO SUB 9000
 120 PRINT ''"     HERE IS A TRIO OF SIMPLE    GAMES THAT ARE RELATED TD
EACH  OTHER IN AN AMUSING AND PERHAPS SUPRISING WAY."
 130 PRINT '"    YOUR AIM IN EACH GAME SHOULDBE EITHER, IF THE COMPUTER
HAS  FIRST GO, TO DRAW WITH IT, OR IFYOU HAVE FIRST GO, TO BEAT IT."
 140 PRINT '"    IF YOU MANAGE TO SUCCEED AT LEAST ONCE IN EACH OF THE G
AMES,THEN THE RELATIONSHIP BETWEEN   THEM WILL BE REVEALED!"
 150 PRINT )0;"      NICK PHILLIPS 1985"
 160 GO SUB 9200
 170 RETURN
 180 REM
 200 REM MENU
 210 GO SUB 7000: IF i$="Y" THEN  LET line=4000: RETURN
 220 IF i$="N" THEN  DIM r(4)
 230 CLS : LET t$="MENU": GO SUB 9000
 240 PRINT AT 4,10;"I/ FIFTEEN";AT 6,10;"2/ JAM";AT 8,10;"3/ HOT"
 250 LET p$="CHOOSE A GAME (1-3)": GO SUB 9100
 260 GO SUB 9300: IF CODE i$<49 OR CODE i$>51 THEN  GO TO 260
 270 LET line=1000*VAL i$
 280 RETURN
 290 REM
 300 REM PLAY GAME
 310 REM line=LINE NO OF GAME
 320 CLS : GO SUB line: REM INSTRUCTIONS
 330 GO SUB 560: REM LOAD ARRAYS
 340 GO SUB 9200
 350 LET i win=0: LET you win=0
 360 GO SUB 730: REM DRAW GAME
 370 LET p$="DO YOU WANT FIRST GO? (Y/N)": GO SUB 9100
 380 GO SUB 9400
 390 LET you first=(i$="Y"): LET your move=1-you first
 400 REM START GAME
 410 PRINT )0;AT 0,6;"PRESS ""X"" TO ESCAPE"
 420 DIM g(9): REM HOLDS MOVES
 430 FOR m=1 TO 9: REM MOVE COUNTER
 440 LET your move=1-your move
 450 GO SUB 5000*your move+6000*NOT your move
 460 IF i$="X" THEN  LET m=9: NEXT m: GO TO 510
 470 GO SUB 820: REM PRINT MOVE
 480 IF i win OR you win THEN  LET m=9
 490 NEXT m
 500 GO SUB 890: REM END GAME
 510 PRINT )0;AT 0,2;"DO YOU WANT ANOTHER GO? (Y/N)": GO SUB 9400
 520 IF i$="Y" THEN  GO TO 350
 530 IF line=4000 THEN  DIM r(4)
 540 RETURN
 550 REM
 560 REM LOAD ARRAYS
 570 LET x=1+(line=3000)*3
 580 DIM g(9): DIM a(9): DIM b(9): DIM a$(9,x)
 590 RESTORE 660+30*(line=3000)
 600 FOR n=1 TO 9
 610 IF line=1000 OR line=3000 THEN  READ a(n),b(n): LET a$(n)=STR$ a(n)
 620 IF line=3000 THEN  READ a$(n)
 630 IF line=2000 OR line=4000 THEN  LET a(n)=n: LET b(n)=n: LET a$(n)=S
TR$ n
 640 NEXT n: RETURN
 650 REM
 660 DATA 2,8,9,1,4,6
 670 DATA 7,3,5,5,3,7
 680 DATA 6,4,1,9,8,2
 690 DATA 8,4,"HOT",2,2,"FORM",7,7,"WOES"
 700 DATA 1,9,"TANK",9,6,"HEAR",5,8,"WASP"
 710 DATA 3,3,"TIED",6,1,"BRIM",4,5,"SHIP"
 720 REM
 730 REM DRAW GAME
 740 CLS : GO SUB 9000
 750 GO SUB line+100
 760 FOR n=1 TO 9: LET move=n
 770 GO SUB line+300
```

```
780 NEXT n
790 INK 0: PAPER 7: BRIGHT 0
800 RETURN
810 REM
820 REM PRINT MOVES
830 FLASH 1: BRIGHT 1: GO SUB line+300
840 FOR n=1 TO 200: NEXT n
850 FLASH 0: BRIGHT 0: INK 7: GO SUB line+300
860 INK 0: GO TO line+400
870 RETURN
880 REM
890 REM END OF GAME
900 LET p$="WE'VE DRAWN!!!"
910 IF NOT i win AND NOT you first THEN  LET r(line/1000)=1
920 IF i win THEN  LET p$="I'VE WON!!!"
930 IF you win THEN  LET p$="WELL DONE-YOU'VE WON!!": LET r(line/1000)=
1
940 IF you win OR i win THEN  GO SUB 8300: REM PRINT WIN
950 GO SUB 9100
960 RETURN
970 REM
1000 REM FIFTEEN
1010 LET t$="FIFTEEN": GO SUB 9000
1020 PRINT ''"    NINE PLAYING CARDS, WITH    VALUES FROM ACE TO NINE, A
RE    DISPLAYED."
1030 PRINT '"    YOU TAKE IT IN TURNS WITH   THE COMPUTER TO PICK A CARD
."
1040 PRINT '"    THE WINNER IS THE FIRST TO  OBTAIN THREE CARDS THAT ADD
 UP  TO FIFTEEN."
1050 RETURN
1060 REM
1100 REM DRAW FIFTEEN
1110 PAPER 1: INK 7: BRIGHT 1
1130 FOR n=4 TO 7: PRINT AT n,2;"                            ": NEXT n
1140 FOR n=0 TO 192 STEP 24
1150 PLOT n+23,119: DRAW 0,17: PLOT n+23,119: DRAW 9,0
1160 PLOT n+23,136: DRAW 9,0: PLOT n+32,119: DRAW 0,17
1170 PLOT n+21,117: DRAW 0,21: PLOT n+21,117: DRAW 13,0
1180 PLOT n+21,138: DRAW 13,0: PLOT n+34,117: DRAW 0,21
1190 NEXT n
1200 LET r=9: GO SUB 7900
1220 RETURN
1300 REM PRINT MOVES
1310 PRINT AT 5,3*a(move);a$(move);AT 6,3*a(move);" "
1320 RETURN
1400 REM PRINT MOVE
1410 PRINT AT 10+INT ((m-1)/2),8+14*(your move);a$(move)
1420 RETURN
2000 REM JAM
2010 LET t$="JAM": GO SUB 9000
2020 PRINT ''"    YOU ARE SHOWN A ROAD MAP IN REPRESENTATIONAL FORM, ON
WHICH ARE SHOWN NINE NUMBERED ROADS."
2030 PRINT '"    YOU TAKE TURNS WITH THE     COMPUTER TO CHOOSE ONE OF T
HESE NUMBERED ROADS."
2040 PRINT '"    THE WINNER IS THE FIRST     PLAYER TO HAVE THREE ROADS
WHICHENTER THE SAME TOWN."
2050 RETURN
2100 REM DRAW JAM
2110 DIM j(9,6): REM HOLDS DATA TO DRAW JAM
2120 RESTORE 2210
2130 FOR n=1 TO 9: FOR m=1 TO 6
2140 READ j(n,m): NEXT m
2150 INK 6-(n>3)-(n>5)
2160 PLOT j(n,1),j(n,2): DRAW j(n,3),j(n,4)
2170 NEXT n: INK 0
2180 FOR n=1 TO 8: READ r,c: PRINT AT r,c; INK 3;" ": NEXT n
2190 PRINT AT 3,25; INK 1;"YOU__";AT 5,25; INK 2;"ME__"
2200 RETURN
2210 REM JAM DATA
2215 DATA 44,140.64,-87.8.9
2220 DATA 108,53,32,31,12,15
2225 DATA 44,28,144,56,14,18
2230 DATA 20,84,24,56,6,3
2235 DATA 20,84,168,0,10,6
2240 DATA 20,84,24,-56,14,2
2245 DATA 44,140,144,-56,5,10
2250 DATA 108,115,32,-31,9,14
2255 DATA 44,28,64,87,14,9
2260 DATA 11,2,11,23,4,5,18,5,15,13,7,13,11,17,11,10
2270 REM
2300 REM PRINT MOVES
2310 PRINT AT j(move,5),j(move,6);a$(move)
2320 RETURN
2330 REM
2400 REM PRINT MOVE
2410 PLOT j(move,1),j(move,2): DRAW  INK 2-your move;j(move,3),j(move,4)
2420 RETURN
3000 REM HOT
3010 LET t$="HOT": GO SUB 9000
3020 PRINT ''"    YOU ARE PRESENTED WITH A    LIST OF NINE WORDS."
3030 PRINT '"    YOU TAKE IT IN TURNS WITH   THE COMPUTER TO PICK A WORD
."
3040 PRINT '"    THE FIRST PLAYER TO HAVE    THREE WORDS WHICH BEAR THE
SAME LETTER IS THE WINNER."
3050 RETURN
3100 REM DRAW HOT
3110 PAPER 1: INK 7
3130 FOR n=3 TO 10: PRINT AT n,0;"                               ": NEX
T n
3140 FOR n=0 TO 2: PLOT 8,96+n*24: DRAW 239,0: NEXT n
3150 FOR n=0 TO 5: PLOT 8+48*n,96: DRAW 0,48: NEXT n
3170 LET r=12: GO SUB 7900
3180 FOR n=1 TO 9: PRINT AT 4+INT (n/6)*3,-3+6*(n-5*INT (n/6));n: NEXT n
3190 INK 0: PAPER 5
3200 RETURN
```

```
3300 REM PRINT MOVES
3310 PRINT AT 5+INT (a(move)/6)*3,-4+6*(a(move)-5*INT (a(move)/6));a$(mo
ve)
3320 RETURN
3410 PRINT AT 13+INT ((m-1)/2),6+13*(your move);a$(move)
3420 RETURN
4000 REM
4010 GO SUB 7500
4020 READ y,c: LET y$=CHR$ y: LET c$=CHR$ c
4030 RETURN
4100 REM
4110 INK 1: PLOT 104,64: DRAW 0,72: PLOT 128,64: DRAW 0,72
4120 PLOT 80,88: DRAW 72,0: PLOT 80,112: DRAW 72,0
4140 PRINT AT 12,0; INK 2;"ME...";c$;AT 14,0;"YOU..";y$
4150 DEF FN r(n)=6+3*INT ((n-1)/3): DEF FN c(n)=8+3*(n-3*INT ((n-1)/3))
4160 INK 4: RETURN
4300 REM
4320 IF you win OR i win THEN  GO TO 4400
4330 PRINT AT FN r(move),FN c(move);a$(move)
4340 RETURN
4400 REM
4410 LET m$=c$: IF your move THEN  LET m$=y$
4430 PRINT AT FN r(move),FN c(move);m$
4440 RETURN
4450 REM
5000 REM YOUR MOVE
5010 LET p$="ENTER YOUR MOVE (1-9)": GO SUB 9100
5020 GO SUB 9300: IF CODE i$<49 OR CODE i$>57 AND i$<>"X" THEN  GO TO 50
20
5030 IF i$="X" THEN  RETURN
5040 LET move=b(VAL i$)
5050 IF g(move)<>0 THEN  BEEP 1,-20: GO TO 5020
5060 LET g(move)=2
5070 REM CHECK FOR YOU WIN
5080 FOR l=1 TO 8: GO SUB 8000
5090 IF g(11)+g(12)+g(13)=6 THEN  LET you win=1: LET win=1: LET l=8
5100 NEXT l: RETURN
5110 REM
6000 REM MY MOVE
6010 DIM s(9): REM SCORE FOR EACH MOVE
6020 LET p$="MY MOVE": GO SUB 9100
6030 IF m=1 THEN  LET r=INT (RND*4): LET move=1*(r=0)+3*(r=1)+7*(r=2)+9*
(r=3): LET g(move)=1: RETURN : REM RANDOM FIRST MOVE
6040 IF m=2 AND g(5)=0 THEN  LET move=5: LET g(move)=1: RETURN
6050 IF m=3 AND g(5)=0 AND move=2*INT (move/2) THEN  LET move=5: LET g(m
ove)=1: RETURN
6060 FOR l=1 TO 8
6070 LET i$=INKEY$: IF i$="X" THEN  LET l=8: NEXT l: RETURN
6080 GO SUB 8000: REM POSSIBLE WINNING COMBINATIONS
6090 LET me=(g(11)=1)+(g(12)=1)+(g(13)=1)
6100 LET you=(g(11)=2)+(g(12)=2)+(g(13)=2)
6110 GO SUB 8100: REM ASSESS MOVES
6120 NEXT l
6130 REM SCORE POSSIBLE MOVES
6140 LET score=0
6150 FOR n=1 TO 9
6160 IF s(n)>score THEN  LET score=s(n): LET move=n
6170 NEXT n
6180 LET g(move)=1
6190 RETURN
7000 REM
7005 LET end=0
7010 FOR n=1 TO 3: LET end=end+r(n): NEXT n
7020 IF end<3 THEN  LET i$="": RETURN
7030 CLS : PRINT AT 7,8; PAPER 2; INK 7;" WELL DONE!!!";AT 10,3; INVERSE
 1;"YOU HAVE NOW SUCCEDED IN        EACH OF THE THREE GAMES!";AT 16,0;"A
RE YOU READY TO LEARN THE TRUE  NATURE OF THESE GAMES? (Y/N)"
7040 GO SUB 9400
7050 RETURN
7500 REM
7510 RESTORE 8380
7520 LET t$=""
7530 FOR n=1 TO 19: READ t: LET t$=t$+CHR$ t: NEXT n
7600 REM
7610 RESTORE 7700
7620 FOR n=4 TO 12: READ length
7630 PRINT AT n,INT ((32-length)/2);
7640 FOR j=1 TO length: GO SUB 7800: NEXT j
7650 NEXT n
7660 GO SUB 9000: PRINT AT 6,6;t$;"!";AT 10,1;t$;" GRID, THEN"
7670 RETURN
7690 REM
7700 DATA 22,65,76,76,32,79,70,32,84,72,69,83,69,32,71,65,77,69,83,32,65
,82,69
7710 DATA 15,73,83,79,77,79,82,80,72,73,67,32,87,73,84,72
7720 DATA 1,32,1,32
7730 DATA 31,73,70,32,89,79,85,32,67,65,78,32,80,73,67,84,85,82,69,32,84
,72,69,32,69,76,69,77,69,78,84,83
7740 DATA 27,79,70,32,69,65,67,72,32,71,65,77,69,32,68,73,83,80,76,65,89
,69,68,32,79,78,32,65
7750 DATA 1,32
7760 DATA 29,89,79,85,32,83,72,79,85,76,68,32,72,65,86,69,32,78,79,32,68
,73,70,70,73,67,85,76,84,89
7770 DATA 24,87,73,84,72,32,65,78,89,32,79,70,32,84,72,69,32,71,65,77,69
,83,33,33,33,79,88
7800 REM
7810 READ q: LET q$=CHR$ q: PRINT q$;
7820 RETURN
7830 REM
7900 REM PRINT ME:- & YOU:-
7910 PRINT  BRIGHT 0; INK 2; PAPER 7;AT r,5;"ME:-";AT r,18;"YOU:-"
7920 RETURN
7930 REM
```

```
8000 REM POSSIBLE WINNING COMBINATIONS
8010 RESTORE 8040+1
8020 READ 11,12,13
8030 RETURN
8040 REM
8041 DATA 1,2,3
8042 DATA 4,5,6
8043 DATA 7,8,9
8044 DATA 1,4,7
8045 DATA 2,5,8
8046 DATA 3,6,9
8047 DATA 1,5,9
8048 DATA 7,5,3
8050 REM
8100 REM ASSESS MOVES
8110 IF me+you=3 THEN  RETURN
8120 GO SUB 8130: GO TO 8200
8130 IF me=2 THEN  LET score=1000: LET i win=1: LET win=1: RETURN
8140 IF you=2 THEN  LET score=500: RETURN
8150 IF m=3 AND g(5)<>0 AND you AND me THEN  LET score=500: RETURN
8160 IF you AND me OR NOT you+me THEN  LET score=10: RETURN
8170 IF me THEN  LET score=50: RETURN
8180 IF you THEN  LET score=25: RETURN
8190 REM
8200 REM SCORE MOVES
8210 IF g(11)=0 THEN  LET s(11)=s(11)+score
8220 IF g(12)=0 THEN  LET s(12)=s(12)+score/1.3
8230 IF g(13)=0 THEN  LET s(13)=s(13)+score
8240 RETURN
8250 REM
8300 REM PRINT WIN
8310 LET l=win: GO SUB 8000
8320 BRIGHT 1: FLASH 1: PAPER 6
8330 LET move=11: GO SUB line+300
8340 LET move=12: GO SUB line+300
8350 LET move=13: GO SUB line+300
8360 BRIGHT 0: FLASH 0: PAPER 7
8370 RETURN
8380 DATA 78,79,85,71,72,84,83,32,65,78,68,32,67,82,79,83,83,69,83
8390 REM
9000 REM PRINT TITLE
9010 LET u$=""
9020 FOR n=1 TO LEN t$: LET u$=u$+"-": NEXT n
9030 PRINT  INK 1;AT 0,FN p(t$);t$;AT 1,FN p(t$);u$
9040 RETURN
9100 REM PRINT PROMPT
9110 PRINT  BRIGHT 1; INK 2;AT 20,0;TAB FN p(p$);p$;
9120 FOR n=FN p(p$)+LEN p$ TO 31: PRINT  BRIGHT 1;" ";: NEXT n
9130 RETURN
9200 REM PRESS ANY KEY
9210 LET p$="PRESS ANY KEY...": GO SUB 9100
9220 GO SUB 9300
9230 RETURN
9300 REM READ INKEYS
9310 IF INKEY$="" THEN  GO TO 9310
9320 LET i$=INKEY$
9330 RETURN
9400 REM ANSWER Y/N
9410 GO SUB 9300
9420 IF i$<>"Y" AND i$<>"N" THEN  GO TO 9410
9430 RETURN
```

Atari Error Message Reporter

by Paul Lay

Although Atari 8-bit computers can differentiate between a large range of Basic errors, these are unfortunately all of the form 'Line 1350 Error 143' which send you scurrying to a manual to find the corresponding text. This program for all Atari machines (including XLs) will generate a full textual error message each time an error occurs.

I've included the assembly listing purely for the interest of readers with a knowledge of machine code. You need only type the first listing to use the program.

```
10 GRAPHICS 2
20 POKE 752,1
30 SETCOLOR 3,4,8
40 ? #6:? #6:? #6;"  ERROR"
50 ? #6;"     message"
60 ? #6;"          reporter"
70 ? #6:? #6
80 ? #6;"  (C)1985 paul lay"
90 COLOR ASC("*")
100 PLOT 0,0:DRAWTO 19,0
110 DRAWTO 19,9:DRAWTO 0,9
120 DRAWTO 0,0
```

```
130 ? :? "          INITIALISING "
140 K=0:FOR I=1552 TO 1668
150 READ J:POKE I,J
160 K=K+J:NEXT I
170 IF K=16277 THEN 230
180 ? :? "      DATA VALUES INCORRECT"
190 ? "       IN LINES 720->880"
200 FOR I=1 TO 800:NEXT I
210 GRAPHICS 0:LIST 720,880
220 END
230 ? :? "       READING ERROR MESSAGES ":?
240 IF PEEK(9)<>0 THEN 290
250 POKE 9,2
260 POKE 2,16:POKE 3,6
270 POKE 12,16:POKE 13,6
280 GOTO 330
290 POKE 9,1
300 POKE 1553,PEEK(12)
310 POKE 1554,PEEK(13)
320 POKE 12,16:POKE 13,6
330 FOR I=0 TO 15
340 POKE 1536+I,PEEK(58368+I)
350 NEXT I
360 POKE 1542,34:POKE 1543,6
370 POKE 838,34:POKE 839,6
380 V=1+PEEK(58374)+256*PEEK(58375)
390 HI=INT(V/256):LO=V-256*HI
400 POKE 1667,LO:POKE 1668,HI
410 DIM MESS$(100)
420 MESSLO=256*(PEEK(106)-32)-2048
430 HI=INT(MESSLO/256)
440 LO=MESSLO-256*HI
450 POKE 1617,LO:POKE 1618,HI
460 MESSHI=MESSLO+66
470 HI=INT(MESSHI/256)
480 LO=MESSHI-256*HI
490 POKE 1623,LO:POKE 1624,HI
500 LENGTH=MESSHI+66
510 HI=INT(LENGTH/256)
520 LO=LENGTH-256*HI
530 POKE 1629,LO:POKE 1630,HI
540 CURRENT=LENGTH+66
550 FOR ERR=2 TO 65
560 READ MESS$
570 IF MESS$="+" THEN 690
580 ? MESS$
590 HI=INT(CURRENT/256)
600 LO=CURRENT-256*HI
610 POKE MESSLO+ERR,LO
620 POKE MESSHI+ERR,HI
630 SIZE=LEN(MESS$)
640 POKE LENGTH+ERR,SIZE
650 FOR I=1 TO SIZE
660 POKE CURRENT,ASC(MESS$(I))
670 CURRENT=CURRENT+1
680 NEXT I
690 NEXT ERR
700 GRAPHICS 0
710 NEW
720 DATA 32,34,6,169,6,141,34,3
730 DATA 169,34,141,70,3,169,6
740 DATA 141,71,3,96,133,203,165
750 DATA 185,240,14,165,203,201,69
760 DATA 240,13,201,155,208,4,169
770 DATA 0,133,205,165,203,76,130
780 DATA 6,165,205,208,247,230,205
790 DATA 134,206,132,207,165,185,201
800 DATA 128,48,3,56,233,106,170
810 DATA 189,255,255,141,103,6,189
820 DATA 255,255,141,104,6,189,255
```

```
830 DATA 255,141,113,6,162,0,134
840 DATA 204,189,255,255,32,130,6
850 DATA 230,204,166,204,224,255,208
860 DATA 242,169,155,32,130,6,169
870 DATA 69,166,206,164,207,76,130
880 DATA 6,76,255,255
890 DATA Memory Insufficient
900 DATA Value Error
910 DATA Too Many Variables
920 DATA String Length Error
930 DATA Out of Data Error
940 DATA Number greater than 32767
950 DATA Input Statement Error
960 DATA Array or String DIM Error
970 DATA Argument Stack Overflow
980 DATA Floating Point Overflow/Underflow     Error
990 DATA Line Not Found
1000 DATA No Matching FOR Statement
1010 DATA Line Too Long Error
1020 DATA GOSUB or FOR Line Deleted
1030 DATA RETURN Error
1040 DATA Garbage Error
1050 DATA Invalid String Character
1060 DATA LOAD program Too Long
1070 DATA Device Number Larger
1080 DATA LOAD File Error
1090 DATA BREAK Abort
1100 DATA IOCB
1110 DATA Nonexistent Device
1120 DATA IOCB Write Only
1130 DATA Invalid Command
1140 DATA Device or File not Open
1150 DATA BAD IOCB Number
1160 DATA IOCB Read Only Error
1170 DATA EOF
1180 DATA Truncated Record
1190 DATA Device Timeout
1200 DATA Device NAK
1210 DATA Serial Bus
1220 DATA Cursor Out of Range
1230 DATA Serial Bus Data Frame Overrun
1240 DATA Serial bus data frame checksum error
1250 DATA Device done error
1260 DATA Read after write compare error
1270 DATA Function not implemented
1280 DATA Insufficient RAM
1290 DATA +,+,+,+,+,+,+,+,+,+,+,+
1300 DATA Drive number error
1310 DATA Too many OPEN files
1320 DATA Disk full
1330 DATA Unrecoverable system data I/O error
1340 DATA File number mismatch
1350 DATA File name error
1360 DATA POINT data length error
1370 DATA File locked
1380 DATA Command invalid
1390 DATA Directory full
1400 DATA File not found
1410 DATA POINT invalid


0000        1000        *=   $0600
            1010 ;
            1020 ; PUT THE NEW 'E' DRIVER TABLE
            1030 ; AT PAGE 6
            1040 ;
            1050 TABLE
            1060 ;
0600        1070        *=   *+$10
            1080 ;
```

```
                1090 ; WARM START CODE
                1100 ; NOTE THAT 'EXIT' CHANGES FOR
                1110 ; DISK BASED SYSTEMS
                1120 ;
                1130 WARMST
                1140 ;
0610 202206 1150         JSR  EXIT
0613 A906   1160         LDA  #TABLE/$0100
0615 8D2203 1170         STA  $0322
0618 A922   1180         LDA  #NEWPUT&$00FF
061A 8D4603 1190         STA  $0346      ; IOCB#0 PUT CHARACTER
POINTER (LOW)
061D A906   1200         LDA  #NEWPUT/$0100
061F 8D4703 1210         STA  $0347      ; IOCB#0 PUT CHARACTER
POINTER (HIGH)
0622 60     1220 EXIT    RTS
                1230 ;
                1240 ; THE AMMENDED 'E' PUT ROUTINE
                1250 ;
                1260 MYPUT
                1270 ;
00CB        1280 CHAR    =    $CB
00CC        1290 COUNT   =    $CC
00CD        1300 FLAG    =    $CD
00CE        1310 TEMP    =    $CE
00CF        1320 TEMP2   =    $CF
00B9        1330 ERROR   =    $B9
                1340 ;
0623 85CB   1350         STA  CHAR
0625 A5B9   1360         LDA  ERROR
0627 F00E   1370         BEQ  EXIT2
0629 A5CB   1380         LDA  CHAR
062B C945   1390         CMP  #'E
062D F00D   1400         BEQ  MESSAGE
062F C99B   1410         CMP  #$9B
0631 D004   1420         BNE  EXIT2
0633 A900   1430         LDA  #$00
0635 85CD   1440         STA  FLAG
0637 A5CB   1450 EXIT2   LDA  CHAR
0639 4C8206 1460         JMP  OLDROUTINE
                1470 ;
                1480 MESSAGE
                1490 ;
063C A5CD   1500         LDA  FLAG
063E D0F7   1510         BNE  EXIT2
0640 E6CD   1520         INC  FLAG
0642 86CE   1530         STX  TEMP
0644 84CF   1540         STY  TEMP2
0646 A5B9   1550         LDA  ERROR
0648 C980   1560         CMP  #$80
064A 3003   1570         BMI  SKIP
064C 38     1580         SEC
064D E96A   1590         SBC  #$6A
064F AA     1600 SKIP    TAX
0650 BDFFFF 1610         LDA  MESSLO,X
0653 8D6706 1620         STA  LOOP+1
0656 BDFFFF 1630         LDA  MESSHI,X
0659 8D6806 1640         STA  LOOP+2
065C BDFFFF 1650         LDA  LENGTH,X
065F 8D7106 1660         STA  ATEND+1
0662 A200   1670         LDX  #$00
0664 86CC   1680         STX  COUNT
0666 BDFFFF 1690 LOOP    LDA  $FFFF,X
0669 208206 1700         JSR  OLDROUTINE
066C E6CC   1710         INC  COUNT
066E A6CC   1720         LDX  COUNT
0670 E0FF   1730 ATEND   CPX  #$FF
0672 D0F2   1740         BNE  LOOP
0674 A99B   1750         LDA  #$9B
0676 208206 1760         JSR  OLDROUTINE
```

```
0679 A945   1770        LDA  #'E
067B A6CE   1780        LDX  TEMP
067D A4CF   1790        LDY  TEMP2
067F 4C8206 1800        JMP  OLDROUTINE
            1810 ;
0622        1820 NEWPUT =    MYPUT-1
            1830 ;
            1840 OLDROUTINE
            1850 ;
0682 4CFFFF 1860        JMP  $FFFF
            1870 ;
FFFF        1880 MESSLO =    $FFFF
FFFF        1890 MESSHI =    $FFFF
FFFF        1900 LENGTH =    $FFFF
```

BBC Format 40/80

by Sean D Kelly

Format 40/80 is a program designed to format disks in a special format; programs properly saved on these disks can be read by both 40 and 80-track drives. This enables BBC owners to exchange programs between others with different drives. A master disk, once created, can be read by any 5¼in BBC Micro disk drive.

The following is a technical explanation of the method used, and refers to disk tracks as for an 80-track disk. Track 0 is reserved for the catalogue (sectors 0 and 1) and a boot file (!BOOT — sectors 2 to 9). The next 19 tracks (1 to 19) are not used: a dummy file name is created for tracks 1 to 19. The following 20 tracks (20 to 39) are numbered 20 to 39 and are read by 80-track drives. The rest (40 to 79) are similarly numbered but in pairs (20, 20, 21, 21, and so on) for reading by a 40-track drive.

The system means that the disk capacity is effectively reduced to 50k, which should be adequate for program exchange. To create a dual format disk, CHAIN or RUN the program and follow the instructions.

