Australian Personal Computer

SINCLAIR'S COLOUR MICRO BENCHTESTED

JUNE 1982 $2.50* NZ $3.00*

AUSTRALIA'S TOP SELLING COMPUTER MAGAZINE

FROM THE MAN WHO GAVE YOU PET...

Sirius: The 3rd Generation Micro

inside

Editors
Sean Howard
Graeme Domm

Technical Editor
Ian Davies

Production Manager
Mike Northcott

Advertising
Gerard Kohne
(03) 818 7508

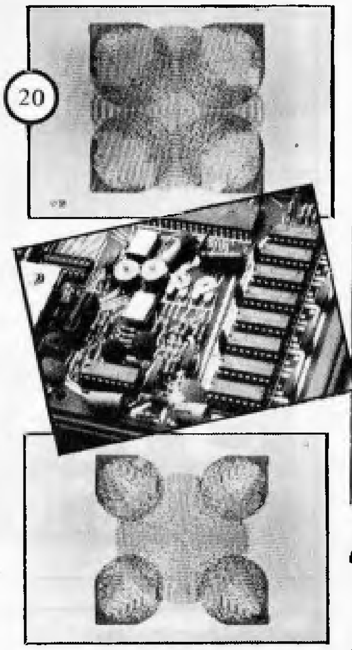
Distribution Manager
Valerie Meagher

Typesetting
Marcia Brown

Printed by
Lewis Printing
500 Clayton Road,
Clayton, Vic.

Produced under licence from Felden Productions.

Published by Howard Productions,
462 Burwood Road, Hawthorn, Vic. 3122,
Telephone (03) 818 1488, Telex: AA 30333 AMJ



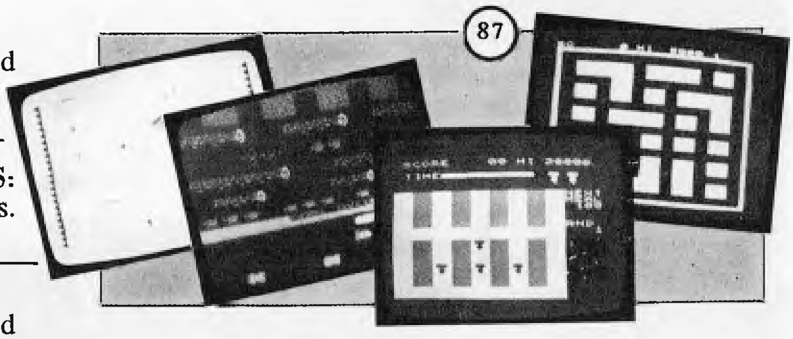
BENCHTESTS

FEATURES AND SERIES

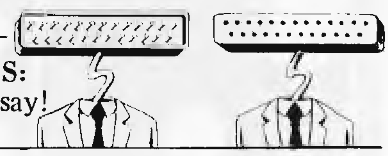
APC

Vol. 3 Number 6 June, 1982.

Registered for posting as a Publication Category B. ISSN 0725-4415 *Recommended Retail Price only.



REGULARS



PRINTOUT

Miriam Cosic and associates report on the latest news from the world micro scene.

New Commodores

After sitting quietly letting the world speculate and shake its collective head in wonderment, Commodore has suddenly leapt in with a new range of computers, both up- and down-market from its existing machines.

Most spectacular is the 720, a smart new business machine which not only looks radically different from anything else on the market but has some very exciting features. Like: 256k of RAM (a 128k model is on its way), 20k of ROM, an integral 80 x 25 display, a full qwerty keyboard with 10 programmable function keys, four cursor controls (at last!) and a numeric pad, twin built-in floppy disk drives (also at last), cassette RS232, IEEE 488 and user ports and – most interesting of all – a slot for a second processor; 8088 and Z80 add-ons are also promised.

Slightly down-range is the 500 series, which looks quite similar except that there's no integral screen or disk drives. The unit plugs into a TV to give a 40 x 25 colour text display or 320 x 200 hi-res colour graphics, there's a music synthesiser with a 90 octave range and three voices, 20k of ROM and 64k RAM (expandable to 256k) and the unit also has a second processor slot.

Both machines come with Basic in ROM and, with the addition of a Z80, both will run CP/M software, with the main processor, a 6509 (a 6502 with memory paging) handling I/O.

Three new home machines have been announced: the VIC 10, VIC 30 and the long-rumoured Commodore 64.

The two VIC models feature similar basic specs (40 x 25 colour text or 320 x 200 graphics, ROM Basic, and sound), but the Basics differ in that the 10 has a plug-in ROM pack Basic which can be replaced with games or music cartridges while the 30 has an 8k Basic buried in 20k of ROM, which includes the operating system and character generator. They have different RAM sizes, too: 2k for the 10 and 16k for the 30.

The Commodore 64 has the same graphics and sound capabilities of the VICs but has a full 64k of RAM in addition to the 20k Basic and operating system in ROM and also has a second processor slot. In addition, it can be persuaded to run software written for other 40-column Commodore machines.

With the new machines, Commodore now has a very large product range. Although there's no official word from CBM at the moment, it rather looks as though the ageing remnants of the PET line will fade in the near future (i.e. as soon as the existing stocks have been sold), to be replaced by the 500 and 700 series.

Commodore's smooth 720

Centuries of computing

Abacus has a number of new products, offering under the ubiquitous Century label which is being slapped on most of its micros.

The most interesting product is a micro floppy disk from Sony which measures 3½ inches internal diameter and gives 300k in single sided format.

A new range of micros utilising this product was officially released at the NCC in America; and according to Theo Sapountzis at Abacus, the release at the Southern California Business show had them standing six deep.

Abacus has three new computers. The Courier, a Personal Computer, costs $3995 plus tax, and bears a slight visual resemblance to the Osborne. A 4mHz Z80A runs processing and disk control, while a separate 8085 controls the screen and keyboard. Two of the micro floppies provide 600k of memory, and there is 64k on board. The inbuilt screen measures 9 inches, more comfortable than Osborne's 5 inches. On the 8-slot STD bus, three of the slots are already used, a serial port goes into printer mode, and another will accommodate a controller for 8 inch and 5¼ inch drives as well as the micro floppy drives. It all closes into a neat carrying case. The machine will run CP/M, Wordstar, Supercalc, Microsoft Basic and Pascal.

The Diplomat is a desk top version of the Courier. It has a 12 inch screen, and a slot for a 5¼ inch Winchester. The extra $500 cost essentially buys greater operator comfort.

The third product is the Century 8800, which is a cash register and computer in one. It is essentially a computer with a cash till, Theo said, and has the same processing power and storage capacity of the Courier. It looks like a pretty effective $8,000 solution for the shopkeeper looking to computerise, the greatest advantage being that data from sales registering does not have to be re-entered into a computer. Once a sale is rung up, the information automatically resides in memory. The keyboard can be configured to user specifications.

Abacus is at 512 Bridge Road, Richmond, Vic. 3121; Tel: (03) 429 5844.

Moving in . . .

People describe Japanese machines in Japan as very cheap but lacking in software. Over here, they still aren't even cheap – but with Mitsubishi's latest idea, maybe they are starting to get some software, because the thing comes complete with a robot.

The robot is the 'Move Master', a robot arm which can move all over the place, under control of a CP/M microcomputer.

Apart from having the CP/M software, the micro also has to have something called a 'semiCentronics interface' which is a little less than the normal parallel wires that control Centronics-type printers but can be connected to a normal Centronics 'outlet' on a micro. So, to make the robot work you need the half-Centronics software to send the right signals down the right half of the wires.

Cost of Move Master and control processor is $5000. Mitsubishi hopes the thing will be used 'for training operators, as a teaching aid, or as a point of sale display' (a shop window gimmick, I suppose).

PRINTOUT

With a little luck, it will also prod lethargic US micro builders into realising what they will be up against in a year or so, when Japanese machines do start spilling over into the West. But I doubt it: they will carry on making 'hand-crafted quality machines'.

Details of Move Master from the company's Tokyo offices at 2-3, Marunouchi 2-chome Chiyoda-ku, Tokyo 100, Japan — or phone (218) 2173.

Satellite centre

Business is obviously doing very well by Damar Management Services — it is opening a new showroom, called the Computer Centre, at 5 Yarra Street, Geelong.

It is a dedicated computer house, providing hardware, computer supplies, and a strong team of software people writing customised systems.

The 2,000 sq. ft showroom will provide hands-on demonstrations of computers, catering for hobbyist to businessman, including Sinclair, the BBC Computer, Cromemco and the full range of HP computers and calculators, soon including the 3000 series minis.

Dave MacFarlane said that he thinks they will be one of the few houses providing calculators through minis, backed up with a team of seven software writers. The hobbyist, as well as the businessman with biccies to spend, will be welcome to browse.

Apple Logo

Just because a language is easy to learn doesn't make it easy to write programs in that language — as a great many Basic programmers are starting to find out. Which accounts for the number of people who are getting enthusiastic about Logo, a teaching language which actually encourages people to write good programs, and to learn good programming habits.

Latest to get enthusiastic about Logo is Apple itself, which has announced an official Apple Logo. Hopefully it is a good version because there are so many different people around, all offering their own versions, that it is going to be very difficult to know which one to recommend — but the Apple version is almost certain to become the standard. Just because Apple is behind it, probably.

Three new 16-bitters

Three 16-bit micros were recently released, one of which looks as though it could ally with the Sirius 1 to give IBM a tough time when the Jolly Giant finally gets round to launching this side of the Pacific (and the latest estimate for that is an early '83 launch).

Office equipment giant Olivetti finally launched its personal computer, the M20.

Instead of first testing the water with a conventional 8-bit machine, Olivetti has produced a 16-bit micro with 128k of RAM and its very own operating system.

The M20 is virtually unique in the micro world in being based around Zilog's Z8000 processor, a decision made back in the early days of the 16-bit chips when the Z8000 was the

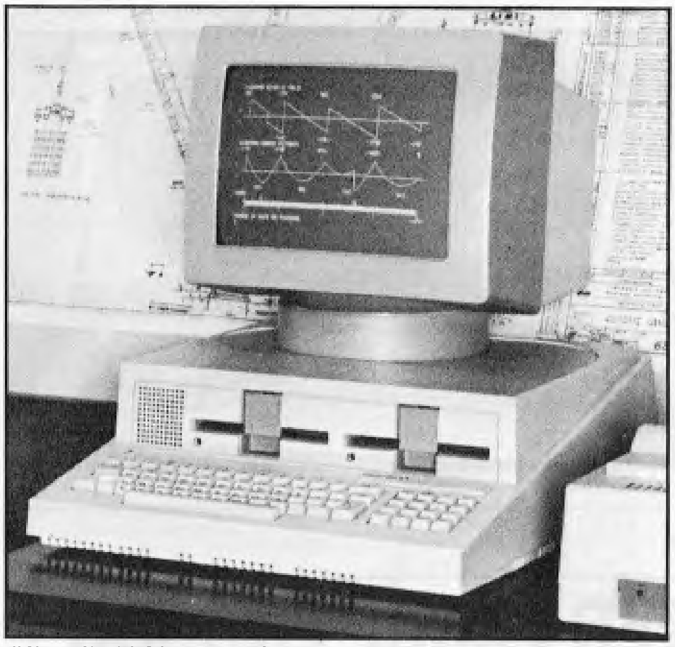
The Move Master — see 'Moving in . . .'

only real product available and looked all set to win the race.

As well as the Z8000 and the 128k RAM, the M20 has as standard Centronics and RS232 interfaces and an 80 x 25 display (software switchable to 64 x 16) with 512 x 256 graphics, and twin 5¼in floppy drives holding a total of 286k (formatted) each.

Upgrades and expansions include memory expansion to 224k by three plug-in 32k cards, IEEE 488 and twin serial port boards, and very nice colour graphics.

Olivetti's operating system, called PCOS, is a single-task, single-user setup



Olivetti's 16-bit contender

Corvus with full page screen.

PRINTOUT

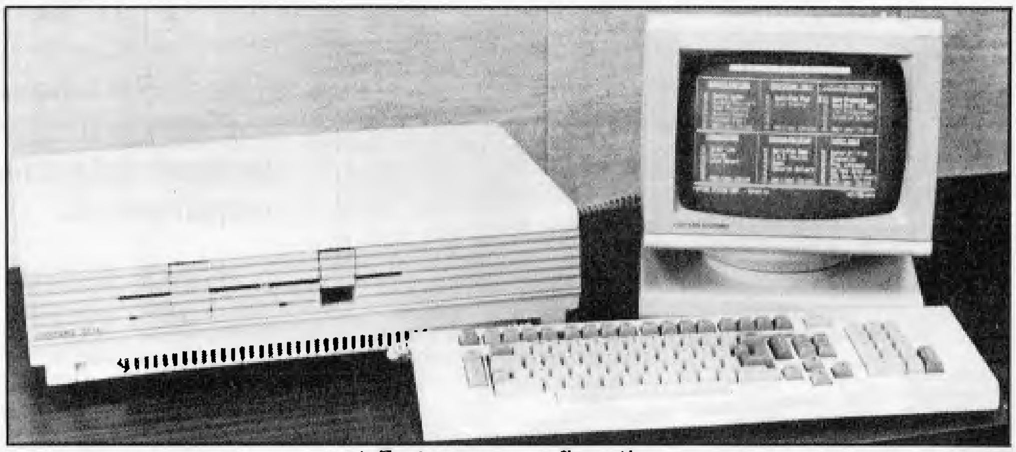
which, from the documentation, appears rather similar in concept to CP/M but is more flexible and friendlier – it incorporates a 'help' command, for example – and includes password protection facilities for files. The Basic used is more-or-less standard Microsoft, with commands to handle the quite sophisticated display, which includes windows and full graphics control.

A set of diagnostic routines is contained in ROM and is automatically executed on power-up, with any malfunctions being reported on the screen or printer. A disk containing more sophisticated diagnostics is also available.

All this is, of course, useless without some good applications software. The deviation from the now almost-standard CP/M family means that the fast-growing pool of 16-bit software won't run on the M20. Recognising this, Olivetti has produced a number of its own packages, including a word processor called Oliword, a 'data entry, retrieval and update package' called Olientry and a spreadsheet package called – no, not Olicalc but Multiplan. A number of utilities are also available, including a scientific subroutines library and a system self-instruction guide. A choice of three printers is available, one thermal and two dot matrix, and, of course, you can hook up one of those stylish Olivetti daisywheel units too.

Rather more up-market is the Fortune 32:16. As the name suggests, it's based on the Motorola 68000 chip, which looks like a 16-bit machine on the outside but has 32-bit internal architecture. With the basic unit you get 128k internal RAM plus a single 800k floppy and a single-user operating system based on Unix but with a rather neat and considerably more user-friendly shell around it which sensibly keeps the user isolated from the nasties of Unix itself. A twin floppy unit and integral hard disk (5Mb or 10Mb) are also both available. Upgrade to a multi-user operating system and you can share the system with other users by adding work stations.

Surprisingly (especially as



A Fortune at your fingertips

the basic machine sells in the US for $4995), no applications software is included in the price. So you'll pay extra for what looked at the demo to be a rather nice word processing package, a spreadsheet and a database system called IDOL.

Another 68k-based micro launched recently also caters for more than one user, but does it with a network rather than splitting the CPU between users. The Corvus Concept is in fact a very upmarket workstation designed to hook in to the Corvus Omninet local area networking system, giving communications with other machines and with printers and hard disks. 'Workstation' isn't really doing the machine justice actually – it's a fully-blown Unix machine just bursting with really interesting features, the most immediately obvious of which is the rotating screen. Well 'rotating' isn't quite the word – it's a two-person job – but in essence, you use the machine with the screen in an upright position (see photo) to give a full A4 display (72 lines of 80 characters) for word processing; if you want to do accounts-type work, for which a very wide display is usually a great help, you simply lift the screen unit up, turn it on its side and put it back after flicking a switch on the back to tell the computer what you've done – the machine then reconfigures itself to the new screen format – 56 lines of 120 chars.

A close look at the photo will show that the Concept has no disk drives. As it's a network-oriented machine, you need to interface it to a Corvus disk drive. Putting more computers onto the network is easy, though, and – impressively – you're not just confined to adding more Concepts; the Apple II can interface to Omninet to give a low-cost terminal the full power of the M68000, and Chuck Peddle recently announced that the Sirius 1 will have an Omninet capability soon.

Software for the Concept will include Pascal (ISO with full UCSD extensions), Fortran 77, the obligatory word processor and spreadsheet and a CP/M emulator to allow you to run Basic and Cobol.

And there's a *very* interesting hardware feature inside the Concept – a bus into which a large range of Appke II cards can be plugged, although you will have to re-write the ROM drivers on the cards . . .

Personal Cromemco

Norman Rosenbaum looked like the cat that got the cream, when I spoke to him at CETIA. Informative Systems is extremely proud of the new 16-bitter that Cromemco has given birth to. By October, it will have full networking capabilities.

I was meditating on the AJA's Code of Ethics when Norman told me (and almost everyone else at CETIA) about a new super-confidential product due here in October. Fortunately, the product was released at the NCC, and embargo lifted in time for this.

Cromemco, that doyen of reliable machinery, with products that are quite expensive but, really, you get quality for money, etc, etc, have released a personal computer which, in a business package will cost the hefty sum of $1,800.

The C10 Personal Computer has a Z80A processor, 64k of on-board RAM, 16k internal ROM, 12 inch green screen, detached keyboard, floppy drives, and a letter quality daisy wheel printer.

The C10 costs $995, the keyboard $195, 5 inch drives $595, and printer $895. The package contains the C10, keyboard, floppy drives, CP/M, 32k structured Basic, word processing and spreadsheet analysis software, all exchangeable for your personal cheque for $1,800.

The C10 will also act as a mode in the networking system, and as a stand-alone is fully upgradeable through the Cromemco range right up through to the 32-bit series.

Informative Systems is already taking orders. Contact them at 337 Moray Street, South Melbourne 3205; Tel: (03) 690 2899.

Aussie action

One company that won't be whingeing about the recession

Microline

SERIES OF
DOT MATRIX PRINTERS

Microline 80

80 CPS UNIDIRECTIONAL 80 COLUMN

STANDARD FEATURES

- 40, 80, 132 columns
- 6 and 8 lines per inch
- 200,000,000 character head life
- Centronics interface
- Graphics
- Continuous operation
- Plain paper - up to 3 parts
- Paper tear bar
- Friction and pin feed
- Upper and lower case

$625 including Tax.

Microline 82A*

120 CPS BIDIRECTIONAL 80 COLUMN

STANDARD FEATURES

- Centronics and RS232C interface
- Upper and lower case with true descenders
- 40, 80, 132 columns
- 6 and 8 lines per inch
- Rear/bottom paper path entry
- Graphics
- Plain paper - up to 4 parts
- Multi-national character sets
- Paper tear bar
- Friction and pin feed

Microline 83A*

120 CPS BIDIRECTIONAL 136 COLUMN

STANDARD FEATURES

- Friction and tractor feed
- Centronics and RS232C interface
- Upper and lower case with true descenders
- 6 and 8 lines per inch
- Rear/bottom paper path entry
- Graphics
- Plain paper - up to 4 parts
- Paper tear bar
- Double width and condensed characters

bankcard welcome here

*CALL FOR PRICES ON OTHERS

DISCOUNT FOR QUANTITY

COMP-SOFT MICROCOMPUTER SERVICES

233 — 235 SWAN STREET, RICHMOND, VICTORIA, 3121. TELEPHONE (03) 428-5269

is local software house, Integrity Management Services. Following on the announcement that Olivetti has selected IMS business software packages to go with its new M20 desk top computers, is the announcement that the ICL personal computer range will be offered with IMS software. It will be carried by ICL dealers in Australia, NZ, Singapore, Malaysia and the Pacific, and will be available to American users through IMS subsidiary, American Integrity Systems in the US. Doesn't it make your patriotic heart throb.

Modular marvel

Modular Information Systems, Australian subsidiary of the UK company, has opened in Australia to market a new system.

The processor is a Z80B running at 6mHz, and a separate processor runs the printer and VDU. The 16-bit architecture will access from 512k to 4Mb of core storage. 64k of RAM is devoted to application programs. Hard disk will take capacity to 32Mb.

According to Miro Philippi, marketing consultant to the new company, the system represents true state of the art technology (have I heard that somewhere before?), is truly user-friendly, and is subject to rigorous quality control before shipment.

Although the system was developed in the UK, the models, both hardware and software, have been redesigned for the Australian environment. It is fully modular and is configured according to user specifications. Because of the modularity, repair is simply carried out by replacement of the offending part. Support is fully local and, according to Mr Philippi, the company already has eight support staff and is still recruiting.

The system was first released to the national photographers body, with a photographers package which was locally written. Apparently, it won an UAP National Merit Award, and has been officially recommended to Australian photographers. There you are.

As an intelligent terminal the computer starts at the $3,000 mark.

Games people play

Just as toy trucks and dolls can help children learn the language of the people and traffic they will have to cope with as adults, playing with video games today may prepare the adults of the next two decades to react with computers.

Not my own idea, but that of someone called Victor Walling, one of a fascinating clutch of academics unearthed by Atari, in a publicity handout designed to show how useful games are.

I hope that some of the academics have been misrepresented – they read like a 'Pseud's Corner' anthology in all too many cases.

For instance, there is one who is quoted as knowing of 'no hard evidence that violence from blowing up a Space Invader would transfer to normal, day-to-day interactions with other people'. Violence? Ah well, probably some strange new usage of the word . . .

Then there is the psychologist who uses video games as a means of treating depression in teenagers. A fascinating report summarised in Atari's publication, but a bit confusing to find that 'he also plans to use the game as a reward for good behaviour' for some of the aggressive boys who have behaviour problems – he feels that 'the game provides a particularly good reward, compared with jukeboxes and snacks'.

Others are disappointing in their lack of results, presumably because the studies are being turned into somebody's thesis. For example, the chief of the Stress Analysis Research Unit at the Aviation Physiology lab in Oklahoma City, has been testing people – about 25 of them – with bad hangovers.

He's lucky to be alive, really: he tested these unfortunates on Atari games to check coordination and response – and I know what sort of response I would have for a doctor if I complained of a severe hangover, and he asked me to play Asteroids. 'Apparently the study evolved from problems with alcohol use among pilots,' notes Atari, unnervingly.

The really interesting thing about it all is that Atari has donated games and cartridges for research purposes, to most of these researchers who had approached the firm for help in studying these areas. Nice to see encouragement of academic effort, yes – but very revealing. It becomes clear that there is starting to be considerable pressure on Atari from a lobby of people who think that games encourage crime, and train children in violence.

Myself, I agree with the researcher who said that this chain of 'logic' was really a lot of sloppy thinking. But throughout the Atari report, it is clear that many of the researchers quoted assume that they have to somehow justify games, against a rising consensus that they are harmful.

One researcher said that 'the negative response of some of the public to video games may stem from some sort of "technological phobia" but video games are less likely to bring out violent responses than TV'.

Another said in an equally defensive tone that 'the games are absorbing, much like music, but they're not addictive in the technical sense of the word'.

I didn't know anybody thought they *were* addictive in the 'technical sense of the word.

Micro DEClaration

It's certainly all happening. After listening to mainframe people pour scorn on personal computers for more than a decade, it's interesting to watch their hardware manufacturers scramble for the mighty micro market.

The respectable DEC is to have a personal counterpart. The Rainbow 100 series, also appearing here in 1983, are business-oriented personal computers in the $4,000 to $6,000 price range.

DEC is following the IBM/ICL approach of playing it safe with industry standards rather than pioneering technology. The new machine will have dual, 8 and 16-bit processing and run CP/M. DEC has already produced a glossy catalogue of programs available for CP/M.

Anadex printers

Anadex has released a large range of impact printers suitable for the industrial market.

Sold by its Australian representative, Datascape, these printers are suited for use in many applications such as ticket printing, weighing and medical.

The various models are:

The DP-750A alphanumeric drum printer with 21 columns and up to 42 alphanumeric characters with parallel and serial interface.

The FP Series of OEM and stand-alone form and ticket printers in both alphanumeric and numeric format, which are capable of printing through several layers of paper and on to card stock. Mechanisms only are available for OEM applications.

The DP-650 and 500 are numeric drum printers for 21 or 18 columns with or without clock and are serial, parallel or DTL/TTL compatible.

On the doorstep

Computer Imports has set up shop vitually on Tandy's door step by opening its new computer centre at 220 Morphett St. in Adelaide. When company founder Brook Hill saw Tandy open there in March the temptation was apparently too much to resist and so he moved quickly to acquire premises and open up in less than three weeks.

Commodore National Sales Manager, Roger Davis, who was in Adelaide for a Commodore Sales program and the opening of Computer Imports' new premises, is delighted with having Commodore's products virtually under Tandy's nose. Computer Imports handles both Tandy and Commodore equipment. One product to which the company attributes its success has been the new "Silicon Office" program which runs on the new 8096 series Commodore with 96k of RAM and 1Mb of disk storage.

"This package", says Hill, "essentially incorporates all the requirements of a modern office such as invoicing, stock control, control of debtors etc., word processing, mailing etc. in one easily modifyable package".

"We can now setup a suite of software packages to suit a small business in a matter of days when previously we would need to look at between 0.25 and one man years of programmer's time at twenty five to thirty five dollars an hour".

"Essentially, this package enables us to suit the software to the existing methods used in the business".

Texas talkie

Texas Instruments devotes a lot of effort and funds to research on synthesised speech; the results are to be seen in its speech chips and in products such as Speak and Spell and Speak and Math. These last products give vocally aided instruction, using ROM based vocabularies; in the case of Speak and Spell you can buy further ROM packs to enlarge the repertoire but the range is still rather limited. TI's latest offering, announced at the end of May, overcomes this limitation by allowing a speech synthesiser to read software encoded as bar-codes on the printed page.

Called the Magic Wand Speaking Reader, the unit comprises a box of tricks with an on/off switch as its only control, and a hand held bar-code reader which uses fibre-optics. This is scanned over a high-density bar-code in one of the books which will be sold for the device, and the result is a spoken sentence.

Currently the books are designed for pre-school age children and have pictures and written words over the bar-codes. Eventually, however, TI hopes to apply the system to language teaching for adults as well. The synthesis system uses principles recently developed in TI's labs, which break down English speech into 128 'allophones'; these are encoded and stored in ROM and can then be used in combination to build up any word. The high-density bar-code used by Magic-Wand allows a certain amount of inflection to be encoded into the speech, so that the result is less robot-like than previous efforts; the book I tried even had a singing voice ('Twinkle, Twinkle Little Star'). The allophone system makes it possible to synthesise other languages by analysing them into a suitable allophone set.