```
 10 REM Dual format disc formatter
 20 REM Written by Sean D Kelly (c) 20th September 1984
 30 REM Formats discs in the special 'dual-format' which allows them
 40 REM to be read by 40 or 80 track drives
 50 REM Formatting program to be used only on 80 track drives
 60 REM To save programs on dual format discs, use a 40/80 switchable
 70 REM drive, save programs with the drive set to 40 tracks and then
 80 REM with it set to 80 tracks (or vice-versa)
 90
100 REM Error handling
110 ON ERROR MODE 7: REPORT: PRINT " at line ";ERL: @%=%A0A: END
120
130 REM Main program
140 DIM idfield% 40, command% 20, track% 79, sector% 9,directory% 511
150 osword=&FFF1
160 @%=4
170 VDU 15
180 PROCreadtracks
190 PROCreadsectors
200 PROCdrive
210 PRINT"Set drive ";D%;" to 80 tracks"
220 PROCcheck
230 PRINT '"Formatting drive ";D%'
240 PROCseek
250 PROCformat
260 PROCdirectory
270 @%=%A0A
280 PRINT
290 END
300
310 REM Order of tracks on disc .
320 DATA 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19
330 DATA 20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39
340 DATA 20,20,21,21,22,22,23,23,24,24,25,25,26,26,27,27,28,28,29,29
350 DATA 30,30,31,31,32,32,33,33,34,34,35,35,36,36,37,37,38,38,39,39
360
370 REM Order of sectors on track
380 DATA 0,5,1,6,2,7,3,8,4,9
390
400 DEF PROCosword
410 X%=command% MOD 256
420 Y%=command% DIV 256
430 A%=&7F
440 CALL &FFF1
450 ENDPROC
460
```

```
470 DEF PROCreadtracks
480 FOR T%=0 TO 79
490   READ track%?T%
500   NEXT
510 ENDPROC
520
530 DEF PROCreadsectors
540 FOR S%=0 TO 9
550   READ sector%?S%
560   NEXT
570 ENDPROC
580
590 DEF PROCdrive
600 REM Check on drive
610 PRINT "Drive? ";
620 *FX 15 1
630 REPEAT
640   D%=GET
650   UNTIL D%>47 AND D%<52
660 D%=D%-48
670 PRINT ;D%
680 ENDPROC
690
700 DEF PROCcheck
710 REM Confirm before formatting
720 PRINT "Are you ready (Y/N)? ";
730 *FX 15 1
740 S%=GET
750 IF S%<>89 AND S%<>121 STOP
760 ENDPROC
770
780 DEF PROCseek
790 REM Tell 8271 to seek track 0 on the disc
800 ?command%=D%
810 command%!4=&C000
820 command%?5=1
830 command%?6=&69
840 command%?7=0
850 PROCosword
860 ENDPROC
870
880 DEF PROCformat
890 REM Format tracks from 0 to 79
900 FOR T%=0 TO 79
910   PRINT,T%;
920   REM Set up ID field
930   FOR I%=0 TO 36 STEP 4
940     idfield%?I%=track%?T%
950     idfield%?(I%+1)=D%
960     idfield%?(I%+2)=sector%?(I%/4)
970     idfield%?(I%+3)=1
980     NEXT
990   REM Tell 8271 to format track with ID field just set up
1000   ?command%=D%
1010   command%!1=idfield%
1020   command%?5=5
1030   command%?6=&63
1040   command%?7=T%
1050   command%?8=16
1060   command%?9=42
1070   command%?10=0
1080   command%?11=16
1090   PROCosword
1100   NEXT
1110 ENDPROC
1120
1130 DEF PROCdirectory
1140 REM Write directory information in sectors 0 and 1
1150 FOR I%=0 TO 511 STEP 4: directory%!I%=0: NEXT
1160 REM Reserve track 0 for catalogue and boot file
1170 REM Then create dummy file, to occupy all space up to sector &C7
1180 REM (i.e. first free sector is C8)
1190 $(directory%+&8)="       $": REM Dummy file
1200 $(directory%+&10)="!BOOT  $": REM Boot file
1210 directory%?&18=32
1220 directory%?&10D=&BE: REM length of dummy file
1230 directory%?&115=&8: REM length of boot file
1240 directory%?&10E=&CC
1250 directory%?&116=&CC
1260 directory%?&10F=10: REM Start sector of dummy file
1270 directory%?&117=2: REM Start sector of boot file
1280 directory%?&104=0: REM Number of writes to disc
1290 directory%?&105=16: REM Position in sector of last catalogue item
1300 directory%?&106=1: directory%?&107=&90:
     REM Space on disc=&190 sectors
1310 command%?0=D%
1320 command%!1=directory%
1330 command%?5=3
1340 command%?6=&4B
1350 command%?7=0
1360 command%?8=0
1370 command%?9=34
1380 REM Tell 8271 to write directory to sectors 0 and 1
1390 PROCosword
1400 ENDPROC
1410
```

Atari Multi-Mode Text

by Garry Whitaker

This program wasn't meant to be published in two parts, but somewhere along the road to publication the second listing of Multi-Mode Text was lost. For those readers struggling to type the machine code strings in lines 550 to 580 of Multi-Mode Text, April *APC*, this program will create them for you.

For full instructions as to the use of this program, see the April issue.

```
90 REM I WOULD ADVISE YOU TO SAVE THIS BEFORE RUNNING
100 DIM MC$(185):TRAP 120:I=0:HASH=0
110 READ A:I=I+1:HASH=HASH+A*I:GOTO 110
120 RESTORE :IF HASH<>2498273 THEN ? "YOU HAVE A DATA ERROR ":END
550 ? " ":? :? "NEW":? :? :? :? "550 MC$=";CHR$(34);:POKE 766,1:FOR I=1 TO 55:RE
AD A:? CHR$(A);:NEXT I:? CHR$(34)
560 ? "560 MC$(LEN(MC$)+1)=";CHR$(34);:FOR I=1 TO 76:READ A:? CHR$(A);:NEXT I:?
CHR$(34)
570 ? "570 MC$(LEN(MC$)+1)=";CHR$(34);:FOR I=1 TO 21:READ A:? CHR$(A);:NEXT I:?
CHR$(34)
580 ? "580 MC$(LEN(MC$)+1)=";CHR$(34);:FOR I=1 TO 33:READ A:? CHR$(A);:NEXT I:?
CHR$(34):POKE 766,0:? "POKE B42,12"
590 POSITION 0,0:POKE 842,13:END
1990 POSITION 0,0:POKE 842,13:STOP
2000 DATA 104,104,133,213,104,133,212,104,104,133,214,104,133,216,104,133,215,10
4,104,133,217,104,133,219,104
2010 DATA 133,218,104,104,133,175,104,104,133,226,169,255,133,221,169,0,56,101,2
21,133,221,168,192,255,240
2020 DATA 6,177,218,201,253,208,1,96,24,201,32,48,10,201,96,176,6,56,233,32,56,1
76,7,201,32
2030 DATA 176,3,24,105,64,56,176,2,176,210,133,225,165,215,133,222,165,216,133,2
23,162,0,165,225,24
2040 DATA 101,222,133,222,169,0,101,223,133,223,232,224,8,208,238,160,0,165,212,
133,85,165,213,133,86
2050 DATA 177,222,162,0,152,72,24,101,214,133,84,177,222,24,42,133,225,165,226,1
44,2,165,175,133,224
2060 DATA 138,72,165,224,32,183,245,104,170,24,165,225,232,228,217,208,228,104,1
68,200,192,8,208,199,165
2070 DATA 85,133,212,165,86,133,213,56,176,154
```

Amstrad See PC

by John Coyne

See PC for the Amstrad CPC464 enables the user to examine and disassemble all 96k of internal memory, both ROM and RAM. Output consists of either disassembled machine code or a hex dump, both accompanied by a character representation of memory contents. The program uses two 40-column windows side by side, so one window can be used for the main routine and the other to display called subroutines.

Upon running, the program is driven by the following commands:

S: start address in hex followed by the number of lines to be displayed (a high number gives continuous scrolling)
X: switches windows
D or H: disassembly or hex dump
A or O: RAM or ROM (defaults to RAM at &4000 and &BFFF)
C: to view next address, can be held to scroll
W: to wipe current screen
Q: to quit

Continuous scrolling can be paused by [ESC] or terminated by [ESC][ESC] without exiting the program.

The program wasn't printed on an Amstrad printer so unfortunately the hash sign appears as a pound sign. Other than that, it's all correct.

```
10 ' ** SEE-PC **
20 ' AMSTRAD CPC464 ROM/RAM DISASSEMBLER
30 ' COPYRIGHT J. COYNE 1985
40 SYMBOL AFTER 256:MEMORY &ABBF:SYMBOL AFTER 248:MODE 2
50 ON BREAK GOSUB 80
60 ON ERROR GOTO 90
70 GOTO 41000
80 L=0:RETURN
90 PRINT CHR$(7):RESUME 51000
1000 ' SUNDRY SUBROUTINES
1200 CLS £0:WINDOW SWAP 0,1:WINDOW SWAP 2,3:WINDOW SWAP 4,5
1230 F=S(5,0):R=S(5,1):A=S(5,4)
1250 CLS £2:PRINT £2,CHR$(10);"FUNCTION: ";F$(R);F$(F)
1255 T1=S(0,F):T2=S(1,F):RETURN
1300 CALL &ABC0,R,@B,@A:PRINT £4,HEX$(B,2);:C$=C$+CHR$(1)+CHR$(B):RETURN
1700 IF Y<4 AND Y<>2 THEN T$=T$+"A,"
1710 RETURN
```

```
1800 IF Y=6 THEN F$="F" ELSE F$=R$(Y)
1810 S$="(C)":RETURN
1900 O$=""
1905 GOSUB 1300:O$="&"+HEX$(B,2)+O$:RETURN
1910 GOSUB 1300:O$=HEX$(B,2):GOTO 1905
1920 GOSUB 1900
1925 O$="("+O$+")":RETURN
1930 GOSUB 1910:GOTO1925
9997 '
9998 ' NOT CB/ED PREFIXED
9999 '
10000 ON X GOTO 11000,12000,13000
10010 ON Z GOTO 10100,10200,10300,10400,10500,10600,10700
10020 IF B=0 THEN T$="NOP":RETURN
10030 IF B=8 THEN T$="EX  AF,AF'":RETURN
10040 IF B=16 THEN T$="DJNZ ":GOTO 10070
10050 IF B=24 THEN T$="JR  ":GOTO 10070
10060 T$="JR  "+C$(Y-4)+","
10070 GOSUB 1300:IF B>127 THEN B=B-256
10080 B=B+A:T$=T$+"&"+HEX$(B,4):RETURN
10100 IF Y AND 1 THEN T$="ADD "+I$+","+W$(V):RETURN
10110 GOSUB 1910:T$="LD  "+W$(V)+","+O$:RETURN
10200 T$="LD  "
10210 IF V=0 THEN F$="(BC)":S$="A"
10220 IF V=1 THEN F$="(DE)":S$="A"
10230 IF V=2 THEN GOSUB 1930:F$=O$:S$=I$
10240 IF V=3 THEN GOSUB 1930:F$=O$:S$="A"
10250 IF Y AND 1 THEN T$=T$+S$+","+F$ ELSE T$=T$+F$+","+S$
10260 RETURN
10300 IF Y AND 1 THEN T$="DEC "+W$(V) ELSE T$="INC "+W$(V)
10310 RETURN
10400 T$="INC "+R$(Y):RETURN
10500 T$="DEC "+R$(Y):RETURN
10600 GOSUB 1900:T$="LD  "+R$(Y)+","+O$:RETURN
10700 T$=M$(4,Y):RETURN
11000 IF B=118 THEN T$="HALT":RETURN
11010 T$="LD  "+R$(Y)+","+R$(Z):RETURN
12000 T$=M$(0,Y):GOSUB 1700:T$=T$+R$(Z):RETURN
13000 ON Z GOTO 13100,13200,13300,13400,13500,13600,13700
13010 T$="RET "+C$(Y):RETURN
13100 IF Y AND 1 THEN 13300
13110 T$="POP "+W$(V+4):RETURN
13200 GOSUB 1910:T$="JP  "+C$(Y)+","+O$:RETURN
13300 K=-1
13310 K=K+1:IF K=12 THEN 40000
13320 IF B<>B(0,K) THEN 13310
13340 ON K GOTO 13351,13352,13353,13354,13355,13356,13357,13358,13359,13360,13361
13350 GOSUB 1910:T$="JP  "+O$:RETURN
13351 GOSUB 1920:T$="OUT "+O$+",A":RETURN
13352 T$="EX  (SP),"+I$:RETURN
13353 T$="DI":RETURN
13354 T$="RET":RETURN
13355 T$="EXX":RETURN
13356 T$="JP  "+R$(6):RETURN
13357 T$="LD  SP,"+I$:RETURN
13358 GOSUB 1920:T$="IN  A,"+O$:RETURN
13359 T$="EX  DE,HL":RETURN
13360 T$="EI":RETURN
13361 GOSUB 1910:T$="CALL "+O$:RETURN
13400 GOSUB 1910:T$="CALL "+C$(Y)+","+O$:RETURN
13500 IF Y AND 1 THEN 13300
13510 T$="PUSH "+W$(V+4):RETURN
13600 T$=M$(0,Y):GOSUB 1700:GOSUB 1900:T$=T$+O$:RETURN
13700 T$="RST"+STR$(8*Y):RETURN
19997 '
19998 ' CB PREFIXED
19999 '
20000 IF X THEN 20100
20010 T$=M$(1,Y)+R$(Z):RETURN
20100 IF X=1 THEN T$="BIT"
20110 IF X=2 THEN T$="RES"
20120 IF X=3 THEN T$="SET"
20200 T$=T$+STR$(Y)+","+R$(Z):RETURN
29997 '
29998 ' ED PREFIXED
29999 '
30000 ON X GOTO 31000,32000,40000
30010 GOTO 40000
31000 ON Z GOTO 31100,31200,31300,31400,31400,31400,31400
31010 GOSUB 1800:T$="IN  "+F$+","+S$:RETURN
31100 IF B=113 THEN 40000
31110 GOSUB 1800:T$="OUT "+S$+","+F$:RETURN
31200 IF Y AND 1 THEN T$="ADC " ELSE T$="SBC "
31210 T$=T$+"HL,"+W$(V):RETURN
31300 GOSUB 1930:IF Y AND 1 THEN T$="LD  "+W$(V)+","+O$:RETURN
```

```
31310 T$="LD   "+O$+","+W$(V):RETURN
31400 K=-1
31410 K=K+1:IF K=12 THEN 40000
31420 IF B=B(1,K) THEN T$=B$(K):RETURN
31430 GOTO 31410
32000 IF Y<4 OR Z>3 THEN 40000
32010 K=Z+4*(Y AND 1):T$=M$(V,K):RETURN
40000 T$="INVALID OPCODE":RETURN
40997 '
40998 ' INITIALISE VARIABLES
40999 '
41000 DEFINT A,B,R,F,S,V,X,Y,Z,I,K,L,P,T
41010 T$="":C$="":O$="":F$="":S$="":I$=""
41020 DIM S(1,4),F$(3)
41025 S(0,0)=2:S(0,1)=0:S(0,2)=21:S(0,3)=32:S(0,4)=0
41030 S(1,0)=2:S(1,1)=0:S(1,2)=36:S(1,3)=32:S(1,4)=0
41035 F$(0)="RAM ":F$(1)="ROM ":F$(2)="DISASSEMBLY":F$(3)="HEX DUMP"
41040 DIM M$(4,7),R$(7),W$(7),C$(7)
41050 DIM B(1,11),B$(11)
41100 FOR K=0 TO 7
41110 FOR L=0 TO 4:READ M$(L,K):NEXT
41120 READ R$(K),W$(K),C$(K):NEXT
41140 FOR K=0 TO 11:READ B(0,K),B(1,K),B$(K):NEXT
42000 DATA "ADD ","RLC ",LDI,LDIR,RLCA,B,BC,NZ
42010 DATA "ADC ","RRC ",CPI,CPIR,RRCA,C,DE,Z
42020 DATA "SUB ","RL  ",INI,INIR,RLA,D,HL,NC
42030 DATA "SBC ","RR  ",OUTI,OTIR,RRA,E,SP,C
42040 DATA "AND ","SLA ",LDD,LDDR,DAA,H,BC,PO
42050 DATA "XOR ","SRA ",CPD,CPDR,CPL,L,DE,PE
42060 DATA "OR  ","ERRR",IND,INDR,SCF,(HL),HL,P
42070 DATA "CP  ","SRL ",OUTD,OTDR,CCF,A,AF,M
43000 DATA 195,68,NEG,211,69,RETN
43020 DATA 227,70,"IM  0",243,71,"LD  I,A"
43040 DATA 201,77,RETI,217,79,"LD  R,A"
43060 DATA 233,86,"IM  1",249,87,"LD  A,I"
43080 DATA 219,94,"IM  2",235,95,"LD  A,R"
43100 DATA 251,103,RRD,205,111,RLD
44997 '
44998 ' LINE 1300 ENABLES ROM IF NEC.,="B=PEEK(A):A=A+1"
44999 '
45000 A!=43968:FOR K=0 TO 57:READ T$:POKE A!+K,VAL("&"+T$):NEXT
45100 DATA DD,6E,00,DD,66,01,4E,23
45110 DATA 46,DD,7E,04,B7,28,19,78
45120 DATA 2F,E6,C0,E2,E8,AB,07,07
45130 DATA 07,32,DD,AB,CD,00,B9,F5
45140 DATA 0A,5F,F1,CD,0C,B9,18,02
45150 DATA 0A,5F,03,70,2B,71,16,00
45160 DATA DD,6E,02,DD,66,03,73,23
45170 DATA 72,C9
46997 '
46998 ' PROMPTS MENU
46999 '
47000 C$=CHR$(13)+CHR$(10):T$=STRING$(27,CHR$(9)):PRINT TAB(33)"AMSTRAD CPC464"
47005 PRINT TAB(30)"ROM/RAM DISASSEMBLER";C$;C$;TAB(20)"PRESS:"
47010 PRINT T$;"X TO CHANGE CURRENT SIDE";C$;T$;
      "D FOR DISASSEMBLY";C$;T$;"H FOR HEX DUMP"
47020 PRINT T$;"A TO READ RAM";C$;T$;"O TO READ ROM";C$;T$;
      "S TO ENTER START ADDRESS";C$;T$;"C TO PRINT NEXT INSTRUCTION"
47030 PRINT T$;"W TO WIPE CLEAN CURRENT WINDOW";C$;T$;"Q TO QUIT";C$;C$;T$;
      "ANY KEY TO CONTINUE"
47040 T$=INKEY$:IF T$="" THEN 47040
47050 CLS
48997 '
48998 ' OPEN WINDOWS, INITIALISE SCREEN
48999 '
49000 WINDOW £6,1,80,1,2:WINDOW £4,1,40,3,22:WINDOW £5,41,80,3,22
49010 WINDOW £2,1,40,23,24:WINDOW £3,41,80,23,24
49020 WINDOW £0,1,40,25,25:WINDOW £1,41,80,25,25
49500 PRINT £6,"ADDRESS:   CODE:    ASSEMBLER:     CHAR ";
49505 PRINT £6,"ADDRESS:   CODE:    ASSEMBLER:     CHAR "
50000 S=0:B=0:GOSUB 1230:GOTO 53000
50997 '
50998 ' MAIN LOOP 51000-53080
50999 '
51000 LINE INPUT £0;"START ADDR. &";T$:T$="&"+T$:A=UNT(VAL(T$))
51010 INPUT £0,"NO. OF LINES";L
51020 CLS £0
51997 '
51998 ' LOOP FOR EACH LINE DISASSEMBLED/DUMPED 52000-52900
51999 '
52000 WHILE L<>0
52010 L=L-1
52020 A!=A:IF A!<0 THEN A!=A!+65536
52030 PRINT £4,MID$(STR$(A!),2);:PRINT £4,TAB(7);HEX$(A,4);" ";
```

```
52040 T$="":C$="":IF F=2 THEN 52100
52049 ' HEX DUMP
52050 FOR K=1 TO 4:GOSUB 1300:GOSUB 1300:PRINT £4," ";:NEXT:GOTO 52700
52099 ' DISASSEMBLER
52100 I=0:P=0:I$="HL":GOSUB 1300
52110 IF B=237 THEN P=2:GOSUB 1300:GOTO 52150
52120 IF B=221 THEN I=1:I$="IX":GOSUB 1300:GOTO 52140
52130 IF B=253 THEN I=2:I$="IY":GOSUB 1300
52140 IF B=203 THEN P=1:GOSUB 1300:IF I THEN K=B:GOSUB 1300
52150 W$(2)=I$:W$(6)=I$
52200 X=(B AND 192)/64
52210 Y=(B AND 56)/8
52220 Z=(B AND 7)
52230 V=(B AND 48)/16
52299 ' DEAL WITH IX,IY OFFSET
52300 IF I=0 THEN 52400
52310 IF NOT(Y=6 OR Z=6) THEN 52400
52320 IF P=1 THEN B=K:GOTO 52350
52330 GOSUB 1300
52350 K=1:IF B>127 THEN K=-1:B=ABS(B-256)
52360 I$=I$+CHR$(44-K)+MID$(STR$(B),2)
52400 R$(6)="("+I$+")"
52499 ' DECODING SUBROUTINES
52500 ON P+1 GOSUB 10000,20000,30000
52600 IF I THEN K=INSTR(5,T$,I$):IF K=0 THEN T$="INVALID PREFIX"
52699 ' PRINT OUTPUT
52700 PRINT £4,TAB(T1);T$;TAB(T2);
52705 FOR K=1 TO LEN(C$):PRINT £4,MID$(C$,K,1);:NEXT:PRINT £4
52800 IF NOT (INKEY(62)) THEN 52020
52900 WEND
52999 ' GET KEY
53000 CALL &BB18:PRINT £0,"ENTER X - D/H - A/O - S - C - W -Q"
53010 T$=UPPER$(INKEY$):IF T$="" THEN 53010
53020 IF T$="C" THEN CLS £0:L=1:GOTO 52000
53030 IF T$="S" THEN 51000
53040 IF T$="A" THEN R=0:S(5,1)=R:GOSUB 1250
53045 IF T$="O" THEN R=1:S(5,1)=R:GOSUB 1250
53050 IF T$="D" THEN F=2:S(5,0)=F:GOSUB 1250
53055 IF T$="H" THEN F=3:S(5,0)=F:GOSUB 1250
53060 IF T$="X" THEN S(5,4)=A:S=ABS(S=0):GOSUB 1200
53070 IF T$="W" THEN CLS £4
53080 IF T$<>"Q" THEN 53000
55000 FOR K=0 TO 6:WINDOW £K,1,80,1,25:NEXT:END
READY.
```

MACHINE CODE ROUTINE CALLED IN LINE 1300.
EQUIVALENT TO 'B=PEEK(A):A=A+1'

```
100 CALL &ABC0,R,@B,@A
105 ******************
110 ABC0 DD6E00 START: LD   L,(IX+0)
120 ABC3 DD6601        LD   H,(IX+1)
130 ABC6 4E            LD   C,(HL)
140 ABC7 23            INC  HL              ;BC←"A"
150 ABC8 46            LD   B,(HL)
160 ABC9 DD7E04        LD   A,(IX+4)      ;ACC←"R"
170 ABCC B7            OR   A
180 ABCD 2819          JR   Z,GETRAM
190 ABCF 78            LD   A,B
200 ABD0 2F            CPL                  ;IF "A">=&C000, THEN ACC←&00
210 ABD1 E6C0          AND  &C0             ;IF "A"<=&3FFF, THEN ACC←&C0
220 ABD3 E2E8AB        JP   PO,GETRAM     ;ELSE PO, SO GETRAM
230 ABD6 07            RLCA
240 ABD7 07            RLCA
250 ABD8 07            RLCA                 ;ACC←&00 OR &06
260 ABD9 320DAB        LD   (ENABLE+1),A ;&B900 FOR UPPER ROM
270 ABDC CD**B9 ENABLE:CALL &B9**          ;&B906 FOR LOWER ROM
280 ABDF F5            PUSH AF
290 ABE0 0A     ROMGET:LD   A,(BC)
300 ABE1 5F            LD   E,A
310 ABE2 F1            POP  AF
320 ABE3 CD0CB9        CALL &B90C          ;RESTORE PRIOR ROM STATE
330 ABE6 1802          JR   POKRAM
340 ABE8 0A     GETRAM:LD   A,(BC)
350 ABE9 5F            LD   E,A
360 ABEA 03     POKRAM:INC  BC              ;("A=A+1")
370 ABEB 70            LD   (HL),B
380 ABEC 2B            DEC  HL              ;"A"←BC
390 ABED 71            LD   (HL),C
400 ABEE 1600          LD   D,&00
410 ABF0 DD6E02        LD   L,(IX+2)
420 ABF3 DD6603        LD   H,(IX+3)
430 ABF6 73            LD   (HL),E
440 ABF7 23            INC  HL              ;"B"←DE
450 ABF8 72            LD   (HL),D
460 ABF9 C9            RET
READY.
```

Color Darts

by Keith Theobald

Darts is not a game I'd immediately think of simulating on a computer, but this program for the Tandy Color works quite well. The game is for two players who, upon running the program, are prompted to enter their initials. After this you can then select whether you want a double start, a double finish, computer scoring, the initial starting score, and skill level (1 = hard, 20 = easy).

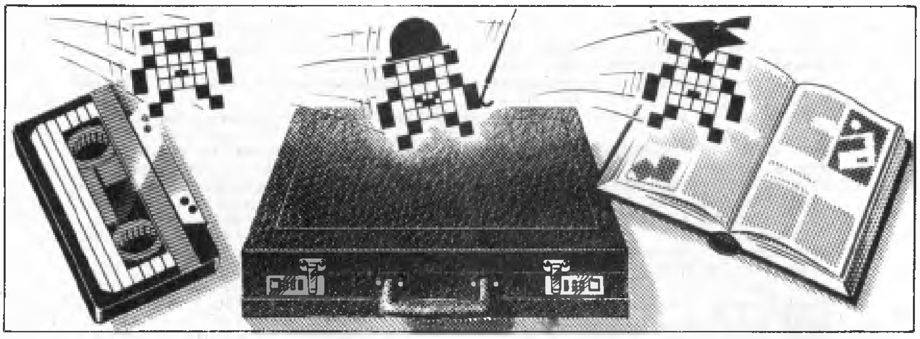
When you have configured your game you can start it by pressing S; the display will change to a dartboard.