Tax time

With a press release that would make any happily married or feminist person twitch, K.C. Moorley (I presume that's the vendor) has announced the availability of a personal tax program for the Atari and Apple.

See Texas talkie.

PRINTOUT

It apparently offers prompts for each item to ensure that nothing is over-looked, a running summary of entries, a full calculation of tax payable and refund or balance payable, a menu drive, summaries of all sections called from the menu, printer control, user adjustable tax scales for amendment in future years; and its available on 16k cassette for Atari or 24k diskette for the Apple II.

Dealers and real people can ring (02) 520 9470.

Apple en masse

A new mass storage system for the Apple II is now around. The Apple III ProFile Personal Mass-Storage System (who thinks up these imaginative names?) is a self-contained unit containing an intelligent controller, a 5¼ inch Winchester-technology disk drive, a power supply, an interface card and driver software. It increases on line storage to 5Mb.

Seven new software packages take advantage of the thing. They are Apple III Pascal, Apple Business Basic, Visicalc III, Apple Writer III, Access III, Script III, and Business Graphics III.

The price of the storage system is $4,500; and bought with the Apple III, it comes to $6,990.

Cheap Conversion

The costs of converting Olympia ES100 typewriters into printer/terminals has been greatly reduced by a manufacturer, SME Systems.

SME, which has designed conversion boards for this purpose, has been manufacturing them for the past two years. Previously the boards had only been available through a tied wholesaler arrangement. Now SME is offering the boards, with manual and fitting instructions, directly to the public, and through its distributors, for $245 (plus tax).

Mike Pratt, managing director of SME, said his company's conversion hardware and software had made the Olympia a highly versatile typewriter.

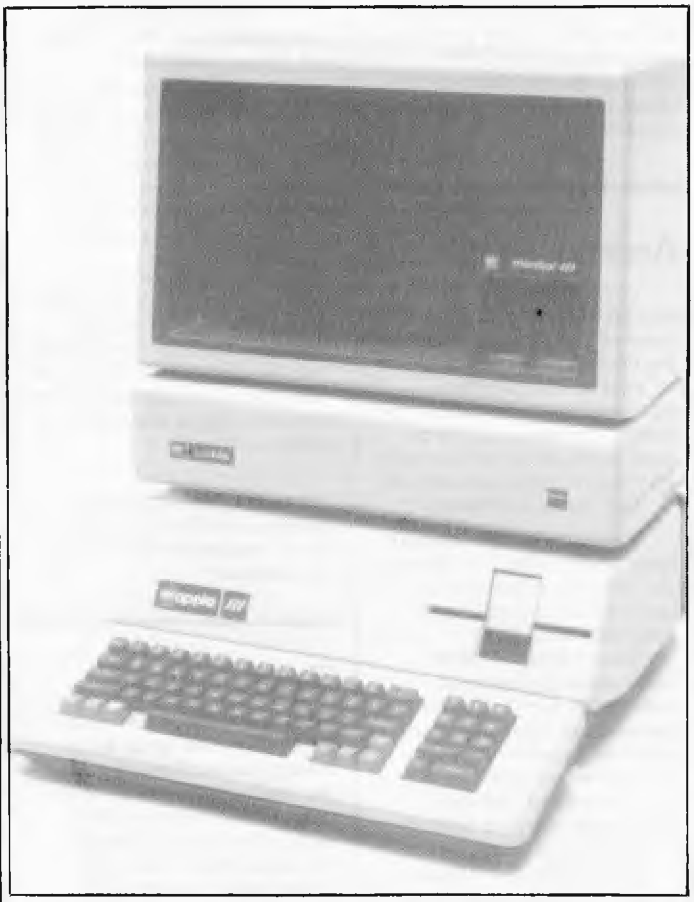
"With the board, the Olympia ES100 is also a word processing quality printer and, a stand-alone hard copy computer terminal. In both functions it is compatible with any machine with RS232C communication protocols.

"As a terminal it can communicate via modem with virtually any computer system, a handy tool for the professional and small businessman who finds it difficult to justify the expense of a "dumb terminal" that may only be used infrequently," Mr Pratt added.

When not directly communicating with a computer system, the typewriter can still be used as a typewriter.

Mr Pratt claimed that "virtually" any typewriter owner could fit the board in less than 20 minutes, although his company was happy to do the installation for an extra $20.

For further trade information contact SME Systems, 22 Queen Street, Mitcham, 3132; Tel: (03) 874 3666.



The Apple III personal computer with new Profile mass storage system.

The hand has writ . . .

Victoria's State Insurance Office has placed an order for seven Micropad hand-writing data entry devices, with a possible further order of 70 Micropads, depending on the success of the initial order.

The Micropad, produced by Quest Automation, enables hand-written data to be captured by a computer at the time of writing. The SIO will use it in the Employers' Liability Department for the registration of workers' compensation claims.

"We carried out an extensive study on our procedures before making our decision, and we see Micropad providing a streamlined approach to our operation," an organisation spokesman said.

Each Micropad, consisting of a pressure sensitive writing surface, a microprocessor and a 40 character line display, will be connected to the SIO's Burroughs mainframe computer through a converter to be supplied by Progeni Systems.

The Micropad hand-writing data entry device.

The Latest One

Inspired, no doubt, by The Last One, a Michigan City firm called Advanced Operating Systems has launched a $500 program called the Programmer. It runs on the Apple II CP/M version, and the IBM PC, and an ordinary Apple version is planned. And yes, it does everything, too.

Details from Kenneth Jones, at 450 St John Road, Michigan City, IN 46360.

Southern awareness

Victoria's Information Technology Week is being extended to a month this year.

Joint sponsors, the Department of Science and Technology and the ACS, have

decided that a week isn't long enough.

"Computer month" will officially start on August 3 and offers exhibitions, free seminars, classes and films. The culmination will be the old Information Technology Week, to be held in Collins Place from August 30 to September 3, which will run displays of micros, videotex and other advanced technology stuff, and lunchtime movies and classes.

Osborne land

Richard Graham, previously of Computerland in Australia, is the man heading up the new Osborne Computer Corporation Australia. Earlier in the year he spent a week at Osborne in the US being fully trained into all aspects of the operation, and then a week in the UK subsidiary learning just how Osborne likes to run its subsidiaries.

Back in Australia Richard has set up the local company in a two-story building with 5,500 sq ft of warehouse on the ground floor, and marketing, administration and software support upstairs. So far, ten people have been employed within the organisation. They include as technical manager, Murray Baker previously at Computerland (oh, yes, how interesting), and Phil Miles, also ex-Computerland (hmm!), as marketing manager. 35 dealers have been appointed.

Padmede accounting software has just been developed for the Osborne, with each module retailing for $370, including tax. Richard said that he believes the price of software should be compatible with the price of the machine, so is busy providing programs for the Osborne which will enhance the point of buying a cheap machine.

Data 82 in August will see the release of a double density disk option which will provide 200k per drive for $300, and an 80 column option.

For those that like their VDU at an orthodox size, there is a monitor adapter, previously costing $65, which will allow other brand monitors to be interfaced. It is now part of the package price.

Wiser Women

With the cost of software soaring under the impetus of government duties, it's terrific to get some good news for a change.

Linda Graham and Heather Miles in Sydney have started a new company, called Wiser Microsoft, which has a full licensing arrangement with Microsoft, with full rights to the source code, they will be duplicating all Microsoft's lines, including products for the Apple, Osborne, Xerox 820, Superbrain, Tandy and IBM-compatible systems, in Australia. On the hardware side, they will also be handling Microsoft RAM cards for the Apple and IBM type systems.

Apart from being cheaper (I was quoted 5% off pre-duty prices, so you're looking at up to 40% off post-duty), the products will be warranty-supported in the country. Wiser Microsoft will be carrying on Microsoft's policy of providing updates for clients, whereby for the cost of an update fee, software will be brought to current status and all manuals provided.

Among new products on the way is Multiplan, an electronic worksheet operating in colour and 80 column width accompanied by a tutorial and a reference manual. It is the beginning of a series of 28 packages which will hook into this first package to provide templates for special applications, i.e. real estate, insurance, and such.

Until Wiser Microsoft moves into new premises, Linda or Heather can be contacted through Wiser Microsoft, P.O. Box 95, Forrestville, NSW 2089.

Business school

Adler has released a teaching aid for schools, business colleges and companies. It is a course called Business Computer Processing, and runs on Adler's Alphatronic Micro.

It is designed to familiarise the student with using a computer to process the full range of day to day transactions encountered in a business environment. It has been structured for self-paced individual study, and uses a method where the student processes realistic docements of a fictitious business over a simulated three month period.

The package is based on three teaching texts (covering debtors, invoicing and stock; creditors and general ledger; and payroll) by Doug Young, former head of the Business Studies department at Devonport Technical College. It has already attracted the support of several large companies.

Contact Adler Business Machines, cnr Lane Cove and Waterloo Roads, North Ryde, NSW 2113; Tel: (02) 888 7644.

High speed CP/M

General Electronic Developments, of Sydney, claim to be one of Australia's most experienced micro-computer manufacturers. One of the first to introduce a CP/M computer back in 1977, the company has produced more than 200 systems, most of which have been marketed through OEMs. But now G.E.D. has decided to market its latest system direct to end users.

Called the System 85, the product is a single board computer which has several unusual features. For example, the internal terminal, instead of interfacing with the CPU via an I/O port (the usual method), is integrated into the CPU itself. This gives System 85 the ability to handle accounting and inventory software (for instance) at speeds as much as 200% higher than those achieved by competitive units, G.E.D. claims.

For word-processing, there is an additional terminal with a specially-designed word-processing keyboard. Despite the provision of two terminals, G.E.D. says the price of a System 85, complete with software, is in the low range for conventional business computers.

To help users of the new System to select appropriate CP/M software, G.E.D. is making available selected packages from the several hundred which the company has evaluated. These cover most normal business functions, plus many special and technical functions.

Micro Wang

Wang has a new computer called the Wang Professional Computer, It is built on Intel's 8086 processor with 16-bit architecture and internal and external data paths.

The basic system has 128k of operating memory, 320k diskettes, keyboard, Microsoft's DOS and Basic and will run under CP/M 80. The new product is designed as a stand-alone computer, or will act as a work station on Wang's network systems.

Australia should see it next year.

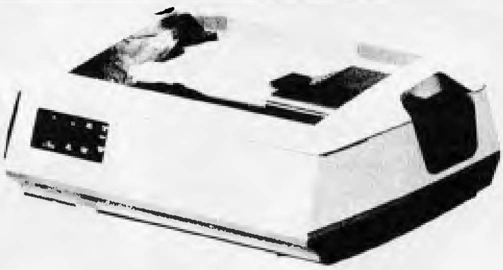
Extra graphics

Provided you have expanded a VIC-20 with an extra 3k memory, you can use GraphVICS to draw high resolution pictures. The package provides a new area of memory in which a screen is held, and the user can look at either the normal VIC memory picture, which is usually used for text, or the expanded GraphVICS area.

The product is released by Michigan software producer Abacus Software, P.O. Box 7211, Grand Rapids, Michigan, 49510, Tel: (616) 241 5510.

DEFOREST SOFTWARE

26 Station Street, Nunawading 3131 **Phone: (03) 877 6946**



EPSON MX-80III

- Bit image printing
- Full 96 ASCII with descenders, 8 International character sets
- 32 Print fonts standard
- Bidirectional printing with logic seeking (text mode)
- Versatile interface options
- 3 way paper handling (F/T type)

$880 + TAX

MANNESMANN TALLY

- 2 print modes
- Draft 200 cps — 7 x 9 character font
- Correspondence 50 cps — 40 x 18 character font
- Paper slew rate — 8 ips
- Wide carriage 15.25 Inches
- Versatile interface options

$2990 + TAX

EPSON MX-100III

- 100 cps
- Wide carriage 15"
- Bit image printing
- 32 Print fonts standard
- Full 96 ASCII with descenders, 8 International character sets
- Corresponding quality printing
- Versatile interface options

$1180 + TAX

CP/M SOFTWARE

American Training International, Inc.,
-*ATI-POWER FOR CP/M, requires CP/M 2.2
-MENU-POWER-operator training wordstar booklet only
AUSTRALIAN SOFTWARE
IMS Version -
-ACCOUNTS PAYABLE
-ACCOUNTS RECEIVABLE
-GENERAL LEDGER
-PAYROLL
-ORDER ENTRY INVOICING
IMS Version 3
-ACCOUNTS PAYABLE
-ACCOUNTS RECEIVABLE
-GENERAL LEDGER
-DRDER ENTRY INVOICING
Practical Computers
-PRACTICAL MAIL MANAGER
-INVENTORY MANAGER

Artificial Intelligence (Requires CBASIC 2)
-MEDICAL
-DENTAL
Ashton-Tate (data base management system)
-dBASE II
-DEMO DISK AND MANUAL
Byrom Software
-BSTAM -micro to micro comm.
-BSTMS -micro to mainfram comm.
Centa Systems
-CP/M NORTHSTAR -horizon only
-CP/M MICROPOLIS
Computer Controi
-*FABS
-ULTRASORT II
Computer Development Inc.
-IMAGE- word processor with graphics
Condor Computer Corp.
-*CONODR I-entry level data base mngmt.system
-*CONDOR II-relational dBASE system
-CONDDR III-extensive report writer
-CONDOR 20-R -report writer can ad to Condor 1 or 11
-CDNDOR 28-Q -indexing allow quick access selected data
Designer Software
-*PLANTIR -WORD PROCESSING
Data Most
-WRITE ON -word processor
Digital Research
-*MAC -macro assembler
-*SID -symbolic debugger for 8880
-*ZSID-symbolic debugger for Z/80
-*TEX -tex formatter
-*DESPOOL -background print ulty
-*PL/I-80 -language
-*BT-80 -record retrieval system
-CP/M 2.2 INTEL MDS -808
-*CB 80 -a true compiler for CRASIC 2
-*C8ASIC -language
-*PASCAL MT+ version 5.5. language requires 56k
-*COMPILER ONLY
-*SPP ONLY -speed programming package
-RMAC, LINKLIB; LIB AND XREF -8" only
-LINK-80, PLILIB,LIB AND XREF -8" only
-XLT86
For the DEC VT-180 (5 1/4" DD)
-PL/1-80 version 1.3
-CBASIC version 2.8
-CB-80 version 1.2
-PASCAL MT+ version 5.5
Ecsoft
-*MICROSTAT-file oriented advanced statistics package
Epic Computer Copr.
-*SUPERVYZ -operating systems expertise via menus
Faircom
-*MICRO B+ -keyed file accessing
Finanacial Planning Association
-*MINI MODEL
Fox and Geller Associates Inc,
-*QUICKSCREEN -screen builder
-*d-UTIL -dBASE II Utility
Freinds Software
-*LEVEL I -report generator/cross tabulator
-*LEVEL II-output and logic processor
Frontier Software
-PROFESSIONAL TIME ACCODNTING
-FAM -fixed asset management
-FAM/ACRS -depriciation, tax, investment, credit etc.
-APPLICATION DEVELOPMENT UTILITIES
-GENERAL SUBROUTINE PACKAGE
Graham Dorian
-APARTMENT MANAGEMENT
-CASH REGISTER
-JOB COSTING
I.S.A.
-SPELLGUARD
-SP LAW
International Software Marketing
-MATHEMAGIC
-ITHACA INTERSYSTEMS
Key Bits
-WORDSEARCH -spell check/dictionary
-STRING 80 -string utility
-STRING 88 -source
-STRING BIT -fortran string handling
Lexisoft Inc.
-*SPELLBINDER -wordprocessor
Mark of the Unicorn
-MINCE -text editor
-SCRIBBLE -formatter
-BOTH
Micro Ap
-SELECTOR III -C2
-SELCETOR IV -data base manager
-SELECTOR V
-GLECTOR III -G/L option for selector III
-GLECTOR IV -G/L option for selector IV
-GLECTDR V -G/L option for selector V
-*S-BASIC -language
Micro Focus
-*CIS COBOL
-*FORMS 2
Micro Pro
-WORDSTAR TRAINING GUIDE
-WORDSTAR -word processor
-MAIL MERG
-WORD STAR/MAIL MERG
-DATA STAR -data entry, ret. update
-DATA STAR CUSTOMIZATION NOTES
-WORDMASTER -text editor
-SUPERSORT I -sort merge
-WORDSTAR CUSTOMIZATIDN NOTES
-SPELLSTAR -spell check/dict.
-CALCSTAR -electronic spread sheet
Microsoft
-BASIC 80
-BASIC COMPILER
-FORTRAN 80 -language
-COBOL 88 -language
-MACRO 80 -macro assembler
-EDIT 80 -line editor
-MU MATH/MU SIMP -symbolic math package
-MU LISP/MU STAR -language
-M-SORT -sort
Microatuf
-*CROSSTALK -data communication
Microtech Exports Inc.
-REFORMATTER
Nexus
-*ZIP -screen and printout formwriter
Northwest Analytical
-*STATPAK
Oaais
-*The Word -spelling checker 45,008 words
Organic Software
-*TEXTWRITER III -text formatter
-*DATEBOOK -appointment scheduling
-*MILESTONE -critical path analysis
Pickles Trout
-CP/M
Reding Group
-*LYNX -microsoft compatible linker
Select Information Systems
-SELECT WITH SUPERSPELL -word proceasor
Soho Group
-MATCHMAKER
-WORKSHEET
Sorcim
-*PASCAL/M -z80 or 8080 language
-PASCAL/M -8886 or 8088 language
-*ACT 80 -cross assembler, z80, 8088 8085
-*ACT 65 -cross assembler, 6502
-*ACT 68 -cross assembler, 6800
-*ACT 69 -cross assembler, 6889
-*ACT 86/88 -cross assembler,8086, 8088
-*TRANS 86 -translator
-*SUPERCALC -CP/M based visicalc
Southern Computers
-*RAID -Interactive debug/emulator
-*RAID FP/S -raid with floating point pkg + sorce code
-*RAID FP/O -raid with floating point pkg + object code
-*FPP -with source code/floating point package
-*FPP -with object code
-*RECOVER -lost data recovery
-ISIS -translator
THE TRIG - PACK -sine,cosine,logs, etc.
-*Relocatable Dbject Code
-*With Source Code
Standard Micro Systems
-*QUIC-N-EASI -business applications development system
-*MAILMAN -mail list manager
Standard Software
-THE PROTECTOR -encode,decode encyption software
Star Computer Systems
-LEGAL TIMEKEEPING, BILLING AND ACCDUNTING
-PROPERTY MANAOEMENT
Structured Systems
-ANALYST
-LETTERIGHT
-NAD
-QSORT
Supersoft
-ENCODE/DECODE II
-*DIAGNDSTIC II -hardware checker
-*FORTH -specify z80 or 8080 -language
-*SSS FORTRAN -language
-*RATFDR -language enchancer fortran
-*SUPER M-LIST -mail list program
-*TERM I -micro to micro comm.
-TERM II -micro to mainframe
-TINY PASCAL -language
-DISK DOCTOR -lost data recovery
-UTILITIES I -general purpose program
-UTILITIES II
-C COMPILER -language
-STAR EDITOR -text editor
-Z-8000 -cross assembler
-NEMESIS
-DUNGEON MASTER
-ANALIZA II -requires C-BASIC 2
-*ADA -compiler
-*SCRATCH PAD -electronic spread sheet
-*DATA-VIEW -electronic filing cabinet
-*STATS-GRAPH -statistical display with graph formats
-SCRATCH PAD, STATS GRAPH DATA VIEW
System Plus
-FMS 80 II
-FMS 80
Woolf Software Systems
-*MOVE IT -micro to micro comm

*-AVAILABLE FOR APPLE II (REQUIRES SOFTCARD)

IBM PERSONAL COMPUTER
American Training International Inc.
-ATI-POWER FOR PC DOS teaches you about PC DOS
-ATI-POWERS FOR CP/M -86

I.S.A.
-SPELLGDARD
-SP LAW
Sorcim
-SUPERCALC

CP/M 86 DISPLAYWRITER
Digital Research
-CP/M 86
-C BASIC 86
I.S.A.
-SPELLGUARD -comes with baisc word dictionary

CP/M 86
Digital Research
-PASCAL MT+ does not include SPP
-SPP only
Sorcim
-SUPERCALC

CAPS SHIFT
SYMBOL SHIFT
BREAK
SPACE
YELLOW
WHITE
GRAPHICS
BLACK
DELETE

BENCHTEST
PERSONAL COMPUTER

SINCLAIR ZX SPECTRUM

Headed "Sinclair ZX Spectrum" this Benchtest is a preview of a machine that will never be marketed in Australia under that name. The "Spectrum" (we'll continue to call it that, despite the existence of Australia's own "Spectrum" from DD Webster, until Sinclair Equipment (Aust.) advise us of the machine's Australian name) is available now in the UK. It will soon be marketed in the USA followed by the 'also-rans' including Australia. While we have to wait longer, when it finally does arrive later this year or early in '83, it will be accompanied by a wealth of software from the UK and USA. David Tebbutt looks at a production prototype.

Once again Sinclair offers value for money with a vengeance – colour, sound and graphics for an expected retail price in Australia of less than $400. Two versions of the Spectrum are being manufactured: 16k and 48k. Upgrading from 16 to 48k costs sixty pounds in the UK and should be around $200 in Australia.

Like most computers at this level, the Spectrum plugs into the domestic television and uses a normal cassette recorder. The ZX printer can be attached and, with a little modification, ZX81 programs will run happily on the Spectrum. Communications facilities, an RS232 interface and a miniature disk drive (the ZX Microdrive) will be announced later on. The good news is that the Microdrive should sell in England for fifty pounds. Translate that to dollars and it sounds like a tidy package.

Hardware

The Spectrum measures just 233x144x 30mm and weighs in at 520 grams excluding the separate power supply and cables. It looks extremely elegant and, unlike its predecessors, it has keys that actually press down.

You'll not be surprised to learn that there are hardly any components inside the machine: 14 chips, a UHF modulator, a piezo-electric 'speaker' and an assortment of capacitors, resistors, diodes, crystals and a coil make up the complement. I swear that some of my crystal sets had more in them. All this is mounted on a single board and, looking underneath this production prototype, I notice that there's not a single patch. The only odd thing about it is that there's a big blob of green plasticine stuck around the coil. The coil on the review machine does whistle a bit but I understand that production machines come with suitably lacquered coils to eliminate this problem.

A hefty edge connector at the back brings out just about every signal you could wish to have. This is used for printers, communications and disk drive connections. Inside there are two spare sockets which accommodate each end of the 32k memory expansion board. This is a great improvement on the ZX81 memory expansion which tended to drop off the back of the machine at the least provocation. Talking of sockets (well I was, just now) every chip except the ULA is socketed. The reason the ULA isn't is because it gets very hot – putting it on the PCB allows the heat to dissipate better.

The keyboard comprises a one-piece grey rubber moulding mounted over a pressure-sensitive membrane. The keys poke up through holes in a black metal plate and I must confess the feel is more that of a calculator than a typewriter. Most keytops have three symbols on them and, in addition, most of them have another two associated inscriptions printed on the metal surround. If you're anything like me you'll find yourself reading the whole keyboard each time you want to find a function. You do get used to it after a while; in my case it took a couple of days. I found that red symbols on grey keytops are quite difficult to read and, thinking my eyesight might be going, I showed the machine to a number of friends, all of whom had the same difficulty. I showed it to my 11-year-old and he thought it was just fine though.

A power supply is included in the price, so there's not a lot of point risking one of your own and blowing the Spectrum up. The two cassette leads terminate in 3.5mm jack plugs so be sure that they work with your recorder before you embark on any major programs. It took me four or five tries before I found the right volume setting on my tape recorder. Once this was found, though, program loading presented no problems.

I tried the Spectrum on three televisions and the results matched the quality of the sets used. The display comprises 24 lines of 32 characters with the bottom two lines reserved for messages and entries. The display can also be regarded as 176 x 256 resolution for graphics work. High resolution graphics work is best done in two colours as you will see in the Firmware section of this review. The screen, border and individual characters can each take on one of eight colours and, in addition to this, characters can be bright or flashing. Other screen attributes like inverse and overprinting relate to the whole screen. More on these later.

The single channel BEEP facility is about what you'd expect from a piezo-electric speaker. It does sound slightly better amplified from the cassette port but it's still pretty awful. A couple of octaves around middle C aren't bad; but the other eight are best used for sound effects. At the high end they warble and at the low end they grate – BEEP is a refreshingly honest description.

Really, there's not a lot more to say about the hardware. It is a very professional job; looks smart, works well and manages to squeeze 191 legends on to just 40 keys!

Firmware

Here's a new section for *APC* Benchtests. All the software on the review machine was in the ROM chip which also contained the character set. This time Sinclair has gone for a basic ASCII set (upper and lower case) with the addition of both built-in and user-defined graphics characters. Outside of the range SPACE to QUOTES (32 to

126), many of the codes have special values relating to Spectrum keys and functions. For example, you'll find a copyright symbol key. (Now why didn't anyone else think of that?) You can define up to 21 characters of your own.

Two screen tables are maintained in memory – one for the displayed characters themselves and the other for the attributes which describe how they're to be displayed. These attributes can be tested from within a Basic program. The character colour is referred to as INK while the background colour is called PAPER. Isn't that sensible? Each character can have its own value for INK, PAPER, FLASHING, BRIGHTNESS, INVERSE and OVER. The last two should be explained: INVERSE simply means that the dots which form the character are printed in the PAPER colour while the PAPER is printed in the INK colour. OVER is special: it allows you to merge a new character with the one already at the screen position. The rules are that two INKs or two PAPERs print PAPER otherwise it prints INK. This means that you have a neat way of removing the last thing printed and restoring what was there before it.

By now you have probably realised why it is best to stick to two colours when doing graphics work. Since the colour of the INK and PAPER relates to a whole character position, then each time a new colour graphics point is set, all other set points within the boundary of that character are set to the new colour. This makes for a very curious effect to say the least.

Mathematical accuracy is to 9½ decimal digits and a fairly full range of mathematical functions is accessible from the keyboard. While on the subject of keyboards, this one has a built-in software 'click', an upper-case lock key and automatically repeating keys. Like the ZX80 and ZX81 before it, the Spectrum makes great use of single stroke keyword entries. In fact, I think every standard function and command is obtainable in this way. You'll even find things like > – and < – occupying their own pieces of grey rubber.

Basic

The Spectrum comes with a very useful version of Basic. It will be quite familiar to anyone who is used to the Microsoft types of Basic and is easy to learn for those new to the language.

Rather than go through all the features and functions of the language, I have summarised them in a separate box. Here I'll just comment on the unusual and interesting aspects of this particular implementation. Unlike some Basics, it is a bit strict about things like using LET before assigning a value to a variable name or putting GOTO after a THEN. My view is that this is good discipline and is more than compensated for by the fact that Spectrum pops in all those spaces which make programs so much easier to read. Of course, once you've found your way round the keyboard, the single stroke keyword entry is a joy. (I've got a feeling I said that in my last two ZX reviews.)

SAVEing and LOADing cassette tapes gives plenty of scope on this machine. You can save a program normally, you can save it so that execution starts automatically when it is reloaded, you can save arrays, you can save particular chunks of memory and if you want to keep a pretty picture you've created then you can use the SCREEN$ option to save that too. All saved programs can be verified after saving. The screen save can't be verified because the display is changed during the verify program and it would not then match that held on tape. The LOAD command can, of course, handle any tape created by SAVE. The MERGE command allows you to merge a program on tape with one already in memory. Program lines which are duplicated are overwritten while all others are suitably interleaved.

The graphics facilities are great fun. You can draw straight lines, curves and circles on the 176x256 pixel (PICture ELement, or dot) window. Position 0,0 is at the bottom left-hand corner of the screen. You can define up to 21 graphics characters of your own which is a superb feature if you're into writing your own Space Invader or Pac-Man games. I had a lot of fun drawing and animating little people on the screen. The nice thing is that you can do all this sort of thing without leaving Basic. A BIN (binary) notation has been introduced to allow you to define numbers as a series of 0s and 1s – just the thing for designing funny characters. Each character comprises eight lines of eight points, so a succession of eight BIN numbers is all you need to define such a character. Another use for user-defined graphics is to squeeze some extra colours out of the machine. If you lay out the 64 pixels like a chess board and choose suitable INK and PAPER colours then you can get some interesting effects.

You won't be surprised to learn that line drawing and circle plotting are achieved using the DRAW and CIRCLE commands. A PLOT command allows you to plot single points. POINT enables you to find out whether a particular pixel is set. You always DRAW from where Spectrum thinks you are on the screen. For example, a command DRAW 10,10,pi would draw a semicircle ending up 10 places to the right

What about the '81?

It was no secret that Sinclair was going to launch another cheap micro – he's put a bomb under the industry twice already, producing machines which brought computing power within everybody's reach at prices which drastically undercut the competition. A slightly upmarket (by Sinclair standards) machine offering colour and sound and reasonable graphics at a price comparable to the VIC-20 is the latest addition to Clive Sinclair's range.

What is interesting, though, is that the Spectrum does not replace the ZX81, as the 81 did the 80 – it's an addition to the range and the ZX81 will continue in production. In fact, production of the 81 is to be *increased* to a target of 150,000 a month by the end of the year.

'The ZX81 will continue to be ideal for the person who wants the lowest possible entry cost into computing', says Sinclair.

Spectrum Basic

Functions

ABS	ACS	AND	ASN	ATN	ATTR	BIN	CHR$	CODE
COS	EXP	FN	IN	INKEY$	INT	LEN	LN	NOT
OR	PEEK	PI	POINT	RND	SCREEN$	SGN	SIN	SQR
STR$	TAN	USR	VAL	VAL$				

Operations

+ – * / = > < <= >= <>

Statements

BEEP	DATA	GO TO	LOAD. .DATA	OVER	RETURN
BORDER	DEF FN	IF. .THEN	LOAD. .CODE	PAPER	RUN
BRIGHT	DELETE	INK	LOAD. .SCREEN$	PAUSE	SAVE
CAT	DIM	INPUT	LPRINT	PLOT	SAVE. .LINE
CIRCLE	DRAW	INVERSE	MERGE	POKE	SAVE. .DATA
CLEAR	ERASE	LET	MOVE	PRINT	SAVE. .CODE
CLOSE	FLASH	LIST	NEW	RANDOMIZE	SAVE. .SCREEN$
CLS	FOR. .TO. .STEP		NEXT	READ	STOP
CONTINUE	FORMAT	LLIST	OPEN #	REM	VERIFY
COPY	GOSUB	LOAD	OUT	RESTORE	

SINCLAIR ZX SPECTRUM

and 10 above the current position. A fraction of pi would provide a different arc while zero, or no third argument (DRAW 10, 10) would draw a straight line. The curve can be drawn on either side of the centre line by making the third argument a positive or negative number.

The CIRCLE command used three arguments: x-axis, y-axis and radius. Remember, the OVER command can be used to erase something already drawn. I used this feature in conjunction with DRAW, PLOT and CIRCLE to create cartoon effects. OVER is also useful for embedding text in a drawing. When set on, the text merges with the existing lines in the drawing. When set off, it prints the full 8x8 character, completely replacing anything already displayed at that position. Incidentally, SCREEN$ can be used to return details of the contents of a character position. Used in conjunction with the PRINT AT command, this could be a good way of making your program find a suitable place to print a sort of 'label' on a drawing. The AT allows you to define the row and column at which printing should start.

A few instructions I particularly noticed as I went through the manual were READ, DATA, RESTORE and VAL$. READ and DATA are old friends although I can't remember them being on previous ZX machines. Using the DATA command you can provide lists of information at the beginning of a program. Each READ instruction takes the next word from this list. RESTORE can be used to set the DATA pointer to any DATA statement. VAL$ baffles me – it strips the outside quotes from string expressions and returns the string value of the result. Perhaps some kind reader would care to suggest a worthwhile application for this feature.

Now let's have a look at our honestly named friend, BEEP. There's not a lot to tell, really, except that you can control both pitch and duration. Notes below middle C are represented by negative numbers, those above by positive. Twelve numbers make an octave. (If you look at a piano keyboard you'll find that there are seven white notes and five black notes per octave.) Middle C is zero. The duration is expressed in seconds or fractions of a second. As I mentioned earlier, the sound isn't brilliant but it has the saving grace of being fairly quiet. You can pick this sound up from the cassette ports if you so wish. I'd say these facilities are more likely to be used for sound effects than composing symphonies.

That's really all I have to say about the Basic. It is a very good implementation for a machine of this size. A PAINT instruction would have been nice to fill in graphics shapes, but I think it would look a bit weird in multicolour mode with the colours changing at each character boundary. A routine to do this should be simple enough. I think the screen resolution is quite adequate for most personal users of the machine. In fact you can churn out some quite stunning effects using DRAW, PLOT, CIRCLE and the user-defined characters.

Before moving on to documentation, here's a list of the disk commands just to whet your appetite: CAT, CLOSE, DELETE, ERASE, FORMAT, MOVE, OPEN. CAT is probably short for Catalogue which lists files on a disk. MOVE probably copies a file from one place to another. The others are self-explanatory.

Documentation

Two manuals come with the Spectrum – a thin but useful introduction for the complete novice and a thicker one which explains things in depth. A lot of effort has been put into this latter manual. It is professionally presented and easy to read. Unfortunately, I was given a photocopy of the final proofs and it contained no index and no table of contents. I read the whole manual a couple of times before starting the review and found it a real problem to find things that I knew were there somewhere. I must admit that the style wasn't to my liking; it's a little verbose and the individual chapters seem to lack structure. I also found the inevitable errors which might cause a beginner problems – things like a minus sign being printed instead of equals, for example. The manual certainly seems to cover everything, so if the table of contents and a comprehensive index are added you'll probably find it adequate. It's certainly an improvement on many manuals on the market.

Potential use

If I were to purchase a system, I would use it for fun, for fooling around with graphics and for programming in Z80 code. I would treat it as a hobby machine, a way of relaxing. My children have already become very interested in the graphics capability and I see this as a way of giving them a real understanding of mathematics. A Logo system on this at the right price would go down well – if anyone out there thinks of doing it, I'd love to review it. Of course, there are those who want to learn to write programs. Once again, this is an excellent machine to cut your teeth on. I think that schools and homes will have to be the prime targets for the Spectrum.

Later on if the disk drives appear, this may change. At a predicted fifty pounds in England for a 100k drive, a lot of people who will have written the Spectrum off as a hobby machine will have to think again. Add to that a combined RS232 and communications facility, and you could be talking about some very interesting and fairly sophisticated networks. At that stage, it becomes a very real prospect for schools looking for a fairly grown-up system, but one which can involve as many pupils as possible.

Without disk drives there is no great office potential for the Spectrum. Once (and if) they're on stream then it's probably just a question of appropriate software. Information management and Visicalc-type applications would seem to be the most likely and, because of the price of the television, they will probably be used with portable black and white machines. No doubt the dedicated will take their computer home to plug into the colour TV when Sinclair finally announces a flat screen colour television. The network idea could then be useful in offices for things like telephone directories, noticeboards and memos.

Prices

It's a bit difficult to forecast exact Australian prices, especially considering unknowns such as delivery dates and resultant inflationary adjustments and fluctuations in international currency between now and the end of the year. However, on today's market, the 16k Spectrum would sell for less than $400 and possibly quite a bit less. Allowing for these minor changes the memory expansion to 48k is expected to cost $200.

Conclusions

Well, for the benefit of those who only read the first and last paragraphs of these reviews here are my conclusions: Clive Sinclair has produced a very good 16k personal computer which offers colour, high resolution graphics and limited sound for less than $400. That represents very good value for money provided that this is the sort of machine you want. It is ideal for people who want to learn about computing and have a lot of fun while they're doing it. Given the right sort of graphics-based educational software, it can bring people very pleasurable ways of learning

Benchmark timings

BM1	4.8
BM2	8.7
BM3	21.1
BM4	20.4
BM5	24.0
BM6	55.3
BM7	80.7
BM8	25.3

All timings in seconds.

Technical data

CPU:	3.5 MHz Z80A
Memory:	16k Dynamic RAM, 16k ROM, 32k expansion option
Keyboard:	40 keys rubber moulding. 183 functions + 8 colour labels. Auto-repeat
Screen:	Domestic colour television. 8 Colours or 6 grey shades.
Cassette:	Domestic recorder.
Disk drives:	To be announced. 100k per drive.
Ports:	RS232 standard. Comms to be announced
Language:	Basic in ROM

SINCLAIR ZX SPECTRUM

subjects such as mathematics and geography. It should arrive in Australia late in '82 or '83 with an assortment of games programs and a lot of people will use it just for that, although it does seem a bit of a waste.

Later on, the provision of disk drives and communications facilities would make it an even more serious contender for the school markets and it will begin to creep into businesses. When the flat screen television appears then I suspect that the business interest will rise because the price will be far more appropriate. Bulletin boards, memos, telephone directories, spreadsheet calculation and information management seem to be the most likely applications.

The 'proper' keyboard is a distinct improvements on its predecessors, but it still doesn't achieve – or try to achieve – the quality of an IBM. All the regular Sinclair features are included – the single keyword entry and the automatic syntax checking as you enter each command, for example.

7th West Coast Computer Faire.

David Tebbutt and Chris Horseman report.

Something like 40,000 souls risked the spring sunshine and the possibility of an earthquake to attend this year's West Coast Faire in San Francisco.

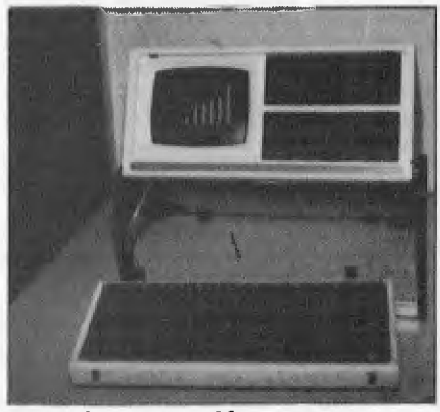
Compared to previous shows, there was less of a revolutionary air about the event – it reflected more of an evolution from what had gone before. Fortune systems weighed in with its 32-bit machine and Osborne clones predictably appeared. The IBM machine has spawned a whole sub-industry of magazines, hardware add-ons and software products since the last show when the IBM Personal Computer was only a very faint rumour.

In previous years I have visited the show as a reporter and have managed to visit every square inch, as well as getting to one or two conference sessions. This year my viewpoint has been that of an exhibitor tethered to his stand about 80 percent of the time. Of course, outside show hours there was the usual round of 'receptions' (i.e. booze-ups) and visits to local companies. From this I offer my impressions.

For me the star of the show was the Otrona Attache. It occupies less than a cubic foot of space and into this manages to pack two high capacity (380k each) disk drives, a full 80-column screen with graphics and all the necessary workings to make it go. The machine is pretty as a picture and would look good on anyone's desk. It comes with some software packages included in the price – CP/M, Wordstar Plus, Basic-80, Valet and Charton. Valet allows you to interrupt whatever you're doing while you perform another task. Charton is a plotting package which takes advantage of the 320 x 240 point display. The reason it won't wipe the floor with Osborne is because the price is more than twice that of the Osborne 1. Further details can be obtained from the Otrona Corporation at 4755 Walnut Street, Boulder, Colorado 80311; Tel (303) 444 8100.

Somebody whispered that an Osborne clone was lurking down by the concession area. Further investigation uncovered the Kaycomp II, which looked remarkably similar to the Osborne prototype shown at last year's Faire. Unlike the Osborne, this machine boasts a 9in 80-column screen and vertically mounted disk drives. It is on sale at exactly the same price as the Osborne, and includes CP/M, a word processor, a high level language (Basic, presumably), a spreadsheet program and a mathematical program. All these descriptions are fairly vague in the supplied literature. Further details are avaliable from meter manufacturers Non Linear Systems Inc, 533 Stevens Avenue, Solana Beach, CA 92075; Tel (714) 755 1134.

Crowds queue to see the Faire



Otrona's neat portable

Durango's desk-top micro

Adam Osborne was much in evidence at the show. He no longer makes awards, he collects them. *Infoworld* presented him with the world's first 'Floppy' award for the Osborne 1. The reasoning behind the name was if other professions can have Oscars and Grammys, why shouldn't we have Floppies? Why not indeed?

Adam took advantage of the show to pre-announce a couple of upcoming Osborne add-ons. An intelligent double-density disk drive option will be available at the end of April. As well as doubling the Osborne disk capacity, it will also be able to handle a number of other formats including the Xerox 820. The option will cost around $185 in the US and will take a dealer about half an hour to install. The battery option will be available from mid-May. This will give up to an hour of power, sounding an alarm five minutes before the power runs out. This 'Portable Power' unit can be used when running off the mains or a car battery to give continuous power in the event of an interruption of the main supply. A charger is supplied with the unit and it takes something like 16 hours to recharge the batteries inside. Portable Power comes in a leather case and costs $390 in the US.

An outfit called Xedex has come up with a Z-80 based peripheral card which allows IBM's Personal Computer to run normal CP/M programs. When not being used for this, it acts as an extra 64k of RAM. With the lovely name of Baby Blue CPU Plus, the card plugs directly into one of the PC's expansion slots. The software which drives Baby Blue is supplied on disk. It allows you to read several different format 5¼in soft sectored disks and to write the CP/M program files in PCDOS disk format. While doing this, Baby Blue tricks a header into the program so that whenever it is loaded it wakes up Baby Blue's own Z80 processor, pops a small translator into BB's memory then transfers the CP/M program to BB. The program is then executed by BB for I/O calls which are passed back via the translator to PCDOS. When the program terminates, BB goes back to sleep and appears to be just another common or garden 64k RAM board.

BB costs $600, unless you'd like Wordstar and MailMerge too, in which case it'll cost you $980.

Further information can be obtained from Xedex Corporation, 645 Madison Avenue, New York, NY 10022; tel (212) 247 1400.

Fortune Systems has launched a 32 bitter. It's built around the Motorola MC 68000 and comes with between 128k and 1Mb of RAM plus 4-16k of ROM. It can drive up to four 5¼in floppy disk drives (800k each, formatted) and up to four hard disks, either 5¼in or 8in.

Bell Labs' UNIX system is what makes this thing tick, so not surprisingly, you'll find it allows multi-user, multi-function operation. Plus, it offers a wide range of communications options.

The main languages offered are Basic, Cobol, Fortran, Pascal and C. The 32:16 allows local networking through the Ethernet system.

I could go on and tell you about the hi res (640x480) colour graphics or the business software available, but I won't. If you're interested, write to Fortune Systems Corporation, 150 Industrial Road, San Carlos, California 94070; tel (415) 595 8444.

Another hardware development which caught my eye was the 'modem on a chip'. Perhaps the rumours I have been hearing of a one-dollar built-in modem will come true. I suspect the dollar is the cost per unit in bulk manufacture rather than a retail figure.

Someone told me there is an 8088 add-on for the Apple which sounds mighty interesting. Graham Hawker also told me about a product which is a dealer's dream. It is called Teach/M and it is a software package which answers all the user's questions about CP/M.

The day following the Faire I found myself at the Machine Intelligence Corporation. This is a company set up a few years ago by some people from Stanford Research Institute. While I was there I was shown some interesting vision systems connected to robots which enabled them to 'see' silhouettes of objects and handle them appropriately. A new subsidiary of MIC, Symantek, is about to be created specifically to implement artificial intelligence applications on microcomputers. If the Editor will let me, I shall write an article about their work later on this year. For the moment the details of this work are a secret.

All right, all you clever bods who know what software publishing is: answer this one. What is the largest software publishing organisation in the micro business?

Those who know their onions will start with a list of people such as Quality Software, Visicorp, Online Systems, Broderbund, and so on — all people who get programs submitted from ordinary users, and who then sell copies and pay the author a royalty.

Those who really know their onions will say that Visicorp, the company which used to be called Personal Software until its product Visicalc got more famous than the company itself since it turns over upwards of $20 million a year, is the largest.

At the Faire, I made the interesting discovery that Appple itself now claims to have around $40 million worth of software publishing business this year. Certainly, on a subsequent visit to Cupertino, I found the offices absolutely crawling with outsiders who were claiming to be freelance programmers, and who were working in the Apple offices on their own products, turning them into Apple products.

It would, said one of these programmers in his laconic Californian drawl, be a much more impressive figure if one didn't have to contrast it with the number of programs that it could publish, actually owns rights to publish, but hasn't yet quite managed to organise.

'The only bigger collection of freelance programmers than the bunch inside the Apple buildings', he said, 'is the bunch who used to be in, but who have stormed out in a rage because Apple can't organise their products onto the market — they don't even seem to know they've got them.'

A lot of people have been very sceptical (except they were Americans,

so they were skeptical) about the latest offspring of CP/M — something called 'concurrent' CP/M-86. This annoys me, because I thought it was the star of the Faire. (See the Xerox 820 Benchtest in the last issue for another example of a 'concurrent' operating system. — Ed.)

What it is to the user is the answer to all his prayers, if it works. It allows you to run several programs at once.

For example, you are typing away at your word processor and you find that you have to include a quotation for a customer. He has ordered 235 units at $4.50 each, plus 345 units at $2.99 each, plus two dozen at 14 D marks, half of each to be discounted at 24 percent and the rest to be discounted at 15 percent which makes them free of tax. . . in short, you need a calculator.

It just happens that your computer, if it were running Basic rather than your word processor, could let you calculate all these things instantly. If you were running your invoicing program, you could not only use it to calculate all these numbers, but could create an invoice, and store it as a file.

At this point, the phone goes. It's the *Daily Snoot*, wanting to know turnover in Afghanistan this year compared with before the Russians moved in. You know the figures are somewhere in your CMS files, or maybe they were in dBase II. If only you could be sure — a DIR command would tell you what the filename was, but your wretched system won't do DIR while running the word processor. . . Under concurrent operating

systems, you can run all these programs at once. Under concurrent CP/M-86, you press control and a number, and you get a fresh screen. On that screen, you can load and run a program. On the next empty screen, you load and run a third. You create your invoice, then switch back to Basic to check the arithmetic. It's okay, so you switch instantly back to the word processor, and call up the new invoice file. You read the figures right out of the invoice file, and merge the invoice and the standard 'pay in 30 days' letter.

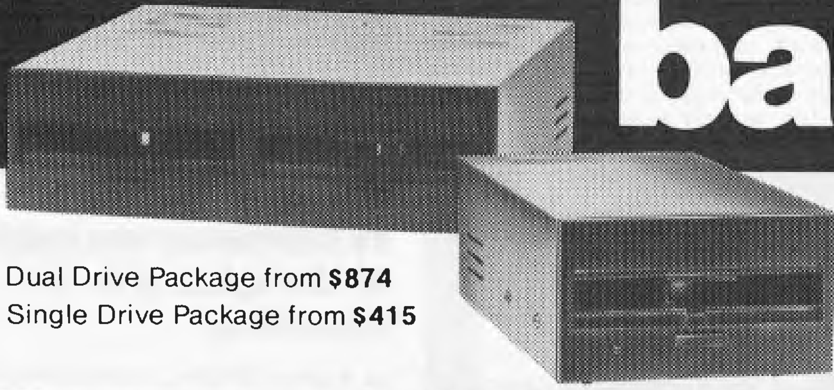
The reason people haven't been jumping up and down enthusiastically is simple enough: concurrent CP/M-86 is based, mainly, on a program which is called MP/M.

That is a multi-user version of CP/M, and it has amply vindicated those sceptics who forecast that it would never work properly (they said nothing that sophisticated would ever work on an eight-bit Z80 or 8085 type micro). It had burned fingers — not just the users' but also Digital Research's fingers, in trying to improve it and make it work.

One day, maybe it will. In the meantime, however, people are nervous about anything that has the smell of MP/M about it. So naturally, they are wary of concurrent CP/M-86.

As I see it, the problems that bedevilled MP/M don't apply to this new system. Mainly, because although it may have four or five or whatever different programs 'running' at one time, they are in fact not running, but waiting. There is no chance that one program will try to read a record which another program is actively updating,

We drive a soft bargain



Dual Drive Package from **$874**
Single Drive Package from **$415**

Disk drive bargains for TRS-80™ Apple™ and Hitachi™

TM APPLE is a trademark of the Apple Corporation. TRS-80 is a trademark of Tandy Electronics HITACHI is a trademark of Hitachi Australia Pty Ltd

Super value disk drive packages for TRS-80.

DRIVE TYPE	NO. OF TRACKS	NO. OF HEADS	CAPACITY	DOSPLUS VERSION	PRICE INC. S.T.
DRIVE Ø					
1xMPI B51	40	1	100K	3.3	$ 499
1xMPI B52	40	2	200K	3.4	$ 639
1xMPI B92	80	2	400K	3.4	$ 799
DRIVE 1					
1xMPI B51	40	1	100K	—	$ 415
1xMPI B52	40	2	200K	—	$ 525
1xMPI B92	80	2	400K	—	$ 695
DUAL DRIVES					
2xMPI B51	40ea	1ea	2x100K	3.3	$ 874
2xMPI B52	40ea	2ea	2x200K	3.4	$1125
2xMPI B92	80ea	2ea	2x400K	3.4	$1454

- **Drive Ø package** includes one disk drive, self-contained single drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.
- **Drive 1 package** includes one disk and self-contained single-drive cabinet/power supply as illustrated.
- **Dual-Drive package** includes two disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of DOSPLUS indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

Disk drives for the Hitachi Peach.

MICRO-80 has developed the necessary electronics to interface MPI disk drives to the Hitachi Peach and save you $00's.

DRIVE TYPE	NO. OF TRACKS	NO. OF HEADS	NO. OF LOGICAL DRIVES	CAPACITY	PRICE INC. S.T.
1xMPI B51	40	1	1	80K	$ 560
1xMPI B52	40	2	2	160K	$ 670
2xMPI B51	40	2x1	2	160K	$ 988
2xMPI B52	40	2x2	4	320K	$1099

All drives are mounted in an attractive dual-drive cabinet with power supply, interfacing and all necessary cabling. Single head systems include programs on disk to allow single drive formatting and copying. Dual head drives require a modified disk controller board. We can modify your MP-1800 controller for $50 or supply a specially modified MP-1800 controller board for $249 inc. S.T.

Disk drives for Apple™! Disk drives for Apple™!

Now there is an alternative. Sensibly priced disk drives from Japan. Completely compatible with standard APPLE™ drives. The AMUST drive can be mixed on the same cable with disk 1 or disk 2 drives. It operates from an APPLE™ controller card or APPLE™ drives will operate from the AMUST controller card.

MODEL AAA-1 INCLUDES CONTROLLER CARD AND DOS 3.3 ONLY $699 inc. S.T.
MODEL AAA-2 DRIVE 2 NO CONTROLLER OR DOS ONLY $560 inc. S.T.

5,000,000 Bytes of hard disk storage. MICRO-80 Winchester drive

The MICRO-80 Winchester Drive sub-system incorporates a new technology 5¼ inch Winchester drive, the proven ACT controller board with HOP data separator and the appropriate interfacing for your micro-computer. This is a complete hardware/software package ready to plug-in and go! Supplied with CP/M disk operating system. **SPECIAL INTRODUCTORY OFFER FOR TRS-80 MODEL 2 or 3 MICROCOMPUTERS COMPLETE SYSTEM ONLY $2995 inc. S.T.**

FOR OSBORNE 1, ZENITH Z89, SUPERBRAIN, NORTH STAR HORIZON ONLY $3799 inc. S.T.
FOR 10 MBYTE SYSTEM ADD $569 TO ABOVE PRICES

TRS-80, TRS-80 Colour Computer and Hitachi Peach owners
MICRO-80 MAGAZINE IS JUST FOR YOU. NEW PROGRAMS AND ARTICLES EVERY MONTH.

MICRO-80

for all your disk drive requirements.

433 Morphett St. Adelaide (08) 211 7244
P.O. Box 213, Goodwood, S.A. 5034. (4 lines)

because they can't work together — they take it in turns.

I hope I'm right! If I am, it will be a tremendous boon for the IBM Personal Computer — IBM was due to announce its availability as this piece went to press — and the Sirius One, where Chuck Peddle's company is negotiating to get the system from Digital Research.

On another stand a weird collection of rods was being waved around, pointing at an apple. Not an

Fortune's micro (above) and a general view of the Faire (below)

Apple, but an apple — green, crunchable, nourishing-looking.

The man behind it was pointing the end of his rods at the apple, and pressing buttons on his computer — and eventually it dawned on me that he was producing a three-dimensional picture of the fruit.

Each point where his apparatus touched the skin was connected to the nearest ones until a 'wire cage' around the surface was built up.

This magic trick, by Penguin Software, works for anything you can put on the plotter table. For more information, contact Penguin Software at 830 Fourth Avenue, Geneva, Ill 60134.

When people are asking $4,000 and more for hard disk drives in Australia, it comes as a bit of a shock to walk past something called 'Davong Systems' and see a hard disk for the IBM Personal Computer, costing $2,000.

That was the price for the disk system installed in one of the holes where you normally find a floppy, with an interface, a controller, and software to let the system boot up from MSDOS, the new IBM operating system from Microsoft.