```
0 CLS0:PRINT@75,"K.THEOBALD";:PRINT@175,"IS";:PRINT@268,"PROUD"CHR$(128)"TO";
1 PRINT@364,"PRESENT";
9 PCLEAR6:PMODE1,5:PCLS:GOSUB400:GOSUB11000
10 PMODE4,1:PCLS:POKE65495,0:CLS
11 PSET(1,0):PSET(2,0)
12 PSET(2,1):PSET(3,1)
13 FORX=1TO9:PSET(X,2):NEXT
14 PSET(2,3):PSET(3,3):FORX=6TO8:PSET(X,3):PSET(X,1):NEXT
15 PSET(1,4):PSET(2,4)
20 DIM S(20),BO(916),SX(20,3),DX(20,3),TX(20,3),B(10),TY(20,3),DY(20,3),SY(20,3)
,DA(10)
21 GET(1,0)-(9,5),DA,G:PCLS
30 CIRCLE(96,96),78:CIRCLE(96,96),86:CIRCLE(96,96),55:CIRCLE(96,96),47
40 PAINT(96,2),3,4
50 SI=1.5:PI=3.141592654:C=PI/30:FR=50
60 A=SI*C
70 SX=96+85*SIN(A):SY=96-85*COS(A)
80 LINE(96,96)-(SX,SY),PSET
90 SI=SI+3:IF SI<=58.5 THEN 60
100 COLOR0:FORD=0TO15:CIRCLE(96,96),D:NEXT:COLOR5:CIRCLE(96,96),15:COLOR5:CIRCLE
(96,96),7:CIRCLE(96,96),15:PAINT(96,96),5,5:PSET(87,100,0):PSET(100,87,0)
110 DATA 96,23,111,26,133,34,150,50,160,67,167,87,166,109,157,133,143,150,123,16
2,103,166,80,167,59,155,41,138,31,124,27,103,27,81,37,60,52,40,69,30
120 C=0:FORI=1TO20:READ X,Y:PAINT(X,Y),C,5
130 IF C=5 THENC=0:GOTO150
140 IF C=0 THENC=5
150 NEXT
170 DATA 96,14,117,18,141,29,159,44,170,63,177,86,175,116,161,144,147,159,118,17
6,97,176,78,176,48,161,33,146,21,124,16,100,16,78,35,42,53,27,67,21
180 C=5:FORI=1TO20:READ X,Y:PAINT(X,Y),C,5
190 IFC=5 THENC=0:GOTO210
200 IFC=0 THENC=5
210 NEXT
220 DATA 96,48,81,50,65,58,54,70,48,85,48,92,50,110,57,123,67,134,80,141,95,144,
110,143,123,136,136,124,141,114,145,92,141,76,134,64,123,55,110,50
230 C=5:FORI=1TO20:READX,Y:PAINT(X,Y),C,5
240 IFC=5THENC=0:GOTO260
250 IFC=0THENC=5
260 NEXT
270 DATA 96,55,108,58,116,67,116,81,116,90,116,96,116,103,116,111,109,111,103,11
5,95,115,91,115,81,115,81,108,78,101,78,96.78,90,78,83,78,73,88,73
280 C=0:FORI=1TO20:READX,Y:PAINT(X,Y),C,5
290 IFC=5 THENC=0:GOTO310
300 IFC=0 THENC=5
310 NEXT
320 COLOR5:LINE(194,0)-(255,180),PSET,B
330 LINE(194,8)-(255,8),PSET
340 LINE(225,8)-(225,180),PSET
350 GOTO830
400 DIMA$(90)
410 DRAW"BM0,0"
420 A$(65)="U4E1R2F1D1NL4D3"
430 A$(45)="BM+0,-3;R3"
440 A$(66)="U5R3F1G1NL3F1D1G1L3"
450 A$(43)="BM+2,-3NU2ND2NR2NL2"
460 A$(67)="BM+3,+0NE1L2H1U3E1R2F1"
470 A$(44)="BM+3,-1;DG"
480 A$(68)="U5R3F1D3G1L3"
490 A$(69)="NR4U3NR3U2R4"
500 A$(70)="U3NR3U2R4"
510 A$(71)="BM+3,+0E1U1L2R2D1G1L2H1U3E1R2F1"
520 A$(72)="U5D2R4U2D5"
530 A$(73)="R4L2U5L2R4"
540 A$(74)="BM+0,-1F1R1E1U4L2R4"
550 A$(75)="U5D3R1NE3U1F3"
560 A$(76)="NR4U5"
570 A$(77)="U5F2E2D5"
580 A$(78)="U5F4D1U5"
590 A$(79)="BM+0,-1U3E1R2F1D3G1L2H1"
600 A$(80)="U5R3F1D1G1L3"
610 A$(81)="BM+0,-1U3E1R2F1D3NF1NH1G1L2H1"
620 A$(82)="U5R3FDGNL3FD"
630 A$(83)="BM+0,-1;F1R2E1U1H1L2H1U0E1R2F1"
640 A$(84)="BM+2,+0;U5L2R4"
650 A$(85)="BM+0,-1NU4F1R2E1U4"
660 A$(86)="BM+0,-5;D3F2E2U3"
670 A$(87)="BM+0,-5D4F1E1U1D1F1E1U4"
680 A$(88)="U1E4BM-4,-0F4D1"
690 A$(89)="BM+2,+0U3NE2H2"
700 A$(90)="R4L4U1E4L4"
710 A$(49)="R4L2U5L1G1"
720 A$(50)="NR4U1E1R2E1U1H1L2G1"
730 A$(51)="BM+0,-1;F1R2E1U1H1L1R1E1H1L2G1"
740 A$(52)="BM+3,+0U5BL3D3R4"
750 A$(53)="BM+0,-1;F1R2E1U1H1L3U2R4"
760 A$(54)="BM+0,-1F1R2E1H1L2G1U3E1R2F1"
```

```
770 A$(55)="U1E4L4"
780 A$(56)="BM+0,-1FR2E1U1H1L2G1D1U1EH1E1R2F1G1"
790 A$(57)="BM+0,-1F1R2E1U3H1L2G1F1R2E1"
800 A$(46)="BM+3,+0U1"
810 A$(48)="BM+0,-1F1R2E1U3H1L2G1D3"
820 A$(32)="BR6"
821 RETURN
830 K$="BSDSTSOS":X=200:Y=190:GOSUB840:GOTO860GOTO860
840 FORU=1TOLEN(K$):A=ASC(MID$(K$,U,1)):DRAW"BM"+STR$(X)+","+STR$(Y)+A$(A):X=X+7
841 IF X>255 THEN X=0:Y=Y+7
850 NEXTU:RETURN
860 K$="20":X=90:Y=8:GOSUB840
870 K$="1":X=120:Y=13:GOSUB840
880 K$="18":X=142:Y=23:GOSUB840
890 K$="SCORE":X=200:Y=7:GOSUB840
900 K$="4":X=167:Y=46:GOSUB840
910 K$="13":X=178:Y=64:GOSUB840
920 K$="6":X=185:Y=99:GOSUB840:K$="10":X=179:Y=134:GOSUB840
930 K$="15":X=166:Y=156:GOSUB840:K$="2":X=144:Y=172:GOSUB840
940 K$="17":X=117:Y=186:GOSUB840:X=92:Y=189:K$="3":GOSUB840
950 K$="19":X=63:Y=186:GOSUB840:K$="7":X=41:Y=171:GOSUB840
960 K$="16":X=11:Y=148:GOSUB840:K$="8":X=9:Y=128:GOSUB840
970 K$="1":X=0:Y=100:GOSUB840:X=5:GOSUB840
980 K$="14":X=1:Y=74:GOSUB840:K$="9":X=16:Y=52:GOSUB840:K$="12":X=28:Y=30:GOSUB8
40
990 K$="5":X=60:Y=16:GOSUB840
1000 CLS:FORY=172 TO 184STEP 6:PUT(162,Y)-(170,Y+5),DA,PSET:NEXT
1001 GET(0,0)-(190,192),BO,G
1010 PRINT@11,"d a r t s":PRINT@64,"ENTER INITIALS.(NO MORE THEN 3)":PRINT:INPUT
"PLAYER1";P$(1):PRINT:IF LEN(P$(1))>3 OR P$(1)="" THEN1010
1020 INPUT"PLAYER2";P$(2):IF LEN(P$(2))>3 OR P$(2)="" THEN1020
1030 K$=P$(1):X=200:Y=16:GOSUB840:K$=P$(2):X=228:Y=16:GOSUB840
1040 LL=15:SC(1)=501:CLS:GOSUB1500:SC(2)=SC(1):K$=RIGHT$(STR$(SC(1)),LEN(STR$(SC
(1)))-1):X=200:Y=24:GOSUB840:X=228:Y=24:GOSUB840:SP(1)=31:SP(2)=31:SC=0
1050 SCREEN1,1:PL=2:DD=1
1060 DATA 25,20,1,18,4,13,6,10,15,2,17,3,19,7,16,8,11,14,9,12,5
1070 IF S(0)>0 THEN1300 ELSEFORI=0TO20:READ S(I):NEXT:GOSUB10070:GOTO1300
1075 GOSUB10000:LINE(0,0)-(40,14),PRESET,BF:IF I$="Q" THENRUN ELSEIF I$="D" THEN
1295
1080 SI=SI/3
1090 X=200:Y=190
1100 GET(X,Y)-(X+6,Y-6),B,G
1105 PUT(X,Y)-(X+6,Y-6),B,PRESET
1106 FORD=1TO LL*5:NEXT
1120 PUT(X,Y)-(X+6,Y-6),B,PSET
1125 IF INKEY$>""THEN1150
1130 X=X+7:IF X>255 THENX=200
1140 GOTO1100
1150 V=1:IF X=200 THENSI=0:V=2
1160 IF X=214 THEN V=2
1170 IF X=228 THEN V=3
1180 IF X=242 THENSI=0
1181 IF CG(3)=1 THEN1220ELSEIF DO(PL)=0 THEN 1182 ELSEIF V=1 OR V=3 THENV=0:SC=0
 ELSE DO(PL)=0
1182 SC=SC+(S(SI)*V):IF CG(2)=0THENIF SC(PL)-SC=0 AND V<>2 THENSC=SC-(S(SI)*V):G
OTO1260
1190 K$=STR$(SC):X=0:Y=14:GOSUB840
1200 K$=STR$(SC(PL)-SC)
1210 X=0:Y=7:GOSUB840
1220 IF V=1THEN X=SX(SI,DD):Y=SY(SI,DD)ELSEIFV=2 THENX=DX(SI,DD):Y=DY(SI,DD)ELSE
X=TX(SI,DD):Y=TY(SI,DD)
1230 P=PPOINT(X,Y):IFP=5 THENDRAW"C0"ELSEDRAW"C5"
1235 DRAW"BM"+STR$(X)+","+STR$(Y)+"NU4ND4NR4NL4"
1236 IF PPOINT(SX,SY)=5 THEN P=0 ELSE P=5
1237 IF CG(3)=0 THENIF SC(PL)-SC=0 THEN1260
1238 IF CG(3)=0 THENIF SC(PL)-SC<2-CG(2)  THEN SC=SC-(S(SI)*V):GOTO1260
1239 LINE(162,(166+(6*DD)))-(170,(165+(6*(DD+1)))),PRESET,BF
1240 IF TT=( THEN RETURNELSEPSET(SX,SY,P):DD=DD+1:IF DD<=3 THEN1075
1260 IF CG(3)=1 THEN1290 ELSESC(PL)=SC(PL)-SC:IF PL=1 THENX=200 ELSEX=228
1270 DRAW"BM"+STR$(X)+","+STR$(SP(PL)-7)+"C5M+17,-5"
1280 Y=SP(PL):SP(PL)=SP(PL)+7:K$=RIGHT$(STR$(SC(PL)),LEN(STR$(SC(PL)))-1):GOSUB8
40:IF SP(PL)>179 THENSP(PL)=24:IF PL=1 THEN LINE(200,17)-(220,179),PRESET,BF:X=X
-21:GOTO 1280 ELSESP(PL)=24:LINE(228,171-(250,179),PRESET,BF:X=X-21:GOTO1280
1290 FORD=1TO2000:NEXT
1295 PUT(0,0)-(190,192),BO,PSET
1300 IF SC(PL)=0 THEN 1340 ELSEIF PL=1 THENPL=2:GOTO1320
1310 IFPL=2 THENPL=1
1320 K$=P$(PL)+" TURN":X=130:Y=10:GOSUB840
1330 SC=0:DD=1:GOTO1075
1339 ' PLAYER PL HAS WON
1340 CLS:PRINTP$(PL)" HAS WON"
1345 LINE(200,9)-(253,179),PRESET,BF:LINE(225,8)-(225,180),PSET
1350 A$="DO YOU WANT ANOTHER GAME (Y/N)?"
1360 PRINT@64,A$:B$="do you want another game (y/n)?"
1370 FOR I=1TO LEN(B$)
1380 PRINT@63+I,MID$(B$,I,1);
1390 I$=INKEY$:IFI$>"" THEN1440 ELSE NEXT
1400 FORI=1TOLEN(B$)
1410 PRINT@63+I,MID$(A$,I,1);
1420 I$=INKEY$:IFI$>"" THEN1440 ELSENEXT
1430 GOTO1370
1440 IF I$="Y" THENCLS:GOTO 1010
1450 IF I$="N" THENPOKE65494,0:END
1460 GOTO1370
1469 ' CHOOSE GAME
1500 CLS:N$(0)="off":N$(1)="on":PRINT@11,"d a r t s":PRINT@67,"WRITTEN BY KEITH
THEOBALD":PRINT@110,"(C) COPYRIGHT 1985":PRINT@160,"1-DOUBLE START "N$(CG(1))
1510 PRINT"2-STRAIGHT OFF "N$(CG(2))
1520 PRINT"3-YOU SCORE(THIS IS GOOD IF YOU WISH TO PLAY ANOTHER GAME E.G. AROUND
 THE CLOCK) "N$(CG(3))
1530 PRINT"4-WHERE YOU WISH TO START. E.G. 501. THIS IS ON"SC(1)
1535 PRINT"L-CHANGE LEVEL. THIS IS ON"LL
1540 PRINT"S-START GAME"
1550 K$=INKEY$
```