'We placed a $2 million order with Tandon, the disk maker,' explained the company boss. It turns out that the price frightens even the Americans, but not because they think it's impossible — just that they think all hard disk prices will be down to that level soon. Tandon, Seagate and Shugart are fighting the price down, in the hope of encouraging a big market, it seems.

Davong Systems is at 1061 Terra Bella Avenue, Mountain View, CA 94043

If you are looking for help in choosing the best way of organising vast amounts of data on a micro, you would have done well to stay away from the most obvious source of help at the Faire — the seminar session on database.

All the big ones were there — dBase II, now best-seller outside the Apple world, MDBS, Access 80 (a new one, but impressive to experts), Selector V, Condor and FMS80.

One by one, each product was 'described' by a senior executive from the company which either sold it or produced it. And, one by one, the audience either fought off sleep long enough to get out of the room, or yielded and went into a dull doze.

My conclusion at the end of the session was that none of the people who were speaking had the faintest idea of what their rivals could or couldn't do. They made little attempt to compare and contrast the abilities of their own database manager with others — instead, they recited a list of 'features'.

The trouble with a list of features is that it conveys nothing about whether the features are useful or not. Is a system with the ability to use five files better or worse than one with one file of infinity size and no restriction on record length? Why?

At the end, I came to the following tentative conclusions:

1. GBS from Quality Software is not available yet.
2. MDBS from Micro DataBase Systems is very powerful, but should be restricted only to programmers of considerable skill. The company says so, and I don't think they say so just to impress customers.
3. Access 80 may be very powerful at storing, and recalling, data you put into it. On the other hand, it may not be suitable for what you have in mind, and the literature, or the program itself, will never tell you anything to help you make up your mind.
4. dBase II is market leader. Some people like it.
5. If Condor is a good product, somebody ought to tell the people who sell it.
6. FMS-80 isn't a database manager, though it may well do more for you than a database manager can.

Meanwhile, Chris Horseman was looking at the lighter side of computing.

As a games designer, I was particularly interested to see what was new on the games scene. The vast majority of new American games fall into the 'arcade' category, although the original fervour for producing copies of 'Space Invaders' has largely died down. Most games producers have finally come around to the idea that arcade games need to be in machine language. We are thus seeing better use of graphics and a generally more professional standard of quality. There are still more games available for the Apple than for any other system, though this is largely due to the time it has been on the market. Most of the top games are now being converted for the Atari (eg, Raster Blaster and Apple Panic), but these conversions do not usually take advantage of the superior graphics capability. There is still much copying of ideas for games with a sort of 'goldrush fever' surrounding Pacman. I saw perhaps ten versions, each with their own strong points and improvements but still copies. This has finally aroused some copyright action from Atari who bought the rights for the game from the designers. There have already been some minor court proceedings over Pacman but no one has been successfully sued yet.

There were some excellent new games on show and some people are beginning to stand out as producers of games software. Possibly the most noticeable of these was Scott Adams Inc, who had a large mock castle set up in the windows. Scott has a large (and still growing) range of software available for all of the popular machines including his increasingly famous adventures. These have been recently enhanced on the Apple by the addition of colour graphic illustrations and a 'voice track' which can be used if you own a Vocoder. The graphic adventures should soon be available for the Atari.

Nasir Gebelli's company Gebelli Software Inc is one of the newer US software producers, and to judge by the high quality of the games will become one of the majors very soon. Nasir (who I believe used to program for Sirius) is producing a range of arcade type games for the Apple and Atari. The games include a high quality racing car game for the Atari called Match Racer.

One of the most original games that I saw was the Swashbuckler by Datasoft; this is a simple game but has stunning animation. In the game you control a

7th West Coast Computer Faire.

man with a sword and must fight off a series of piratical and animal opponents whilst descending through the decks of a pirate ship. Swashbuckler runs on Apple II. Datamost also produces a range of other games including the obligatory Pacman-type game called in this case Snack Attack; all Datasoft's programs so far are for the Apple.

Broderbund produce software for the Apple, Atari and TRS-80. This includes an excellent pinball game called David's Midnight Magic for the Apple (and shortly for the Atari); also a very compulsive game called Apple Panic. This is a 'man with a hammer' game, and also involves knocking rampaging apples through floors (honestly!). This is also available for Apple and Atari. Their TRS-80 games are less arcade-oriented and include Galactic Empire and Galactic Trader.

On-line Systems produce highly polished and professional games for Apple and Atari, including Jawbreaker — which is arguably the best Pacman-type game. Mouse Attack is an excellent Pacman-inspired game with additional features such as two-player option. On-line are also producing a version of the arcade game Frogger for the Atari which looks remarkably close to the original. They are also producing a mammoth graphic adventure called Time-Zone which is retailing for a hundred dollars and is reckoned to take over a year to play. (Does this mean that it will be obsolete by the time you finish it?)

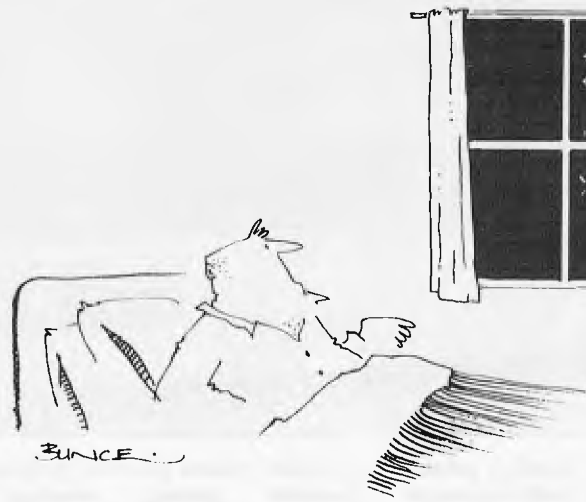
Sirius Software produces games for the Apple II with excellent cover art work and some amazing captions on the games. The games don't look bad and were certainly attracting interest from the crowd. Titles include Hadron, Space-Eggs and Cops and Robbers.

One company producing solely for the TRS-80 is Big Five Software, which makes imaginative use of of the limited graphics facilities. It produces six games, including Supernova and Robot Attack, which has voice sound.

Automated Simulations continues its EPYX adventure range with Crush, Crumble and Chomp. Datamost was showing an original game called Tumble Bugs (or sometimes Dung Beetles), which uses an unusual 'lens' effect to magnify part of a maze on the screen and has some simple voice synthesis.

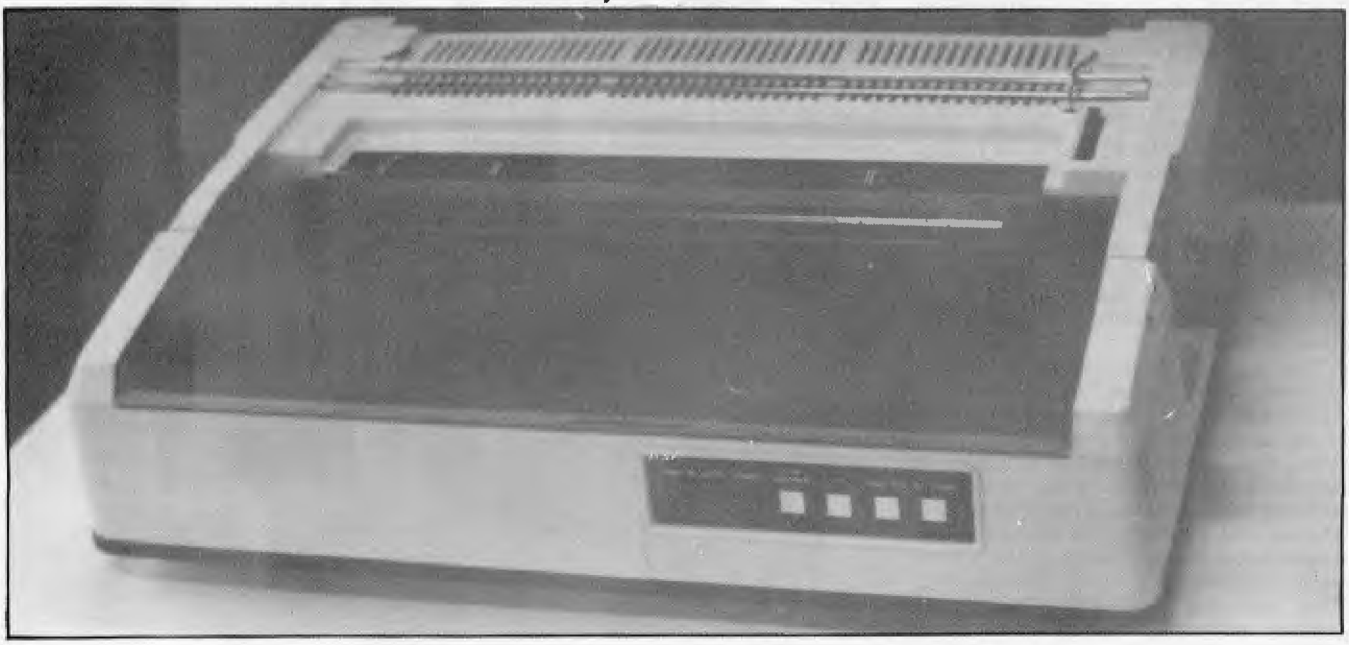
Bill Budge's Raster Blaster has been available for some time, but only for the Apple II. This game has just been released for the Atari, although the colours don't come out well on the television system. Bill received *Softalk* magazine's game of the year award for this game, and I was astonished to see that Bill is only 19, although an old man of the software industry.

On the subject of awards, Chris Crawford, head of Atari's games research group won the Creative Computing game of the year award for his innovative and excellently engineered Atari war game Eastern Front.



TEC PRINTER REVIEW

by Ian Davies



As a follow-up to last month's printer survey, we take a closer look at a couple of printers produced by the Japanese manufacturer, TEC. Both are daisy-wheel devices, suitable for home or business word processing applications.

Both units are capable of 136 characters per line using Pica pitch, or 163 characters in Elite pitch. Spacing between lines is adjustable down to 0.0208 inches and print spacing is to a minimum of 0.0083 inches. They provide 96 printable characters in a Diablo/Qume font and can handle paper widths up to 15.5. inches, normally through a friction feed, but with tractors as optional.

The main difference in the two models is print speed. The lesser of the two, the F10-40 can manage 40 characters per second, whereas the F10-55 can (you guessed it) perform at 55 cps. The model 40 can handle three pieces of paper at once, and the 55 can squeeze six pieces through. Both come with an RS232C interface as standard and can accept data at 300, 600, 1200 or 2400 baud, with the model 55 also providing 110, 200, 4800 and 9600 baud. Both models can optionally be equipped with 8 bit parallel, 12 bit parallel or RS422 interfaces for faster data transfer. The units have a 256 byte buffer in their standard configuration, but this can be extended up to 2,048 bytes.

For more serious applications, the F10-55 accepts black and red ribbons which are software selectable — an essential feature for printing debtors statements. Additionally, a word processor option provides proportional character spacing, meaning the characters such as 'i' take up less space than characters such as 'm'. I would consider this option essential for any word processing applications.

The units appear pretty well put together, weighing in just under 14 kgs. Their overall dimensions are 57.4 x 40.5 x 15.35 cms. Internally, all the printing mechanism is towards the front of the case with the three circuit boards layed vertically across the rear wall. A rather large heat sink also runs along the rear wall, but does not appear close enough to give the ICs a sun-tan. The power supply sits roughly in the middle of the unit. An 8085 CPU sits on one of the circuit boards, directing operations from the rear. The printer looks as though it should be quite reliable, but of course any real test of reliability would take months. All I can say is that it didn't break down while I was using it.

The printers are reasonably quiet in operation, being measured at 65 Db at one metre from the platen. They are both equipped with "silent covers", but I suspect that a larger external noise cover (available from most office suppliers) could be in order if one was to use it in an open office situation. This is true of most impact printers.

A small control panel on the front of the units provides a number of operator controls such as PRINT ON/OFF, LINE FEED, PAGE FEED and SET PAGE. Three indicator lights show printer status in terms of PRINT ON/OFF, POWER and ALERT. The ALERT light comes on for one of five possible reasons, three of which are optional features and one is only for the model 55.

The printers provide all the normal paper movement controls, including one unusual escape sequence which puts the unit in ONLINE mode. Comprehensive data protocol options are installed, as are self-test procedures. In fact, there are five test procedures for testing print, circuits, print wheel homing, print wheel indexing and RAM.

The documentation supplied with the printers is concise and to the point. It describes printer operation and control codes in a very precise manner, and gets into the guts of the data communication protocols. The manual we were provided with was a preliminary only and, while its nice to see a manual that does not begin with "Thank you for purchasing our printer . . . ", I think that it could be made more readable by less technical detail and a little more overview. Often this sort of printer is used by secretaries and the like, and in this environment a section (near the front) on general operating procedures and hints is essential.

The printers are distributed in Australia by Ampec, and sell for $2300 (F10-40) and $4200 (F10-55), excluding sales tax. The model 40 seems like quite good value for money, but I would find it hard to justify the extra expense of the model 55. Both models have options that really should be provided as standard.

HITACHI
MEANS BUSINESS

BUSINESS PACK 1
PEACH CPU — EXTRA MEMORY
HI RES GREEN SCREEN
DUAL 5 INCH DRIVES
80 CPS PRINTER
$3688* ex tax

BUSINESS PACK 2
PEACH CPU — EXTRA MEMORY
HI-RES GREEN SCREEN
DUAL 8" DRIVES
80 CPS PRINTER
$5248* ex tax

Program modules
$310 EACH

DELUX COLOUR BUSINESS PACK 3
PEACH CPU — EXTRA MEMORY
HIGH RES COLOUR SCREENS
DUAL 8" DRIVES — 80 CPS PRINTER
$6018* ex tax

HI WRITER WORD PROCESSOR PACK
INC. HIWRITER PROGRAM
PEACH CPU — EXTRA MEMORY
HI RES GREEN SCREEN
D/S 5" DRIVE, PRINTER
$3676* ex tax

NOW AVAILABLE PEACH INVADERS

AND SO DO WE, AT

* plus sales tax
if applicable

robs COMPUTER CENTER

**295 THOMAS STREET,
DANDENONG
ph: (03) 791 2900**

AUTHORISED
HITACHI
AND
OSBORNE
DEALERS

**IF YOU NEED PORTABILITY
PLUS POWER, THEN**

$2595 inc tax
PRINTER $500 inc tax

OSBORNE IS FOR YOU

NOW AVAILABLE DOCTORS, DENTISTS OR REAL ESTATE PACKAGES — $500.00 each.

MTH The Microcomputer House

WE'VE MOVED

(FROM 1st JULY 1982)

Thanks to the thousands of people who have supported The Microcomputer House — we have now moved to new, larger premises at 116 - 120 Abercrombie Street in Chippendale. The new premises have two showrooms — one for the businessman and the other for the home computer enthusiast. More room means expansion into new systems, new package systems, a wider range of software for the businessman and the enthusiast and all of this will be coming your way as from the 1st of July. In appreciation of the support given to The Microcomputer House by our valued customers we are offering some great 'WE HAVE MOVED' specials — YOU REAP THE BENEFIT.

SAVE ON COMMODORE CBM 4032*

32K RAM

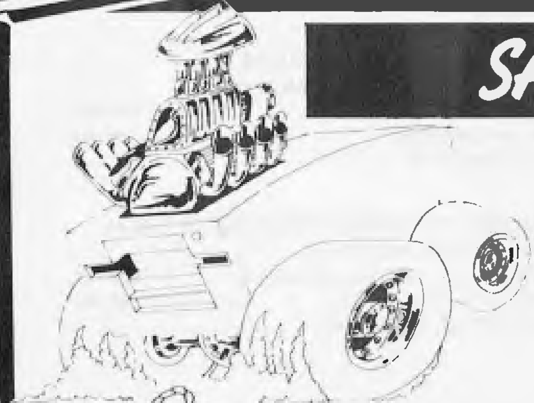
A super special buy of the **CBM** 4032 enables The Microcomputer House to offer this 32K RAM unit for a fraction of its original price! This versatile computer is a completely integrated processor, memory (32K), keyboard, and 9" VDU. Now is the time to act fast — limited stocks of this fantastic bargain for the businessman or the enthusiast.

* CBM 4032 with 9" VDU

ONLY $700

LIMITED STOCKS

SAVE A MASSIVE $1,079.00



SAVE ON DISK DRIVES

Bankcard Mail Orders Welcome

PETDISK II is a 5¼" Mini Floppy system that has a high speed plus high capacity. Complete with PDOS II disk software with a full set of BASIC commands with standard syntax.

540/1 Single drive/Double density (143K)
Normally $650 SAVE $100 NOW $550
540/2 Dual drive/Double density (286K)
Normally $1105 SAVE $255 NOW $850
580/1 Single drive/Quad density (286K)
Normally $895 SAVE $195 NOW $700
580/2 Dual drive/Quad density (572K)
Normally $1438 SAVE $238 NOW $1200

SAVE UP TO $255.00

This is the ideal package for the small businessman — a complete system, right down to the connecting cables. Included also is the VISICALC software package designed for the busy executive — replaces the pen and calculator.

System includes:
CBM 4032 computer with 32K RAM
Commodore 2031 disk drive
Commodore 4022 printer
Cable and VISICALC
Total value of system $4,086.00
CBM 4032 with 9" VDU

NOW $2,800.00

SAVE A MASSIVE $1,286.00

The Microcomputer House Pty. Ltd.

116 - 120 Abercrombie Street, Chippendale, N.S.W. 2008
Phone: 698 7866 or 698 7076

CURSOR

THE BEST FRIEND FOR YOUR COMMODORE PET

NOW YOU CAN SUBSCRIBE

For $43 you will receive a 6-issue subscription which includes a C-30 cassette with 5 programs plus the cover program. Each subscription means a total of 36 programs at around $1.19 each!!

FOR THE VIC 20 ARFON EXPANSION

by I. Davies

Anyone who has got a VIC 20 will know that it is a reasonably good machine which could be expanded into quite a nice system. The problem, however, is that the VIC has only got one plug-in port on the rear of its case. The Arfon expansion unit solves this problem by providing seven slots for plug-in cartridges and, at the same time, attractively housing the VIC, cartridges, power supply and modulator.

The Arfon unit is a cleverly designed chasis which the VIC 20 slides into. The VIC's expansion port mates with the Arfon mother board, which itself, is extremely simple, consisting of just five integrated circuits. It uses three of these to buffer the address and data lines to give them the necessary fan-out drive up to seven cartridges simultaneously. The other two ICs provide gating logic for the buffers. It also contains a couple of capacitors across the supply rails and some pull-up resistors on the data lines. The machine control bus is not buffered at all. The buffered and unbuffered lines then run straight across seven expansion slots in parallel. These slots are exactly compatible with the one on the rear of the VIC 20.

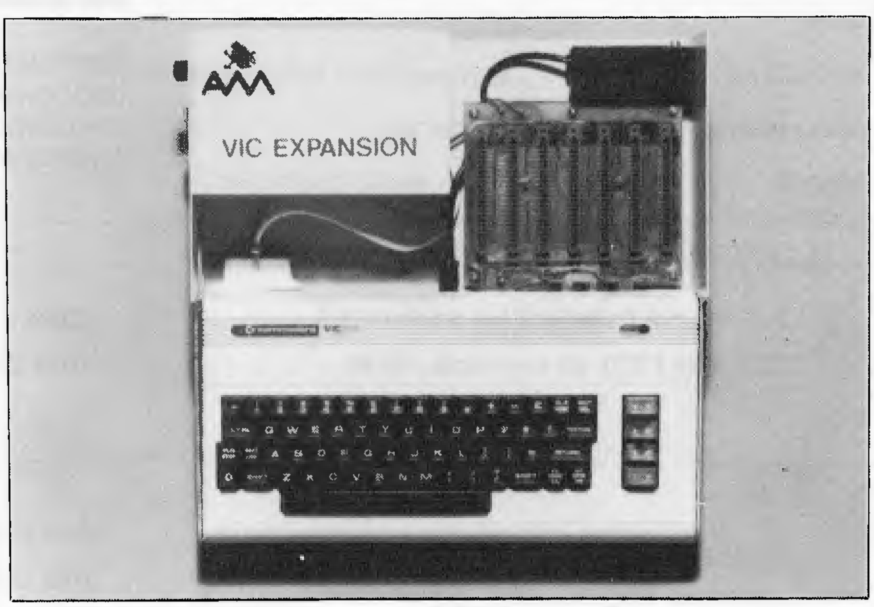
The expansion interface also contains a new power supply to provide the necessary current to drive the fully expanded system, and includes an extra 24 volt rail to supply the Arfon VIC 20 printer, which we will be seeing soon. The chassis includes an "on-off" switch on the rear of its left side. This switch is used instead of the one on the VIC unit itself.

The expansion unit includes a handy little clip to hold the VIC 20 modulator and is designed so as to allow access to the VIC's disk cassette and joystick ports. The unit we were provided with did not have a plug on the end of its 240 volt lead, and this is one thing that I feel very strongly about. The days are gone when computers were only used by people who knew about electronics, and I feel that it is irresponsible to provide "bare wires" to computer consumers; I only hope that this fault was restricted to our demo system.

The chassis itself is made out of sturdy sheet metal with an optional aluminium cover which, when in place, allows the television or monitor to be placed on top.

In conclusion, the main thing the unit has to offer is the seven extra slots.

Arfon is a newly established company from north Wales and its expansion unit is available in Australia from Computer Imports, of Adelaide, and Computer Cellar, of Newcastle.



The Arfon expansion unit showing the seven expansion units and RF Modulator.

Now in stock

- **LARGE SELECTION OF DRIVES (5.25 AND 8 INCH)**

- **HI-WRITER COLOUR WORD PROCESSOR (5.25 AND 8 INCH)**

  Many sophisticated features including merge, mailing list, global search, on screen help, instant formatting etc.

- **HI-CALC VISUAL CALCULATOR (5.25 AND 8 INCH)**

  Full colour, many advanced features

- **OUTSTANDING FEATURE !!!**

  Hi-Calc will integrate with Hi-Writer so you can merge statistics and text.

HITACHI PEACH

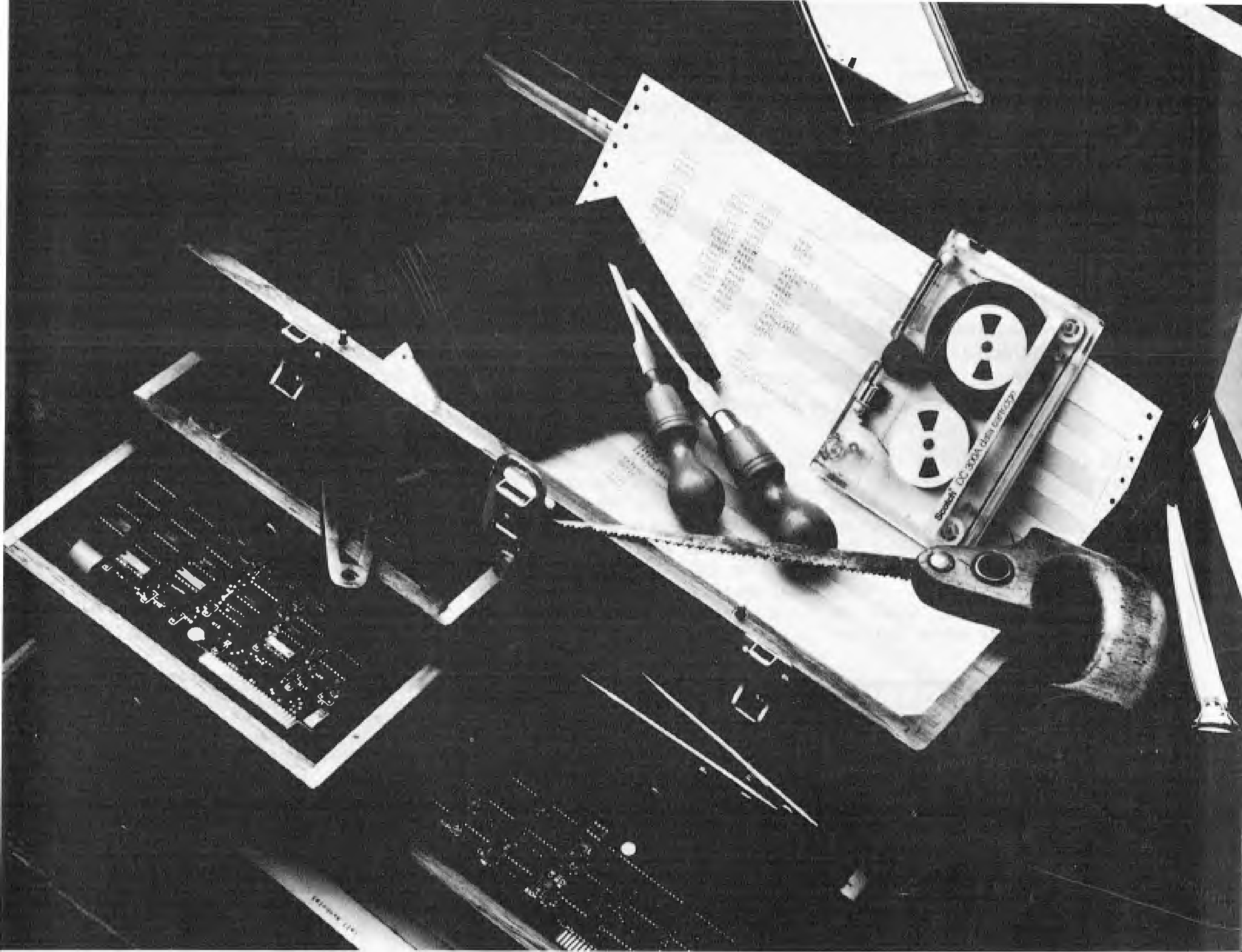
RADARO COMPUTER CENTRE

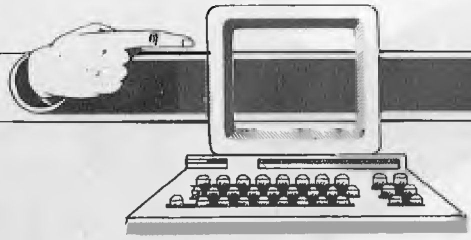
1100 TOORAK ROAD, HARTWELL, VIC.

TEL. 29-4466 29-5570

WE ARE HITACHI PEACH DEALERS.

WE DON'T SELL OTHER BRANDS.





MICROMARKET

More than once a day, our office receives a call regarding machine specifications or availability or requests for advice on software, so we've decided to release an updated version of MICROMARKET and complement its bi-monthly appearance with a new section called PACKAGES. The first appearance of MICROMARKET will be in the September issue of APC followed by PACKAGES in October, each being published thereafter in alternate months.