```
1560 IF K$="1" AND CG(1)=0 THEN CG(1)=1:DO(1)=1:DO(2)=1:GOTO1500 ELSE IFK$="1" T
HENCG(1)=0:DO(1)=0:DO(2)=0:GOTO1500
1570 IF K$="2" AND CG(2)=0 THEN CG(2)=1:GOTO1500 ELSEIFK$="2"THENCG(2)=0:GOTO150
0
1580 IF K$="3" AND CG(3)=0 THEN CG(3)=1:GOTO1500 ELSE IFK$="3"THEN CG(3)=0:GOTO1
500
1590 IF K$="4" THEN PRINT@448,"HOW MANY";:INPUTSC(1):IFSC(1)>999 OR SC(1)<1 THEN
1590 ELSE1500
1595 IF K$="L" THENPRINT@448,"CHANGE LEVEL ENTER 1-20";:INPUTLL:IF LL<1 OR LL>20
 THEN1595 ELSE1500
1600 IF K$="S" AND SC(1)>1 THENRETURN
1610 GOTO1550
9999 SCREEN1,1:GOTO9999
10000 C=PI/30:SI=RND(19)*3:A=SI*C:GOTO10020
10010 A=SI*C
10015 LINE(X2,Y2)-(SX,SY),PRESET
10020 SX=96+43*SIN(A):SY=96-43*COS(A)
10030 X2=96+17*SIN(A):Y2=96-17*COS(A)
10040 SI=SI+3
10044 P=PPOINT(SX,SY):IF P=5 THEN COLOR0,5 ELSE COLOR5,0
10045 I$=INKEY$:IF I$>"" THENPSET(SX,SY):COLOR5,0:RETURN
10046 LINE(X2,Y2)-(SX,SY),PSET
10050 IFSI=60 THENSI=0
10060 GOTO10010
10070 DATA 94,85,84,93,90,104,90,24,97,34,102,24,112,26,113,37,121,31,132,35,128
,44,139,42,142,56,150,51,150,62,161,69,152,73,160,79,155,91,166,90,161,99,155,11
1,163,114,157,121,157,131,146,129,150,140,134,144,131,152,141,150,118,154,111,15
9,121,163,90
10080 DATA 165,100,165,96,157,77,152,78,163,69,159,61,144,57,154,52,148,49,131,4
2,140,38,131,28,113,30,122,39,115,37,93,28,100,24,90,26,78,30,69,36,76,35,59,41,
51,47,60,49,41,58,36,58,48,66,29,77,28,74,38
10100 DATA93,93,99,95,94,100,88,15,98,15,105,15,113,17,119,19,126,21,135,25,141,
28,147,33,156,41,161,47,165,54,170,62,172,69,175,77,176,87,177,96,176,103,174,11
2,172,120,169,128,166,138,162,143,158,149,151,157,147,161,140,165,128,171,122,17
4,113
10110 DATA177,97,177,104,176,88,176,78,176,69,174,62,171,54,166,48,161,41,157,34
,149,30,145,27,139,21,128,18,121,15,113,15,103,14,96,15,88,15,79,18,72,21,64,27,
55,31,49,35,42,42,35,46,31,53,26,63,21,70,19,76,17
10130 DATA91,45,97,43,100,46,106,49,112,48,115,50,121,52,125,55,128,58,133,62,13
6,65,139,69,142,76,145,79,143,84,146,91,145,95,147,101,143,107,145,112,141,115,1
39,122,136,125,133,129,129,135,125,138,121,140,116,141,113,144,107,144,100,144,9
6,148,91
10140 DATA146,85,146,81,144,76,142,70,139,66,135,62,133,58,128,54,124,51,121,49,
116,46,110,49,107,45,100,43,95,47,91,45,85,48,81,51,77,53,70,56,67,59,64,61,58,6
5,56,70,53,76,50,80,49,84,47
10150 DATA94,85,84,93,90,104,93,93,99,95,94,100
10160 FORI=0TO20:FORJ=1TO3:READ X,Y
10170 SX(I,J)=X:SY(I,J)=Y
10180 NEXTJ,I
10190 FORI=0TO20:FORJ=1TO3:READX,Y
10200 DX(I,J)=X:DY(I,J)=Y
10210 NEXTJ,I
10220 FORI=1TO20:FORJ=1TO3:READX,Y
10230 TX(I,J)=X:TY(I,J)=Y
10240 NEXTJ,I:RETURN
11000 K$="D":X=13:Y=56:DRAW"S40":GOSUB840
11010 K$="A":X=X+40:Y=56:GOSUB840
11020 K$="R":X=X+40:Y=56:GOSUB840
11030 K$="T":X=X+40:Y=56:GOSUB840
11040 K$="S":X=X+40:Y=56:GOSUB840
11050 CIRCLE(128,192),130,,1,.5,0
11060 CIRCLE(128,192),110,,1,.5,0
11070 CIRCLE(128,192),100,,1,.5,0
11080 CIRCLE(128,192),60,,.9,.5,0
11090 CIRCLE(128,192),50,,.9,.5,0
11100 SI=2.8:PI=3.141592654:C=PI/30
11110 A=SI*C
11120 SX=128+110*SIN(A):SY=192-110*COS(A)
11121 S1=128+40*SIN(A)
11125 IF SX>255 OR SX<0 OR SY>192 OR SY<0 THEN11140
11130 LINE(S1,192)-(SX,SY),PSET
11140 SI=SI+4.2:IF S1<=58.5 THEN11110
11150 PAINT(45,192),4,4:PAINT(128,192),2,4:PAINT(84,190),4,4:PAINT(125,108),2,4:
PAINT(63,140),2,4:PAINT(88,125),4,4:PAINT(102,178),4,4:PAINT(148,178),4,4:PAINT(
167,137),4,4:PAINT(196,180),4,4:PAINT(172,189),4,4:PAINT(88,178),2,4:PAINT(167,1
78),2,4
11155 PAINT(196,137),2,4
11156 GOTO11250
11250 PAINT(199,92),3,4:PAINT(220,140),4,4:PAINT(127,87),4,4:PAINT(54,118),4,4:P
AINT(230,169),2,4:PAINT(176,99),2,4:PAINT(91,93),2,4:PAINT(28,162),2,4:PAINT(178
,174),4,4:PAINT(128,143),4,4:PAINT(82,167),4,4:PAINT(183,185),2,4:PAINT(165,157)
,2,4
11260 PAINT(103,149),2,4:PAINT(77,186),2,4
11270 K$="2":X=113:Y=76:DRAW"C2S10":GOSUB840:K$="0":X=130:Y=76:GOSUB840
11280 K$="1":X=181:Y=93:GOSUB840
11290 K$="1":X=213:Y=116:GOSUB840:K$="8":X=228:Y=135:GOSUB840
11300 K$="4":X=240:Y=183:GOSUB840
11310 K$="5":X=60:Y=95:GOSUB840
11320 K$="1":X=16:Y=141:GOSUB840:K$="2":X=31:Y=121:GOSUB840
11330 K$="9":X=6:Y=186:GOSUB840
11340 COLOR1:LINE(116,145)-(81,175),PSET:LINE(128,145)-(128,175),PSET:LINE(142,1
45)-(180,175),PSET
11350 LINE(124,150)-(132,150),PSET:LINE(124,150)-(128,175),PSET:LINE(132,150)-(1
28,175),PSET:PSET(126,153):PSET(130,153):PSET(126,155):PSET(130,155)
11360 COLOR3:LINE(128,169)-(123,172),PSET:LINE(128,169)-(134,172),PSET:LINE(123,
172)-()23,186),PSET:LINE(134,172)-(134,186),PSET:LINE-(129,182),PSET:LINE-(123,1
86),PSET:PAINT(129,178),3,3:COLOR1:LINE(128,169)-(128,182),PSET
11370 LINE(105,146)-(113,154),PSET
11380 LINE(105,146)-(81,174),PSET:LINE-(113,154),PSET:COLOR3:LINE(93,164)-(107,1
52),PSET:COLOR1:PAINT(103,154),1,1:PSET(89,166,1)
11390 LINE(87,168)-(67,178),PSET:LINE-(71,184),PSET:LINE-(75,190),PSET:LINE-(87,
168),PSET:PAINT(75,180),3,1:LINE(87,168)-(71,184),PSET
11400 LINE(147,154)-(155,146),PSET:LINE-(180,175),PSET:LINE-(147,154),PSET
11410 COLOR3:LINE(153,152)-(165,162),PSET:PAINT(163,160),1,1:PSET(167,164,1):COL
OR1
11420 LINE(175,170)-(183,190),PSET:LINE-(197,178),PSET:LINE-(175,170),PSET:PAINT
(185,180),3,1:LINE(175,170)-(189,184),PSET
11430 DRAW"S4C4":SCREEN1,0:RETURN
```

PROGRAMS



APC is interested in programs written in any of the major programming languages for all home and small business micros. When submitting programs please include a cassette or disk version of your program, brief but comprehensive documentation, and a listing on plain white paper — typed if you have no printer.
Please ensure that the software itself, the documentation and the listing are all marked with your name, address, program title, machine (along with any minimum requirements) and — if possible — a daytime phone number.
All programs should be fully debugged and your own original, unpublished work.
We prefer to receive programs with a maximum 80-column width printed in emphasised typeface.
Please keep a copy of everything.
Programs are paid for at the rate of $20 per page of published listing.
Send your contributions to APC Programs, 77 Glenhuntly Road, Elwood, Victoria 3184.

BACK ISSUES

This Back Issues listing has been updated to include all of the available 1985 issues of APC. An order form is included at the end of the listing. Please allow up to four weeks for delivery.

Volume 2 No 3, 1981
Checkout: Sinclair ZX81/APC-80: Recovering lost programs, JUMP command (allows a GOTO "numeric expression")/ Building a Bigboard/Keystroke reduction for EDTASM users/ Sanders Printer reviewed/Profile of BS Microcomp/CP/M explained by Rodnay Zaks/The rapid bubble sort for the Apple/ Encryption for any Microsoft Basic/An imagined 6502 "Dream Machine" specs/Vectors explained on the Challenger IP/ Programs: TRS-80 flashing cursor and non-destructive backspace, Treasure Hunt (PET).

Volume 2 No 4, 1981
Benchtest: VIC-20, Tandy TRS-80 III/TRS-80 Monitor software compared/Computer Games: Backgammon on micros/Tree access routines explained/Gateways to Logic, Part 8: Peripherals/How Computers Communicate, Part 1: What is I/O?/ Profile Gary Blom of the Computer Company/Part 1 of 2: Defining program specification needs/6502 Assembler in Basic/ Wordpower wordprocessor program for the PET/Programs: PET Arithmetic Test, Apple Mondrian.

Volume 3 No 2, 1982
Checkout: Apple III/Fitting a smooth curve to complex data plots/Speech synthesis for TRS-80s, System 80s, Part 2/"Bridge" on micros/Relocating assembly language programs/Binary sort explained/Programmable rhythm generator project for PET/Large number calculations on micros/ Basic interpreters explained/ Checkout: ZX81 printer/APC-80 overview and debounce routine/Storing arrays on tape/ Frames of Reference, Part 1: A DP manager's guide to micros/ How Computers Communicate, Part 4: The IEEE interface/ Overview of micro-computer databases/Programs: TRS-80 Alien Seabattle.

Volume 3 No 4, 1982
Benchtests: Osborne 01, Micro Bee/APC-80: Command mode syntax error recovery/How Computers Communicate Part 6: The RS232 interface/80 x 24 display controller project/Preview of the Commodore 64/Atari 400 games reviewed/Profile: Adam Osborne/ ANS Basic's features/Solving the hidden surface problem in 3D graphics/Frames of Reference, Part 3: Micros in mainframe company/Hewlett Packard's networking capability/Programs: TRS-80 Reaction Timing, ZX81 Graphplot, PET Cheese, Superboard Spin-Fighter, TRS-80 Extra.

Volume 3 No 5, 1982
Benchtests: Texas Instruments TI 99/4A, Xerox 820/Database Benchtest: FMS-80/TRS-80 Model 1 games reviewed/Frames

of Reference, Part 4: Software standards/How Computers Communicate, Part 7: Interrupts in micro systems/How to use 3D graphics/Equation solving program/80 x 24 display controller project, Part 2/"Logo" Overview/ Printer survey/Casio's calculator printer/Programs: TRS-80 Double Precision Maths and Trig, Apple 3D Maze, Atari Sums for Kids, Apple Air Flight.

Volume 3 No 6, 1982
Benchtests: Sinclair ZX Spectrum, Sirius 1/Database Benchtest: dBase II/7th West Coast (micro-computer Faire)/ Checkout: F-10 Daisywheel printer, Arfon Expandaboard/ How Computers Communicate, Part 8: Direct memory access/ Frames of Reference, Part 5: Buying micro hardware in a DP department/Self learning program/80 x 24 display controller project, Part 3 (end)/How to get more on Apple disks/Lisp — an artificial intelligence language/ VIC-20 games reviewed/Implementing CP/M system calls from Microsoft Basic/APC Subset (first on new monthly column for assembler language routines)/ Programs: TRS-80 Invader, PET Mini-animate, VIC-20 Trailblazer, ZX81 Book Index, Weebug Monitor (TRS-80), VIC-20 Large Characters.

Volume 3 No 7, 1982
Benchtests: Sharp MZ80B, Monroe OC 8820/Checkout: Sharp PC1500, The Micro-Professor/Apple II games reviewed/APC-80: Various PEEKs and POKEs explained/ Reversing images on computer screens/Frames of Reference, Part 6: Putting your micro to work/How Computers Communicate, Part 9: Character codes/Educational arcade-type game/Programs: ZX81 Hypocycloids, TRS-80 Truth, PET Doc, TRS-80 Screen Dump, PET Boxes, Atari Earth.

Volume 3, No 9, 1982
Benchtest: ICL Personal Computer/Checkout: E40(CP/M data compression utility) Daisywriter printer, HP 11C & 120 calculators/BBC micro graphics capability/Best of APC's cartoons/ How to use Benchmarks/Logo Program (Microsoft Basic) Computer generated textures/RS232 overview, Part 2/Memory-saving utility for Apple/How Computers Communicate, Part 11: Interrupts and buffers/Programs: Sytem 80 Extended Basic, Apple Trees, ZX81 Alphabetising, PET File Companion, PET German Game.

Volume 3 No 10, 1982
Benchtests: Hewlett Packard HP-86, National Panasonic JB3000/ Checkout: Sharp PC-1211/UCSD p-System overview, Part 3 (end)/ How to implement 3D graphics on a micro/CP/M-86 vs MS-DOS: Relative merits of these 16-bit operating systems discussed/ Designing your own database/ Monitor for TRS-80/System 80/ File searching method/"Laws of Form" — a novel form of logic/ How Computers Communicate, Part 12 (end)/Benchmarking high level languages/Programs: TRS-80 Cardshuffler, PET Knockout, PET Trains

Volume 3 No 11, 1982
Benchtests: Hewlett Packard HP75C, Kaypro II, DEC Rainbow/Programs for the HP41C and Casio fx702p/Algebra checking program/More on MS-DOS vs CP/M-86/Predictions in the micro industry/Clock/ calendar card for the Apple II, Part 1/Benchmarks summary/ Programs: Apple II Piano Computer, Moon Module (Apple II, correction in Vol 4 No 1), Walls (Atari, correction in Vol 3 No 12).

Volume 4 No 2, 1983
Benchtests: Sharp PC1251/Database Benchtest: Hi Data/Micros as best friends/A major boost to the standards of 'user friendliness'/Computing can be a health hazard/Expert Systems' — part two: appraisal of 'intelligent' computers/Networks: Part 1/The Logo Turtle checked-out/Getting the most from the BBC's graphics/Are home computers just a passing fad?/ The Prestige vs The human: micro chess/Programs: Apple Character Plotter, System Tape Copier (TRS-80/System 80).

Volume 4 No. 8, 1983
Benchtests: Apple Lisa, DOT/ Checkouts: Osborne Executive, Epson FX-80 printer/Consumer Electronics Show Report/Will the Computer be the next dominant species on Earth/Milton Bradley's chess computer that moves its own pieces/Choosing suitable disks for your computer/ Cryptography on a micro/ Warnier Orr structured programming, Part 1/How to use the six function keys on the PC1500/ Programs: ZX81 Least Squares, System 80 Loading tapes from an external cassette player, TRS-80/ System 80 Adventure program, Apple II Pascal menu generation.

Volume 4 No. 9, 1983
Benchtests: Sord M5/Checkout: Tandy Model 100, Lisawrite/ Screenplay: TI 99/4A games/ Steve Wozniak returns to Apple/ Choosing a home micro/Warnier Orr programming, Part 2/Graph plotting and curve fitting on the BBC Computer/Bemoaning the mechanical teller/Programs: VIC-20 Snake line, ZX81 Surround, Apple II Screenplay, PET Histogram.

Volume 4 No. 10, 1983
Benchtests: Archives PC/Home Computer Survey — 15 micros selling for less than $1000 checked out by Steve Withers in an exhaustive market survey/ Checkout: Simons Basic, T/Maker III — office tool for the IBM PC, Digital Research Personal Basic/Computerising Your Business — a light and practical guide/Beginners Guide to Basic Program Conversion/ Clever trick with TI Sprites/ Cocktail program/Warnier Orr programming, Part III/How portable is portable/Programs: Atari No-Trons, TRS-80/System 80 Multi-Maths, Apple Text Maker, VIC 20 Spider.

Volume 4 No. 11, 1983
Benchtest: Apricot/Checkouts: Atari 600XL, Ashton Tate's Financial Planner, Condor database, Atari Writer/Which Spreadsheet? PerfectCalc/ Profiles: Clive Sinclair, Nolan Bushnell/Set up your own computer learning centre/Basic Converter Chart/Warnier Orr Programming Part 4: Techniques (end)/Programs: PET Wave Simulation, Apple II Aplist, Microsoft Basic Calendars, TI99/ 4A Breakout, Commodore Testing Your Fingers, Apple Dotter Puzzle, VIC-20 Starship, Commodore Maths Test.

Volume 4 No. 12, 1983
Benchtest: TANDY MC-10/ Checkout: Executive 816, Lotus 1-2-3, VisiOn, Gemini 15X Printer/Computerising Your Business: Part 2, Setting up/Sort Trees for beginners/Printing big on an Epson printer/Bulletin boards/Programs: VIC-20 Robotank, VZ-200 Missile Command, New 'Bee Screen, MicroBee Grooble Grab, Apple French Test Card, TRS-80 Road Rally.

Volume 5 No. 1, 1984
Benchtest: NEC PC-8201A/ Checkouts: Coleco Adam, Kaypro 10, Atari Paint, Desq/ Micro music — how it's done: Part 1/"Check Digits" — methods of ensuring correct data entry/Building models using surfaces, not lines/Column sort algorithm/Graphics on Tandy's Co Co/Spectrum listing: photofit/Locking Apple Listings/ Programs: Commodore 64 Fast Sprites, IBM PC Sheepdog Trials, VIC 20 Variable List/ Spectrum Lower CLS, Commodore 64 Mnnitor, Oric City Defense, MicroBee Tunes.

Volume 5 No. 2, 1984
Benchtest: Workslate, Commodore 720/Checkouts: Visuall, Sord's Falc, '64 Vizawrite, Brainstorm/DIY Apple Interface/ TRS-80 Disassembler/Benchmark summary to-date/Basic Program Conversion, Part 2 (Part 1 in Vol.4 No. 11)/BBC Music, Part 1/Could speech synthesisers do long term damage to the language?/Programs: TRS-80 Pascal Procedures, PET Maths Maze, BBC Logic Tree, VIC 20 Grid Bike, '64 Heli-bomber, '64 Battlestar Fighter, Apple Bridge Builder.

Volume 5 No. 3, 1984
Benchtests: IBM PC Junior, Sharp MZ-700/Checkouts: Androbot's Topo, Homeword word processor, TK!Solver, Educational Games/Basic Program Conversion, Part 3: Apple II graphics/Atari memory (for patterns of colour and sound) tester/Teach Yourself Assembler,

BACK ISSUES

Part 1/BBC Music. Part 2/View of the future from the author of VisiCalc/Give your program cassettes a spoken index/ Programs: Apple Stargo. PET Areas. Spectrum Jackpot. Atari Split-screen. TRS-80 Sound Synthesiser. 3D 'Bee. '64 Sprite Designer.

Volume 5 No. 4. 1984
Benchtests: Macintosh. Spectravideo/Checkouts: IBM Portable PC. Unix. Visiword Plus. Spectravideo/Teach Yourself Assembler. Part 2/Basic Program Conversion. Part 4: TRS-80 and Apple II graphics/Sharp PC1500 game scoring listing/ Interview: Bill Gates of Microsoft/Microchess: 4th World Chess Championship results/ Inside Atari's research laboratory/Programs:(Microsoft Basic) Inlay Cards. BBC Splash!. VIC 20 Invaders. Commodore 64 Reversi. VZ-200 Moon Lander. ('64) Gary the Guitar.

Volume 5 No. 5. 1984
Benchtests: Hewlett Packard 150 Touch Screen. Dick Smith Challenger. Canon X-07/ Checkouts: Revelation. Concurrent CP/M. StarBurst and StarIndex. Sendata modem. Commodore SFD 1001 disk drive. Brother EP44 personal typewriter/Basic Program Conversion. Part 5: Atari/ Compaction techniques. examples in Commodore Basic/ Teach Yourself Assembler. Part 3/Text vs Graphics adventures/ Operating Systems. Part 1/ Microchess: Superstar vs Constellation/Programs: '64 Plane Attack. Commodore Wordsquare. Atari Flash Simulator. Atari Pseudo-Dos. Sord M5 Charpatt. VIC 20 Ape King. MicroBee Hires Editor. Apple II Oisterads. 'Bee Label Printer.

Volume 5 No. 6. 1984
Benchtest: Sharp PC5000/ Checkouts: Codewriter. Microsoft Word. Dick Smith Cat. Apple ProDos. Knowledgeman. Autocad/Play Battleships on two Commodore computers/The History of the Keyboard/Teach Yourself Assembler. Part 4/ Artificial Intelligence: a report from Japan/Basic Program Conversion. Part 6: Spectrum/ Spectrum "wide screen" word processing/Software Copyright: the debate/The dangers of reviewing software/Programs: TRS-80 Compiler. TRS-80 Braille Writer. VIC 20 Deathwall. Basic-86 Marvin. PET 3D O's & X's. Five W 'Bee.

Volume 5 No. 7. 1984
Benchtest: Epson PX-8/ Checkouts: Memotech. Framework. HP Ink Jet Printer. Expert-Ease. Apple's Instant Artist/Teach Yourself Assembler. Part 5/Operating Systems. Part 2/ Designing and selling programs. Part 1/Calling routines available in CP/M/The story behind MSX/ Basic Program Conversion. Part 7: BBC/Programs: '64 Balloon. Atari Function Keys. BBC Sected. MicroBee Slalom. VZ-200 Blockout. '64 Split Screen Graphics. VIC 20 Monster Hunt.

Volume 5 No. 8. 1984
Benchtest: Sinclair QL/ Checkouts: Perfect Link. Friday!. Knowledgeman (Part 2). PlanStar. Commodore 64 Flight Simulator. Constellation. Pick/ Modem protocols: XModem/ Exploring WordStar/Input and Output on the Atari/$25,000 competition: Brun's Constant/Teach Yourself Assembler. Part 6/ Designing and selling programs. Part 2/Teach Yourself Lisp. Part 1/Detente between DP departments and standalone users — the Information Centre/ Programs: VIC Hatchery. BBC RAM Editor. VIC 20 Life Game. Commodore 64 Connect-Four (note correction to this program in Bludners Vol 5 No. 10. 1984). VZ-200 Database. TRS-80 Color Grafx Editor. Atari Basic System Reset.

Volume 5 No. 9. 1984
Benchtests: Hewlett Packard 110/ Checkouts: Framework vs Symphony. overview: Portable Computers. Jane vs Appleworks. Pick/Profile: Wayne Wilson/ Teach Yourself Lisp. Part 2/ Logic of assembly language. written in convertible Basic/ Teach Yourself Assembler. Part 7/Braindump: Defence of the Gotogoto Bird *(this is a really excellent one page article — Ed)*/ Microchess: Cray Blitz vs David Levy/Programs: '64 Defuse. BBC Mindwaves. VIC 20 Gothic and Greek. '64 Brackets (an updated version appears on page 76 of Vol 5 No. 11. 1984). Spectrum File. VIC Star Scramble (note: a correction to this program appears in the Bludners section of Vol 5 No. 10. 1984).

Volume 5 No. 10. 1984
Benchtests: Commodore Plus/4. Osborne Encore/Checkouts: Model 100 disk drive and video interface. Open Access. GSX from Digital Research. Netcomm modem/Is this education software any good: opinion/DIY robotics for the BBC/Combining video and PC output on the one screen/Sorting useful-sized files without a disk/Bubble memory: has it been worth the wait/Teach Yourself Lisp. Part 3/Pirate Bulletin Boards/DIY PC-video connection/How to write great software. Part 1/Programs: Atari Autorun. Commodore 64 Basic Assembler. Apple II Menu. BBC Equation Solver. Commodore Honeypot. Atari Snake. Spectrum Voyager. VZ-200 Mini calc Spreadsheet (improved in Bludners Vol 5 No. 12. 1984). IBM PC Microcomputer Graphics. Animated '64. Spectrum Graphics and sound.

Volume 5 No. 11. 1984
Benchtest: Apricot F1/Checkouts: Olivetti M24. Sperry PC. ITT Xtra. Commodore 16. AAP's microwave news service/The demise of the philosophy of the scholar: opinion/Artificial Intelligence: mind over matter/ DIY Micro Music Circuit (to plug into a parallel port)/ Computer Musicians/Teach Yourself Lisp. Part 4/Compilers: How they work and how to buy the best/Improving Commodore 64 programming skills/How to write great software. Part 2/ Molecular electronics/Programs: Commodore 64 Superfile. Commodore 64 Mouse Master. '64 Sprite Editor. BBC graphics compiler/interpreter. Spectrum Crib Player. VZ-200 MON-200. Duelling VICs.

Volume 6 No. 1. 1985
Benchtests: ComputerPhone. Data General One/Checkouts: Filevision. Juki 6100 printer/ Spreadsheet product overview/ Data validation in Microsoft Basic/Teach Yourself Lisp. Part 6/Voice synthesis and recognition as a viable alternative to the keyboard/Programs: Brimstone Part 2, Spectrum Golf, Basic Sort At Input, Commodore 64 Defkeys, Commodore 64 Space Drop, Commodore 64 Raster Interrupts, VZ-200 Sketcher.

Volume 6 No. 3, 1985
Benchtests: North Star Dimension, Digital Research GEM/ Checkouts: Microsoft Macintosh Basic 2, WordStar 2000/Preview of new Commodore and Atari machines/Moving up from dBase II to III/Teach Yourself C, Part 2/Data transmission methods/ Random number and arithmetic algorithms/Computer controlled lasars/Cray Blitz at the North American Computer Chess Championship/Programs: Spectrum Qix, Atari Disk Sector Editor, BBC Basic Line Profiler, TI/99A Submarine Hunt, Crazy Caverns, Part 1 (Commodore 64).

Volume 6 No. 4, 1985
Benchtests: Epson PX-4/ Checkouts: Pascal, Modula 2, Apple MacBasic, Electronic Mail Systems, Electric Desk, Sargon III/Microtex 666 electronic publishing/Spreadsheet warnings/New screen technologies/ Searching and sorting techniques/Teach Yourself C, Part 3/ Computer jargon explained/

Program more quickly and reliably with object-oriented design/Using Basic to write business programs/Programs: Commodore 64 Scroffs Basic, Atari Multi-Mode Text (completed in Vol. 6 No. 6, 1985), BBC Smooth Scroller), BBC Anti-List Utility, Controller, Crazy Caverns Part 2 (Commodore 64).

Volume 6 No. 5, 1985
Benchtests: Commodore PC/ Checkouts: Yamaha CX5 MSX music micro, Enable, Smart, Seiko wristwatch micro, Sidekick, Spotlight, Omni-Reader/Do you need a multi-user system?/Online information services/Lasar disks as computer memory/PC-Mainframe links, Part 1/Digital Research Concurrent DOS 286/ Teach Yourself C, Part 4/APL computer language/World Microcomputer Chess Championship/Programs: Split MicroBee, VIC-20 Programmable Keys, dBase II Preprocessor, BBC Screen Designer-Input Validation, VZ-200 Micro Type.

BACK ISSUES ORDER FORM

Please supply the following back issues:

Vol. 2 No.	☐ 3	☐ 4				
Vol. 3 No.	☐ 2	☐ 4	☐ 5	☐ 6	☐ 7	☐ 9
	☐ 10	☐ 11				
Vol. 4 No.	☐ 1	☐ 2	☐ 8	☐ 9	☐ 10	☐ 11
	☐ 12					
Vol. 5 No.	☐ 1	☐ 2	☐ 3	☐ 4	☐ 5	☐ 6
	☐ 7	☐ 8	☐ 9	☐ 10	☐ 11	
Vol.6 No.	☐ 1	☐ 3	☐ 4	☐ 5		

Volumes not listed are no longer available.
Please check carefully and use block letters when ordering.

Any one issue $4.50; any two issues $8.70; any three issues $12.70; any four issues $16.40; further issues $3.50 each. Prices include post and packing.
Cheque or P.O. payable to *Australian Personal Computer*,
77 Glenhuntly Road, Elwood, Victoria 3184.
Please allow up to four weeks for delivery.

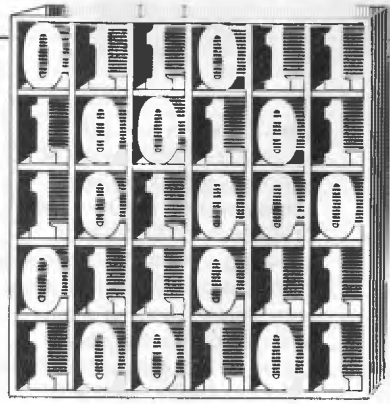
Please charge my Bankcard. Bankcard No ....

Expiry Date .... Signature ....

Name ....

Address ....

.... Postcode ....



SUBSET

David Barrow presents more documented machine code routines and useful information for the assembly language programmer. If you have a good routine, an improvement or conversion of one already printed, or just a helpful programming hint, then send it in and share it with other programmers. Subroutines for any of the popular processors and computers are welcome but please include full documentation. All published code will be paid for. Send your contributions to Sub Set, APC, 77 Glenhuntly Road, Elwood, Victoria 3184.

UNTRAPPABLE DIVISION

DIV32S (Datasheet 1) and DIV32U (Datasheet 2) by TA Browning are the response to the challenge made in *APC*, January 1985, for 68000 divisions that don't run the risk of ending up in system mode for exception processing.

The M68000 series processor does have 32-bit by 16-bit division instructions but gives only 16-bit quotients and remainders. Overflow, or division by zero, is trapped and dealt with in supervisor mode — not particularly useful if the system limits you to user mode.

DIV32S and DIV32U are full 32-bit operations giving 32-bit quotient and remainder. They both exit immediately with the overflow flag (V) set on input of a zero divisor. The 32-bit results cannot overflow the 32-bit result registers and so error trapping ought not to occur. The problem of a zero divisor can be dealt with by your own error routine in user mode.

DATASHEET 1

```
;=================================================================
:= DIV32S   32-bit signed division (non-trappable).
;=================================================================
:JOB        To divide one signed (2's complement) number by
:           another, returning signed quotient and remainder, or
:           division by zero flag.
:ACTION     IF divisor = 0:
:           THEN [ Set division by zero flag. ]
:           ELSE [ Clear division by zero flag.
:                  Calculate quotient sign.
:                  Make arguments absolute.
:                  Call unsigned division subroutine.
:                  IF quotient sign negative:
:                  [ Negate quotient and remainder. ] ]
;-----------------------------------------------------------------
:CPU        M68000 series.
:HARDWARE   None.
:SOFTWARE   DIV32U - 32-bit unsigned division (non-trappable).
:           (Assembled within an 8-bit offset range of DIV32S.)
;-----------------------------------------------------------------
:INPUT      D0 = signed dividend. D1 = signed divisor.
:OUTPUT     V = 1: division by zero, arguments unchanged.
:           V = 0: division completed.
:                  D0 = signed quotient.
:                  D1 = remainder (negative if D0 is negative).
:ERRORS     None.
:REG USE    CCR D0 D1 (A7)
:STACK USE  (USP) 18 (including call to DIV32U).
:RAM USE    None.
:LENGTH     46
:CYCLES     Not given.
;-----------------------------------------------------------------
:CLASS I     *discreet       *interruptable      *promable
:******      *reentrant      *relocatable        *robust
;=================================================================
:
DIV32S MOVE     SR,-(A7)      :Save Condition Codes and set     40E7
       BSET     #1,(A7)       :stacked overflow flag V for      08D7
                              :if divisor is zero.              0001
       TST.L    D1            :Test divisor, if zero exit       4A81
       BEQ      DIVS0         :immediately with V = 1.          6722
       BCLR     #1(A7)        :Else clear stacked V flag to     0897
                              :show division valid.             0001
       MOVE.W   #1,-(A7)      :Put a positive "sign-word" on    3F3C
                              :stack for sign calculation.      0001
       TST.L    D0            :Test sign of dividend, skip      4A80
       BPL      DIVSA         :if positive as okay,             6A04
       NEG.W    (A7)          :else change sign-word and get    4457
       NEG.L    D0            :absolute dividend value.         4480
:
DIVSA  TST.L    D1            :Repeat process to get absolute   4A81
       BPL      DIVSB         :value of divisor in D1 and       6A04
       NEG.W    (A7)          :stacked sign-word positive only  4457
       NEG.L    D1            :if arguments had same sign.      4481
:
DIVSB  BSR      DIV32U        :Go divide absolute values.       61dd
       TST.W    (A7)+         :Test sign-word while clearing    4A5F
       BPL      DIVSC         :it off stack, skip if positive,  6A04
       NEG.L    D0            :else negative so negate          4480
       NEG.L    D1            :quotient and remainder.          4481
:
DIVSC  :Label for positive result skipping negation.
DIVS0  :Early exit label for division by zero.
:
       RTR                    :Exit, restoring CCR.             4E77
;=================================================================
```

DATASHEET 2

```
;=================================================================
:= DIV32U   32-bit unsigned division (non-trappable).
;=================================================================
:JOB        To divide one unsigned (absolute) number by another,
:           returning unsigned quotient and remainder, or
:           division by zero flag.
:ACTION     IF divisor = 0:
:           THEN [ Set division by zero flag. ]
:           ELSE [ Clear division by zero flag.
:                  Clear remainder accumulator.
:                  FOR 32-bit count:
:                  [ Shift dividend left 1-bit into remainder.
:                    Clear dividend lowest bit as quotient bit.
:                    IF divisor < remainder:
:                    [ Remainder = remainder - divisor.
:                      Set quotient bit. ] ] ]
;-----------------------------------------------------------------
:CPU        M68000 series.
:HARDWARE   None.
:SOFTWARE   None.
;-----------------------------------------------------------------
:INPUT      D0 = unsigned dividend. D1 = unsigned divisor.
:OUTPUT     V = 1: division by zero, arguments unchanged.
:           V = 0: division completed.
:                  D0 = unsigned quotient.
:                  D1 = absolute remainder.
:ERRORS     None.
:REG USE    CCR D0 D1 (A7)
:STACK USE  (USP) 10.
:RAM USE    None.
:LENGTH     46
:CYCLES     Not given.
;-----------------------------------------------------------------
:CLASS 1     *discreet       *interruptable      *promable
:******      *reentrant      *relocatable        *robust
;=================================================================
:
DIV32U MOVE     SR,-(A7)      :Save Condition Codes and set     40E7
       BSET     #1,(A7)       :stacked overflow flag V for      08D7
                              :if divisor is zero.              0001
       TST.L    D1            :Test divisor, if zero exit       4A81
       BEQ      DIVU0         :immediately with V = 1.          6722
       BCLR     #1(A7)        :Else clear stacked V flag to     0897
                              :show division valid.             0001
       MOVEM.L D2/D3,-(A7)    :Save working registers D2 & D3   48E7
                              :on user stack.                   3000
       MOVE.L   D1,D2         :Move divisor to D2 and clear     2401
       CLR.L    D1            :D1 for remainder register.       4281
       MOVEQ    #31,D3        :Set count for 32 bit division.   761F
:
DIVULP ASL.L    #1,D0         :Loop, shifting next dividend     E3B0
       ROXL.L   #1,D1         :bit to remainder and clearing    E391
                              :quotient bit (bit 0,D0).
       CMP.L    D2,D1         :Test if divisor will subtract,   B282
       BCS      DIVUA         :skipping if not.                 6504
       ADDQ.B   #1,D0         :Else set quotient bit in D0      5200
       SUB.L    D2,D1         :take divisor from remainder.     9282
```

```
DIVUA  DBF      D3,DIVULP   :Always decrement counter, loop  51CB
                            :until D3 = -1 (32 times).        FFF2
       MOVEM.L (A7)+,D2/D3 :Restore working registers        4CDF
                            :D2 and D3 from user stack.       000C
DIVU0  :Early exit label for division by zero.
:
       RTR                  :Exit, restoring CCR.             4E77
:================================================================
```

Z80 BANK SWITCHING

I wrote in March that the Z80 block repeat instructions, LDIR and LDDR, could not be used for moving data between parallel banks of RAM/ROM. Since then I have been overwhelmed by letters from programmers whose machines do switch data from ROM to a RAM bank occupying identical addresses — I think I've got the message!

Keith Bremer utilises such a facility which allows memory banks to be independently set for 'read' or 'write'.

With one of two parallel blocks read-disabled (write-enabled) and the other write-disabled (read enabled), LDDR or LDIR is used with the source pointer HL having the same value as the destination pointer DE to copy data.

FRAMED

Keith Bremer is also due an apology. I didn't mention the most obvious use of his routines ENTRY and EXIT in my introduction to them in April.

Although I stressed the possibilities of using the routines to set up and index stack frames, Keith submitted them primarily as a solution to the perennial problem of assembler programming — pushing more than you pop. The Z80 stack is the most obvious place for temporary workspace, but is also very dangerous. All stack used within the routine must be tidied up before exit. If not, the most likely result will be an irrecoverable attempt to execute data in video RAM as the PC is loaded with the high order word of your square root, or whatever. Keith's routines ensure against this possibility by saving the top-of-stack address (that is, the location of the return address) in IX at the start of a routine and reloading the stack pointer from it on exit from the routine.

PASCAL'S TRIANGLE

PASCAL is a routine to calculate the value in any indexed cell of the well known Pascal's Triangle.

The triangle is constructed by adding two adjacent values to form the value that lies between them on the next row.

This is shown (skewed into an orthogonal grid) in Fig 1.

Gamblers will recognise the triangle as a tabulation of the probabilities for (X) heads uppermost among (Y) tossed coins. A-level students will also recognise the values as the coefficients of binomial expansions for $(b+a)^y$ and calculated by the formula $C(x,y) = y!/(x!(y-x)!)$.

The '!' is a factorial sign denoting that all integers from x or y down to unity be multiplied together. Using the coefficient formula can be complex, very slow, and can produce intermediate values of astronomical proportions. Robert's algorithm uses only simple three-byte comparisons to achieve the same result.

PASCAL is recursive: since each cell value is the sum of the two values above it in the triangle, the method simply decrements Y and X in turn and calls itself to calculate the new (X,Y). Recursion ends when X=0, or when X = Y and the cell value is 1. Each return but the last is to the next higher level where the 1s are accumulated. The routine does not recurse when X = 0 or X = Y on input, but returns the correct value of 1 (if input X is greater than Y the routine still returns a default value of 1, although logically these values should be 0 — the probability of five heads showing among four coins is zero). Recursive routines are very heavy users of stack and not particularly well suited to the small 256-byte stacking facility of the 6502.

	Columns (X):	0	1	2	4	5	6	7	8	9
Rows (Y):	0	1	1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1	1	1	1
	2	1	2	1	1	1	1	1	1	1
	3	1	3	3	1	1	1	1	1	1
	4	1	4	6	4	1	1	1	1	1
	5	1	5	10	10	5	1	1	1	1
	6	1	6	15	20	15	6	1	1	1
	7	1	7	21	35	35	21	7	1	1
	8	1	8	28	56	70	56	28	8	1

Fig 1 Pascal's triangle

USEFUL TRICKS

Alastair Macdonald would like to see more space given over in Subset to the unspecified instructions that seem to come with every processor, and all those nifty programming tricks that aren't quite so obvious. As an example, he notes that the CLR instruction cannot be used to zero a 68000 address register, but the instruction 'SUBA An,An' will do the job without affecting the condition codes.

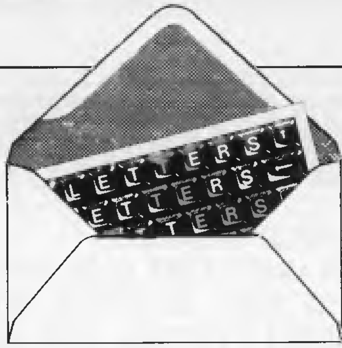
Subset has pointed out quite a few wrinkles in the past. If you do discover any new ways to ease the programmer's task, send them in and let everyone know.

DATASHEET 3

```
;================================================================
:= PASCAL    Calculate Pascal triangle cell value.
;================================================================
:JOB         To calculate the value of a single cell in the
:            Pascal triangle (or coefficient of a binomial
:            expansion) by recursively calculating preceeding
:            cell values and adding.
:ACTION      (To calculate the Pascal number P(x,y) where x is
:            the column value and y is the row value, x and y are
:            repeatedly reduced until x=0 or x=y where P(x,y)=1.
:            The number of reductions gives the required value.)
:            IF x=0 OR x>=y:
:            THEN: [ P(x,y) = 1. ]
:            ELSE: [ y = y - 1. Call PASCAL, compute P(x,y-1).
:                    Push current result.
:                    x = x - 1. Call PASCAL, compute P(x-1,y-1].
:                    Result = current result + stacked result.
:                    y = y + 1. x = x + 1. ]
;----------------------------------------------------------------
:CPU         6502
:HARDWARE    None.
:SOFTWARE    None.
;----------------------------------------------------------------
:INPUT       X = Pascal triangle cell column.
:            Y = Pascal triangle cell row.
:OUTPUT      The cell value, P(X,Y), is in M1,2,3.
:            P(X,Y) = Y! / (X! * (Y-X)!).
:            (If X=0 or X>=Y then result = 1.)
:            M0, P and A are changed.
:            X and Y are unchanged.
:ERRORS      Large input X or Y values could cause the Stack
:            Pointer to wraparound the 256-byte stack or an
:            overflow of the 3-byte result variable.
:REG USE     P A X Y
:STACK USE   3X + 2Y - 2
:RAM USE     M0 M1 M2 M3
:LENGTH      56
:CYCLES      IF X=0: 27.  IF 0<X>=Y: 32.
:            IF 0<X<Y:   (27 * (Y-1)!) / (X! ((Y-1)-X)!)
:                      + (32 * (Y-1)!) / ((X-1)! (Y-X)!)
:                      + (88 * (Y!-1)) / (X! (Y-X)!).
```

SUBSET

```
;------------------------------------------------------------------
;CLASS 2     -discreet      *interruptable      *promable
;-###--      *reentrant     -relocatable        -robust
;==================================================================
;
PASCAL STY  M0          :Save y value for comparison.        84 M0
       TXA              :First test x value for zero,        8A
       BEQ  EQUAL       :if so, P(x,y) = 1.                  F0 28
       CPX  M0          :Else test if x>=y,                  E4 M0
       BCS  EQUAL       :if so, P(x,y) = 1.                  B0 24
;
       DEY              :Else compute P(x,y-1) by a          88
       JSR  PASCAL      :recursive call to PASCAL.           20 lo hi
;
       LDA  M3          :M1,2,3 (M3 hi-byte) contains        A5 M3
       PHA              :value of preceeding cell at         48
       LDA  M2          :(x,y-1) so save it on stack         A5 M2
       PHA              :before computing other cell         48
       LDA  M1          :which adds to it to give value      A5 M1
       PHA              :at (x,y).                           48
;
       DEX              :Compute value at cell (x-1,y-1)     CA
       JSR  PASCAL      :by recursive call to PASCAL.        20 lo hi
;
       CLC              :M1,2,3 now contains value of        18
       PLA              :cell at (x-1,y-1), so add the       68
       ADC  M1          :top 3-bytes of stack, value at      65 M1
       STA  M1          :cell (x,y-1), to form value         85 M1
       PLA              :of cell at (x,y), storing           68
       ADC  M2          :result in M1,2,3.                   65 M2
       STA  M2          :  The recursive algorithm used      85 M2
       PLA              :repeatedly reduces x and y until    68
       ADC  M3          :either x=0 or x=y ... in either     65 M3
       STA  M3          :case giving a cell value of 1.      85 M3
;
       INY              :Set x and y to index current        C8
       INX              :cell and exit either to compute     E8
       RTS              :next cell or exit routine.          60
;
EQUAL  LDA  #1          :Either x=0 or x>=y. In both         A9 01
       STA  M1          :cases P(x,y) = 1 so set result      85 M1
       LDA  #0          :in M1,2,3 to 1. Then exit this      A9 00
       STA  M2          :level to compute cell value at      85 M2
       STA  M3          :next level, or exit routine if      85 M3
       RTS              :required cell value computed.       60
;==================================================================
```



LETTERS

This is the chance to air your views — mail to 'Letters', Australian Personal Computer, 77 Glenhuntly Road, Elwood, Victoria 3184. Please be as brief as possible and add 'not for publication' if your letter is to be kept private.

Ancient bugs

You may like to note that in the early 1870s Edison had a favourite expression: 'Well, boys, now let's find the bugs' (GG Bryan, *Edison* 1926). Can anyone point to an earlier use of the term, or is it (as I suspect) yet another of Edison's inventions?

Michael Behrend

Hot and cold

Without wanting to blow out of all proportion a very mundane problem (for most people anyway) of whether to put on the woollies or the short sleeves (I am by the way referring to the misprinted °F—°C conversion formulae which appeared in 'Letters', *APC* March), it is true that quite a lot of people who have been brought up to think in terms of one temperature scale cannot so easily adapt to, or think in, the other. I have known people who have shivered at the thought of going for a swim at temperatures of around 30°, and I for one was shocked to find that people have a body temperature close to the boiling point of water.

To cut the story short, I have since worked out a quick way of converting temperatures by mental arithmetic:

$$°F = (°C \times 2) - 10\% \, (°C \times 2) + 32$$

$$°C \simeq \frac{(°F-32)}{2} + 10\% \, \frac{(°F-32)}{2}$$

$$\therefore 30°C = 60 - 6 + 32 = 86°F$$

and $98°F \simeq 33 + 3.3 = 36.3\,°C$

This is good enough for everyday use.

F Khatir

Multiple comments

You are to be congratulated on your recent series of reviews by Mike Liardet of Knuth's "The Art of Computer Programming". These types of books and reviews can only lead to a more efficient approach to programming, and a greater utilisation of equipment. It is also pleasing to see some more general notes about programming, rather than just machine specific programs, which are largely undocumented. In that spirit, I wish to make some comments on the binary search routine contained in figure 4 on page 82 of the April, 1985 issue. That routine will not necessarily find the first or leftmost match if the list contains more than one possible match. If, for example, the list contains the following data:

1 2 3 3 3 4 4 5 6

and the search is for "3", it will find a "3", but not necessarily the first "3". It is necessary to cover the possibility of potential multiple matches. Because the list is sorted, any possible multiple matches will be adjoining, so a fan should be performed to find multiple matches. One way of doing this is by inserting the following code into figure 4 of that article. See listing 1.

```
35052 LPOSN = POSN : UPOSN = POSN : REM Store found location
35053 LPOSN = LPOSN - 1 : IF LPOSN < 1 THEN 35055 : REM go down
35054 IF K = K(LPOSN) THEN 35053 : REM Further match. Try again
35055 UPOSN = UPOSN + 1 : IF UPOSN > N THEN 35058
35056 REM Execution drops to here if no further lower, and starts to go up
35057 IF K = K(UPOSN) THEN 35055 : REM Further match. Try again
35058 REM This returns the following values in K
35059 REM POSN = -1 : no matches   /  (LPOSN+1 = POSN) AND (UPOSN-1 = POSN)
      : 1 match at POSN  /  ELSE multiple matches in the range LPOSN+1 TO UPOSN-1
```

Listing 1

Of course, the calling routine would have to be adjusted to take these alterations into account.

It should also be noted that the binary search routine will also work on a list that is not sorted, but indexed. This may be useful in some situations, for example if it is not desired to move strings around in memory (to avoid the garbage collection routines) or if more than one array is used for related data.

A sample program to do this is in listing 2.

J Lamich

```
10 CLS : DIM K$(10),K(10) : REM BINARY SEARCH DEMONSTRATION
20 N = 10
30 FOR I = 1 TO N : READ K$(I) : NEXT
50 DATA E,D,J,T,L,J,G,Z,A,L
60 PRINT "Unsorted list "
70 FOR I = 1 TO N
80 PRINT K$(I);
90 NEXT
100 PRINT
110 REM ======= sort
120 FOR A = 1 TO N
130 P = 1
140 FOR B = 1 TO N
150 IF K$(A) > K$(B) THEN P=P+1
160 IF K$(A) = K$(B) AND A>B THEN P=P+1
170 NEXT
180 K(P) = A
190 NEXT
200 PRINT
210 PRINT "Sorted list"
220 FOR I = 1 TO N
230 PRINT K$(K(I));
240 NEXT
250 PRINT
260 INPUT "Target ";K$
270 GOSUB 35000
280 IF POSN = -1 THEN PRINT "Not found ": GOTO 200
290 IF (LPOSN+1 = POSN) AND (UPOSN-1 = POSN) THEN PRINT "One match at " POSN : GOTO
```

```
200
300 PRINT "Multiple Matches at "LPOSN+1" to "UPOSN-1
310 GOTO 200
35000 REM binary search
35010 L = 1 : U = N
35020 IF U < L THEN POSN = -1 : RETURN
35030 POSN = INT((U+L)/2)
35040 IF K$ < K$(K(POSN)) THEN U = POSN -1 : GOTO 35020
35050 IF K$ > K$(K(POSN)) THEN L = POSN +1 : GOTO 35020
35052 LPOSN = POSN : UPOSN = POSN : REM Store found location
35053 LPOSN = LPOSN - 1 : IF LPOSN < 1 THEN 35055 : REM go down
35054 IF K$ = K$(K(LPOSN)) THEN 35053 : REM Further match. Try again
35055 UPOSN = UPOSN + 1 : IF UPOSN > N THEN 35060
35056 REM Execution drops to here if no further lower, and starts to go up
35057 IF K$ = K$(K(UPOSN)) THEN 35055 : REM Further match. Try again
35060 RETURN
```

Listing 2

can be used. POKE characters and string lengths into memory so that they can be read by machine code.

The aim of the process is to keep things simple. It's quick and easy to write Basic, and harder to write machine code, so it's a good idea to adapt the Basic to cope with the idiosyncracies of machine code. Another advantage of using PEEK and POKE is that the Basic commands mimic the machine code LDA and STA.

Once you've broken the data shown into bytes or values between 0 and 255, you can use any organisation you like for the stored data.
SG

Data efficiency

I am writing a program for the unexpanded VIC-20, some of which is in Basic and some in machine code, but I'm having trouble using the data in the arrays (dimensioned in Basic) in the machine code part of the program.

Could you enlighten me on the easiest or most efficient way to read data from arrays when in machine code.
EM

The key to this problem lies in the data you are processing, but unfortunately you don't say much about that. There are routines within the ROM which locate array values, but they're complicated and not really designed for the purpose you have in mind.

If you're using floating point numbers (decimals) then you're probably best advised to write your entire program in Basic and forget about the code: machine code to manipulate decimals tends to be verbose and slow, so there's little to be gained by using it.

If you're working with whole numbers only, the obvious solution is to make things easy for the code. Rather than use Basic arrays, you should store the data in memory along with your machine code, using PEEK and POKE to read and write values from Basic, and LDA and STA from machine code.

*It is quite easy to simulate an array with PEEK and POKE. If you're using an array of dimensions (5,6), the Basic command A(J,K) = A(L,M) would become POKE BASE + J * 6 + K, PEEK (BASE + L * 6 + M). We've assumed that stored values fall within the range 0 to 255, and BASE contains the address of an area of 42 otherwise unused bytes. Notice that, since array subscripts in VIC-20 Basic start at zero, we've had to multiply by six, rather than five, when working out the address of the data.*

If you need to store positive and negative values, you can add 128 to each number before you POKE it, giving an effective range of — 128 to 127. The limit of 256 possible values need not hinder you either — if you use two bytes for each value there's room for numbers between 0 and 65535.

As an example, the following line will PRINT the value stored at position J in a table of 100 values. We've assumed that values of J start at 0 for the first entry in the table, and BASE contains the address of an area of 200 otherwise unused bytes:

*PRINT PEEK (BASE+J*2)+256*PEEK (BASE+J*2+1)*

In this case the data is stored in pairs of bytes, like addresses in machine code: the second byte contains the number of multiples of 256, and the first contains the remainder. To store the value of K at J, use:

*POKE BASE+J*2+1,INT (K/256):*
*POKE (BASE+J*2),K-256*PEEK(BASE+J*2+1)*

If you need to manipulate strings from machine code, the same approach

Soughts of sorts

I read with great interest

your recent Mike Liardet series review of Donald Knuth's "The Art of Computer Programming". However, in view of its rather spectacular performance against other sorts and because I had never even heard of the distribution sort before, I was more than a little disappointed that Mr Liardet dismissed it with the comment that it was "not universally applicable".

One of the monotonously repetitive tasks of my computer (and, I'm sure, of many of your readers') is the sorting of string arrays in which the keys are the string elements themselves. I'm therefore constantly on the lookout for a faster sort routine and immediately determined to try to harness the distribution sort.

After a full day of hacking (and tearing my hair out!) I finally settled on the following algorithm as being probably optimal:

1 the use of the distribution sort as a presort of a randomly-ordered string array to arrange the array so that all string elements would be clustered with others sharing the same initial (naturally in ascending order); and then

2 the use of an "intelligent" bubble sort to arrange each cluster of elements in lexicographic order.

The following program, written on a Commodore 64, generates a randomly-ordered array of strings, each of 30 characters' length, and applies this distribution/bubble sort technique to it. The TI$ reference is, of course, the onboard C64 timer; interested readers using other computers will have to work out their own timing devices.

```
1  N=300 : DIM A$(N),C(26) : FOR X=1 TO N : FOR
   Y=1 TO 30
2  Z=INT(RND(0)*26)+65 : A$(X)=A$(X)+CHR$(Z) :
   NEXT : NEXT
3  TI$="000000" : PRINT TI$" SORTING . . . "
4  FOR J=1 TO N : A=ASC(A$(J))-64 : C(A)=C(A)+1
   : NEXT
5  FOR J=2 TO 26 : C(J)=C(J)+C(J-1) : NEXT :
   R=N+1
6  R=R-1 : IF R=0 THEN 13
7  A=ASC(A$(R))-64 : IF C(A)<R THEN 6
8  IF C(A)=R THEN C(A)=C(A)-1 : GOTO 6
9  A$=A$(R) : J=C(A) : C(A)=C(A)-1
10 B$=A$(J) : A=ASC(B$)-64 : K=C(A) : C(A)=C(A)-1
   : A$(J)=A$ A$=B$ : J=K : IF J<>R THEN 10
11 A$(J)=A$ : GOTO 6
12 REM *** NOW FOR THE "INTELLIGENT" BUBBLE
   SORT! ***
13 FOR J=1 TO N-1 : FOR K=J+1 TO N
14 IF ASC (A$(J))<ASC(A$(K)) THEN K=N : GOTO 16
15 IF A$(J)>A$(K) THEN A$=A$(J) : A$(J)=
   A$(K)=A$(K)=A$
16 NEXT : NEXT : PRINT TI$" SORTED!" : END
```

The Benchmarks:

# OF ELEMENTS	MY SORT	QUICKSORT
50	00' 05"	00' 08"
100	00' 14"	00' 20"
150	00' 24"	00' 41"
200	00' 45"	00' 55"
250	01' 09"	01' 17"
300	01' 45"	01' 45"
500	07' 05"	04' 50"

I have chosen 300 as the size of the array because, as the following Benchmark chart shows, that size appears to be the break-even point between my sort and the old faithful Quicksort. However, my sort has the advantage that, if the randomly-ordered array just happens to be more or less truly ordered to begin with, it will perform considerably faster, whereas — as everyone knows — the Quicksort will be disastrously slower in such a circumstance.

For instance, one Quicksort run I performed on a 500-element array took 9 minutes 45 seconds as opposed to the mean 4 minutes 50 seconds shown in the chart shown, leading me to suspect that that particular randomly-ordered array was not as random as it might have been!

I do hope these observations will be of interest to some of your readers.

K Riordan

BLUDNERS

Commodore 64 Scoffs Basic in the April issue wasn't helped by an illegible line at 1600. The line reads :AZ=AZ+(V*16^(DG-T1)). And the Atari Multi-mode Text listing is reprinted in full this month.

Datagramming

Geoff Wood demonstrates how to enhance VisiCalc by datagramming, or print-to-file programming, which increases its power and speed.

Few users of spreadsheets are aware of a simple technique that enhances the power of VisiCalc and some (but not all) other spreadsheets.

The technique is known as datagramming, or print-to-file programming. It is not mentioned in the VisiCalc manual but it is a very powerful and useful aid. In effect, it is a way of automatically typing labels and values into a worksheet to obtain results that would take much longer by hand.

This article is written in the context of VisiCalc or FlashCalc on Apple micros, but it can be readily converted for use with other micros such as the IBM PC, and other spreadsheets.

In order to understand datagramming, you should know how VisiCalc saves and loads worksheets. A worksheet file on a disk is simply a special type of text file similar in structure to the text files used by many word processing programs such as Apple Writer. When you instruct VisiCalc to load a file, the program reads the test file and carries out the programmed instructions.

A simple example is shown in Figs 1 and 2.

The formula in D3 is +B3+C3, replicated into D4 on a relative basis. The formula in B6 is B3+B4, replicated across on a relative basis. Fig 2 shows this worksheet printed out with the /SS,S1 command.

There is one line for each cell (except the blank cells) starting with the lowest right cell and working to the left and upwards. (The reason for starting with the lowest right cell rather than the top left is to speed up the loading. If VisiCalc started by loading in the top left cell, it would have to re-establish the overall size of the worksheet for each new column or row.)

```
>D6:+D3+D4        >A3:"1983
>C6:+C3+C4        >D1:/FR"Profit
>B6:+B3+B4        >C1:/FR"Costs
>A6:"Totals       >B1:/FR"Sales
>D4:+B4-C4        >A1:"Summary
>C4:1600          /W1
>B4:2000          /GOC
>A4:"1984         /GRA
>D3:+B3-C3        /GC9
>C3:800           /X!/X>A1:>A1:
>B3:1000
```

Fig 2 VisiCalc worksheet as a text file

The first entry reads >D6:+D3+D4. This is exactly what you would type on the keyboard to enter the formula +D3+D4 in cell D6. The > sign means 'Go to' and the D6 after the > sign gives the coordinates of the cell to go to. The colon has the same effect as pressing the RETURN key. The formula +D3+D4 is then typed into the cell as a value. Similar formulae are entered in C6 and B6.

The next entry reads >A6:"Totals. This means 'Go to cell A6 and enter the label "Totals.' The inverted commas before the word Totals mean that VisiCalc treats the entry as a label rather than a formula. (Note that the year entries in cells A3 and A4 are labels, not values.)

The next few entries are self-explanatory until you arrive at >D1:/FR"Profit. The /FR means 'Format the entry on the right'. Four of the last five rows are simple: they mean 'Set the Window Command at 1', 'Set the Global Order of Recalculation to Columns (rather than Rows)', 'Set the Global Recalculation to Automatic (rather than Manual)', and 'Set the Column Width to nine characters'.

Column>	A	B	C	D
Row 1	Summary	Sales	Costs	Profit
2				
3	1983	1000	800	200
4	198	2000	1600	400
5				
6	Totals	3000	2400	600

Fig 1 Summary sheet

```
>A3:/IR"1982
>B3:500
>C3:400
>D3:+B3-C3
>B7:@SUM(B3...B5)
>C7:@SUM(C3...C5)
>D7:@SUM(D3...D5)
```

Fig 3 File for changing the summary worksheet

The last entry is not so clear. It uses the /X command which is not mentioned in the VisiCalc manual. /X! means 'Change the direction indicator to vertical'. /X>A1 means 'Move the worksheet so that cell A1 is in the top left-hand corner', and >A1 means 'Move the cursor to cell A1'.

(If you have not used the command /X before, try typing it at the keyboard followed by any valid cell coordinates, then press the RETURN key: it moves that cell to the top left-hand corner of the screen. If the command is issued while the cursor is in the second window, it moves that cell to the top left-hand corner of the window.)

The file shown in Fig 2 can be created and saved with Apple Writer (or other word processing programs) or with VisiCalc itself. With Apple Writer, each entry must be on a separate line. With VisiCalc, you must type each entry on a separate row, preferably using only one cell per entry. The easiest way is to use column A and, if necessary, widen the column. (You can use two or more

Column >	A	B	C	D
Row 1	Summary	Sales	Costs	Profit
2				
3	1982	500	400	100
4	1983	1000	800	200
5	1984	2000	1600	400
6				
7	Totals	3500	2800	700

Fig 4 Summary worksheet after loading the change file

Column>	A	B	C	D	E	E	G
Row 1	Budget Statement						
2							
3		This month:— October			Cumulative		
4		Budget	Actual	Variance	Budget	Actual	Variance
5	Sales	1000	1100	100	10000	9800	−200
6							
7	Material	500	550	−50	5000	4900	100
8	Wages	200	225	−25	2000	1900	100
9	Overhead	250	250	0	2500	2600	−100
10							
11	Profit	50	75	25	500	400	−100

Fig 5 Budget statement