MICROMARKET will be a comprehensive list of all microcomputers available in Australia including single board micros, and kits. There will be five main headings under which the most important specifications of each machine will be listed.

For space reasons it would be impossible to list all prices and configurations of available microcomputers, so the minimum configuration and its price will be listed. Where disk drives, speech synthesiser or graphics cards, for example, are available, they will be listed in the "Miscellaneous" column with the relevant price.

A number of abbreviations are to be used and they are listed in the table below.

The following example shows the order in which components, options and software will be listed. We ask that companies submitting information do so in the standard form shown, using abbreviations wherever possible.

MACHINE (PRICE FROM)	MAIN DISTRIBUTOR/S (NO. OF DEALERS)	HARDWARE	SOFTWARE	MISCELLANEOUS (DOCUMENTATION) (BENCHTEST DATE)
Full Name of Basic Machine (Minimum Retail)	Name of Main Australian Distributor (No. of Dealers Including Franchised Dealers)	RAM : PROM : EPROM Processor/s: C : Disks (Capacity) (either F/D or H/D if standard : TV Int : P/P (incl. no. of ports) : S/P (incl. no. of ports) : G/c : Tone Generator	Standard Languages : Optional Languages (listed in italics)	The contents of this section will be left largely up to the supplier. (One of the following abbreviations will be used to describe the general level of documentation : E, H (i.e. hardware details included), I S (i.e. Details of ROM included). BP – Date of full APC test).

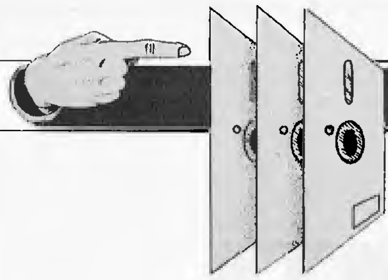
List of Abbreviations

A	Assembler	G/C	Graphics Card	M/A	Macro Assembler	S	Software
BT	Bench Tested	H	Hardware	N/A	Not available	S/P	Serial port
C	Cassette	H/D	Hard disk	N/P	Numeric pad	T/E	Text editor
E	Extensive	I	Introductory	O/S	Operating system	TBA	To be announced
F/D	Floppy disk	Int	Interface	P/P	Parallel port	U	Utility

Please note: Software items listed in *italic* are not included in the basic price of the equipment. All prices are inclusive of Sales Tax.

This is an example of how an entry will appear in MICROMARKET:

MACHINE (PRICE FROM)	MAIN DISTRIBUTOR/S (NO. OF DEALERS)	HARDWARE	SOFTWARE	MISCELLANEOUS (DOCUMENTATION) (BENCHTEST DATE)
Tandy TRS-80 Color ($599)	Tandy Electronics (02) 638 6633 (200)	4–16k RAM : 6809 : 8–16k ROM : C : 16 x 32 TV int : RS232 port	Color Basic : Extended Color Basic	With 16k RAM, 16k ROM & Extended Color Basic $250 (I). BT 1/82
TRS-80 Level I ($499)	Tandy Electronics (02) 638 6633	4-16k RAM; Z80; C; 12", 16x64 optional: B/W VDU	Basic: Games: A	Basic in 4k ROM; upgradeable to Level 2 (I)



PACKAGES

This software directory will be confined to business packages which have been in use for at least six months in a minimum of five sites.

The layout has been designed to allow you to discover which packages are available for the application you have in mind and to show you which packages are available for your computer if you already have a machine. In either case the code enables you to look up the supplier's name and telephone number in a separate table.

So, the directory will be divided into two sections. The first will index software packages according to application. If you require a program to provide credit control, then under this heading you'll find a list of machines for which this software is available and a code referring to the package's supplier.

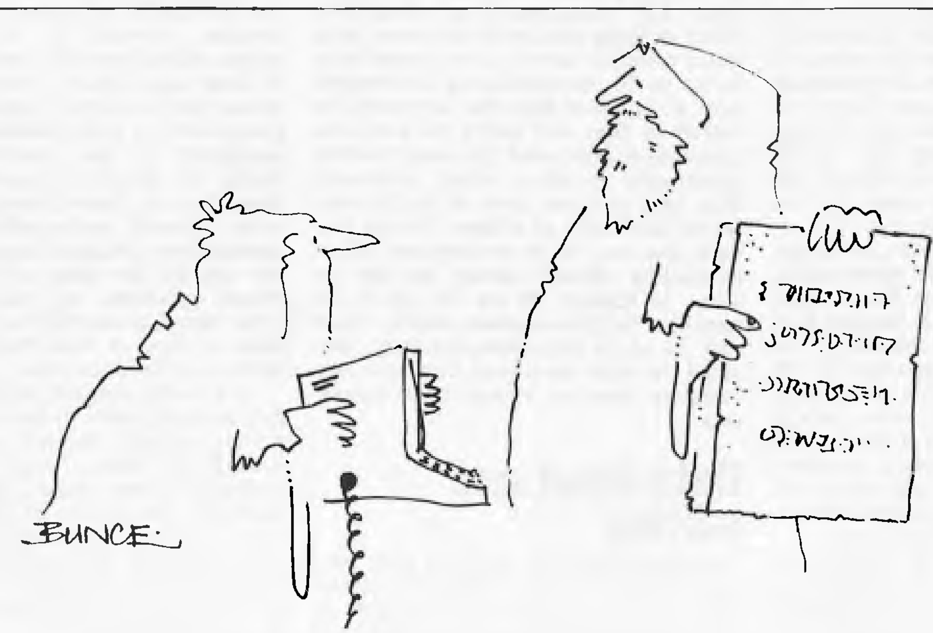
The second section will be an alphabetic list of microcomputers. Under each machine will be a list of available software packages and supplier code, so that if you are contemplating the purchase of a machine or already own a computer, it will be easy to identify available software.

Producers of packages which fall within the first mentioned constraints should send details to PACKAGES, *APC*, P.O. Box 115, Carlton, 3053. We require only the category of application, computer for which the package is intended, price, supplying company name and telephone number.

Below is a suggested list of applications (CP/M and other widely used operating systems are listed as separate "microcomputers" because of the large amount of software written specifically for them).

Applications

Appointments Planner
Assembler dev
Bank accounts
Bill of materials
Bonds/pension quotations
Budgeting package
Bureau change
Cash flow
Cash register
Company secretary
Container accounting
Contract costing
CP/M & utilities
Credit control
Customer file
Database management/
Information retrieval
Dental Records
Disk operating system
Estate agent
Equipment lease/rent/HP
File Handling
Financial modelling
Financial planning
General ledger/NL
Hotel management
Incomplete records
Industry Factory loading
Industry work study
Integrated accts
Investment portfolio
Invoicing
Job costing
Job order control
Legal precedents
Letter writer
Lotteries
Mailing List
Mail shot
Membership accting
Motor Dealer
Order entry/invoicing
Order processing
Office admin
Pad to plotter systems
Payroll
Perpetual Inventory
Personnel records
Petaid report generator
Planning/Maintenance
Postal advertising
response package
PR/advertising package
Price lister
Printers job control
Production analysis
Prof appts groups
Prof appts individ
Prof client billing
Programming aids
Property management
Purchase ledger
Quotation estimating
Sales ledger
Salesman
Sales Tax analysis
S/L, P/L & stock control
Solicitor's complete
record accounting
Solicitor's package
Statistics
Stock control/recording
Surveying
Text file librarian
Time/cost recording
Travel agency accts
Travel Agents Diary
Travel Ticket Sales
Utilities
Utility set
Various engineering
Vet package
Video message
Warehousing
Word processing
Work in Progress



'But think about it Dad! With one of these we could have millions of commandments!'

BENCHTEST SOFTWARE

dBASE II: A DATA MANAGEMENT SYSTEM UNDER CP/M

Kathy Lang continues her series of database reviews

dBaseII gives users of CP/M systems the ability to store, process and access data in a wide variety of ways. It has a clean, well-constructed user image, making it unusually easy to learn and use even its most powerful facilities. The authors call it a relational data base management system but, strictly speaking, it is a 'file management system with data base connections'. While the term 'data base' is increasingly being used to mean the same as 'file of data', I don't find this a very helpful trend, so I shall stick to the word 'file' when I'm talking about a single file of information which need not be connected with any other data file. 'Data Base' will in these articles continue to be reserved for a system in which the data is stored in one or more files which are interconnected in such a way that the end user (as distinct from the data base administrator) does not need to know which physical file contains the data he needs to access.

dBaseII stores data in sequential fixed record length files, and must be given information about the record structure for the file. This information is stored in the first physical record of the file, so there is no need for a separate record definition file, and hence no danger of getting the two out of step (for instance by deleting the record definition file by mistake). dBaseII can handle two files of data at once, but not more. Each data file may have one or more indexes for fast access and for updating; these indexes are of equal status, that is, dBaseII regards as the primary index which ever index is invoked first on a particular access of the file. If the user wishes it, he can arrange for all indexes to a particular file to be kept up to date automatically when data is being updated, a distinct improvement on other data management packages. Keys used to construct indexes do not have to be unique but the rapid access commands work better if they are.

dBaseII has a variety of ways of updating, accessing and displaying the data, making it one of the most flexible packages on the market. It originates in America and is distributed and supported in this country by a number of companies, so the price can vary somewhat.

Constraints

Records stored in dBaseII files must all have the same structure and be of fixed length, with the maximum length 1000 characters. No record may contain more than 32 fields, which could be a serious limitation in some applications; a field may contain up to 254 characters. Data items may be numbers, character strings or logical variables (ie, taking the value True or False); no special 'date' type exists, so dates must be stored as numbers in year/month/day order to sort correctly. Arithmetic in dBaseII is performed to an accuracy of 10 digits; numeric values may be treated as integers or real numbers, but for length calculations each digit occupies the same space as a character.

Index keys may be constructed from several data fields, but may not be more than 100 characters long altogether. When defining the record structure, data items must be named; these names may be up to 10 characters long. Commands may be invoked from the keyboard, or stored in files and called in with one instruction; command files may include statements invoking other command files, and you may have up to 16 command files open at a time. One or two data files may be in use together. When calculating dBaseII allows the user to store, in memory or on file, up to 64 variables for intermediate results; these may be up to 254 characters long, provided the total space used for temporary variables does not exceed 1536 characters.

Data input and editing

Creating a data file is a two part process: the CREATE command allows the user to specify the structure of each record and then to input the data using this structure. Records are displayed with a named field on each line, with the type shown and a delimiter used to show how long the field may be. Formatted screens to allow more sophisticated displays can be set up using a set of commands giving full control over the screen format.

Once created, data may be edited using an EDIT command; this involves specifying the record to be edited by a variety of methods and then using simple screen editing to amend the record displayed. The screen editing uses the same conventions as the popular word processing package Wordstar — CTRL-E for moving up a line, CTRL-D for moving right a character and so on. It would be nicer to be able to use the terminal's own cursor keys, but at least the Wordstar convention will already be familiar to many dBaseII users. Records may be deleted in EDIT mode but not added; addition is done with INSERT within the data file and APPEND at the end.

Multiple changes are possible too: the REPLACE command lets you, for instance, increase by 10 percent all prices which have not been changed for at least six months, while CHANGE allows you to display each record in a group turn, to allow fields named in the command to be modified without having to specify a record key each time. For all these commands, and all other dBaseII commands which can operate over a range of records, the user can specify the range of operation by record numbers, by relative position ('the next five records') or by characteristics of one or more fields — see the section on Selection later.

One useful feature, which makes online updating safer to use is that records are not actually deleted when you tell dBaseII to delete them — you can 'undelete' them again, provided you have not issued the PACK command, when the records marked for deletion are expunged. Deleted records which have not been expunged can be displayed (they are shown marked with an asterisk) but will not be copied, sorted or appended to another file.

dBASE II

Until a file has data in it, the data structure can be edited without hindrance. Once the file contains data, editing the structure would destroy the data. So dBaseII makes it possible to set up a new, modified data structure and copy data into it. The data may be from an existing dBaseII file, or from an external file in a variety of formats. This makes it possible to add fields to an existing data file which can have data added later, or to create a data file which is a subset of another, as well as to import data files written by other software. In each case, the operation takes just three or four simple commands, which use the standard dBaseII structure.

It is also possible to use two dBaseII data files in conjunction to modify data. One file may be updated by another, with the user specifying the key to be used to match the files. Or, using a similar technique, two data files can be merged to form a third.

Displaying data

Two kinds of command are used to show data on the screen. The user can either choose which record to show and display it all in one command, or locate the chosen record(s) and then display them. This gives greater control over location and display. Using these commands, you can display a set of records matching particular criteria, either as a list or one at a time, move around in the file using the range specifiers or such position identifiers as Top, Bottom or Skip. You can display a whole record, or just selected fields.

The 'selection only' commands can either use the index currently selected for the file, or use fields for which you haven't created an index, although the latter is, of course, slower. Where several records match a specification, dBaseII displays the first and permits you to 'continue' through the rest one at a time. The command which is normally used for printing reports can also display on the screen, so that you can show summaries on the screen too. Some of the commands used to display data from files can also be used to display information from memory, so you can carry out calculations on the current data file and display the results.

Reporting

The REPORT command allows the user to create layout specifications for summaries on either screen or printer. These specifications are stored for subsequent use but cannot be edited. Reports are laid out with fields listed across the page. Column and row headings are allowed but all specification of rows and columns is, as with other packages I've reviewed so far, in absolute terms – line 3, column 42 etc – so you have to do a lot of counting to make sure the spacing is all right and a lot more each time you change the layout. Records may be selected according to specified criteria and there are some powerful calculating facilities. There is no provision for letting column headings take their names from the field names used in the record definition and the calculation facilities fall short of creating sub-totals when specified fields change. More sophisticated reporting features, such as formatted field display using pictures – a bit like the PRINT USING command in Basic – and sub-totalling when fields change, are available through the use of command files (ie, rather than through the REPORT command in its standard form). I wasn't able to make a full test of the report feature, as it didn't work properly on my version of the package.

Selection sets

Nearly all the commands for file access can be modified with a selection parameter. For instance, ifyou want to select only people over 40 years old from a file containing age as a field, and show on the screen the name, age and sex of people in those records on the screen, you can give the command 'DISPLAY FOR AGE 40 NAME, AGE, SEX' and the relevant records will be listed, 15 at a time. So you don't have to decide in advance which fields you will want to select on, and set up a selection criteria file; you just add a FOR parameter to the display, location, reporting and other commands. Brackets can be used in conditions to ensure the correct order of evaluation and you can use the logical operators AND, OR and NOT as well as the usual comparison operators. Comparisons involving strings may also use an operator which searches within strings as well as comparing complete fields.

Sorting records

In dBaseII, indexing is used to carry out the kinds of operation which in other packages often involve both indexing and sorting, and I was able to do all my tests without using SORT except as a straight benchmark. So it is little hardship that sorting is a rather cumbersome difficult operation in dBaseII; you can sort on only one field at a time, so sorts on multiple fields involve sorting on each field in turn, starting with the least significant. You can sort on parts of fields and in ascending or descending order. I couldn't discover how to get dBaseII to ask the user to change disks, so I was limited also to sorting within a single disk, as dBaseII takes up most of one disk. However, this isn't really a problem, since when a file is opened in conjunction with an index, it is accessed in the order indicated by that index. So if you wanted a file displayed in a particular order, you would index it using the desired ordering fields as keys.

Any field or combination of fields can be used in an index up to a total length of 100 characters. The only other limitation is that this access technique must use the order in which the index was constructed. For instance, if a file was indexed by age, years of education and salary, records could be found by specifying age, age and years of education or all three, but not by years of education and salary alone. A slightly irritating feature of INDEX is that you can only index on character variables, so numeric items must be converted with the STR function. This is specified in the INDEX command, but the specification must include the length of the numeric item, even if you want to use the full length of the field as given in the record definition. So the instruction to carry out the indexing example just suggested could look like this:

```
INDEX ON STR(AGE,3) +
STR(LENGTHED,2) +
STR(SALARY,5,2)
```

When you bring a data file into use, you can specify up to seven indexes to be used with it. Only one will be used to provide the keys for accessing data, but all those specified will automatically be kept up to date when data is changed. This is an unusual and powerful facility, and the integration of updating and indexing makes it much easier to ensure that indexes and data stay in step.

Calculations

The user can perform calculations using data fields, items typed in from the keyboard and constants freely intermixed, using the normal arithmetic operators and brackets to ensure correct ordering. Items can be counted, as well as totalled. Results can be stored in memory variables or in data fields in files, and can be 'one-off' single results or a series resulting from a calculation performed once for every record in the file. Memory variables are referred to by name. Calculations stored in memory can be saved on a separate file for continued use. You can also create a file which consists entirely of aggregates; for instance, if your employees work on several jobs at a time, you can record the job information on a 'session' basis, and then ask dBaseII to create a new file consisting of one record for each job containing the totals of time spent, resources used etc.

Security

This is probably the weakest area in the package. You can of course build protection into command files through which operators invoke dBaseII but the package itself provides no facilities through the ordinary commands to prevent unauthorised access to data. All transactions may be logged, either in the usual fashion on the printer, or on a disk file.

Tailoring

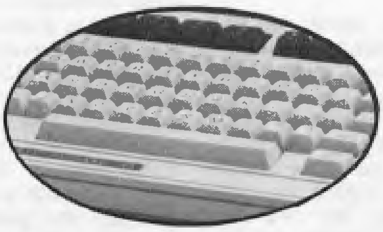
Any command which can be executed from the keyboard may instead be put in a command file for later execution. Other commands, particularly looping instructions, may only be used from

SAVE 61¢ an issue

SUBSCRIBE TO APC

2nd BIRTHDAY REVELRIES

To celebrate APC's second anniversary we are offering, for a limited period only, reduced rates for Australia's largest selling microcomputer publication.



You don't need it to calculate the savings we are offering readers who subscribe, renew or extend their subscription to AUSTRALIAN PERSONAL COMPUTER right now!

1 Year Subscription (Australia only)	*$24.00*	*We will send you a full 12 issue subscription at the reduced rate of $24.00, saving 50c an issue over newstand price.**
2 Year Subscription (Australia only)	*$46.00*	*You receive 24 issues, saving yourself $14.00 off the full subscription price. Plus you protect yourself against future increases in cover prices and postal rates which will probably amount to a saving of an additional $8.00 over the period.**
3 Year Subscription (Australia only)	*$68.00*	*You receive 36 issues, saving yourself $22.00 (or 61c an issue) off the full subscription price. Plus you protect yourself against future increases in cover prices and postal rates which will probably amount to a saving of $11.00 over the period.**

The 'APC Subscription Department' has now been relocated and is no longer directly accessible from our publishing offices. Therefore, we ask that any communication be made in writing, including changes of address and renewals.

Existing subscribers can take advantage of any of the adjacent offers. Just tick the 'This is an extension subscription' box on the coupon and enclose your payment. You do not have to wait until it is time to renew (by which time this offer may have been withdrawn). Save money by extending your subscription today!

** Offer expires July 31, 1982*

Why do more people read APC than any other microcomputing magazine in Australia? Subscribe today!!

I would like to subscribe to *APC*.

☐ This is a new subscription ☐ This is a renewal ☐ This is an extension subscription

☐ 1 year (12 issues) Aust $24.00 ☐ 1 year rest of the world (12 issues) Aust $40 surface

☐ 2 years (24 issues) Aust $46.00 ☐ 3 years (36 issues) Aust $68.00

☐ I enclose my cheque, made payable to Australian Personal Computer for $. . . . .

☐ Please debit my Bankcard. Bankcard Number . . . .

Expiry Date . . . . .Signature. . . . .

Name . . . .

Address. . . . .

. . . . .Postcode . . . .

Please send this order form, with your remittance, to Australian Personal Computer, Subscriptions Department, 462 Burwood Road, Hawthorn, Vic. 3122

PLEASE USE BLOCK CAPITALS

dBASE II

command files. The instructions look Basic-like at first but have more structure than is common in Basic. For instance, the IF statement must have a matching ENDIF, so you are allowed to nest these pairs. There is a limited form of GOTO, but only to jump to the end of a DO . . . WHILE loop — so it's basically 'no GOTO' programming without being obsessive about it, which I personally feel is about the right balance. There are also commands to allow flexible input from and display to the terminal, and fancy layout on the printer; one form of the input commands makes it easy to build menus. I found the command files easy to construct and use, and flexible in their application, though not quite as powerful as those of FMS-80. One helpful feature typical of the consistency found almost everywhere in dBaseII is that throughout, the commands fields are referred to by the names they are given in the record definition.

Stability and reliability

dBaseII has been in use for some time in this country and in the States. The previous release reportedly had some bugs in it; the latest release, which I tested, had a couple of glitches that I discovered. I didn't lose any data, or come close to it, but I did have trouble using the REPORT feature, nor could I get the command for editing command files to work properly. I am assured, however, that a revised version with the REPORT bugs corrected is on its way from the States and will be forwarded to end users.

User image

As distributed, dBaseII is not a menu-driven package, which means that the user must know the form of at least the simplest commands before he can get started. To some extent, it is a matter of taste whether you prefer menu-driven or user-initiated systems. Personally, I like to have the choice and in dBaseII you only have the option to construct menus — they aren't provided. However, the format of the commands was almost entirely internally consistent, and most operations only required two or three commands. For instance, to find particular records you simply issue one command to tell dBaseII which file to use, and another to specify the keys, and to limit the scope of the search and the display of variables. So I would expect a user with some motivation to find it pretty easy to use. This is largely confirmed by reports from two users, one who had a particular application in mind and became a fluent user in a few days, largely self-taught, and another who got hold of it 'to try it out' and never got very far.

The documentation comes in two parts. The first was written by an experienced user and provides a good, well-paced introduction yet goes right through to the most complex commands. The second part is a reference manual written by the software designer, but is of a much higher standard than usual. I thought the two-level approach a good idea which worked well but it's a pity there is no index for either part, only a list of commands and the pages on which they are described.

There are of course some sillies, though I had to work harder to find some than with previous packages I've seen. I don't like the use of the word QUIT to indicate normal ending of a session, especially when the keys CTRL-Q are used to abort in an EDIT command. Even the tutorial manual starts with 'how to install dBaseII for your terminal', including prompts such as 'are you going to use hex or decimal to specify . . .' at a stage when the user is hardly likely to know the difference. I feel that installation information should be in an appendix and no user should buy software from a dealer who is unable or unwilling to install it, unless the user is experienced enough to find and make sense of the appendix. And please, when are software writers going to make it easy to specify dates in the format preferred by the user, which is not necessarily that of the country of origin of the software?

dBaseII costs $695 (as quoted by Archive Computer Services) for the complete package, and it needs a standard CP/M system with a minimum of 48k to run it.

Conclusions

dBaseII is a powerful and flexible data management system, with a well-designed and consistent user interface. Its strengths and weaknesses reflect an approach typical of good software engineers. Among its strengths are the clear command design, which makes it possible to deduce the formats of commands you haven't used before from the familiar ones, which avoids the hierarchy problems of menu systems while remaining easy to use, and which uses the same format for the 'programming' commands so that the user's growth path is smooth and logical. The selection facilities are good and well integrated, the calculation facilities are excellent (even brackets are supported, not to mention real numbers) and the feature which allows you to request automatic updating of all indexes is invaluable.

On the other hand, the reporting features are more limited than those of more commercially oriented packages, unless one devises command files to construct more fancy reports. The lack of any protection against unauthorised access to files, in the software as provided, could be a problem in some settings. But if you can cope with the limitation of 32 fields per record, and a maximum of two data files in use at any one time, I think you would find dBaseII a powerful and flexible package on which to build a data management system.

'I still think it's a bit of a risk, leaving it on the doorstep all day.'

NEC PC-8000

A low cost, high performance microcomputer system designed for your needs..from $1,500

Includes sales tax

PC8000 combines
the most wanted
features of several
leading microcomputers
together with a few
wonders of its own

Check
these
features..

- 5 user-programmable function keys. Up to 10 different user-defined routines can be activated using these keys and the shift key.
- 82-key keyboard features function keys and numeric keypad.
- 248-symbol character set.
- Z-80A equivalent microprocessor, running at 4MHz.
- CP/M™ disk operating system compatibility.
- N-key rollover.
- Standard memory features 24K bytes of ROM and 32K bytes or RAM
- 64K RAM is available with the PC-8012AI/O unit.
- 160 x 100 matrix of high resolution graphics, powerful control directly from the keyboard.
- Interfaces include color monitor, monochromatic monitor, audio cassette, Centronics™ printer.
- Screen formats of 20 or 25 lines and 36, 40, 72 or 80 characters per line, with adjustable scrolling window.
- A powerful editor simplifies programming and includes [illegible]-way cursor control, character insert and delete.
- Terminal mode available with a single command.
- I/O unit for plug-in circuit boards.

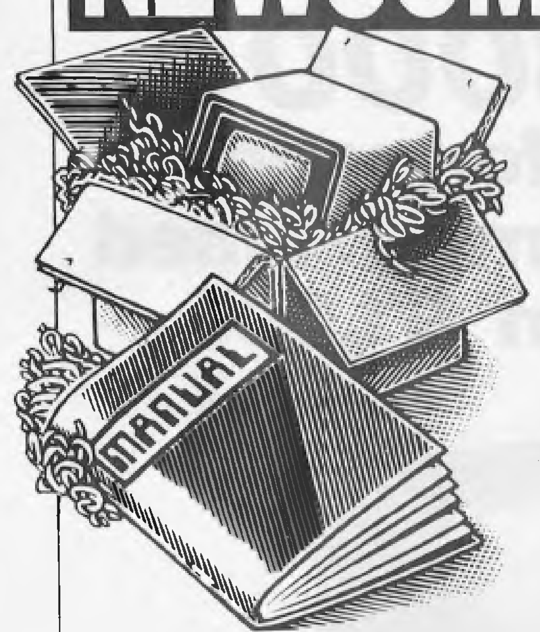
HANIMEX

means business

For Authorised Dealers phone:

SYDNEY 938-0400 ● MELBOURNE 64-1111 ● BRISBANE 262-7555 ● ADELAIDE 46-9031 ● PERTH 381-4622

NEWCOMERS START HERE



This is our unique quick-reference guide, reprinted every month to help our readers pick their way through the most important pieces of (necessary) jargon found in APC. While it's in no way totally comprehensive, we trust you'll find it a useful introduction. Happy microcomputing!

Welcome to the confusing world of the microcomputer. First of all, don't be fooled; there's nothing complicated about this business, it's just that we're surrounded by an immense amount of necessary jargon. Imagine if we had to continually say 'numbering system with a radix of 16 in which the letters A to F represent the values ten to 15' when instead we can simply say 'hex'. No doubt soon many of the words and phrases we are about to explain will eventually fall into common English usage. Until that time, APC will be publishing this guide — every month.

We'll start by considering a microcomputer's functions and then examine the physical components necessary to implement these functions.

The microcomputer is capable of receiving information, **processing** it, storing the results or sending them somewhere else. All this information is called **data** and it comprises numbers, letters and special symbols which can be read by humans. Although the data is accepted and output by the computer in 'human' form, inside it's a different story — it must be held in the form of an electronic code. This code is called **binary** — a system of numbering which uses only 0s and 1s. Thus in most micros each character, number or symbol is represented by eight binary digits or **bits** as they are called, ranging from 00000000 to 11111111.

To simplify communication between computers, several standard coding systems exist, the most common being **ASCII** (American Standard Code for Information Interchange). As an example of this standard, the number five is represented as 00110101 — complicated for humans, but easy for the computer! This collection of eight bits is called a **byte** and computer freaks who spend a lot of time messing around with bits and bytes use a half-way human representation called **hex.** The hex equivalent of a byte is obtained by giving each half a single character code (0—9, A—F): 0=0000, 1=0001, 2=0010, 3=0011, 4=0100, 5=0101 . . . . E=1110 and F=1111. Our example of 5 is therefore 35 in hex. This makes it easier for humans to handle complicated collections of 0s and 1s. The machine detects these 0s and 1s by recognising different voltage levels.

The computer processes data by reshuffling, performing arithmetic on, or by comparing it with other data. It's the latter function that gives a computer its apparent 'intelligence' — the ability to make decisions and to act upon them. It has to be given a set of rules in order to do this and, once again, these rules are stored in **memory** as bytes. The rules are called **programs** and while they can be input in binary or hex (**machine code** programming), the usual method is to have a special program which translates English or near-English into machine code. This speeds programming considerably; the nearer the **programming language** is to English, the faster the programming time. On the other hand, program execution speed tends to be slower.

The most common microcomputer language is **Basic**. Program instructions are typed in at the keyboard, to be coded and stored in the computer's memory. To **run** such a program the computer uses an **interpreter** which picks up each English-type **instruction**, translates it into machine code and then feeds it into the **processor** for execution. It has to do this each time the same instruction has to be executed.

Two strange words you will hear in connection with Basic are **PEEK** and **POKE**. They give the programmer access to the memory of the machine. It's possible to read (PEEK) the contents of a byte in the computer and to modify a byte (POKE).

Moving on to **hardware**, this means the physical components of a computer system as opposed to **software** — the programs needed to make the system work.

At the heart of a microcomputer system is the central processing unit (**CPU**), a single microprocessor chip with supporting devices such as **buffers**, which 'amplify' the CPU's signals for use by other components in the system. The packaged chips are either soldered directly to a printed circuit board (**PCB**) or are mounted in sockets.

In some microcomputers, the entire system is mounted on a single, large, PCB; in others a **bus system** is used, comprising a long PCB holding a number of interconnected sockets. Plugged into these are several smaller PCBs, each with a specific function — for instance, one card would hold the CPU and its support chips. The most widely-used bus system is called the **S100**.

The CPU needs memory in which to keep programs and data. Microcomputers generally have two types of memory, **RAM** (Random Access Memory) and **ROM** (Read Only Memory). The CPU can read information stored in RAM — and also put information into RAM. Two types of RAM exist — **static** and **dynamic**; all you really need know is that dynamic RAM uses less power and is less expensive than static, but it requires additional, complex, circuitry to make it work. Both types of RAM lose their contents when power is switched off, whereas ROM retains its contents permanently. Not surprisingly, manufacturers often store interpreters and the like in ROM. The CPU can only read the ROM's contents and cannot alter them in any way. You can buy special ROMs called **PROMs** (Programmable ROMs) and **EPROMs** (Eraseable PROMs) which can be programmed using a special device; EPROMs can be erased using ultra-violet light.

Because RAM loses its contents when power is switched off, **cassettes** and **floppy disks** are used to save programs and data for later use. Audio-type tape recorders are often used by converting data to a series of audio tones and recording them; later the computer can listen to these same tones and re-convert them into data. Various methods are used for this, so a cassette recorded by one make of computer won't necessarily work on another make. It takes a long time to record and play back information and it's difficult to locate one specific item among a whole mass of information on a cassette; therefore, to overcome these problems, **floppy disks** are used on more sophisticated systems.

A floppy disk is made of thin plastic, coated with a magnetic recording surface rather like that used on tape. The disk, in its protective envelope, is placed in a disk drive which rotates it and moves a **read/write head** across the disk's surface. The disk is divided into concentric rings called **tracks**, each of which is in turn subdivided into **sectors**. Using a program called a **disk operating system**, the computer keeps track of exactly where information is on the disk and it can get to any item of data by moving the head to the appropriate track and then waiting for the right sector to come round. Two methods are used to tell the computer where on a track each sector starts: **soft sectoring** where special signals are recorded on the surface and **hard sectoring** where holes are punched through the disk around the central hole, one per sector.

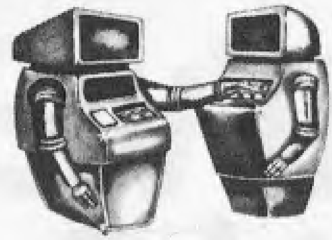
Half-way between cassettes and disks is the **stringy floppy** — a miniature continuous loop tape cartridge, faster than a cassette but cheaper than a disk system. **Hard disk** systems are also available for micro-computers; they store more information than floppy disks, are more reliable and information can be transferred to and from them much more quickly.

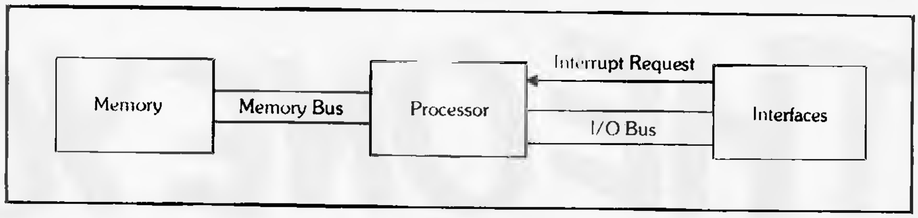
You, the user, must be able to communicate with the computer and the generally accepted minimum for this is the visual display unit (**VDU**), which looks like a TV screen with a typewriter-style **keyboard**; sometimes these are built into the system, sometimes they're separate. If you want a written record (**hard copy**) of the computer's output, you'll need a **printer**.

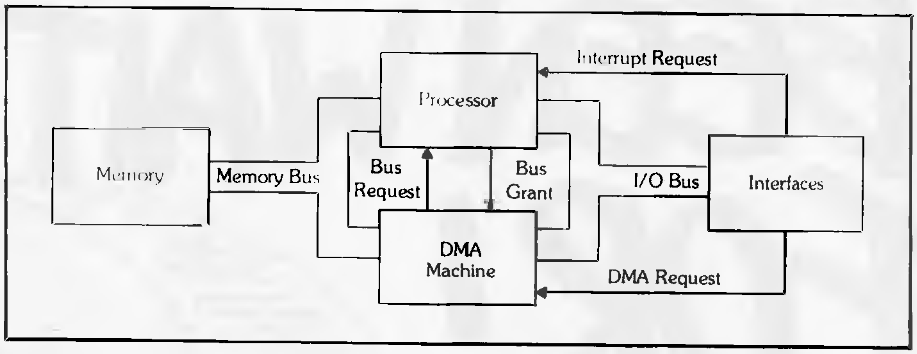
The computer can send out and receive information in two forms — **parallel** and **serial**. Parallel input/output (**I/O**) requires a series of wires to connect the computer to another device, such as a printer, and it sends out data a byte at a time, with a separate wire carrying each bit. Serial I/O involves sending data one bit at a time along a single piece of wire, with extra bits added to tell the receiving device when a byte is about to start and when it has finished. The speed that data is transmitted is referred to as the **baud rate** and, very roughly, the baud rate divided by ten equals the number of bytes being sent per second.

To ensure that both receiver and transmitter link up without any electrical horrors, standards exist for serial interfaces; the most common is **RS232** (or **V24**) while, for parallel interfaces to printers, the **Centronics** standard is popular.

Finally, a **modem** connects a computer, via a serial interface, to the telephone sytem allowing two computers with modems to exchange information. A modem must be wired into the telephone system and you need Telecom's permission; instead you could use an **acoustic coupler**, which has two obscene-looking rubber cups into which the handset fits, and which has no electrical connection with the phone system — Telecom isn't so uppity about the use of these







Diagrams above illustrate the differences between a system that does not include a direct memory access machine, top, and one that does, bottom.

machine. All that remains is to select which device will have control of the busses, the processor or the DMA machine.

CONTROLLING THE DMA

Normally, the processor will have control of the busses because the DMA I/O must be infrequent enough to allow at least some processing to be done. It is therefore necessary for the DMA machine to acquire bus control from the processor whenever necessary.

The processor can enable the DMA machine to request bus control, but it is the interface that must actually request service through the DMA machine. Only the interface knows when the attached peripheral requires DMA service. Thus we must add some connecting signals between the interface and the DMA machine, and between the DMA machine and the processor.

DMA HANDSHAKE

The interface must have some means of requesting service from the DMA machine. A signal called DMA Request (DMAR), added to the collection of signal lines on our I/O bus, will be sufficient. Upon receipt of this request, the DMA machine must request bus control from the processor.

The processor may decide that the time of the request is inopportune and wish to hold off the transfer of control temporarily – This is a job for the ever-present handshake!

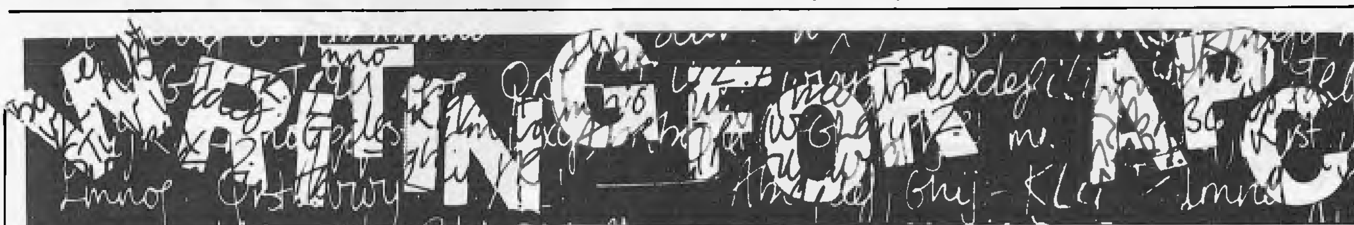
We will .create two handshake lines called Bus Request and Bus Grant. The DMA machine will ask for bus control with Bus Request and wait to actually take control until it receives a signal on Bus Grant. Thus the processor can maintain control of the memory and address busses as long as required.

BURST AND CYCLE-STEAL

The DMA that we have been discussing is called burst DMA because data transfer is done in a burst where the DMA machine totally controls the I/O with the full speed of the memory bus at the expense of completely halting any processor activity.

If half the memory bus bandwidth is sufficient to solve the high speed I/O problem, another type of DMA can be employed. Called cycle-steal DMA, the DMA machine alternates control of the busses with the processor, each unit using every other memory cycle. Cycle-steal DMA allows the processor to operate at 50% efficiency while still providing relatively high speed I/O.

At this point in the I/O series, we have discussed the basic hardware needed for interfacing computers to peripherals. We have covered the four basic interfaces: Parallel, BCD, HP-IB, and Serial and we have discussed specialized I/O; interrupt and DMA. Now that we have our devices talking, we will discuss how to overcome the language barrier. Next issue: character codes.



APC welcomes approaches from would-be writers, even those who may never have appeared in print before. In this game it is often those with practical experience who have important things to say so we don't mind too much if their prose is less than perfect. Providing that submissions have a sensible structure and follow a logical sequence, we can take care of the polishing. Here are some tips:

If the article is already written, simply send it in, making sure that your name, address and 'phone number appear on both the article and the covering letter. If you have submitted the same work to other magazines you should tell us – it would be embarrassing (to say the least) if the same article appeared in more than one.

If you have an idea for an article or a series, write us a letter outlining your ideas. A one or two page synopsis giving the proposed structure, sequence and content will give us a sound basis for discussion. Please give us a daytime 'phone number if possible.

If you have nothing specific in mind but feel qualified to conduct case studies, Benchtests or whatever then drop us a line saying what you'd like to do and why you think you're qualified to do it. We're not particularly looking for strings of academic qualifications – experience carries just as much weight.

Dick Pountain is always on the look out for interesting calculator features and we wouldn't mind seeing one or two readers getting on their soapboxes but remember: even articles such as this need a structure.

Reading *APC* will give you a good idea of the style we prefer. You may notice that we try to avoid pomposity at one extreme and flippancy at the other.

Finally, have a look through back issues indexes and try not to re-invent any wheels.

THE ONE YOU'VE BEEN WAITING FOR!

Australian Personal Computer
SPECIAL ISSUE

MICROCOMPUTER BENCHTEST SPECIAL

AUSTRALIA'S LARGEST SELLING MICROCOMPUTING TITLE BRINGS YOU COMPREHENSIVE BENCHTESTS ON 21 LEADING MICROCOMPUTERS

IN ONE COMPLETE VOLUME

USE THE COUPON BELOW TO ORDER YOUR COPY OF APC'S BENCHTEST SPECIAL

AVAILABLE END OF JUNE.

YES, I'VE BEEN WAITING FOR IT. PLEASE SEND ME ____ COPIES OF APC'S BENCHTEST SPECIAL. I ENCLOSE $3·80 TO BENCHTEST SPECIAL OFFER, 462 BURWOOD RD. HAWTHORN. 3122. CHEQUES SHOULD BE MADE PAYABLE TO — AUSTRALIAN PERSONAL COMPUTER

NAME ______________________

ADDRESS ______________________

____

____

____

FRAMES OF REFERENCE

A DP MANAGER'S GUIDE TO MICROS

By Alan Wood

PART 5 HARDENING ON THE HARDWARE

George Orwell had a vision that we would all be supervised by Big Brother in 1984. The micro industry has a view that we will all be assisted by Little Brother in about the same timescale. DP departments have, subsconsciously at least, been identified as the agents of Big Brother, offering increasingly vast, inflexible and expensive systems with which users are unable or cannot afford to tamper.

The rapid acceptance of a desktop micros in big companies is in part a reaction against mainframe/DP department/Big Brother. The further progress towards Little Brother on all desktops with access to big databases is the likely outcome of revised DP attitudes, the incessant demands of users and the pressure of the new technology. There are conceptually two steps to take the large organisation into the Little Brother era.

Step 1 is the computer literacy programme. For the motor car to become a utility automobile manufacturers had to break the everyman price barrier. And when the barrier was broken, everyman had to learn how to drive. The micro manufacturers have broken every person's computer price barrier. but not every person knows how to 'drive' a computer. A massive programme of, mainly, self-education is well underway. The first wave of cheap microcomputers, as well as doing useful jobs, can be regarded as the main impetus towards computer literacy in companies. The Apples, PETs, Osbornes and SuperBrains are reaching and teaching parts that DP has never penetrated.

Step 2 takes place at what I call the 'computer desk'. The computer desk of the mid-'80s will provide office workers with access to automated filing cabinets, electronic inter-office communication and external databases, be they on the organisation's mainframe or on bureau machines. The computer desk will comprise all the software programs for office, managerial and professional staff and will provide multi access through networked facilities. There will be no practical limit to RAM and add-on store: you will have as many calculators and filing cabinets as you can afford/justify. The computer desk will offer new solutions to cheap mass store (winchester plus video disk) and new solutions to data entry (voice and direct image entry). In conventional computing terms we will be using 16 or 32-bit engines in desk enclosures with 1 to 2 megabytes of RAM. We will have strung fast communication cables and connect points around our buildings on a par with our telephone systems.

No crystal ball is needed to see these two steps towards mass computerisation. What is needed is time. Many of the elements for Step 2 are already with us in the form of raw technology. But they are seldom found altogether in one coherent, reliable and affordable system. And if they were, many organisations would not be ready because they have not even begun the first faltering step towards computer literacy. Some smaller companies who have never previously computerised will get the computer desk first; some large organisations (BP, ICI,) are well into experi mental systems.

Standard hardware

DP executives brought up on ICL bread and IBM milk have a craving to shelter in some standard micro house. There is no such house and, with entry costs a fraction of mainframe costs, there is no need to start on an odyssey to find one. Micros range from the small personal computer to the powerful networked facility, and no one company offers a credible range from top to bottom. Your standard, your bridge and your comfort is in software not in hardware. That said, there are some valid pointers to selection of standard hardware.

1. Be prepared to discard cheap experiments. We are surprised by the number of DP executives who have experimented on home computers and are not prepared to discard them in favour of more powerful, better-suited office machines.
2. Be prepared to write off computer literacy computers. You can afford to sprinkle your organisations with Osbornes as education and personal productivity aids. In two to three years they will be homework computers, terminals on networks and education tools. Just be prepared to write them off in that timeframe.
3. Don't get the body of your micro thinking set in an IBM-like straightjacket. Experimentation with several machines is a valid process towards assessing the best and seeing how they develop. Standard software will keep your options open.
4. Don't let your supplier think for you. Frankly, there are far too many developments in technology for any one supplier to have a monopoly — or even a majority. Mainframe suppliers offering micros are late in the field, more expensive and considerably less fleet of foot with new offerings than the specialist micro suppliers.

Microcomputer architecture

Two schools of architecture (the S100 bus and the proprietory/own bus) have been adopted by microcomputer manufacturers. The common operating systems and languages are available on both schools so that the choice you make will not inhibit your software standards.

Many microcomputers use separate printed boards for the processor, memory, interfacing, disk controllers, etc. These boards are connected by means of a bus, the standard one being the S100 bus which simply provides 100 internal connecting points. In a typical single-user microcomputer, about half of those connecting points are used.

The S100 bus has been adopted by the Institute of Electrical and Electronic Engineers (IEEE) and there is now an international standard for the bus. More than 200 manufacturers make products that can be plugged into S100 computers, e.g. graphics cards, memory cards, interface cards, videotex cards. As new technologies develop they frequently appear in S100 form because of the large market for products. The advantage of the S100 Bus can be summarised as: an international hardware standard; access to a variety of cheap add-on and alternative boards; easy to configure from single to multi user; maintenance and fault finding made easy by board replacement.

The disadvantages of the S100 bus stem from its origins as an amateur design. There were flaws in the original design which have been reduced in subsequent implementations. As always, there is a trade-off to be made by adopting a standard. The disadvantages can be summarised thus: it's less reliable than single board architecture and it's less suited to more powerful processors than special-purpose architectures.

Apple software from Micro-Educational

APPLE SOFTWARE APPLE PERIPHERALS
FROM MICRO–EDUCATIONAL

Australia's leading mail-order supplier for Apple Software and Peripherals. We offer wide range, phone advice, fast delivery ex stock, 10-day money-back warranty on Software and very competitive pricing.

HARDWARE (S.Tax exempt prices. Add 17.5% if S.Tax applies.)

APPLESPEAKER	195.00
NATIONAL 12" B&W MONITOR	135.00
SANYO 12" GREEN SCREEN	339.50
SANYO 9" GREEN SCREEN	339.50
STAR PRINTER	550.00
EPSON MX-80+I/F+CABLE	1050.00
EPSON MX-80TYPE2+I/F+CABLE	1175.00
EPSONMX-80F/T+I/F+CABLE	1225 00
MX-80F/T TYPE 2+I/F+CABLE	1275.00
EPSON MX-100+I/F+CABLE	1495.00
ANDROMEDA 16k EXP CARD	195.00
LEGEND 64K EXPANSION CARD	395.00
MICROSOFT Z80 SOFTCARD	379.50
VISTA VISION 80	295.00
PAL COLOUR CARD	225.00
CCS ASYNCH SERIAL CARD	185.00
CCS PARALLEL CARD	159.50
CCS A/D	145.50
CCS CLOCK/CALENDAR	149.50
CCS COMMUNICATIONS CARD	195.00
CCS GPIB IEEE 488	320.00
CCS ARITHMETIC PROCESSOR	495.00
MICROSPEED I	550.00
MICROSPEED II	725.00
MH CPS MULTIFUNCTION CARD	275.00
WESTSIDE SUPERCLOCK	195.00
ELECTROMED SENDATA COUPLER	325.00
PAYMAR LOWER CASE (+ DISK)	69.95
3,3 DOS UPGRADE	64.50
DOS SWITCH	27.50
16K RAM UPGRADE	49.95
LIBRARY BOX	4.50
T&G PADDLES	44.95
EXPANDAPORT	42.95
3G LORES LIGHT PEN	42.95
JOYSTICK	69.95
ABT BARCODE READER	199.50
NUMERIC KEYPAD	159.50

SOFTWARE (S. Tax Paid)

ADMINISTRATION

PFS	150.00
PFS:REPORT	125.00
VISICALC 3.3	375.00
VISIPLOT/TREND	440.00
VISIDEX	375.00
VISIFILE	375.00
VISI SCHEDULES	440.00
* TARGET BUSINESS PLANNER	275.00
REAL ESTATE ANALYSER	193.00
CONTEXT CONNECTOR	215.00
THE GENERAL MANAGER	121.00
DB MASTER	270.00
DB MAS. UTILITY PAK 1	121.00
DB MASTER STAT PAK	121.00
WINCHENDON STATS DATABASE	242.00
* WORDSTAR	385.00
SANDY'S WORD PROCESSOR	193.00
* SPELLBINDER	545.00
* THE WORD	110.00
SUPERSCRIBE II	160.00
THE DICTIONARY	121.00
SCHOOL TIMETABLE	55.00
TIME MANAGEMENT	215.00
MICRO MEMO	44.00
* DBASE II	770.00
CORP SYSTEM	209.00
THE HOME ACCOUNTANT	91.00

EDUCATION

APPLE PILOT	193.00
FRACTIONS	44.00
DECIMALS	44.00
ARITHMETIC SKILLS	55.00
ELEMENTARY MATHEMATICS VI	55.00
CARTELS & CUTTHROATS	49.00
SENTENCE DIAGRAMMING	28.00
LINGUIST	44.00
TELLSTAR II	94.00
THE PLANETARY GUIDE	37.00
COMPU-SPELL	44.00
TYPING TUTOR II	31.00
MASTERTYPE	42.00

WARP FACTOR	43.95
TORPEDO FILE	65.95
COMPUTER AIR COMBAT	65.95
OPERATION APOCALYPSE	65.95
SHATTERED ALLIANCE	65.95
BATTLE OF SHILOH	43.95
PRESIDENT ELECT	43.95
TIGERS IN THE SNOW	43.95
COMPUTER BASEBALL	43.95
SOUTHERN COMMAND	43.95
NAPOLEON'S CAMPAIGNS	65.95
CONFLICT	27.50
OLYMPIC DECATHLON	29.95
GLOBAL WAR	29.95
HIRES 0 (MISSION ASTEROID)	23.95
HIRES 1 (MYSTERY HOUSE)	27.50
HIRES 2 (WIZARD & PRINCESS)	37.50
HIRES 3 (CRANSTON MANOR)	39.95
HIRES 4 (ULYSSES & GOLDEN FL)	39.95
HIRES 5 (TIME ZONE)	109.95
SOFTPORN ADVENTURE	32.95
STREETLIFE	32.95
ZORK	46.95
ULTIMA	43.95
OO TOPOS	37.50
CYBORG	37.50
GALACTIC EMPIRE	29.95
GALACTIC TRADER	29.95
GALACTIC REVOLUTION	29.95
TAWALA'S LAST REDOUBT	32.95
CRUSH CRUMBLE CHOMP	32.95
AKALABETH	39.95
STAR WARRIOR	42.95
ROBOT WAR	39.95
CASTLE WOLFENSTEIN	32.95
DUNGEON CAMPAIGN/WILDERNESS	34.95
DRAGON'S EYE	24.95
WIZARDRY	54.95
TARTURIAN	29.95
GOBLINS	29.95
CREATURE VENTURE	29.95
ALKENSTONE	43.95
KAVES OF KARKHAN	43.95
SORCERER OF SIRA	31.95
DARK FOREST	32.95
STONE OF SISYPHUS	32.95
ADVENTURE IN TIME	32.95
BIRTH OF THE PHOENIX	19.95
RACE FOR MIDNIGHT	32.95
SPACE ADVENTURE	32.95
SARGON II	37.95
BACKGAMMON	31.95
BRIDGE	29.95
MONOPOLY	39.95
SCRABBLE	39.95
REVERSI	32.95
HIRES CRIBBAGE	29.95
HIRES GOLF	32.95
RUBIK'S CUBE SOLVER	27.50

ACTION AND ARCADE GAMES

THE ARCADE MACHINE	49.95
RASTER BLASTER	32.95
DAVID'S MIDNIGHT MAGIC	36.50
POOL 1.5	37.50
TRICK SHOT	43.95
GORGON	42.95
APPLE-DIDS	32.95
PHANTOMS FIVE	37.50
SPACE EGGS	32.95
STAR THIEF	32.95
FALCONS	32.95
APPLE PANIC	32.95
SNOGGLE (PAC-MAN)	29.95
HEAD ON	27.50
INTERNATIONAL GRAND PRIX	32.95
AUTOBAHN	32.95
FENDER BENDER	29.95
HIRES FOOTBALL	43.95
HIRES SOCCER	34.95
MISSILE DEFENSE	32.95
GOBBLER	32.95
THIEF	31.95
BILL BUDGE SPACE ALBUM	44.95
EPOCH	37.50
SNEAKERS	32.50
BEER RUN	31.95
SNACK ATTACK	31.95
PULSAR	32.95
STAR CRUISER	29.95

ORBITRON	32.95
OUTPOST	32.95
RED ALERT	31.95
RINGS OF SATURN	43.95
ESCAPE FROM ARCTURUS	38.95
SHADOW HAWK 1	34.95
HADRON	38.50
CYBER STRIKE	43.95
COPTS AND ROBBERS	39.95
SPACE QUARKS	32.95
SPACE RAIDERS	29.95
SHADOWHAWK I	54.95
ABM	29.95
GENETIC DRIFT	32.95
TRACK ATTACK	32.95
BUG ATTACK	32.95
FIREBIRD	32.95
SHUFFLEBOARD	32.95
GAMMA GOBLINS	32.95
SABOTAGE	29.50
PEGASUS II	32.95
THRESHOLD	43.95
CROSSFIRE	32.95
MAZEMAN	32.95

GRAPHICS

BUDGE GRAPHICS	49.00
SUBLOGIC 3D GRAPHICS	77.00
SUBLOGIC 3D GRAPHICS ENHAN	33.00
SUBLOGIC GRAPHICS EDITOR	44.00
HIGHER TEXT	44.00
HIGHER GRAPHICS II	44.00
COMPLETE GRAPHICS SYSTEM	77.00
DATASOFT MICROPAINTER	44.00
EZ DRAW 3.3	64.00
** PASCAL GRAPHICS EDITOR	121.00
GRAPHTRIX	83.00

UTILITIES & LANGUAGES

GRAFPAK	105.00
STATGRAPH FILER	44.00
DISK FIXER	37.00
MONITOR EXTENDER 2006	28.00
** THE INSPECTOR	61.00
3.3 PROGRAMMING AIDS	99.00
DOSOURCE	72.00
DOS BOSS	31.00
EXPEDITER COMPILER	121.00
HAYDEN COMPILER	242.00
MICROSOFT COMPILER	215.00
APPLE PILOT	209.00
APPLE FORTRAN	242.00
* MICROSOFT FORTRAN	242.00
FORTH II	110.00
TRANSFORTH II	149.00
ALD SYSTEM II	149.00
SUPERKRAM	187.00
PHIL'S CHEAP EDITOR	66.00
LISP	121.00
TINY PASCAL	61.00
HANDHOLDING BASIC	176.00
LISA ASSEMBLER 2.0	69.00
APPLE-CILLIN	55.00
ACTION SOUNDS & HIRES SCRL	28.00
THE VOICE	49.00
ASCII EXPRESS	88.00
* Z-TERM (PROFESSIONAL)	187.00
THE SOURCE	154.00
VISITERM	193.00

BOOKS

ASSEMBLY LANGUAGE (HYDE)	28.00
APPLE MACHINE LANGUAGE	20.00
BENEATH APPLE DOS	28.00
APPLE II USERS GUIDE	22.00
MOSTLY BASIC (APPLICATIONS)	20.00
PASCAL PROGRAMMING (LEWIS)	20.00
APPLESOFT LANGUAGE	15.00
APPLE BASIC FOR BUSINESS	22.00
COMPUTER GRAPHICS PRIMER	22.00
APPLE MONITOR PEELED	28.00
WHAT'S WHERE IN THE APPLE	24.95
COMPUTERS FOR KIDS	9.00

SUBSCRIPTIONS

SOFTALK — MONTHLY NSW 48.00
All other States 60.00

* (Requires Z80 card)
** (Requires 64K)

For our free catalogue send the attached coupon or write to:
MICRO–EDUCATIONAL, 17 PARK ROAD, GARDEN SUBURB, NSW 2288.