```
>D3:/B        >E7:#+B7
>B5:0         >E8:#+B8
>B7:0         >E9:#+B9
>B8:0         >F5:#+C5
>B9:0         >F7:#+C7
>C5:0         >F8:#+C8
>C7:0         >F9:#+C9
>C8:0         /X>A1:
>C9:0         >D3
>E5:#+B5
```

Fig 6 Datagram to update budget statement

columns if you wish because the entries are saved with the /PF command.) Each entry must be typed as a label and the file must be saved as a Print File. (It is a good idea to also save the file with the /SS command so that you can recall and edit it. Print Files cannot be recalled into VisiCalc.)

If you prefer, you can type your entries in a logical order from the top left cell to the bottom right. VisiCalc will still load such a file but it may take longer than if you start with the bottom right cell. However, if having loaded the worksheet you then save it with the normal VisiCalc /SS command, it will re-load at normal speed.

Once you grasp the idea that VisiCalc commands can be issued in the form of lines of instructions, you can create files to carry out a wide range of tasks. For example, Fig 3 shows a file to insert another row into Fig 1 and to amend the formulae for the totals.

The first line moves the cursor to A3 and inserts a row; it then enters 1982 as a label. The next two lines insert values in columns B and C and the following line inserts a formula in column D. The last three lines amend the formulae on row 7 to add all three rows of figures together.

The effect of loading this file in on top of the summary worksheet is shown in Fig 4. Of course, this is a very simple example designed to illustrate the principles of datagramming. You could achieve the same effect faster by loading in Fig 1, then typing the commands shown in Fig 3 one at a time.

Where datagramming comes into its own is for repetitive work which would otherwise involve typing in many commands. It saves time and reduces the chances of error. Once a datagram is correctly set up, it will automatically carry out the commands that would normally be typed in one at a time.

Updating budgets

A more sophisticated example of datagramming is shown in Figs 5 to 8. This involves updating monthly budget figures in such a way as to automatically enter the cumulative figures.

A simple budget statement with three columns for the current month and three columns for the cumulative data is shown in Fig 5. (In practice, the budget statement could have many more rows and columns.) The figures in columns D and G are calculated with simple formulae to show the difference between the budget figures and the actual figures. (Note that where actual sales are lower than budget the difference is shown with a minus sign, but where costs are lower than budget the difference is shown as positive.) The profit figures are calculated with formulae to show the difference between sales and total costs.

This worksheet could be saved and reloaded next month for updating. But when you enter next month's figures, the cumulative totals will not be automatically updated. The only way to do this manually would be to set the global recalculation order to manual (/GRM), amend the data in columns B and C, set the cursor in each of the cells in columns E and F (except row 11) and use the /#

Column>	A	B	C	D	E	E	G
Row 1	Budget Statement						
2							
3		This month:—			Cumulative		
4		Budget	Actual	Variance	Budget	Actual	Variance
5	Sales	0	0	0	10000	9800	−200
6							
7	Material	0	0	0	5000	4900	100
8	Wages	0	0	0	2000	1900	100
9	Overhead	0	0	0	2500	2600	−100
10							
11	Profit	0	0	0	500	400	−100

The formula in E5 is 10000+B5 | The formula in F5 is 9800+C5
The formula in E7 is 5000+B7 | The formula in F7 is 4900+C7
The formula in E8 is 2000+B8 | The formula in F8 is 1900+C8
The formula in E9 is 2500+B9 | The formula in F9 is 2600+C9

Fig 7 Budget statement after loading datagram

Column>	A	B	C	D	E	E	G
Row 1	Budget Statement						
2							
3		This month:— November			Cumulative		
4		Budget	Actual	Variance	Budget	Actual	Variance
5	Sales	1100	1200	100	11100	11000	−100
6							
7	Material	550	600	−50	5550	5500	50
8	Wages	220	225	−5	2220	2125	95
9	Overhead	275	260	15	2775	2860	−85
10							
11	Profit	55	115	60	555	515	−40

Fig 8 Budget statement after updating

command to hold the numbers, then type in a + sign and the coordinates of the corresponding cell in column B or C.

The datagram shown in Fig 6 will automatically delete the current month's sales and costs (both budget and actual), and insert formulae in columns E and F to automatically update the cumulative figures when next month's figures are entered.

The first line of the datagram puts the cursor in cell D3 and deletes the date entered there. The next eight lines take the cursor down columns B and C and replace this month's entries with a zero. The next eight lines take the cursor down columns E and F and 'fix' the numbers therein (using the # command) but also enter a formula to add the next month's figures from columns B and C. Fig 7 shows the result of loading this datagram into Fig 5.

The final stage is to enter the label November in cell D3, then enter the budget and actual figures in columns B and C. As you do so, assuming the global recalculation order is not set to manual, the figures in columns D to G and row 11 will be automatically updated. The effect is shown in Fig 8.

★ president

The Multi-Station PC System

A multi-station computer system has never been cheaper. The President 16-210 IBM PC compatible computer and President Planet terminal can be linked to create a powerful and exceptionally cost-efficient office system.

The President system is built on a solid foundation. The P16-210 and Planet terminal are designed to be highly compatible with unmodified IBM PC hardware and software. Not merely a lookalike, the P16-210 includes a variety of enhancements making it an attractive alternative to the PC. The innovative Planet terminal PC Server enables users to substantially upgrade existing systems without making equally substantial investments. An advanced local area network built into the Planet terminal makes linking as many as 127 stations to a file serving PC both convenient and inexpensive.

For more information on the President 16-210, President Planet terminal and related products, see us at **Data 85 Show**, Sydney, Centrepoint, Stands 202-203, 22-25 May, 1985 or contact us directly.

N.S.W.
President Computers Pty. Ltd.
100 George St., Hornsby, 2077
Ph. (02) 476 2700

SOUTH AUSTRALIA
Computer Marketing H.Q.
2/19 North Terrace, Hackney, 5069
Ph. (08) 42 4933

A.C.T.
President Computers ACT
Shop 2, Warramanga Centre
Warramanga, 2600
Ph. (062) 88 2000

NORTHERN TERRITORY
Darwin Business Machines
47 Stuart Highway, Darwin, 5790
Ph. (0B9) B1 5905

VICTORIA
President Computers Victoria
Suite 1, 609 St. Kilda Road,
Melbourne, 3004
Ph. (03) 529 1788

WESTERN AUSTRALIA
Micro EDP
9a/1 Leura Ave., Claremont, 6010
(09) 3B4 57B7

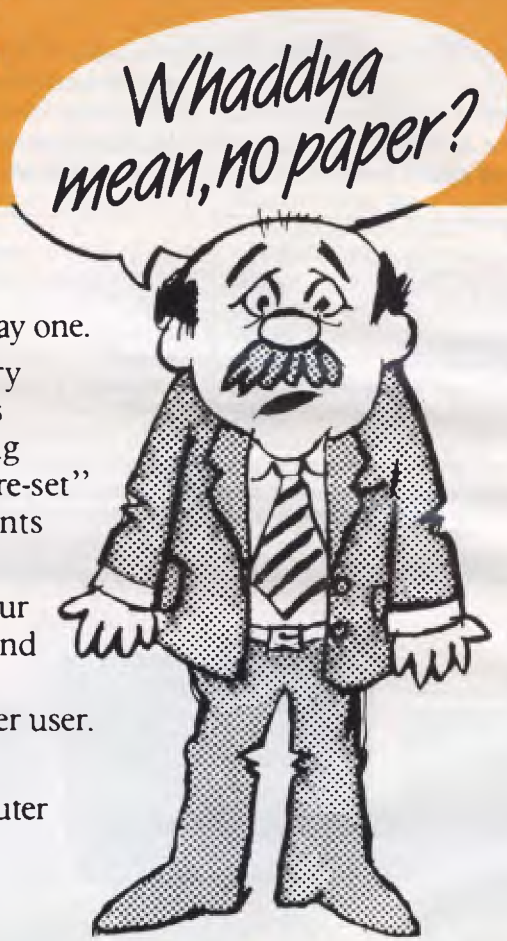
QUEENSLAND
President Computers Queensland
54 Robertson St.,
Fortitude Valley, 4006
Ph. (07) 52 3288

TASMANIA
Northern Business Machines
177 Brisbane St., Launceston, 7250
Ph. (003) 31 8388

P16-210 Features Summary: *•Microprocessar: Intel 8088, 4.77MHz •Disk Operating System: MS-DOS 2.11 •Memary: 512K RAM (expandable ta 704K) •Expansian Slats: Eight •Disk Drives: Twa flappy, hard disk •Keyboard: Erganamic design, Eurapean versions apt. •Display Adapter: Mono & colaur suppart •Parts: Parallel & serial •Miscellaneaus: Clock/calendar, convenient layout.*

PLANET Terminal Features Summary: *•Micrapracessar, Display Adapter. Keybaard: Same as P16-210 •Memory: 44BK RAM, 16K ROM •Parts: Parallel & serial, apt. • Expansian Box: Far add-ans, apt. •Netwark: 1.2 mega bits/sec. transfer rate, CSMA/CD bus cannectian, twisted pair cable, file lock/unlack, security lagan, electranic mail, and mare •Monitar sold separately.*

Chances are when you get your new computer, there'll be something missing.



It can take over six weeks for custom produced stationery to arrive.

So you won't be able to compute on day one.

Rediform instant computer stationery complements all of the popular business accounting packages for micro's, covering everything you're likely to need from "pre-set" invoices, remittance advices and statements to plain print out paper.

Rediform "pre-sets" have space for your company name, address, phone number and other necessary identification so they're particularly suitable for the small computer user. In the interim or long term.

And remember, don't leave the computer store without Rediform.

Then you can start on day one.

RediForm

Off-the-shelf Computer Stationery.

For further information, including your nearest dealer, phone:

(02) 212 1711

Expert knowledge

Artificial intelligence programs need not present a daunting programming challenge — you can use simple algorithms to operate on knowledge representations. Graham Storrs explains intelligent knowledge-based systems in the light of their central role in AI.

The Japanese Fifth Generation computer project aimed at stimulating the development of the next generation of intelligent and powerful computer systems, has laid great emphasis on the importance of Intelligent Knowledge-based Systems (IKBS). To many, the term simply means 'expert systems', but the use of knowledge in all kinds of intelligent programs has been a central theme in artificial intelligence (AI) for almost two decades.

Historical domain

Early in the history of AI, researchers attempted to build programs which could reason in general ways about problems. They produced 'problem solvers' and 'theorem provers' which worked on arbitrary problems to find solutions; this work led to little real success but it did reveal an important principle. This was that, to reason intelligently about some aspect of the real world (a 'domain' in the jargon), a computer program needed to have access to a greal deal of knowledge about the world it was working in. Abstract theorem provers were fine for abstract domains (such as logic and mathematics) where the fundamental laws of the system (its axioms) can be stated. But in the world of people, objects and events where we don't know the axioms, the only way to encode the expected behaviour of the system is in large collections of facts, and of relationships between the facts which the program can then examine and use.

The pattern for most modern AI programs was set by Terry Winograd at the beginning of the 1970s. He wrote a program that simulated a robot (called SHRDLU) which inhabited a world of simple blocks, boxes and pyramids. SHRDLU could be given commands in English to move the blocks about and could be asked to describe the current arrangement of the blocks, its own goals, and its reasons for having done something. Its performance was very impressive and had been achieved by separating SHRDLU's knowledge about the 'blocks world' from its reasoning processes. With this arrangement SHRDLU appeared extremely clever even though its reasoning ability was rather limited. Almost all its apparent intelligence was coded in its knowledge of its domain; ask it about any other subject and it would fail dismally.

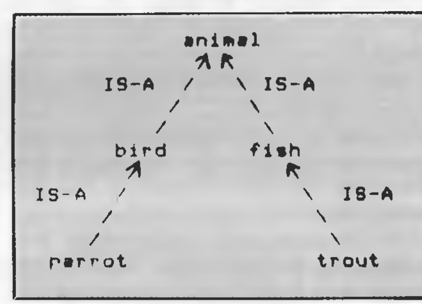
Having realised that domain knowledge is necessary for high performance, the emphasis in AI shifted away from creating general intelligence to finding ways of putting knowledge into programs. Ways of representing the ideas and relationships we use to describe the real world were looked for so that they could be given to computers to reason out. There was also a need for programming languages with enough expressive power to tackle this formidable job. Lisp was one of the first languages to be adopted for this task, with Pop2 appearing about the same time (currently having a revival in the guise of Poplog) and Prolog following a few years later.

Knowledge and networks

The most popular of all knowledge representations has been the 'semantic net'. The basic idea is simple. Concepts (which roughly correspond to nouns in everyday use) are the 'nodes' in the network, and the links between nodes represent relationships between concepts (we can think of these as verbs). One of the most important types of link is called the IS-A link and is used to categorise concepts thus:

parrot ——IS-A——> bird
trout ——IS-A——> fish

We can add other IS-A links to create an hierarchy, like this:



and we can add other kinds of links to give properties to our concepts, as in Fig 1.

Such a network would translate into some kind of data structure in a programming language. The exact structure is not important — it could be Lisp lists or Prolog clauses; in Basic it might be arrays of strings or records in a file. For the network to have any meaningful interpretation (semantics) we must have a way of interpreting the different types of link in terms of operations in the programming language. To see how it works, think of what it means for a trout

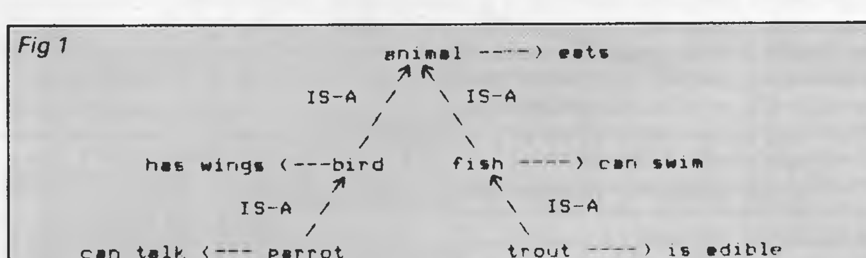


Fig 1

to be a fish. We would expect, for example, that trout would have any properties that fish (as a group) have. That is, trout should *inherit* the properties of fish. Yet just saying a trout IS-A fish does not accomplish this. We need a piece of code which takes the properties of fish and attaches them to trout. More generally, the routine should attach to any particular concept all the properties of all the concepts above it. Trout should get the 'can swim' property from fish and the 'eats' property from animal (but nothing from bird or parrot).

Basic intelligence

In Fig 2, I have represented this network as a set of arrays in Basic. There is an array (OBJECT$(i)) to hold the names of our concepts, and two other arrays (PROPERTIES$(i) and ISA$(i)) to hold a property name for each object and to point to the name of the type of object that it is. (Note that 'animal' IS-A 'topnode'; this is for the benefit of programs using the network as we will see.)

In Fig 3 there are some simple routines, written in Basic, to operate on the network. The function FNOBJNUM(O$) returns the array index of the object called O$; the procedure PROCPRINTPROPS(O) prints the property associated with the object indexed by O and then works its way up the IS-A hierarchy, recursively, printing the properties of all the objects it finds.

The thing to note about these procedures is that they do very simple things — they find an index or search a tree. It is from such simple routines for graph searching and list processing that AI programs are built. No-one would suggest that a tree searching routine is 'intelligent'. Rather, it's what such routines can do when they operate on a suitable knowledge representation that makes us think they are clever in some way.

This is not the most efficient implementation of a semantic network: it was mainly designed to be easy to understand. It is also rather limited in that each object can only have one property, inheritance can only be from a single parent node, and so on, yet it is a surprisingly powerful representation. You might like to consider ways of extending it, enabling it to 'learn' new concepts, say, or of using it in teaching programs or diagnostic systems.

Semantic nets first appeared in language understanding programs in the late Sixties. Since then their widespread use has led to increasing sophistication and considerable effort to tie down their formal properties. In particular, ideas such as strictly limiting the *types* of links and nodes allowed and of partitioning off sections of a net to represent notions such as context or scope, have been incorporated in the past 10 years.

```
10 REM ** some data for a semantic net in BASIC **
20 NUMOBJS=5:REM        the number of objects known about
30 REM  first the object names ...
40 OBJ$(1)="animal"
50 OBJ$(2)="bird"
60 OBJ$(3)="fish"
70 OBJ$(4)="parrot"
80 OBJ$(5)="trout"
90 REM  next the properties ...
100 PROP$(1)="eats"
110 PROP$(2)="has wings"
120 PROP$(3)="can swim"
130 PROP$(4)="can talk"
140 PROP$(5)="is edible"
150 REM finally, the IS-A tree ...
160 ISA$(1)="topnode":REM        the "root" of the tree
170 ISA$(2)="animal"
180 ISA$(3)="animal"
190 ISA$(4)="bird"
200 ISA$(5)="fish"
```

Fig 2 A representation in Basic of a semantic net

Expectations

The demands of writing programs to understand natural language have been a constant spur to the development of knowledge representations. In the 1970s, Marvin Minsky suggested a scheme based on 'frames'. A frame is meant to model the idea that when we think of a concept, we expect that certain other pieces of knowledge will be available. If we were about to walk into a room and someone told us it was a bedroom, we would expect to find a bed, walls, a window, wardrobe, and so on, inside. That is, we have a frame for a bedroom which has a number of 'slots' in it, each slot being for a particular piece of information we are expecting. When we open the door and see the room, these slots are filled by the particular bed, window, and so on, that we see.

Frames have the useful property of allowing defaults. If some of our information is missing (for example, we did not notice what kind of wardrobe was in the room), the appropriate slot can be temporarily filled by a default value. Modern frame-based systems also allow pro-

```
1000 REM ** a routine to print the properties    **
1010 REM ** of an object                          **
1020 PRINT
1030 PRINT"Enter the name of an object for which"
1040 PRINT"you would like to see the properties ":
1050 INPUT ONAME$
1060 OBJID=FNOBJNUM(ONAME$):REM find array index of
                                named object
1070 PRINT
1080 IF OBJID=0
        THEN PRINT"That object is not known":
             GOTO 1110
1090 PRINT"The properties of ";ONAME$;" are "
1100 PROCPRINTPROPS(OBJID):REM  print the properties of
                                the indexed object and
                                its ancestors
1110 END
2000 REM ** to find the array index of an object **
2010 DEF FNOBJID(OB$)
2020 FOUND=0
2030 FOR INDEX=1 TO NUMOBJS
2040     IF OB$=OBJ$(INDEX)
              THEN FOUND=INDEX:INDEX=N
2050 NEXT INDEX
2060 =FOUND:REM          the function returns its result
                         in the variable FOUND
3000 REM ** to print the properties of an object **
3010 DEF PROCPRINTPROPS(OB)
3020 PRINT TAB(15) PROP$(OB)
3030 IF ISA$(OB)( )"topnode"
        THEN OB=FNOBJNUM(ISA$(OB)):
             PROCPRINTPROPS(OB)
3040 ENDPROC
```

Fig 3 Some routines in Basic to operate on the network in Fig 1

cedures to be attached to slots so that different kinds of information will be expected according to what the procedure does.

In other attempts to bring expectations into the representation, ideas such as 'scripts' and 'plans' were introduced. These are meant to describe situations, such as dining in a restaurant, where the participants and the events are so stereotyped that they can be expected to appear every time these situations are encountered. Thus, in a restaurant, you would expect diners, waiters, tables, chairs and menus. The waiter would serve the diners, the diners would have conversations across the tables, eat food, drink wine, and so on. What is more, the events in a script will unfold in a well-defined sequence.

'SHRDLU appeared extremely clever even though its reasoning ability was rather limited . . . ask it about any other subject and it would fail dismally.'

Scripts have been represented as a special kind of semantic net called 'conceptual dependency' nets. The important thing about these is that they are based on the idea that there are only a very few 'primitives' in the language of knowledge with which all knowledge can be described. The primitive INGEST, for example, is the basis of all processes where something passes from outside to inside of something else (like eating, or putting a tape in a cassette player). In many ways this is similar to the idea of case structures where verbs have a number of *cases* (rather like slots in frames) which must be filled with the appropriate nouns. A verb like 'hit' might have cases for an actor (the thing that does the hitting), an object (the thing that gets hit) and an instrument (which is used for the hitting). If the actor, object and instrument cases for hit are filled with 'man', 'nail' and 'hammer' respectively, we get an idea like 'the man hit the nail with a hammer'.

Logic and expert systems

Recently there have been several attempts to show that the various representation techniques are formally correct and complete. It is important to show this because we need to be able to trust the inferences that programs make from such structures as semantic nets. Unfortunately, this is proving very difficult and some researchers argue that, to be safe, we should stick to systems which we know are formally sound, such as predicate logic. This comes at a time when logic programming (using a language based on predicate logic, such as Prolog) is becoming very popular.

In logic programming, it is simple to encode knowledge as either simple facts as in
happy(John)
or
loves(Mary,John)
which we can take to mean 'John is happy' and 'Mary loves John', or as 'if — then' rules which relate facts together, such as
happy(John):-loves(Mary,John)
where ':-' can be read as 'if', giving us 'John is happy if Mary loves John'.

Facts and rules in this form have a natural appeal, and form the basis of another knowledge representation technique called a 'production system'. A production is a rule which says that if a certain condition is true, then perform a particular action (which might be to make some consequence true). Expert systems such as Mycin and Prospector use this type of representation and it has proved to be a very powerful one in that it can capture the knowledge of a human expert (in a small domain) very concisely, and in a way that allows rapid computation of inferences. The rules naturally form tree structures (since they each call upon rules lower down for justification of their condition part) and simple tree searching routines can then follow the paths of inferences through the rules. These routines are often referred to as the system's 'inference engine' for this reason.

Languages

These days, the AI programmer has a number of languages at his disposal specially designed for representing knowledge. Some of the more popular are KRL, KL-ONE (pronounced 'clone'), KIPS and SRL. Each is based on a particular representation technique (semantic nets or frames) and allows the programmer to set up a knowledge base more easily than a more general-purpose language would.

The various expert system 'shells' (expert systems with no rules in them, just the inference engine and a rule editor) such as EMYCIN and OPS-5 must also be considered as knowledge representation languages. Shells like these are now available for the larger micros in the form of APES (a Prolog-based expert system shell from Logic Based Systems Ltd) and a Mycin-like shell called MicroExpert (which has been available for the Apple and other micros for some years now).

Conclusion

Once it is known that AI programs use simple searching algorithms to operate on knowledge representations, the mystique disappears and people tend to see artificial intelligence as something of a con. A recent attack on expert systems likened them to a man in a room taking in symbols he did not understand, manipulating them according to rules he had been told by others, and passing out answers which made no sense to him. The criticism is fair but it misses the point that, whether the man (or the computer) understands what he is doing or not, the *system as a whole* is behaving intelligently.

'Once it is known that AI programs use simple searching algorithms to operate on knowledge representations, the mystique disappears and people tend to see artificial intelligence as something of a con.'

What is more, the separation of knowledge representation and inference mechanism is thought, by psychologists, to be part of the basic organisation of the human mind.

THE 'NEW' PANASONIC PORTABLE

The world's most portable P.C.

STANDARD FEATURES INCLUDE:

- IBM compatibility
- Dual 360K Disk Drives
- 256K RAM
- 9 inch x 80 col high resolution monitor
- 80 col built-in thermal printer

SPECIAL INTRODUCTORY OFFER

only **$3995***

Plus FREE

O-P-E-N ACCESS
OR
SATCHELL FULL BUSINESS SOFTWARE PACK

Panasonic
Office Automation OA

Limited Offer
Until 26th July 1985

MARQUE computing

(03) 419-1344

134 Cambridge St. Collingwood, Vic. 3066

* Prices Excluding Sales Tax

INQUIRY 189

COMPAK

Compak Computer Centre

350 SOUTH ROAD,
MOORABBIN 3189
Telephone: (03) 555 9844

DON'T CHUCK AWAY YOUR PRINTER — JUST CHANGE YOUR CARD

Plugging the NICEPRINT printer card into your Apple II+ or IIe or compatible produces brilliant near letter quality results from your present printer. You can also choose from five fonts actually built into the card.

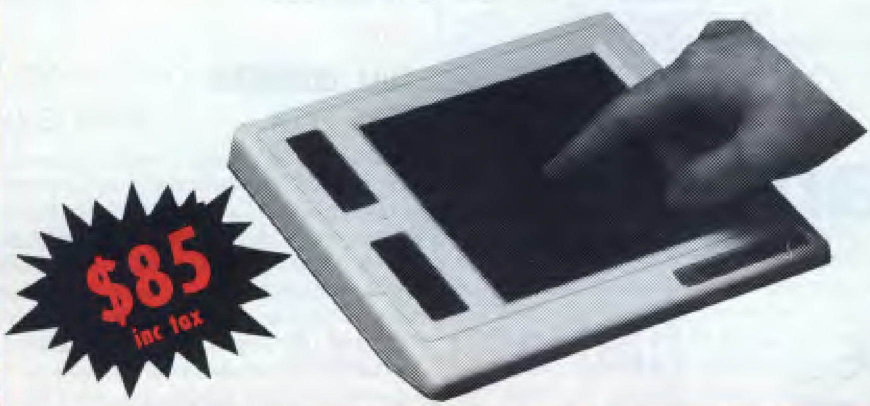
Roman font is smart and good for business correspondence.
Orator is attractive and legible.
Gothic is pretty nifty.
Script is great for personal letters.
OLDE ENGLISH is really impressive.

$150 inc tax
WORKS WITH WORDSTAR, APPLEWORKS, ZARDAX, BANK ST WRITER, EVERYTHING!

Each font can be printed with 5 pitches.
5, 6, 8.5, 10, 12 and 17cps
This sentence seems smaller with every word
No problems mixing fonts on the same line.
It works with DOS 3.3, Pascal and CP/M.
super and subscript
UNDERLINE and **BOLD** for emphasis.
Graphics dumps, inversion, rotation buit in.

This advert printed using an MX-80 with NICEPRINT and Zardax.
This is what the MX-80 prints like with the Niceprint features switched off!
NICEPRINT works with the DT-80, CP-80, PX-80, DT-100 series of printers from Shinwa as well as the MX-80, RX-80 from Epson. Gemini and Star codes are supported. It also improves the NP-2200, Olympia NP-165, PX-120, Logitek 5002 and many more.

EASY PAD



With a touch of your finger, the EASY PAD tablet takes control of your computer, more versatile than any joystick, and easier than a keyboard.
The EASY PAD tablet is compatible with most popular game software. Depending on the applications program you are using, the EASY PAD tablet can be used to make menu selections, move game players, make musical notes, draw computer graphics and much more. EASY PAD makes computing more fun than before.

[redacted]

COPY HOLDER WITH ADJUSTABLE ARM

$33 INC

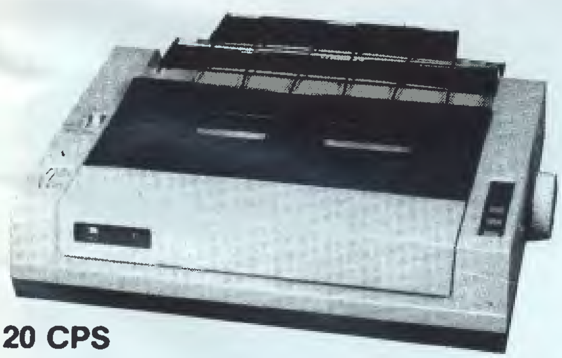
LOCKABLE DISK BOXES

LOWEST PRICE EVER!

Holds 85 Disks

$19.50

LINGO PX120 DOT MATRIX PRINTER ALSO SOLD AS P88/2 AND LOGITEK 5002



120 CPS

Switch Selectable front panel
Fast 120 cps for drafts. Quality for final copy.
Italics, superscript/subscript, graphics, 1k buffer, 5, 6, 8.5, 10, 12, 17 CPI. Definable characters. Special long life ribbon (3 million characters rating)
A brilliant new technology creating a fabulous new printer.

$469 inc.

Mode 1 120 CPS
Mode 2 Near letter quality
Mode 3 Proportional

165 CPS!

NAKAJIMA NP 2200

★ Max speed 165 caps
★ Superb "Near letter quality mode"
★ Sound proofed! You can use it in a busy office or in the middle of the night!
★ 2K buffer

$550

The quick and the quiet one!

FREIGHT $10.00

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

On the air

The idea of mobile phones communicating through cellular radio raises tantalising possibilities, but what effect will it have on data transmission? Adrian Morant examines the problems and capabilities.

So far, radio telephones have been an expensive gimmick, but cellular radio promises to change them into real business tools with a seemingly obvious potential for data transmission. But although cellular radio hit a burst of publicity earlier this year, it was never made clear what this latest example of digital technology meant for data transmission. An understanding of the technology involved helps explain this reticence.

Telecom already operates a 'first generation' cellular system covering the mainland capital cities plus the Gold Coast. As this is a large cell system, its capacity is limited to 6000 subscribers.

In December 1986 a new, high capacity system will be introduced in Sydney. Telecom has identified the top 50 markets for such a service, and expects to cover half of them by the end of 1988.

Over the longer term, Telecom expects around 150,000 subscribers in 10 years; naturally this will depend on the price of the mobile units. Current estimates suggest $2,000–$2,500 (at today's prices) would be about right. While Telecom will be providing the network, subscribers will be able to buy their units elsewhere, in the same way as we now purchase ordinary telephones.

What is it?

Cellular radio is so called because the service area is divided up into a series of small cells, each of which is served by its own base station. Each one has a relatively low power transmitter, compared with the high power transmitter that is needed for each of the comparatively small number of base stations used in conventional radiotelephone systems. The size of these cells ranges from about 4km in cities to 30km in the country. The size of cell will be dictated by the anticipated number of users and the necessity of providing overall good propagation characteristics.

The advantage of this system is that the radio channels used in one cell can be re-used, with appropriate precautions such as not using them in adjoining cells, time and time again across the country without interference. Thus, the limited number of radio channels available can be put to better use than is done in existing systems; the outcome being that the cellular networks will be able to support hundreds of thousands of subscribers compared with the 6000 on the present system. Consequently, it will be possible to set prices in anticipation of the large number of subscribers who will be forthcoming and which will enable economies of scale to be achieved.

Adopted system

The system adopted is known as AMPS (advanced mobile phone service) and is already in operation in some parts of the US. 666 channels in the 800MHz band have been allocated for cellular radio. The bulk of these channels is available for speech, but a small number is reserved for control functions.

The boundaries of a cell are not lines that can be readily drawn on a map: they just indicate the area over which reception from one base station is better than from the next. Consequently, while a set is switched on it monitors and compares signals from the surrounding base stations so that, when it is required for a phone call, it knows which is the appropriate one to use; this is not necessarily the nearest one, as it is rare for the radio waves to travel directly by line-of-sight between the base station and the mobile, and *vice versa*. In general, there are multiple reflections off buildings and other vehicles so that the propagation predictions are not always borne out in practice. In fact, as cellular operates at 800MHz it is possible that conditions could vary with a physical movement of just a few centimetres. In addition to the mobile's antenna, on a hand-held portable or mounted on a moving vehicle in the case of a car set, the changing position of other vehicles could produce a noticeable effect.

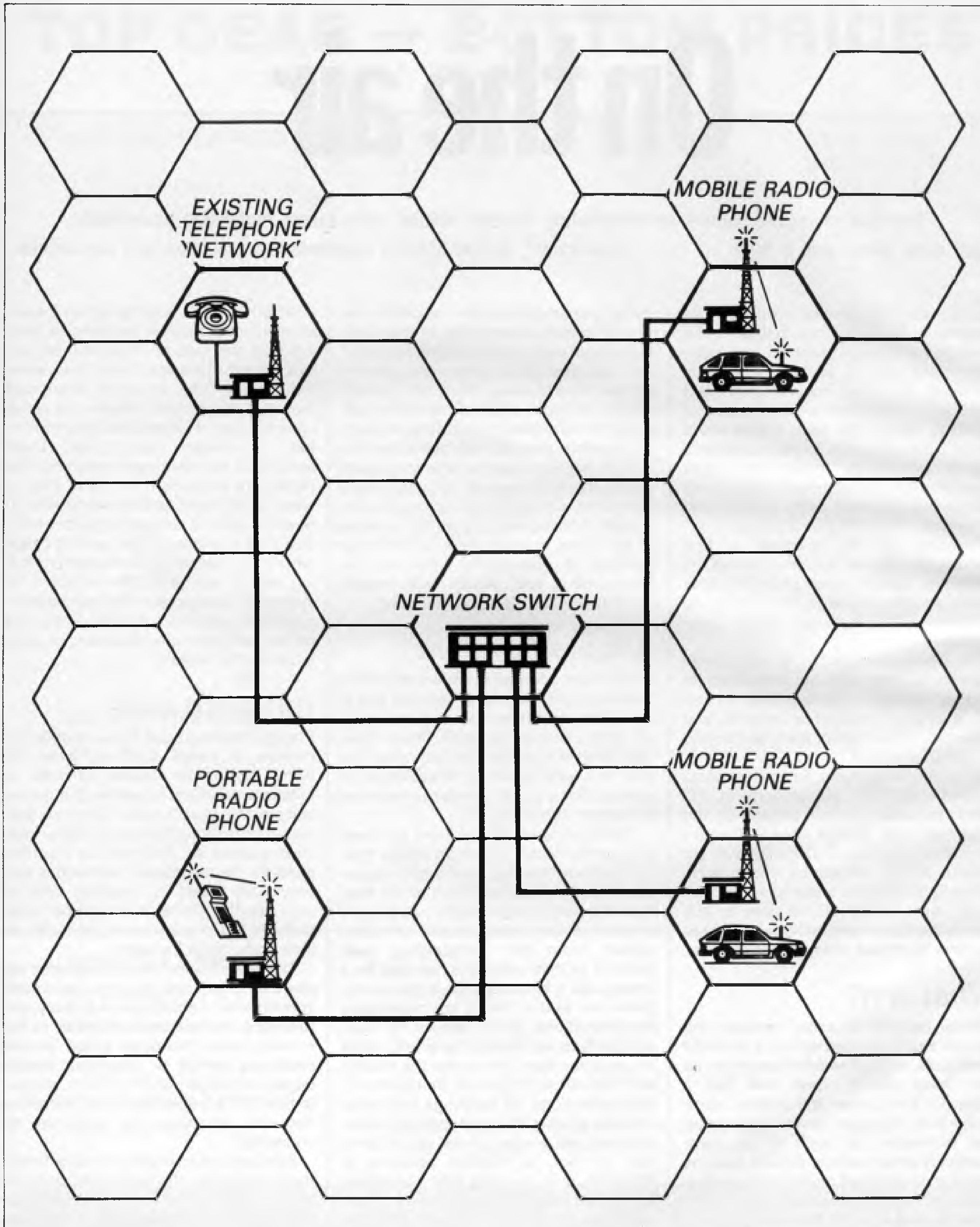
When a mobile is switched on it scans all the control channel frequencies, but it will find only two or three of them and lock onto the strongest one. Then, when it detects an 'idle' condition which says that the appropriate reverse direction control channel is free, it will transmit its full registration particulars. These include its mobile identification number (MIN), its serial number, class (that is, whether it's a hand-portable and what its power output is), and its home identification (the number of the control centre where it is registered and where its billing details are held). This enables the system to keep a record of the mobile's current location — information that will be needed when it is necessary to direct a call to that mobile.

Initiating calls

Prior to initiating a call from a mobile, the number is keyed in. Then, when the SEND button is pressed in order to initiate a call from a mobile, the mobile selects the base station with the best signal from those that are in the vicinity. Having done so, it sends, on a control channel, the telephone number that was previously keyed in, together with its own identity. (Here, it should be noted that the full number, complete with its area code, must be used.)

The base station then designates the channel to be used for the subsequent conversation, sets its status to busy, and forwards the telephone number to the control centre (variously called 'mobile switching centre' or 'electronic mobile digital exchange') which, in turn, sends it to the PSTN (public switched telephone network) so that the call can be connected.

Similarly, when calling a mobile from a fixed telephone, the subscriber dials the access code for the network followed by the required mobile identity number. At the control centre, a look-up table is referred to in order to find the nearest base station to that mobile; the control centre then requests two or three base stations in that vicinity to page that



mobile. The mobile replies to the one with the best signal, and a handshake between the base station and the control centre designates the channel that will subsequently be used for the call. The alert/ringing tone is applied over the voice channel, and speech can commence as soon as the called party in the mobile takes the call.

As a mobile moves it must be 'handed-off' from cell to cell. This means that as the radio signal strength falls, the control

Take advantage of the 18% investment allowance
Bayside are offering the following system to suit specific needs:

WORDPRO

HARDWARE

★ NEC APC III computer including —
— 2 x 720 KB floppy disk drives
— 1 x monochrome green screen
— 256 KB RAM
★ 1 x NEC e.l.f letter quality printer
★ 1 x cut sheet guide
★ 1 x printer cable
★ 1 x diskette storage box
★ 1 x 10 blank diskettes

SOFTWARE

★ MS DOS operating system
★ Wordstar Professional including —
Wordstar, Correctstar, Spellstar, Mailmerge, Star Index

$4995
including sales tax

ACCOUNT MANAGER

HARDWARE

★ NEC APC III computer including —
— 2 x 720 KB floppy disk drives
— 1 x monochrome green screen
— 256 KB RAM
★ 1 x P2 Pinwriter printer 180 c.p.s.
★ 1 x P2 pintractor
★ printer cable
★ cut sheet guide
★ 1 x diskette storage box
★ 1 x 10 blank diskettes

SOFTWARE

★ MS DOS operating system
★ Business manager accounting software including
— General Ledger, Debtors Ledger, Creditors Ledger, Invoicing and Stock (over 1400 sites around Aust. and overseas)

$5395
including sales tax

CAD STATION

HARDWARE

★ NEC APC III computer including —
— 1 x 720 KB floppy disk drive
— 1 X 10 Mb internal hard disk
— 1 x colour graphics board
— 1 x colour screen monitor
— 384 KB RAM
★ 1 x Roland A3 plotter DXY880
★ 1 x Plotter cable
★ 1 x GTCO 1212C Digitiser including —
— 4 button cursor
— power pack
★ Digitiser cable
★ 1 x diskette storage box
★ 1 x 10 blank disks

SOFTWARE

★ MS DOS operating system
★ AutoCad from Autodesk Inc.

$12,595
including sales tax

NEC has released the e.l.f letter-quality printer and the colour dot matrix printer series.

- ☐ Please send me information about other NEC systems.
- ☐ Please find enclosed $500 cheque/money order as deposit on
  - ☐ WORDPRO
  - ☐ ACCOUNT MANAGER
  - ☐ CADSTATION

NAME ______________________

COMPANY ______________________

ADDRESS ______________________

____

PHONE ______________________

★ **Installation and training can be arranged**
★ **Hire purchase or leasing available to approved clients**

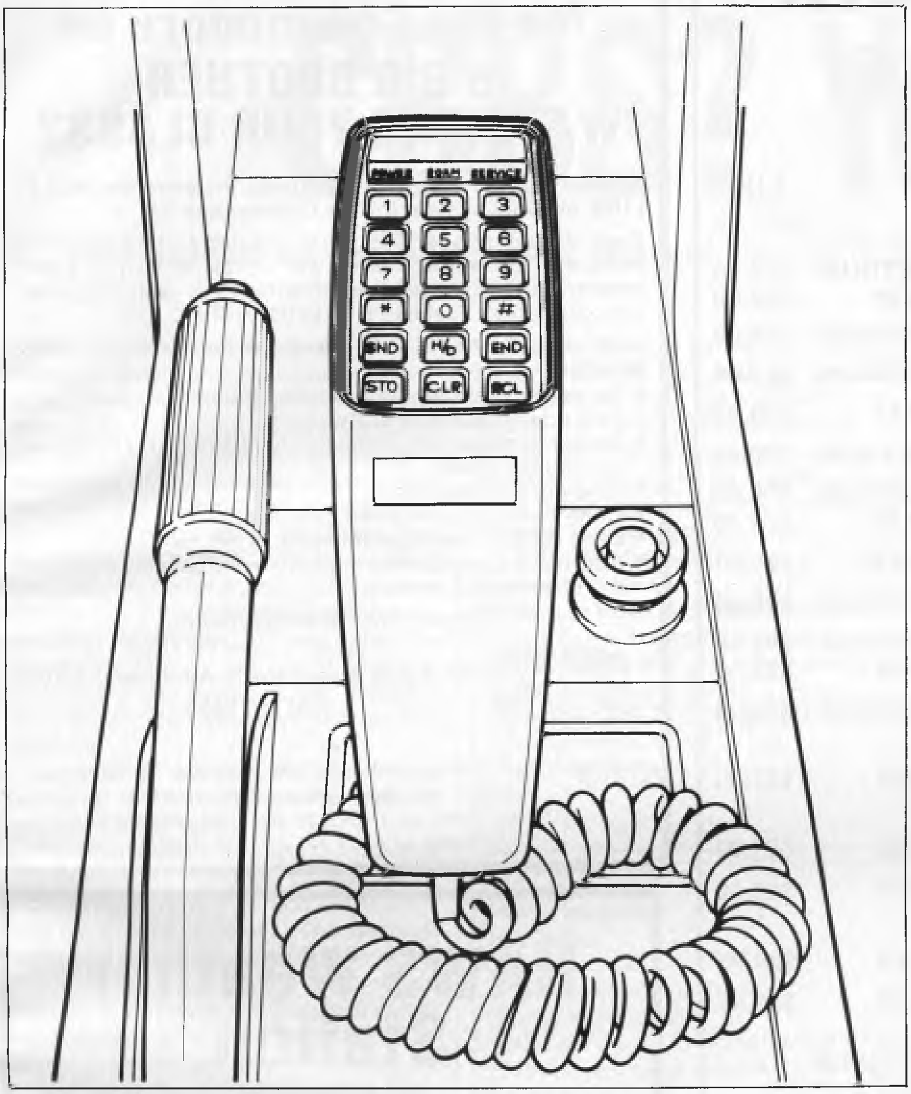
NEC

BAYSIDE COMPUTER SYSTEMS
Incorporated in Victoria
AUSTRALIA

SUITE 1, CNR. SKYE RD. & FARRELL ST. FRANKSTON
Postal Address:- P.O. BOX 647, FRANKSTON 3199

TEL. 781 4011

APC III
ADVANCED PERSONAL COMPUTER III



A range of companies is expected to produce and market the in-car equipment

centre will switch that conversation over to a free channel on an adjoining base station. As previously stated, the control channels are monitored to see which is the best base station for the mobile to communicate with. Consequently, when it is detected that it would be best to change base station, and thus channel, the control centre makes the necessary preparations. Once this is done, it uses the voice channel to send a retune command to the mobile. The mobile acknowledges this by sending a signalling tone; it switches off its transmitter, and retunes to the new channel before switching back on again to allow the call to resume. This hand-off takes about 300 milliseconds and can occur up to five times per minute, so even though the quality of the speech obtained via cellular radio is appreciably better than that normally received via the existing radiotelephones, there are problems in respect of data transmission.

Data transmission

It can be seen that data transmission via cellular radio cannot be better than a dialled-up call via the PSTN. This is because the PSTN forms part of the connection used for cellular radio so that, no matter how good the radio path is, the telephone portion is still in circuit.

As the data will have to be sent via a combined PSTN and radio path, it is necessary to cater for the vagaries of both media. While the performance of the former is well known, the words of Richard Jarvis, technical director of Racal Vodata Ltd (Racal operates one of the two cellular networks in the UK), at a recent conference on Cellular Radio provide a good summary:

A good way of understanding the characteristics of a cellular radio channel from a non-voice viewpoint is that it is either very good or very bad with little in between. Outside the period of Rayleigh fading and when there are no hands-off or "blank and burst" signalling, the channel is better than a typical telephone channel. Otherwise the channel is so bad that no transmission is possible at all.'

The 'blank and burst' signalling referred to here is similar in effect to hand-off. It can occur at any time during a call when, for example, the base station wishes to instruct a mobile to reduce its power output.

While specialised, and expensive, modems have been used for data transmission via radio in the past, it is logical to consider the use of the CCITT V series modems because part of the path is via the PSTN. To date, V21 (300bits/sec) and V23 (1200/75) have been most commonly used. However, V22 and V22bis (1200 and 2400bits/sec respectively) are becoming popular because of their full-duplex capability.

In view of the variable quality of the path, it is necessary to carry out more advanced forms of error detection and correction procedures than parity checking. In data transmission where there is a high probability that line quality will be good, it is normal to use block mode protocols such an Bisync (binary synchronous) and SDLC (synchronous data link control) which use some form (such as block check character or cyclic redundancy check) for checking the integrity of a block. The result of an error being detected is that an automatic repeat request (ARQ) is generated and returned to the sender.

This is probably going to be inadequate for cellular radio because, as the block error rate increased, the number of repeat requests would rapidly rise and, before long, the system would crash. It can be avoided by the expedient of still using the same V series modems but using forward error correction protocols. These involve the provision of redundancy within the transmitted data so that, on receipt, it is possible to detect and correct at the receiving end an appreciable number of errors in transmission.

This would reduce repeat requests to an acceptable level. In addition, if transmission is carried out at 1200bits/sec or above, the effective data throughout should not be less than 300bits/sec and thus provide an effective means of data communciation via cellular radio.

END

CBBS? Yes!

CBBS explained and new bulletin boards introduced, courtesy of Peter Tootill and Steve Withers.

CBBS systems have the problem that there is a certain amount of variation between systems. The main CBBS function menu (Fig 1) is more or less standard, but does vary in that not all systems implement every possible feature. For example, the B command on the standard CBBS menu alters baud rate, but few Australian systems have this facility. Fig 2 lists the commands in full. Only one of these commands leads to a sub-menu, and that is M for 'modem' (the file transfer section — see Fig 3). You will also find a sub-menu in the message entry mode, after you have entered the message.

Some CBBS systems are introducing menus as found on some other types of system. These menus are usually parts of the main function menu (Fig 1) broken into groups: for example, 'utilities' could include the C, D, N, P, V, and X functions (see Fig 2 for more details). The message entry and modem submenus are much the same as standard CBBS systems, but with the commands in full instead of just initial letters. When you become accustomed to the system, you can choose the expert mode (key 'X') and just the function summaries, as in a standard CBBS system, will be provided. You can escape the submenus by hitting CARRIAGE RETURN.

Entering messages manually while online is a similar process to other BBSs. There is normally no provision for private messages (electronic mail), messages prepared offline can only be uploaded to CBBS systems one line at a time which can be a problem if your software doesn't allow you to send prepared messages in this way. You can usually get around it by putting half-a-dozen or so spaces at the start of each line, which gives the BBS time to process each line while the spaces are being sent, and you shouldn't lose any message content.

The message editor on CBBS is very flexible. It gives you the option of retyping the line or replacing a portion of it. If finger trouble or line noise meant that 'Frod' appeared in your message when it should have been 'Fred', then R/Frod/Fred should do the trick.

CBBS has keyword searching facilities which enable searches through messages for particular items of interest. As an example, '35,BBS & software' would start at message number 35 and search for messages containing the words 'BBS' and 'software'. All CBBSs allow you to search the subject, to and from fields, and some will even search through the full message text.

CBBS also have a number of built-in commands such as NEWS, CHAT, MINE, and HELP. The first is obvious, the second pages the sysop to chat to you via the keyboard, and the third checks for messages to you. The help system is easy to use — you just type 'help' followed by the function you need help on: for example, typing 'help messages' could give you information on entering messages.

There are other keywords, but there is some variation here between CBBS systems.

Some CBBSs support file transfer using modem protocols. The commands are in the modem sub-menu, and are 'S' for send a file to you, and 'R' for receive a file from you. Text files can be downloaded in standard ASCII mode (but without buffer control codes.) Upload is only possible by using Xmodem protocols.

Function: /, A, B, C, D, E, F, G, H, K, M, N, O, P, Q, R, S, T, U, V, W, X, #
Please select option:
(or ? if not known) ???

Fig 1 The main CBBS function menu

(/) Quick logoff
(A)lter baud rate
(B)ulletins
(C)ase change
(D)uplex, echo on/off
(E)nter a message
(F)irst-time user information
(G)oodbye. Leave system
(H)elp
(K)ill/erase messages
(M)odem section
(N)ull selection
(O)ne-line summary of messages
(P)rompt bell on/off
(Q)uick summary of messages
(R)ead messages
(S)ummarise messages
(T)ime and date
(U)sers' flags
(V)ideo backspace
(W)elcome
(X)pert user mode
(#) Show caller number

Fig 2 Full list of CBBS commands

Online drinks

The Americans have a new line in 'intelligent' vending machines, which apparently phone the depot when they are getting low and request more stocks. Two of these clever machines in Omaha caused a bit of a stir by repeatedly dialling the depot on a Sunday — alert telephone operators thought that it was burglars. Obviously there are intelligent vending machines and not-so-intelligent vending machines.

New Modem

We recently had the opportunity to try out a new modem produced in Melbourne by Delphic Digital. As we have come to expect, it is a multi-

A = Access to another software area
D = Display files available
M = Messages for this section
R = Receive File from you * Xmodem protocol
S = Send file to you * Xmodem protocol
T = Type a file (use if you don't have Xmodem)

Fig 3 Available modem functions

TEST DRIVE APC!

MONEY BACK GUARANTEE

Subscribe today, and take APC for a 'Test Drive'. See how it handles, revel in the sleek design, the exclusive benchtests, the blinding editorial, the hot tips, the humour, the top writers—You won't be able to put it down!! If after three months you don't wish to continue your subscription we will refund your payment in full. So, get into gear and 'Test Drive' APC today.

SEE PAGE 48 FOR A FREE POST SUBSCRIPTION CARD

standard device (CCITT V21 and V23, plus Bell 103 and 202). Really, there's not much you can say about a modem, as long as it works. This one worked just fine.

Two features make it particularly convenient to use. The case is very solid, and the extra weight means you don't need to hold it down while operating the controls. Instead of using the sub-miniature toggle switches fitted to some modems, Delphic chose push-buttons which are easier to operate and give a clearer indication of the selected modes.

The attention to detail shows in the completely unobstructed DB-25 connector and the built-in power supply. There's nothing amazing about any of these features, but Delphic has avoided the pitfalls that have caught some other companies. The result is a modem that is compact, convenient, and attractive.

We obviously can't make any comments about reliability after a short trial, but the two year warranty suggests that Delphic has confidence in its product.

The Delphic 2123 modem retails at $380, with a handset and twin socket adaptor as optional extras. A 1200 to 75 baud adaptor will soon be available for use with micros that don't support split baud rates. Dealers are being appointed, and you can contact Delphic Digital at PO Box 1184, Carlton 3053.

Microtex 666

We have also been testing the hardware and software components of Computer Publications first "package deal" for prospective Microtex 666 subscribers.

The modem itself is the Commodore-compatible unit from Sendata. Plug one end into the 64's user port, the other into the phone socket and you're in business. You don't even need to worry about a power supply, as the modem, is fed by the 64.

The software has been specially written for Microtex 666, and does the job nicely. The program takes care of your identity number, but for security leaves you to type the password. Using the package as a regular videotex terminal is uneventful (as it should be), and software downloading is achieved at the press of a function key.

You can save screens into a memory buffer for later perusal, and the buffer may be stored in a disk file if necessary.

Since these functions involve the use of disk files, it's good to know that pressing f2 results in a directory listing. Another feature allows you to assign regularly used key sequences (like *666#) to single alphabetic keys.

All in all, a very straightforward system. Being disk-based, it's marginally less convenient to use than a cartridge, but firmware tends to be rather expensive when the time comes for an upgrade. You can give either of us a disk any time.

Incidentally, the package includes a simple terminal program so you can use the modem at 300 baud to access bulletin boards and similar systems.

Turning to the Microtex 666 telesoftware library, we find dozens of new programs. At this stage the team is still concentrating on the Commodore 64, with top-class games from Ozi Soft (I *still* haven't got that joystick — SJW) and other games and utility software from Sydcom (a users' group in Sydney) and other sources. There are also some serious sounding programs like Typing Practice and Finance Package, but we haven't examined them yet.

New bulletin boards

The many Commodore 64 owners in Melbourne now have a BBS dedicated to their interest. Alan Miles' C64-BBS is available round the clock providing the usual message service plus public domain software up/downloading and online sales.

Another Melbourne-based system is that operated by MicroPro Computers. While they don't charge users, they do require registration of name, address and phone number. In the meantime you can dial (03) 568 8180 and sign on as "visitor".

Looking overseas, we have just heard of two systems in Scotland, both (coincidentally) Atari based. The numbers are shown in the UK list under LSBBS and SABBS.

System Listings

Each entry shows the available information in this order: name, phone number, access control ("P" for public access and "M" for member, possibly with a "V" for visitor access), operator's name, operating times, and any special notes.

Systems outside Australasia are only listed if we have been informed that they are available to the public 24 hours per day.

BULLETIN BOARDS

Australian Systems

Micro Design Lab RCPM. (02) 663 0150. P. Stephen Jolly. 5pm–7am week days, 24 hours weekends.

MI Computer Club BBS. (02) 662 1686. MV Evan McHugh. 24 hours daily. Program downloading.

Sydney Public Access RCPM. (02) 808 3536. MV. Barrie Hall and David Simpson. 24 hours daily.

Prophet RBBS. (02) 628 7030. P. Larry Lewis. 24 hours daily.

TISHUG BBS. (02) 560 0926. MV. Shane Anderson. 7pm–7am weekdays, 24 hours weekends.

AUGABBS. (02) 451 6575. MV. Mathew Barnes and Andrew Riley. 24 hours daily.

AUSBOARD. (02) 955 377. P Daniel Moran. 24 hours daily.

Club-80 RTRS. (02) 332 2494. MV. Michael Cooper. 24 hours daily.

Omen I. (02) 498 2495. P. Ted Romer. 4.30pm–9am weekdays, 24 hours weekends.

Oracle.(02) 960 3641. P Rowan Evans. Midnight–8am weekdays, Midnight–6am weekends.

Infocentre. (02) 344 9511. MV. 24 hours daily.

Dick Smith Electronics RIBM. (02) 887 2276. P. Ian Lindquist. 24 hours daily. Program downloading.

Sorcerer Users Group RCPM. (02) 387 4439. MV. John Woolner. 6pm–8am weekdays, 24 hours weekends. Ring-back system.

Date BBS (02) 550 1004. MV. Steven Williams. 9am—11pm weekdays, 24 hours weekends. Computer dating.

Keeboard TBBS. (02) 631 3282. P. Philip Keegan. 6pm–8.30am daily.

RUNX Unix System (02) 487 2533. MV. Mark Webster. 24 hours daily. Call (02) 48 3831 for system status.

Tesseract RCPM. (02) 651 1404. MV. John Hastwell-Batten. 24 hours daily.

Tomorrowland's DIRECT. (02) 411 2053. MV. Mike Kidson. 24 hours daily. Helpline: (02) 412 3909.

Newcastle Microcomputer Club RCPM RBBS. (049) 68 5383. MV. Tony Nicholson. 5pm–8.30am weekdays, 24 hours weekends. RBBS free to all, RCPM for members only — $4/year to PO Box 293, Hamilton, NSW 2303.

Canberra RBBS. (062) 88 8318. 24 hours daily.

MICOM RCPM CBBS. (03) 762 5088. MV. Peter Jetson. 24 hours daily.

Melbourne PIE. (03) 878 6847. P. Len Gould. 24 hours daily.

Sorcerer Computer Users Association CBBS. (03) 434 3529. MV. David

Woodberry. 24 hours daily. Program downloading for members.

PC Connection IBBS. (03) 528 3750. Lloyd Borrett. 24 hours daily. IBM PC program downloading.

Omen IV. (03) 846 4034. Philip Westh. 24 hours daily.

Hisoft IBBM. (03) 799 2001. Richard Tolhurst. 24 hours daily. IBM PC program downloading.

Computers Galore IBBM. (03) 561 8497. Bob Cooban and Martin Scerri. 24 hours daily. IBM PC program downloading.

East-Ringwood RCPM. (03) 870 4623. Mick Stock. 4pm–midnight. Monday–Friday ONLY.

C64-BBS. (03) 489 4557. P. Alan Miles. 24 hours daily. Commodore 64 software up/downloading.

MicroPro BBS. (03) 568 8180. MV.

Gippsland RCPM. (051) 34 1563. Bob Sherlock. 24 hours daily.

Mail-Bus. (051) 27 7245. M. Max Moore. 24 hours. Membership virtually essential. Write to PO Box 234, Newborough, Vic 3825.

Software Tools RCPM. (07) 378 9530. Bill Bolton. 24 hours daily. CP/M, MS-DOS, Unix program downloading.

Adelaide Micro Users Group BBS. (08) 271 2043. MV. 9am–9pm weekdays, 10am–10pm weekends and public holidays.

Computer Ventures CBBS. (08) 255 1946. Daniel Schumacher. 24 hours daily.

The Electronic Oracle. (08) 260 6686. MV. Don Crago and Grayham Smith. 24 hours daily. Program downloading. Membership $35/year to 12 Brentwood Road, Flinders Park, SA 5025.

Omen II. (089) 27 4454. Terry O'Brien. 24 hours daily.

Outback RCPM. (089) 27 7111. Phil Sampson. 24 hours daily.

Omen III. (09) 279 8555. Greg Watkins. 24 hours daily.

New Zealand systems

NZ Micro Club RBBS. 0011 64 9 762 309. Chris Cotton. 24 hours daily. Software up/downloading. Type "help" to log in.

These listings are believed to be correct, but we welcome new information. Please mention whether you have first-hand knowledge of the systems you tell us about, or are simply passing on the information. Viatel users can send messages to Mailbox 063000030, users of RUNX and other systems on ACSnet can mail to stephenw@murdu, and letters may be sent to Steve Withers, C/- Computer Publications, 77 Glenhuntly Road, Elwood, Victoria 3184.

Overseas systems

North America

SYSTEM	NUMBER	NOTES
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
CLEO	0011 1 213 618 8800	Job vacancies
Mindstorm Network	0011 1 812 235 0908	Networked BBSs

EUROPE

SYSTEM	NUMBER	NOTES
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

SYSTEM	NUMBER	NOTES
BABBS Felixstowe	0011 44 394 276306	Apple Users' Group
BABBS TWO Basildon	0011 44 268 778956	Apple Users' Group
Blandford Board	0011 44 258 54494	
CABB	0011 44 631 3076	300/300 and 1200/75 baud
CBBS South West	0011 44 392 53116	
CBBS Surrey	0011 44 4862 25174	
Clinical Notes Online	0011 44 524 60399	
Computers Incorporated Newcastle	0011 44 207 543555	
LSBBS	0011 44 506 38526	
Liverpool Mailbox	0011 44 51 428 8924	
Manchester Open BBS	0011 44 61 736 8449	300/300 and 1200/75 baud
Microweb TBBS	0011 44 61 456 4157	BBC Micro
NBBS Birmingham	0011 44 827 288810	
SABBS	0011 44 698 884 804	
Stoke Information Technology Centre RCPM	0011 44 782 265 078	

Africa

SYSTEM	NUMBER	NOTES
Connection 80, Cape Town	0011 27 21 457 750	
TRShop, Cape Town	0011 27 21 5367	
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.

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	Videotex '85 Contact: Riddell Exhibitions, 135-141 Burnley Street, Richmond 3121 (03) 429 6088	July 16–18, 1985
Melbourne	5th *Australian Personal Computer* Show Contact: Australian Exhibition Services Pty Ltd Suite 3.2 Illoura Plaza, 424 St Kilda Road, Melbourne 3004 (03) 267 4500	July 17–20, 1985
Melbourne	Automach Australia '85 Contact: A Greco and Associates, PO Box 1399, Crows Nest 2065 (02) 439 4014	July 2–5, 1985

LAZING AROUND

Brain-teasers from J J Clessa

Quickie

No answers, no prizes.

Farmer Brown told his wife: 'I went to the stable this morning and there were the same number of heads and arms in there, as there were legs and tails.

What is the least number of horses and riders that there must have been in the stable?

Prize Puzzle

Probably the most well-known triangular number these days is 15 — the number of red balls used in snooker to give an equilateral triangle of balls with five balls in each side. On smaller snooker tables 10 balls are used, to give a triangle with four balls in each side.

Can you find the smallest three consecutive triangular numbers whose product is a perfect square greater than 1000? Answers, please on postcards only (letters are automatically disqualified) to *APC* Prize Puzzle, June 85 Lazing Around, 77 Glenhuntly Road, Elwood, Victoria 3184. Entries to arrive not later than 28 June 1985.

Prize Puzzle February 1985

A rather low response — about 20 entries — possibly because the problem was quite difficult. In fact, 20 per cent of the entrants had the wrong answer. There were four possible solutions — each with 22 digits:

a) 2 173 913 043 478 260 869 565
b) 4 347 826 086 956 521 739 130
c) 6 521 739 130 434 782 608 695
d) 8 695 652 173 913 043 478 260

You can make other solutions with 44,66,88 digits, and so on, by repeating the above sequences of digits.

The winner chosen at random from the 16 correct entries was G Hincksman of Mt Gravatt. Congratulations, your prize is on its way.

NUMBERS

Mike Mudge delves into Euler's Totient function.

Euler's Totient function

The great mathematician Leonhard Euler (1707-1789) had his dormant interest in number theory awakened by certain results of Pierre de Fermat (1601-1665). From 1747 to his death, the last thirteen years suffering total blindness, he made many valuable contributions in the field of number theory.

Theory and Definitions Euler's Totient function Ø(n), is defined to be the number of numbers not greater than n and prime to n. (That is, the number of numbers less than or equal to n and sharing no factor with n.)

n 1 2 3 4 5 6 7 8 9 10 11......50
Ø(n) 1 1 2 2 4 2 6 4 6 4 10......20

Nontotients are those positive values of n for which Ø(x)=n has no solution for example: 14,26,34,38.

Noncototients are those positive values of n for which x− (x)=n has no solution for example: 10,26,34,50.

Now we define f(n)=n−Ø(n) and observe that f(n) is less than n; thus if we iterate the function f to obtain f(f(...f(n)...)) we *must* eventually reach 1. For example: f(6)=6−2=4, f(4)=4−2=2, f(2)=2−1=1. Write s(k) to be the smallest integer which reaches 1 after k iterations.

k 2 3 4 5 6 7
s(k) 4 6 10 18 30 42
2.2 2.3 2.5 2.3.3 2.3.5 2.3.7

Question 1 Is there a pattern to the factorisation of s(k)? Is s(8)=2.3.5.7 or 2.3.3.5 or 2.3.11?

Question 2 (a) Are there infinitely many pairs of consecutive numbers n and n+1 such that Ø(n)=Ø(n+1)? For example: n=1,3,15,104.

Note that 18 solutions are known less than 10^4 and 59 less than 10^6.

(b) What about solutions of Ø(n)=Ø(n+1)=Øn+2)? (c) Consider Ø(n)=Ø(n+2)=Ø(n+4) (d) Similarly Ø(n)=Ø(n+3)=Ø(n+6) and so on.

Note that Schinzel has conjectured (1958) that Ø(n+k)=Ø(n) has an infinity of solutions for every k. However for k=3 only the solutions n=3 and n=5 are known.

Question 3 Determine the number N(y) of nontotients less than y as a function of y, extending the following table.

Y	10^3	10^4
N(y)	210	2627

Question 4 How many noncototients are there less than a given y?

Question 5 Is there a non-prime integer n, such that Ø(n) is a divisor of n-1?

Readers are invited to submit their program listings, output and hardware details together with their conclusions relating to some or all of the above questions to Mike Mudge, C/- *APC*, 77 Glenhuntly Road, Elwood, Victoria 3184.

A suitable prize will be awarded to the best entry received by 1 August 1985. Criteria will include accuracy, originality and efficiency, not necessarily in that order.

Please note that submissions can only be returned if a suitable stamped addressed envelope is included.

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
```