FRAMES OF REFERENCE

The S100 standard is specially useful in technical, scientific and educational establishments. The knowledgeable user can get into the machine, change the boards and do his own maintenance. On the other hand, the office user will rely on an engineer and will seldom look under the bonnet of his machine. Nevertheless, ease of maintenance and expansion by board replacement are important factors in selecting office computers. For example, upgrading to a multi-user system is simplified with S100 add-on boards.

Few of the widely used personal computers use the S100 bus, primarily because it is not compact enough and offers more facilities than are required. The following are among the manufacturers using it: Cromemco, North Star and Vector Graphics.

Some microcomputer manufacturers have developed their own bus structures, particularly the personal computer vendors. Many put all the essential electronic components of the computer on a single printed circuit board. Although there may be provision for adding special boards, they are basically single board machines.

The primary virtues of the 'own bus' machines are compactness, good reliability record, special purpose design. Disadvantages compared to S100 architecture are that it's more difficult to diagnose faults and maintain them, they're more difficult to re-configure and there's less choice of hardware add-ons.

The following are some of the machines with their own bus structure: Apple, Atari, PET and Zilog.

8-bit processors

Two processors dominate the 8-bit microcomputer market — the MOS Technology 6502 and the Zilog Z80. The 6502 is widely used in personal computers. The best-known are the PET and Apple, with Atari a more recent member of the club. CP/M is not available on the 6502 but there are CP/M adaptors for PET (Softbox) and Apple (Softcard) which introduce a Z80 processor for those popular machines.

The Zilog Z80 processor has outsold even the 6502 and is to be found in such machines as the Tandy and the Sinclair at the lower end and just about all the popular office microcomputers. The Z80 incorporates the instruction set of Intel's 8080 and provides a faster processor. CP/M is widely available. All the S100 micros mentioned use the Z80 processor.

A few years ago it was not apparent that the Motorola 6800 and the Intel 8080 would be also-rans in commercial microcomputing. Both these processors are widely used in intelligent controllers and industrial devices. A few computer manufacturers have used them, e.g. Rair uses an 8085 processor and South West Tech, one of the early micro suppliers, uses the Motorola chip.

16-bit processors

The market for 16-bit processors is a battlefield from which no clear victor has yet emerged.

Intel was first in the field and its 8086/88 processor has been adopted by a number of the existing Z80 suppliers, e.g. Altos, as well as by some of the new entrants: IBM, Sirius. The 8088 cheap entry point offers only 8-bit external data paths but with 16-bit internal data handling. Available on IBM and the Sirius, it is not much more powerful than the Z80 but it does offer larger internal RAM capability. CP/M-86 is available for the 8086/88.

The Motorola 68000 chip offers a bridge from 16 to 32-bit and is being preferred by the more sophisticated system vendors, particularly those offering Unix. CP/M is not available yet, but a more advanced operating system is needed to take full advantage of the features of the chip and bridge to 32-bit. Manufacturers offering Motorola-based systems include Cromemco and Tandy. Apple is rumoured to be working on one also but after the Apple III difficulties it will be careful about premature release.

Zilog, manufacturer of the Z8000, has been one of the first to market a credible Unix system. Its system outperforms the PDP 11/44 and compares favourably with the PDP 11/70 minis, at a fraction of the cost. The Zilog processor has not been widely adopted by other suppliers. Onyx being one of the few Z8000 users.

The 16-bit market is settling down into two dominant streams: Intel for mass 16-bit with a big impetus from IBM and Motorola for the more 'sophisticated' user, particularly spreading Unix and needing an impetus from a volume supplier, e.g. Apple, Tandy.

Establishment versus specialist suppliers

Advantages of establishment micros (eg IBM, HP)

* Security that your supplier is going to stay in business
* Communication links with own mainframe/minis.

Disadvantages of the establishment's micros

* A further tie in to your main vendor.
* A lost pressure point for better service and pricing.
* Much more expensive hardware and software.
* Slow reaction to new low cost technology.
* Less flexibility and fewer options.

Advantages of the specialist micro suppliers

* Keen pricing kept sharp by competition.
* Pursuit of industry standard hardware and software.
* Large and growing bank of inexpensive software.
* Record of innovation.

Disadvantages of specialist micro suppliers

* Only the best will thrive.
* Comms software not so quickly available except for IBM protocols.

Mixing mainframes and minis with micros

Understandably the mainframe and minicomputer suppliers feel threatened by the advancing hordes of micros. Genghis Khan could not have struck more fear into his enemies than the fright the technology has given the computer corporation executive. They are beginning to respond: some with carefully measured defences, some with ill-equipped ramparts and others by buying off the enemy. Their main hope for success lies in offering a coherent set of systems to their loyal followers.

At first sight, Data General appears to have a coherent set of systems. It made its mark in minis by offering an upward compatible set of software and it has continued this philosophy with cut-down versions of its software. The Enterprise and MPT desktop machines are based on the old technology 16-bit microNova and run under MP/OS, a cut-down version of AOS.

Data General introduced the Enterprise in the US with business packages developed by one of its OEMs on the bigger Nova range. The Enterprise has not been a commercial success. The offering is less competitive than the specialist micro suppliers with a lot less software. They may be attractive to some existing users but Data General has not struck the mass market chord. It has so far ignored the cheaper and more powerful new microprocessors, as well as the CP/M software bandwagon. It does not presently offer a bottom end home/personal computer although it could release a Z80 plug-in card to partially rectify the omission.

DEC has been offering microcomputers based on the LSI 11 chip set for some while. The offerings range from a dual 8in floppy system up to a 40 megabyte/8-user PDP 11/23. The LSI 11 has been very successful but is now dated in comparison with the new 16-bit processors. DEC has developed a special 12-bit chip for its personal computer, the DECmate, which is selling in the US through its computer stores to offices, small business and professional users. The DECmate executes existing PDP/8 software but does not access the cheap micro software bank. For those who want a CP/M machine, DEC offers a board that converts the VT100 terminal into a Z80 desktop computer supporting two floppy disks.

One tends to think of a large computer supplier as one company. In truth, because of size, large suppliers operate in divisions and departments. Consequently it is not so surprising that the

OUR GAMES ARE SO EXCITING THAT OUR CUSTOMERS ARE NERVOUS WRECKS...

The games depicted are for 16K TRS80/System 80.

But they still keep coming back to COMPSHOP!

What can Compshop do for you? PLENTY! We are the best games software distributor in Australia today. We import the best games from overseas at great prices AND we have a team of fun-loving programmers who spend their days dreaming up devious games to trip you up.

We sell by Mail Order and through our hideaway shop in South Melbourne. (Dealer enquiries welcome.)

Write or call us and we will send you a new catalogue for your computer be it TRS80/System 80, Atari, Sinclair, Apple or Vic.

The games above are only a fraction of our range of TRS80/System 80 games (obviously we couldn't fit them all on the page, let alone what we have for the other micros!). Join our user/subscriber list and keep well-informed as new products become available.

FREE CASSETTE For 1 Month Only!

To: COMPSHOP AUSTRALIA
4/75 Palmerston Cres., Sth. Melbourne 3205.
Ph: (03) 690 5336.

I have a .... micro
☐ Please send me your new software catalogue. (Just send us a 27c stamp.)
☐ Please send me ....
AND a FREE blank cassette with my order.
I enclose a cheque/money order for $ ....
PLUS $1 for postage.
Please debit my bankcard A/C for $ ....

A/C No. ....

Commencement date .... Expiry Date ....

Name ....

Address ....

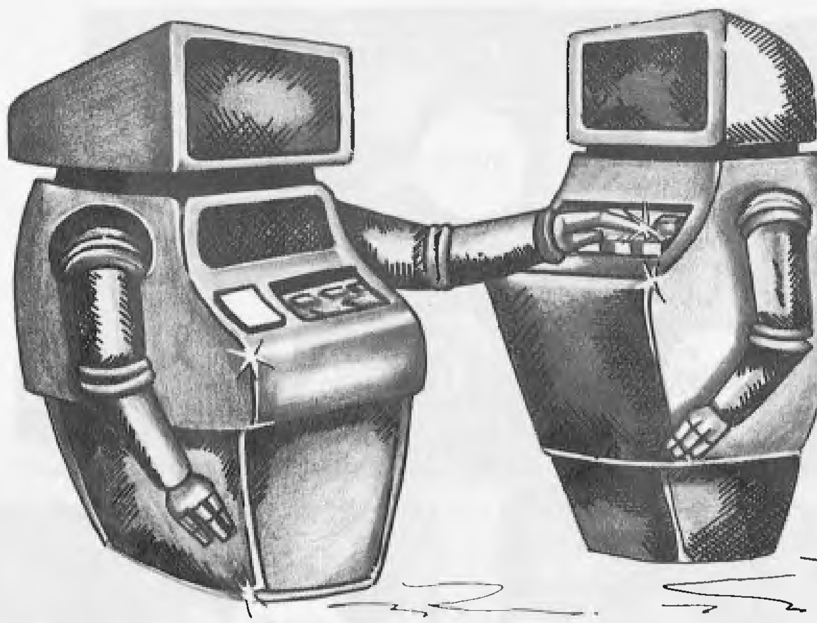
.... P'code ....

Signature ....

Compshop guarantees to replace any faulty tapes immediately!

APC/6

HOW COMPUTERS COMMUNICATE.



PART VIII

DMA: THE I/O SUPERHIGHWAY

by Hewlett-Packard's Steve Leibson

The articles in this series have described the hardware and circuitry necessary to interface peripheral devices with computers.

All the discussions thus far have assumed that the computer processor is in control of the data transfer process. This is true for many of the devices interfaced. The processor is usually fast enough that the peripheral device determines the data transfer rate.

Some devices, however, are too fast for processor-controlled I/O. These devices are capable of data rates approaching the speed of the computer memory and require a different I/O technique. The technique for interfacing such fast peripherals is called direct memory access (DMA).

In the previous article, we discussed interrupt I/O, which is used for interfacing with devices so slow that it is very inefficient to have the processor wait for the completion of each I/O transfer.

Instead, the processor initiates a transfer and then continues with other processing. When the peripheral device is ready for the next transaction, it interrupts the processor and reminds it of the previous I/O committment.

The interrupt I/O technique is used as a software transformer to match the slow peripheral with the fast processor. If the peripheral device is faster, the computer processor may only be able to execute the few machine instructions necessary to perform the I/O transfer before the peripheral is ready for another. Here there is a good match between the I/O software and peripheral speeds, and programmed I/O is sufficient for the task.

SPEEDY PERIPHERALS

Ultimately, there is a class of peripherals too fast for even the few instructions needed to perform programmed I/O. As long as these peripherals are not faster than the computer's basic memory cycle, there should be a method for performing the required I/O. There is and it is called direct memory access (DMA).

In order to discuss DMA and how it works, we must return to the model of the processor-memory-I/O system discussed in the first article in this series. Recall that the processor is linked to the memory via a set of lines called a memory bus and to the I/O interfaces via an I/O bus.

Both busses require the processor to generate address signals and control signals to synchronize the flow of data over these busses. Generally, I/O consists of taking information from the interfaces through the I/O bus and transmitting this information to the memory using the memory bus or vice versa.

INEFFICIENT THROUGHPUT

During this transfer the processor is also using the memory and memory bus to supply machine instructions so that it knows how to effect the data transactions. If we assume that it takes only nine machine instructions to perform one data transaction, we can see that the effective I/O throughput is only 10% of the rate that the memory could support.

That is, for every 10 memory cycles, nine are used to instruct the processor and only one is used to place data for I/O. Only very simple data transactions can be performed with nine machine instructions. If formatting or code conversions are necessary, many more instructions are needed.

BYPASSING THE PROCESSOR

The only way to speed up the I/O process is to eliminate the slowest link in the data path. For high-speed peripherals, the slowest link is clearly the processor itself! How can we eliminate the processor when that is the component that links the I/O and memory busses and is required for the generation of the signals that actually make these busses work?

The answer is to build a specialised circuit that is designed to transfer data at the full memory speed. Because the only function this circuitry must perform is this transfer, the capability may be wired into the circuit and instructions from memory are not needed and do not reduce the effective memory bandwidth.

If we place this specialised circuitry so that it, too, bridges the I/O and memory busses and if we also give it the capability of generating the address and control signals required by these busses, then we have a machine that is capable of performing I/O at the full memory speed. This specialised circuitry is called a direct memory access or DMA coherent set of systems often fall short of what you convinced yourself would be offered when you went their way in the first place. Hewlett-Packard's different machines illustrate the point.

It first produced a personal computer, the HP85, of wide appeal to those using its calculators. About twice the price of the equivalent from Apple or PET, this is nevertheless a quality offering. It then announced the HP 125, a CP/M micro at about twice the price of its equivalent from the specialist Z80 suppliers, but with the ability to connect to HP's larger machines. Next, it announced the HP 9826, a Motorola 68000 desktop machine for technical applications such as graphics. After the 9826 came the HP87, an upgrade of the HP95 with an optional Z80 + CP/M card. Now, if all those machines could talk to each other and later be networked, there might be a coherent strand or an upward compatible set. Of course they are not designed to do so but they are directed to satisfy specific market niches . . . which takes us right back to the specialist micro suppliers, at half the price.

IBM's entry into the personal computer market surprised observers, not because of its technical innovation, (its 8088 processor is only half a 16-bit) but because it read, inwardly digested and applied the basic principles on which the micro industry has grown so rapidly.

It offers a low-cost personal computer with upgrades so that you can start learning and working at home. It is selling through retail outlets to reach the mass market as well as directly to its massive user base. It has chosen industry-standard hardware (Intel) and software, such as VisiCalc. And it is encouraging the spread of low cost software through its publishing arm.

What IBM does not offer is much in the way of storage; it has the distinction of providing larger RAM than floppy disk capacities. But surely the inventor of the winchester cannot be far away with its own hard disk. There is also a growing sub-industry marketing add-on goodies for the IBM personal computer. We hardly expect to see a powerful multi-user of networked micro from IBM for some while, although perhaps it will pleasantly surprise us again, and Xenix from Microsoft is a likely multi-user development.

Like DEC and Data General, Texas Instruments has had an old technology 16-bit micro on the market for some while, the 9900. It also has some novel chips found in its 'speak and spell' devices. Its first personal computer was a disappointment, the pricing and lack of software making it uncompetitive. But it has reduced prices and a new technology 16-bit processor, the 99000, is on the way.

Other suppliers

There is not a lot to be said about the other mainframe/mini suppliers' offerings in the micro market.

So far neither Burroughs, Honeywell, NCR or Univac have entered into the micro market in Australia, although Burroughs and NCR have entered into an OEM deal with a European company to produce an excellent 16-bit distributed system product. Xerox withdrew from the mini market but is back with micros. It has Ethernet, the local network architecture; a very fancy office micro; and the 820, a moderate CP/M machine available through retailers.

Wang has struck gold in office automation where it stole a march on all the other mini vendors. CP/M is now available on the Wangwriter and it looks a likely contender to bring out a personal computer to fit into its office automation range.

Good news, bad news

: The good news is we have got the new improved MP/M-II :
: The bad news is Digital Research has withdrawn it and will release 11.1. :

: The good news is that Microsoft supports MBasic on both CP/M and CP/M-86 :
: The bad news it doesn't support either MP/M or MP/M-86. :

: The good news is that we have received our first newsletter on :
: Processing Technology. :
: The bad news is that it has gone out of business. :

: The good news is that Imsai has not gone out of business. :
: The bad news is that it has just stopped trading for a while. :

Company confusion

You could be forgiven at this stage if you feel like someone who has been through a *Who's Who* in the computer industry. But it may help to remember that, whatever the colour and shape of the boxes you like, the majority will have the same engines underneath: 6502 and Z80 8-bit, 8086 and M68000 16-bit. You can also proceed knowing that no set supplier offers everything and no existing supplier has a coherent set of systems. Again, the strands of connecting standard software emerge so that if you are building skills in CIS-Cobol, Pascal and Basic, they will provide wide access to different eauipment.

The response of the existing computer suppliers to the micro is similar to the response of the mainframe suppliers to minis 10 years ago. It took a long time then for some of them to catch up and now the same pattern is being repeated. Realistically, you could not expect the existing suppliers to offer the same power for a quarter of the price or to undermine their existing product lines. They have made their response at the lowest end of the micro market. They will not take the lead in powerful network and multi-user machines that make their present offerings redundant on price performance. When they do finally get their response right by buying or emulating the successful micro vendors, e.g. IBM, a few new micro names will be firmly entrenched in the minds and on the desks of users.

Microcomputer manufacturers

Space allows for only a brief profile of the more established suppliers, those that have been selling business micros in Australia for some time. The comments reflect the author's experience and opinions and should not be regarded as unassailable fact based on exhaustive research.

More than 30,000 Apples have been sold worldwide since this remarkable company first started in 1977. The Apple II is a good personal computer made better by add-ons from independent vendors, such as packages from Personal Software, hard disks and networking from Corvus and Zynar, and the Z80 card from Microsoft. The Apple III is a disappointment, with poor price performance and so far with little software except in emulation mode.

The Commodore PET is one of the most popular personal computers in Australia though not quite so successful in its home country, the USA. Commodore offers a range of machines and the 8000, 96k PET is a popular option with wide software support, such as Wordcraft, and Silicon Office. The PET is a 6502 machine and CP/M is only available through a Z80 add-on, the Softbox.

Cromemco was one of the early suppliers of heavyweight micros and was the first to offer winchester disks and a Unix look-alike operating system, Cromix. Cromemco ranges from a minifloppy machine up to a four-user hard disk configuration. It has suffered from poor reliability and an appearance more suited to industrial environments. Although based on the Z80, Cromemco has not, until recently, directly supported CP/M, so that users have had to rely on independent vendors providing CP/M software. Cromemco has announced in the USA a Motorola 68000 machine with Cromix.

The Tandy Model I is an often-copied (e.g. System 80) popular home and personal machine with wide software availability for both TRS DOS and CP/M. Tandy also produces a business micro, the Model II, and has recently announced a top-range Motorola 68000-based machine.

Vector Graphics started about the same time as North Star but has developed a range well beyond North Star and comparable to Dynabyte. It has a good reputation for reliability and support.

Some new contenders

An analysis of the micro suppliers would not be complete without at least a passing reference to a few of the newer contenders.

Adam Osborne first made his name as a writer and publisher of some of the standard works in microcomputing. He established Osborne Computer Corporation partially in co-operation with the industry standard software manufacturers, Digital Research and Micro-Pro. His portable personal computer costs only $2595 and includes $1500 of industry standard software in that price! It is very suitable as a personal work

First time on Earth.

Sharp brings you the MZ80B.
A machine that offers you functions previously associated with more powerful, more expensive computers, that gives you versatility to handle a huge range of software and hardware applications in scientific, business and personal use.

The MZ80B opens up a new world of graphic display potential, more flexible data storage and retrieval, and ease of operation.

Here is the computer from the future. Available today.

Stunning Graphic Display

Seeing is believing. The large-screen high-focus, green-face display incorporated in the MZ80B gives you high resolution graphics of 320 x 200 dots.

An additional graphic RAM can be added which allows another 320 x 200 dot resolution pattern to be displayed.

This dual high-resolution graphic ability is especially useful for stimulating and displaying a dynamic picture. It can display 40 characters x 25 lines or 80 characters x 25 lines via software switching.

In addition there are facilities for full, on-screen editing, reverse video, partial scrolling and a full range of graphic symbols.

Character and Graphic Printer

This fast, quiet printer will reproduce your graphic displays and, of course, print-out upper and lower case letters and symbols. A tractor/friction feed version is also available.

Data Storage/retrieval

The MZ80B has a remarkable memory. 64K of RAM. And that constitutes all the memory area, giving flexible storage of any computer language and its software.

The cassette deck is electromagnetically-controlled, with a data transfer speed of 1800 bits/sec. combined with a unique programme search facility to make data storage and retrieval super-fast.

Software

Software includes Basic compiler, PASCAL, COBOL, FORTRAN and numerous accounting applications.

Floppy Disk Drive

A twin Floppy Disk Drive unit can be added which will give you 560 bytes of storage on double-sided, double-density disks.

Comprehensive Documentation

Each MZ80B comes complete with a full set of documentation including an owners' manual giving full circuit diagrams, a monitor reference manual and programming manuals.

Interfaces

RS232C and IEEE Interfaces are available allowing the MZ80B to communicate with scientific instruments and other peripherals.

CP/M* 2·2

CP/M* is also available making a wide range of packages immediately available including wordprocessing, financial modelling, data base management to mention but a few. CP/M* also increases the disk capacity to 680K.

(CP/M* is a Trade Mark of Digital Research Ltd.)

SHARP
We put you a step ahead

FREE Basic compiler with every purchase of the SHARP MZ80B

Please send me full information on the Sharp MZ80B computer.

NAME . . . .

ADDRESS . . . .

. . . . Postcode . . . .

Telephone . . . .

Lasercord Computer Company Pty. Ltd.
CNR HIGH STREET & BLACKBURN ROAD, MT WAVERLEY 3149
Telephone: (03) 232 7621

FRAMES OF REFERENCE

What's what in processors and bus structures

Microcomputer manufacturer	8-bit processors		16-bit processors		Bus structure	
	Z80	6502	8086	M68000	Own bus	S100
Apple		*		?	*	
Commodore		*			*	
Cromemco	*			*		*
North Star	*					*
Osborne	*				*	
Sharp	*				*	
Sirius			*		*	
Tandy	*			*	*	
Vector Graphics	*					*

? = Rumoured but not released

machine and as a terminal to bigger systems.

Sirius is the brainchild of Chuck Peddle, the 6502 and PET designer and is a dead ringer for the IBM personal computer. It is based on the quasi 16-bit 8088 and offers CP/M-86 and industry classics CBasic, MBasic, CIS-Cobol, SuperCalc and MicroModeller. A flat, soft image, high resolution screen and good graphics are features.

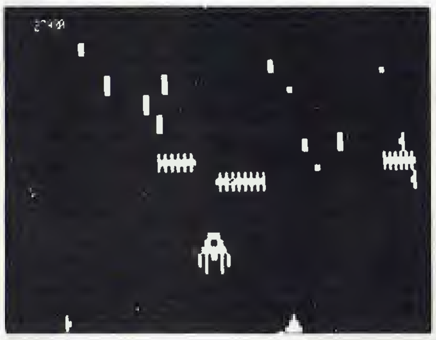
A personal view substantiated by achievement to date is that the best of the micro industry will repeat if not surpass the success of the best in the mini-industry. Similarly, there will be a number of 'second division' micro companies strong in specialist areas. The shifting sands of silicon chips make it impossible to provide a global micro recommendation, which in any case must relate to the needs and status of computing in each organisation.

The fifth commandment of microcomputing

The fifth commandment of microcomputing is: *thou shalt not worship at the altar of your main supplier.* IBM is rightly admired for its technical innovation and marketing muscle. But it does not have all the answers; some of the answers it provides are too late: and some of the answers are downright wrong. Mainframe users tell stories about waiting 18 months for announced equipment, only to find it fulfils half its specification. They also tell horror stories about the IBM Series 1 which took several years before it became a good product. Of course there is some risk in going with the specialist micro suppliers but the payoff in price performance, technical advancement and user benefits should amply repay the risk.