USER GROUPS INDEX

ATTENTION ALL USER GROUPS

A letter will be sent to all User Groups which appeared in the last full listing in December *APC*. We'd ask that Secretaries of User Groups who receive this letter update all details and return these to us no later than 14th June.

Please send your update to: User Group Index, *Australian Personal Computer*, 2nd Floor, 215 Clarence Street, Sydney 2000.

A full User Group listing will be published in the July issue of *APC*. Unfortunately the updates which were received since the last full listing, have been misplaced during the office move.

WRITING FOR APC

Your chance to contribute to the magazine.

We're offering readers a chance to get rich (well, at least richer) and to influence what's published in the magazine — by writing for it. We welcome approaches from would-be writers, including those who have never appeared in print before. It's often users with practical experience who have the most interesting things to say, so don't worry if your prose is less than perfect, we can take care of the polishing.

If you have an idea for a feature write, with a brief synopsis, outlining the proposed structure and content. If your article is already written, then send it in for consideration. Remember to put your name and address on both the covering letter and the manuscript — along with a daytime phone number if possible. Manuscripts should be typed or printed out (dot matrix output is fine), in double-line spacing with ample margins top and bottom and on each side.

Any accompanying program listings should be supplied on disk or cassette, ideally with a printout as well.

We'll try to return all submissions sent in with a suitable sae, but make sure you keep a copy of everything you submit as well.

Bear in mind that it's worth taking a look at the Back Issues advertisement to see what sort of things we have already published — after all there's no point in reinventing the wheel. And please be sure to tell us if you've contacted another magazine (perish the thought): it would be very awkward if the same article appeared elsewhere. Frankly, we're more likely to accept something which has been offered exclusively to us.

We do pay for published work — the rate is $100 per published page, and payment usually follows about four-six weeks after publication.

Finally, our apologies to readers who responded to our last request for submissions and who are still awaiting our reply. Every letter will receive our attention but, owing to the large number of submissions, it is taking some time to wade through them.

Computer Paper

IN MINI & MICRO PACKS AVAILABLE FROM LEADING COMPUTER STORES NOW

11 x 9½/70 WORD PROC. PAPER

W250 Pack — $7.85
W500 Pack — $15.39
W1000 Pack — $29.50
Also available in boxes of 2,000 & 2,500

A4 WORD PROC. PAPER

A4 250 Pack — $8.28
A4 500 Pack — $18.20
A4 1000 Pack — $35.80
Also available in boxes of 2000

11 x 15 PLAIN OR B.H.S.

LP 250 Pack — $8.45
LP 500 Pack — $16.60
LP 1000 Pack — $31.50
Also available in boxes of 2,500

COMPUTER ADDRESS LABELS

37 x 102 — 2000 Labels — $35.00
24 x 89 — 2000 Labels — $23.15

Also available in boxes of 10,000

COMPUTER BINDERS

11 x 9½ — $3.60
11 x 15 — $3.65

(Also pre printed STD inv/stat *formats. All prices include S.T. — Plus packing & postage)*

COMPUTER PRINT & PAPER CO. PTY. LTD.

For Quality Computer Paper Look For This Label

PHONE (03) 584 5488

DEALER ENQUIRIES WELCOME

96B Herald Street, Cheltenham 3192

All prices R.R.P.

VZ-200/300 C-16/PLUS-4 AMSTRAD

We urgently need programs for the above computers and pay a generous 25% royalty to market your software. Please submit your programs to us or write for full terms.

The following are some of the top quality programs we already have available for the VZ-200/300:

$7.00 Speed and Accuracy — This program tests your typing speed and accuracy. Words appear on the screen and you have to type them out as fast and as accurately as possible. A great way to improve those typing skills.

$9.00 The Lost Sword — The land of Erindor is slowly being destroyed by the evil sorcerer Feamor. You have been chosen by the king to recover the lost sword, the only hope of destroying him. (Requires 16K RAM expansion on VZ-200).

$8.00 Lemonade Stand — A great game for any future entrepreneur. How much money can you earn running a lemonade stand? This educational game requires strategy and pre-thought and ensures many hours of fun.

$8.00 UFO'S — Can you save earth from the waves of attacking aliens before you run out of fuel? This game features four waves of increasingly difficult-to-shoot-down aliens, a high score facility, graphics, sound and has a joystick or keyboard option.

We also send a free periodical newsletter to anyone on our mailing list full of helpful programming hints. Write now for a complete catalogue (specify computer).

Trade Enquiries Welcome

LYSCO P.O. Box 265, Bunbury, W.A. 6230

Talk to the World with . . .

Relay

Communication for IBM PC, Wang PC.

— Background mode
— Unattended mode
— Full range of protocols

Kowhai Systems
165 QUEENS ROAD, CONNELLS POINT
N.S.W. 2221 PHONE: (02) 546 6499

NEW!

Paper Tamer

PRINTER STANDS

2 MODELS AVAILABLE

PT1 — Suits all Matrix Printers using 210mm (8¼") A4 paper.
PT2 — Suits all 136 col. Matrix Printers using 380mm (15") wide paper.

● "Paper Tamer" saves space and restores order to your desk top micro-centre. ● Neatly stores up to 900 sheets continuous paper under the printer. ● Allows perfect input paper feed and collects out-put efficiently in special tray. ● Manufactured from **sturdy** white colour coated steel rod. ● Excellent value for money.
NEW in Australia NOW available direct from Supplier

LinTek *Computer Accessories*
23/243 Ernest St., Nth Sydney NSW 2060 (02) 929 3217

QTY	SIZE	PRICE inc freight	
	80 col. suits 210mm (8¼") paper	$49.95	
	136 col. suits 380mm (15") paper	$59.95	
		TOTAL	

Name ....
Address ....
☐ CHEQUE ☐ MONEY ORDER ☐ BANKCARD
☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐ Expiry Date ....
Signature ....
If I am not fully satisfied, I may return goods within 30 Days for full refund

CAS007LA

DEALER ENQUIRIES WELCOME

CHIP CHAT

We've had a hard time convincing some sceptics that last month's exclusive antique-look cover was a carefully planned subtle comment on Commodore's bog standard PC clone.

It's completely unfair to suggest the cover was a balls-up, a mistake, a mutant from the printer's laser scanner.

It cost a fortune to produce something that looked so old, so faded, in so short a period of time.

What was that? You don't believe it! But it's true . . . no, really.

We weren't going to publish this after we said we'd given up embarrassing other magazines for kami-kaze style attacks on APC (ChipChat, January 1985 APC) and threatening to publish the Morgan readership figures on computer magazines (APC 225,000; YC 128,000; TC 77,000), but we're now desperately thrashing around to prove that other magazines can make bludners too; so we have to compromise ourselves by publishing this early 1985 cover shot (from a magazine that purports to make micros more understandable) of a confused-looking young lady. There's free 12 months' subscriptions in the offing to the first ten readers to write in and tell us why she's so confused.

Not to be outdone by Guy Kewney, ChipChat is proud to present an exclusive preview of the new Bushlitt micro. Developed by Paul Hardy and his dedicated team at Bushcat, the system went into limited field testing on 1 April. Remember, you saw it here first.

Bushcat has come up with a set of extra Basic extensions and constructs which are intended to increase the speed and accuracy of the Bushlitt's Basic.

REPEAT . . . UNTIL CORRECT: a very powerful construct. Execution repeats until the right answer is generated.

IF x CLOSE ENOUGH TO y: causes premature exit from the above if the answer is almost right.

DON'T CRASH: a rather more powerful version of the classic ON ERROR GOTO . . .

DON'T DO . . . UNTIL: omits a section of code until a condition has been met.

DON'T DO . . . AT ALL: a special version of the above, where execution speed is all-important.

FUDGE: a system command which can be used at the head of a program to tack on bits here and there where it looks as though there might be a zero-divide on the way, or something else that might lead to the wrong answer.

IF SHBT . . . THEN TOT . . . :IF should have been this THEN tack on that. For experts. Requires prior knowledge of the answer. While in its simplest form it behaves very much like a simple assignment, its full power is realised if FUDGE has been previously specified.

The Bushlitt uses the Bushcat Virtual Time Operating System, the next-generation operating system. It does for time management what the virtual machine does for memory management. It has immense multi-tasking capability, where execution speed remains constant independent of the number of jobs in the system. BVTOS makes use of the fourth dimension (time) to achieve this. The user submits jobs, and the operating system carefully schedules them to maintain the optimum run time. To the user, all jobs appear to be running at once. In fact, the operating system has subtly rescheduled them and some of them may be running in the middle of next week.

Windows are, of course, an integral part of the operating system. The windowing software was written by Bushcat's French subsidiary.

The Bushlitt uses a local area network (LAN) which, while restricted to fairly short distances requires virtually no installation and retains the portability of the micro. With the Bushlitt LAN, the micro can quite happily be moved from desk to desk without extensive rewiring.

This Bushlitt ACOULINK (for ACOUstic LINK) plugs into the micro. It consists of a loudspeaker, microphone, and communications protocol software on ROM. The protocol includes message collision detection and the automatic retransmission of garbled data.

For example, user 1 wishes to transfer file to user 2:
User 1 types
> SEND filname. ext TO micro 2

System action:
"HELLO HELLO MICRO2 ARE YOU THERE?"
(this repeats, getting louder each time, until there is a reply)
"YES I'M HERE."
"ABOUT TO SEND filname. type"
"OK"
"LINE TEN CLEAR SCREEN COLON DEFINT A TO Z"
"OK LINE TEN CLEAR SCREEN COLON DEFINT A TO Z"
"LINE TWENTY INPUT QUOTE DO YOU WANT INSTRUCTIONS QUOTE SEMI-COLON N STRING"
"OK LINE TWENTY INPUT QUOTE DO YOU WANT INSTRUCTIONS QUOTE SEMI-COLON N STRING"
(and so on).

The system includes error correction by both transmitting and receiving micros:
"NO NO I SAID" (micro 1)
or
"HEY?" (micro 2)

With some of the longer-range LANs, Bushcat does, of course, provide earplugs.

Bushcat can also provide mass-storage devices suitable for any market. One of its Far Eastern clients required a system that could operate in a hot, humid climate, miles away from the nearest DS catalogue, to run a small business package for his string factory. While mass-storage was important, it would not be possible to obtain floppy disks.

Bushcat came up with the knotted string reader and writer, this used the client's own string surplus as the storage medium. An elegant solution.

And with each system purchased in June, Bushcat is giving away a free Swiss Army Knife with a cyclist remover: just right for those embarrassing occasions when you have been out shopping and return home to find a cyclist embedded in the radiator grille. Leave it in the glove box and it's ready for use.

ADVERTISERS INDEX

A
Acme Software 19
ACT .... OBC
Activision 81
AED IFC,1
Applied Technology 52,59
Archives 178
Arcom Pacific 4,7,105,113
Attache Software 88
Australian Exhibition Services 10
AWA Thorn 45
Award Software 66

B
Barmet 198
Barson 14,15,39
Bayne and Trembath 164
Bayside Computer Systems 191
Benson Computers 76
Brainstorm 27,128
Brother Industries 40,108

C
Calypso Jane 169
Cash and Carry 51,IBC
Caulfield Business Computers 196
Centre Industries 121
Cerebral Solutions 136
Chendai 205
Commandglen 115
Commodore 131
Compak 186,187,188
Compudat 58
Computer Consultants 41
Computer Club 10 198
Computer-K 198
Computer Print and Paper 206
Computer Warehouse 78
CPM 205
Cybex Computing 194

D
Data Parts 66,138,139
Data Power 41
Datacare 205
Datascape 112
Datatran 93
Dick Smith 125,140
Discware 143

E
EAI 116
East-West Electronics 92
Electrical Equipment 196
Electromark 104
Entercom 25
Epsom 110
Ericsson 177
Esselte Dymo 116

G
Gametronics 192
Geeban 104
Godfrey Deane 201

H
Hanimex 114
Hi Tech 133
High Technology 192
Hills Computer Service 31

I
Ian Huntley 142
IBM 98
ILC Systems 75
Imagineering 12
Info Line 53
Intelligence 119
Interface Technology 56.175
Interfaceware 17

J
John Sands/Sega 32

K
KCM 173
Kent Design 200
Kowhai Systems 206

L
Lintek 206
Logi-Com 54
Lysco 206

M
Mac Designs 141
Magna Data 198
Marque Computing 185
Maxwell Office Equipment 198
Megatron Computers 164
Memorex 94,95
MF Computers 82
Micro Educational 183
Micromail 63,83
Micro Station 122
Microtex 666 22,23,34,35
Minicomp 79
Moore Paragon 180

N
Natwick 96
New Dimension 130
Nucleus 200

O
Optsoft 101
Osborne 72,135,172,202,203
OziSoft 30

P
Penguin Books 43
Personal Computer Services 33
Peter Fox Computers 196
Porchester Computers 103
President Computers 179

Q
Qubie 28,29,64

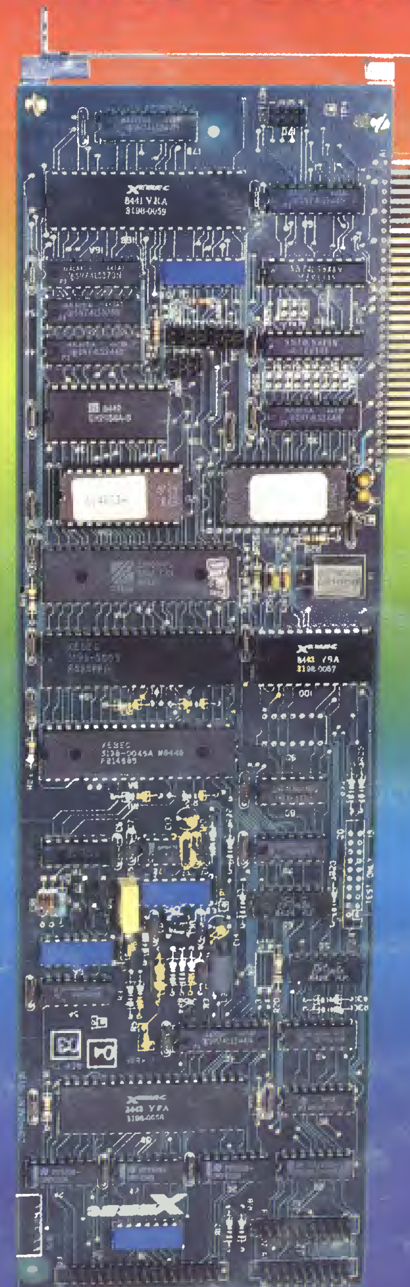
R
Ram Supply 204
Robs Computer Centre 70,71

S
Scarlan 54,120
Silicon Craft 198
Sirimath 43
Software Source 144
Software Specialists 104
Software Supplies 9
Spacestation 208
Speedit 43

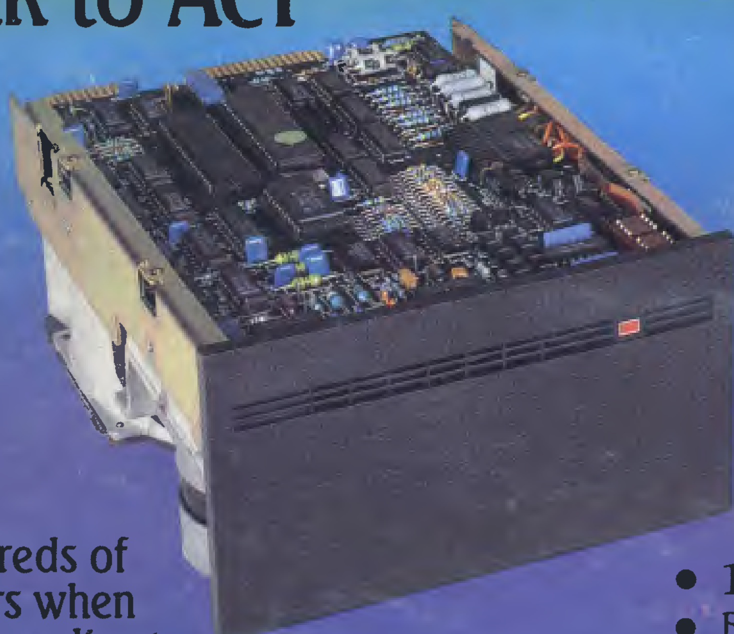
T
Tedita 198
Telecorp 36
Termicom 178
Texas Instruments 48
The Computer Spot 169
The Computer Trader 67
The Logic Shop 80
Thinking Systems 38
Top of the Chart 194
Toshiba 41

U
Unique Micro Design 178
Unlimited Computer Solutions 205
Utilico 171

An affordable Winchester for your IBM



If you want to talk Winchesters and Controllers... Talk to ACT



SAVE hundreds of dollars when you buy direct from the wholesaler!

FROM $1595

(Includes all taxes)

- 10-31 megabytes
- Boots directly to Winchester
- 1 year warranty
- Don't wait months, buy ex-stock
- ACT is an Australian organisation committed to backup support and service

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