ARCADE FAVOURITES
for trs-80/system 80

* 4 GAMES, ALL BASED ALONG THE SAME LINES AS THE NOW FAMOUS BIG FIVE SOFTWARE
* ALL GAMES WRITTEN IN MACHINE CODE FOR FAST REAL TIME GRAPHICS
* ALL GAMES HAVE SOUND AND ARE BASED ON AN ARCADE FAVOURITE
* ALL HIRES AND LOW-RES GAMES HAVE BOTH JOYSTICK AND KEYBOARD CONTROL

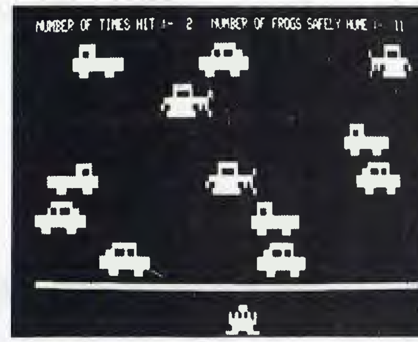


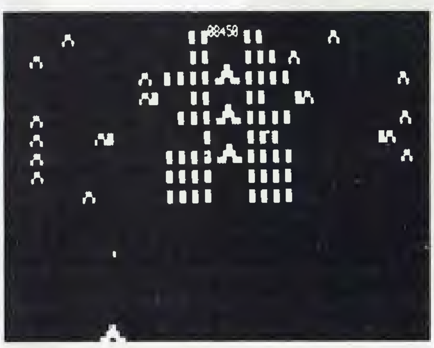
CENTIPEDE ATTACK

Fend off attack of savage centipede. Be wary of the giant spider. Hi and Low-Res.
16K LII
$19.95

HOPPY

Get your frog across the busy highway. After that, cross the torrent stream.
16K LII
$19.95





JUNGLE RAIDERS

Protect your bases from the jungle raiders.The end is enevitable but see how long you can last.
16K LII
$19.95

ALIEN CRESTA

Wave after wave of alien space creatures are attacking. Successfully dock your ship to increase your fire power and chances.
16K LII $19.95

ALSO AVAILABLE ARE: KILLER BEATLES * SPACE SHOOTOUT * BATTLE STATION * GHOST HUNTER

AVAILABLE FROM

All Dick Smith Stores

Sea Horse Computers
10 Mitchell St
Camden. NSW.
Ph: (046) 66 6406.

City Personal Computer
75 Castlereagh St
Sydney. 2000.
Ph: (02) 233 8992.

Computer Country Pty Ltd
338 Queen St
Melbourne. 3000.
Ph: (03) 329 7533.

Software-80
Shop 11/200 Moggill Rd
Taringa. Brisbane.
Ph: (07) 371 6996.

HS Electronics
Cnr. York & Kingsway Sts
Launceston. 7250.
Ph: (003) 31 8942.

Computer Campus Pty Ltd
11 Rundle St
Kent Town. 5067.
Ph: (08) 42 4826.

Micro Base
127 Fitzgerald St
West Perth. 6005.
Ph: (09) 328 9308.

CPU
279 Junction Rd
Clayville. Brisbane. 4011.
Ph: (07) 57 8023.

WD Leslie Pty Ltd
363-75 Raymond St
Sale. 3850.
Ph: (051) 44 2677.

DeForest Software
26 Station St
Nunawading. 3131.
Ph: (03) 877 6946.

Universal Services
Cunningham St
Dalby. 4405.
Ph: (074) 62 3228.

Outback Electronics
71 Barkly Hwy
Mount Isa. 4825.
Ph: (077) 43 3475.

John Hart Electronics
397 Argent St
Broken Hill. 2880.
Ph: (080) 88 3148.

Not all stores stock our entire range of software.

Post to: DUBOIS & McNAMARA SOFTWARE,
P.O. Box 358, EASTWOOD. NSW. 2122.

Trade enqiries welcome: Ph (02) 888 7151.

NAME ....

ADDRESS ....

....

Enclosed $ .... Bankcard/Cheque/Money Order

Expiry Date: .... Signature ....

Bankcard Number: ....

ITEM	QTY	PRICE EACH	AMOUNT

COMPAK
MAIL ORDER

BUY DIRECT FROM IMPORTER/MANUFACTURER

*FREE DELIVERY *MONEY BACK GUARANTEE

DEALER ENQUIRIES PHONE RON HARRIS 03 592 6285

STAR PRINTER

JUST LANDED DIRECT

* 80 CPS BI-DIRECTIONAL
* TRACTOR/FRICTION
* MANY GREAT FEATURES

COMPAK PRICE

COMPAK SPECIAL ONLY

$495·00

EX TAX

Cat DP8480

$591 INC TAX

XXX SCOOP XXX PURCHASE

NEW APPLE™ COMPATIBLE DISK DRIVES

NEW

SLIMLINE — FAST ACCESS

NEW JAPANESE DRIVE AND CONTROLLER

IMPORTED DIRECT BY COMPAK

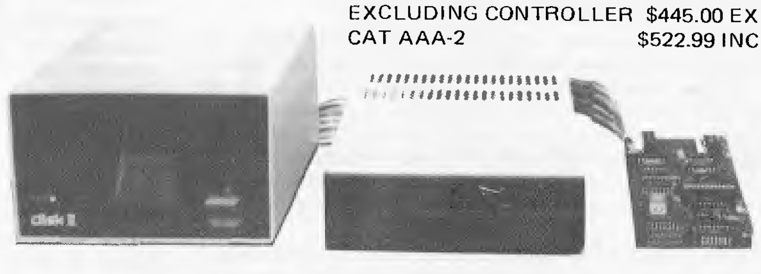
$595·00 EX TAX

* READS ALL APPLE DISKS
* RUNS ON APPLE CONTROLLER
* AUTOMATICALLY READS DOS 3.3/3.2
* APPLE DISK RUNS ON COMPAK CONTROLLER

INC CONTROLLER $699.00 INC TAX CAT AAA-1

EXCLUDING CONTROLLER $445.00 EX TAX

CAT AAA-2 $522.99 INC TAX



APPLE FDD COMPAK FDD CONTROLLER

ALSO COMPATIBLE WITH ORANGE SYSTEMS

APPLE DOS DISKETTE & MANUALS SUPPLIED WITH AAA-1

DIRT CHEAP TOP QUALITY

APPLE CARDS

	CAT NO	EX TAX	INC TAX
* Parallel Interface with graphics dump	NZ101	122	132.35
* RAM Card doesn't require cable	NZ102	145	170.37
* PAL Colour Card	NZ103	165	193.88
* Comms/Serial Card	NZ104	155	182.13

KEYBOARDS!!! FOR ORANGE MICRO

* REPEAT ON ALL KEYS
* NUMERIC KEYPAD
* 10 FUNCTION KEYS
* INCLUDING METAL CASE

$185.00 ex tax

Cat AKB-02

$217.38 inc tax

MITSUBISHI 8" FDD

HIGH QUALITY JAPANESE FLOPPY
* DOUBLE SIDED
* DOUBLE DENSITY
* UP TO 1.2 MBYTE

GREAT VALUE $595.00 EX TAX

$699.00 INC TAX

ALSO

JA-751 8" Slimline FDD $695 ex tax

* Quietest FDD yet manuf. $747.13 inc. tax

M1550 120 CPS dot matrix ptr. $1236 ex tax

* 132 Col/full graphics $1452 inc tax

F-10 Daisywheel 40 CPS $1990 ex tax $2338.25 in tax

CAT NO	QTY	PRICE	TOTAL	GRAND TOTAL
....	....	$............	$............	
....	....	$............	$............	
....	....	$............	$............	$............

NAME..

ADDRESS.......................................

PHONE...

Please find enclosed my cheque, money order. $.............

Please debit my BANKCARD, NUMBER ....

Date of expiry.................Signature...............

POST ORDER TO
COMPAK MAIL ORDER
211 BAY STREET,
BRIGHTON 3186
Phone: (03) 592 8343
592 6285

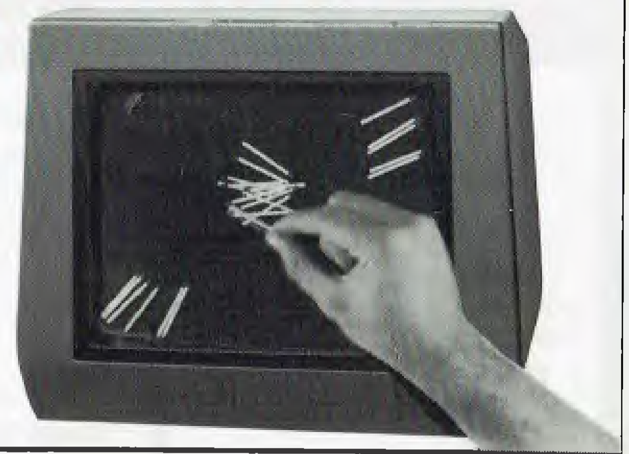
COMPLETE SATISFACTION GUARANTEED OR RETURN WITHIN 7 DAYS FOR FULL REFUND

Cut out here

A SIMPLE APPROACH TO GAME LEARNING PROGRAMS

by David L. Craig

If computers are ever to behave "intelligently", then they must be able to build up a store of experience based on past events and learn from their own mistakes and the mistakes of others that they observe. A simple approach to machine learning is demonstrated by a learning program applied to the well known game of "23 Matches". The possibility of extension of this approach to more complex games such as chess is discussed.



Introduction

Many human decisions are made on the basis of "experience"rather than on the basis of logical reasoning. This is often because there is no algorithm which leads to the best solution of a problem, because the human does not know the algorithm, if one exists, or because evaluation of the algorithm is too difficult. In these circumstances the human will choose the solution to the problem which he knows from experience has the highest probability of being correct.

The human seems to maintain some kind of probability table about the world with which to evaluate incoming information and make decisions. When the result of a decision is known the probability table is altered to reflect the result. If the result is favourable the probability of the decision leading to the result will be increased giving positive reinforcement. If the result is unfavourable the probability of the decision leading to the result will be reduced, reducing the likelihood of the same decision being made the next time that situation is encountered. This is akin to the "operant conditioning" principle of B.F. Skinner.

This is a very simple model of human learning. The process can be observed in a young child learning, for example, the names of colours. At the start of the learning process the child has no entries in his probability table relating names to colours. Initially, he readily accepts what he is told. After being told a number of times that a certain colour is blue, the child establishes an entry in his probability table for that colour. If someone then tells the child that the same colour is red, he will vehemently protest, "That's not red, it's blue!" As the child has been told that the colour is blue more often than he has been told it is red, he decides that it is more likely to be blue than red.

It is possible to make a machine (or a computer program) "learn" in a similar way. The aim of the learning process is to improve the performance in attaining a desired objective. To investigate machine learning, game playing is often used. This has a number of advantages over real life situations, since the objective is well defined (i.e. to win), the allowable moves are well defined, and the game can be as simple or as complex as desired. There have been a number of different games used for demonstrating machine learning, and also a number of different techniques used for implementing the learning process.

One approach was demonstrated by Martin Gardner (Ref. 1) with the game "Hexapawn". The machine's response to each game state is selected at random from the set of legal moves whose individual selection probabilities are the result of previous experience. After each game in which the machine wins, the moves that it used are increased in probability. After each game that it loses, the machine's moves are reduced in probability. Gardner implemented the learning process using coloured beads representing legal moves in a number of matchboxes representing the various game states. Beads were either added to or removed from the matchboxes depending on whether the moves led to wins or losses for the machine. The number of beads of a given colour in a matchbox represented the probability of a particular move being chosen.

Robert Weir (Ref. 2) has written a game-learning computer program version of Hexapawn which functions on a slightly different principle. Losing moves by the computer are entirely eliminated immediately. Russell Yost Jr. (Ref. 3) has implemented a game-learning computer program version of another game, "Eighteen with a Die". This program constructs a table of winning moves for each position of the game by learning from games played. The strategy used is for the computer to choose moves which prevent the opponent from reaching a winning situation.

Other more complex games such as checkers have also been investigated, e.g. Samuel (Refs. 4, 5) and Griffith (Ref. 6). These game-learning programs have differed considerably from those referred to above. These have used "static evaluation functions" to calculate the strength of a position with the importance of various factors in the evaluation function being altered based on experience from games played.

A learning version of 23 Matches

A different game learning procedure is proposed here, and demonstrated by application to the well known game of "23 Matches" (Ref. 7). In the computerised version of "23 Matches" the player and the computer take turns in removing 1, 2 or 3 matches from a pile which initially contains 23 matches. The object of the game is to avoid taking the last match from the pile, i.e. the one who takes the last match loses. There is a simple algorithm from which winning positions can be calculated and the computer is generally programmed to play using this algorithm. As the player with the first move cannot lose if he knows the algorithm, the computer plays second to give the human player a chance to win.

In the version of the "23 Matches" program presented here, the computer is not programmed with the winning algorithm. Rather it is only given the rules of the game and made to learn how to win from experience gained by playing the game against a human opponent.

"23 Matches" is a very suitable game for beginning to experiment with game-learning programs. It contains only 23 states (i.e. each position with a different number of matches in the pile is a state) and there are only 3 states which can legally be reached from any other state (i.e. only 3 legal moves at each point in the game). To win the game it is important at each move to choose the legal state from which the probability of winning is highest.

In this program a table is maintained with one entry per state which contains a value related to the probability of winning from that state. This will be called the "experience table". The experience table is initially zeroed except for the winning state and is updated after each game is played against a human opponent.

During each game that is played a "game record table" is maintained with one entry per game state. Those states which are selected during the game are recorded in the game record table as +1 for the computer's moves and −1 for the human's moves. The other entries are zero. At the end of each game the game record table entries are added to the experience table entries if the computer wins, and subtracted if the computer loses. This means that game states from which a win was achieved (whether by the computer or the human player) have their experience table entries incremented, and those game states which led to a loss have their experience table entries decremented. After a number of games, game states from which wins are regularly achieved will have positive experience table entries, while those which lead regularly to losses will have negative experience table entries. Unexplored states or states with equal numbers of wins and losses following will have experience table entries of zero. The winning state is initialised to a positive value so that the computer can identify the desired end state.

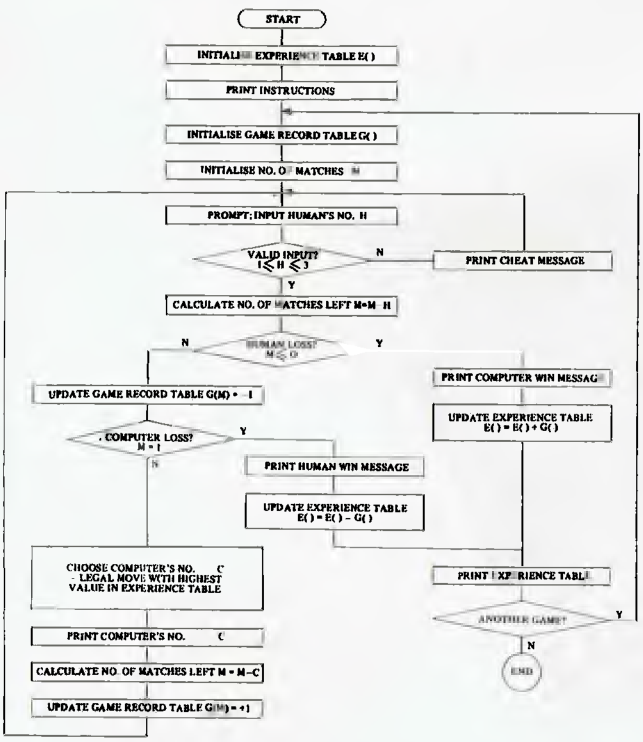


FIG. 1 : FLOWCHART FOR 23 MATCHES GAME-LEARNING PROGRAM

The computer's strategy in choosing a move is to move to the next legal state whose experience table entry is the greatest. This is choosing the move that results in a position which has led to the greatest number of wins over losses in past games. If all legal moves result in states having equal values in the experience tables a fixed move is chosen. (Alternatively a random move could equally well be chosen.)

A flowchart of the 23 Matches Game Learning Program is given in Fig. 1. The program itself is written in Basic and a full listing is provided. The version of Basic used is SWTPC 4K integer Basic which is a very cut-down version of Basic. Because of this it should be possible to run the program without any changes on almost any system supporting Basic. The program is quite straightforward and should be easily followed from the flowchart and the listing commands. The experience table and the game record table are represented as 23 element one dimensional arrays E( ) and G( ) respectively, with entry I of each array representing the state containing I matches. At the end of each game the experience table is displayed to allow the learning process to be observed. The experience table is maintained only during the series of games following the one entry to the program from the Basic monitor. Each time the program is entered from the Basic monitor the experience table is re-initialised.

The more games that are played the more reliable the experience table becomes and the better the machine plays. Also the better the human plays the faster the computer learns. The computer learns to play a perfect game surprisingly fast, and it is interesting to attempt to slow down the learning process by playing against the computer using a strategy aimed at creating errors in the experience table. The computer always learns to play a perfect game much faster than a human who does not know the winning algorithm because it remembers all the moves from the past games.

The game-learning procedure here is somewhat similar to that of Gardner in his Matchbox Games Learning Machine (Ref. 1) in that moves which lead to a win are "rewarded" and those that lead to a loss are "punished". However, there are two very significant differences between Gardner's procedure and the one described here:

(i) the move that is chosen is the one with the highest probability of leading to a win, rather than a random choice based on weighted probabilities.
(ii) the opponent's moves are used to adjust the probabilities as well as the computer's moves.

These two differences should make the procedure described here learn much faster than Gardner's.

Extension to more complex games

Having experimented with this learning version of 23 Matches and found how well the learning procedure works for the simple game, it is logical to ask if the same procedure will work with more complex games. The flowchart of Fig. 1 for 23 Matches can be generalised to cover any game which allows only wins or losses as shown in Fig. 2. This could be generalised further to cover games which also allow draws. In the case of a draw, the experience table would not be updated after a game. In principle the learning procedure would appear to be able to be applied to any game.

I have applied the learning procedure described to Hexapawn. Its performance is much superior to the procedure described by Gardner (Ref. 1) and Weir (Ref. 2). But 23 Matches and Hexapawn are relatively simple games for which there are known winning algorithms. Will the procedure work in complex games where there are no known winning algorithms, for example checkers and chess? It was suggested at the beginning of this article that this is the sort of situation where humans would rely on "experience".

All current chess programs, for example, rely on a tree-search procedure to evaluate a position, but none use information obtained from previous games (except perhaps for fixed pre-programmed openings and endings). Without a store of experience, a computer chess program is like a brilliant analyst who has been taught the moves but never before played a game. It is as though every game played is the first game played and there is no improvement in performance with the number of games played. It has been stated by Frey (Ref. 8) that to become a chess master requires at least 20 hours of study per week for 5 years. The result of this study is a huge store of board patterns of significance as strong or weak positions (i.e. win or loss states) in the game. The experience of thousands of games played by the world's best chess players and the analyses of these games by other top players is assimilated into the memory of the player. Until a chess program is given a similar store of experience it seems unlikely that a computer will be the world chess champion.

The principle of the experience table and of evaluating positions or game states by the probability of winning from that position still appears valid in a game such as chess.

Here's to a longer life!

This is new Verbatim Datalife.™ Both 8″ and mini flexible disks with 7 data-shielding improvements for a longer lifetime of trouble-free data recording, storage and retrieval than ever before possible.

All to assure users throughout the world the best possible performance, time after time, for a long, long time to come.

Users in Australia get an added benefit, because Verbatim Datalife disks are being made right here.

So now you won't have to wait for deliveries coming from afar. And you can have technical support that's nearer, getting you responses faster.

Verbatim Datalife disks. The best flexible disks made in the world, now being made right here.

Support your local disk.

Available through **magmedia**. Contact them at (02) 428-1100, (03) 699-9688, (07) 229-1941 or (062) 48-6751.

Verbatim

Verbatim Australia Pty Ltd., 52-54 River Street, South Yarra 3141. Phone (03) 241-7422.

COMPUTER OSBORNE BOOKS

A McGraw-Hill Company

OSBORNE•McGRAW HILL IS PROUD TO ANNOUNCE THAT THEY ARE NOW THE EXCLUSIVE DISTRIBUTORS IN AUSTRALIA OF THE SUPERB RANGE OF OSBORNE MICROCOMPUTER BOOKS. THESE NEW TITLES PICTURED BELOW, AS WELL AS A BACKLIST SELECTION ARE AVAILABLE FROM:

TECHNICAL BOOK SHOP

114 Castlereagh Street, Sydney
(02) 264 2250

WordStar™ Made Easy

OSBORNE

order **NOW** on the form below!

Send this form to: **TECHNICAL BOOK SHOP**
GPO BOX 2065 SYDNEY. 2001

	NEW THIS SEASON	
72·1	Trade Secrets	$27.95
71·3	A User Guide to the UNIX System	$22.40
79·9	Discover FORTH	$21.00
64·0	Volume 0: 3rd Ed., Revised (4/82)	$11.15
59·4	6502 Assembly Language Subroutines	$22.35
51·9	Assembly Language Programming for the Apple II	$28.15
50·0	VisiCalc Home and Office Companion	$22.40
65·9	Your ATARI Computer	$23.75
74·8	Practical Pascal Programs	$22.40
73·X	Some Common Pascal Programs	$21.00
	USER'S GUIDE SERIES	
46·2	Apple II User's Guide	$23.75
55·1	PET/CBM Personal Computer Guide, 2nd Edition	$21.00
70·5	PET Fun and Games	$16.75
44·6	OSBORNE CP/M User Guide	$22.35
61·6	CBASIC User Guide	$23.75
69·1	WordStar Made Easy	$16.75
47·0	Business System Buyer's Guide	$13.95
28·4	Running Wild — The Next Industrial Revolution	$ 6.95
	ASSEMBLY LANGUAGE PROGRAMMING	
27·6	6502 Assembly Language Programming	$23.80
62·4	68000 Assembly Language Programming	$23.80
29·2	The 8086 Book	$23.80
10·1	8080A/8085 Assembly Language Programming	$22.40
35·7	6809 Assembly Language Programming	$23.80
21·7	Z80 Assembly Language Programming	$23.80
36·5	Z8000 Assembly Language Programming	$28.00
12·8	6800 Assembly Language Programming	$22.40
	TECHNICAL REFERENCE	
34·9	Volume 1: Basic Concepts, 2nd Edition	$22.35
18·7	Volume 3: Some Real Support Devices	$21.00
98·5	Volume 3: 1978/79 Update	$35.00
37·3	Interfacing to S-100/IEEE 696 Microcomputers	$21.00
57·8	Microprocessors for Measurement and Control	$22.40
42·X	OSBORNE 4 & 8-Bit Microprocessor Handbook	$27.95
43·8	OSBORNE 16 Bit Microprocessor Handbook	$27.95
39·X	8089 I/O Processor Handbook	$13.95
45·4	CRT Controller Handbook	$13.95
41·1	68000 Microprocessor Handbook	$13.95
31·4	PET and the IEEE 488 Bus (GPIB)	$22.40
	PROGRAMMING SERIES	
06·3	Some Common BASIC Programs	$21.00
38·1	Practical BASIC Programs	$22.40
68·3	Some Common BASIC Programs — Apple II Edition	$21.00
66·7	Practical BASIC Programs — Apple II Edition	$22.40
63·2	Science and Engineering Programs — Apple II Edition	$22.40
53·5	Some Common BASIC Programs — Atari Edition	$21.00
54·3	Some Common BASIC Programs — TRS-80 Edition	$21.00
67·5	Practical BASIC Programs — TRS-80 Edition	$22.40
40·3	Some Common BASIC Programs — PET/CBM Edtion	$21.00
22·5	Payroll With Cost Accounting — CBASIC	$28.00
23·3	Accounts Payable/Accounts Receivable — CBASIC	$28.00
24·1	General Ledger — CBASIC	$28.00

Name ....

Address ....

....

Please send me books numbered .... as above

Total value $ .... plus postage NSW $2.50, elsewhere $3.50

I enclose a cheque/postal order for $ ....

for (no.) .... books or

Bankcard No. ....

Expiry Date........... Signed .... **APCO/M – 1**

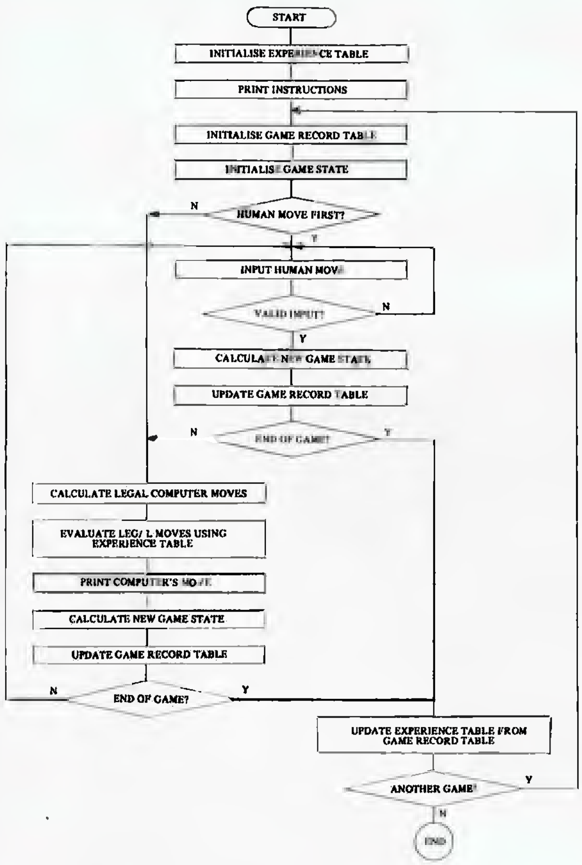


FIG. 2 : GENERALISED GAME-LEARNING PROGRAM FLOWCHART

The biggest problem in applying the learning procedure described here to such a game appears to be one of scale. Since there are a huge number of legal states (board positions) in chess it becomes impossible to assign an entry in an "experience table" to each possible state of the game as was done in the 23 Matches example. In reality, however, there are very many fewer states which have any likelihood of occurrence in a chess game (current chess programs spend most of their time examining ridiculous moves). There are probably very many fewer states again in top-level chess games which are passed through in reaching a win or lose position. Since these are the only positions which are really important in the experience table, it may be possible to reduce the size of the experience table to a manageable level even for a game such as chess. A well organised experience table of, say, 100,000 to 1,000,000 entries would be quite within the capabilities of current computers. Whether this would be an experience table of sufficient size for chess would remain to be seen. Also it is not known if the frequency of occurrence of particular board positions is high enough to produce significant entries in an experience table.

To fill the experience table in the 23 Matches example required the computer to play against a human opponent. To create a reliable experience table quickly requires play against a skilled opponent. Because of the huge number of possible board positions in chess it would require a very large number of games to be played by the computer against top players to generate a reliable experience table. This problem is easily overcome in principle. Because the chess games of the best players in the world have been recorded for many years, a huge library of chess experience is available. This could be converted directly into the form of the required experience table which would then be used as the basis on which the computer could begin playing, and which would be refined further by playing against top players. Creating the experience table from the record of past games would, however, be a formidable task (though perhaps taking fewer hours than required to create a human grandmaster).

I would not expect a computer program using only an experience table to be capable of being a world chess champion, just as top chess players do not rely solely on experience. However, a computer program combining both tree-analysis capability (such as Chess 4.7 or even Sargon) and an experience table should represent an extremely powerful chess player. The use of experience does not directly address the problem of programming strategic planning into a chess program, but indirectly should improve strategy by copying strong strategic lines of play which will be embedded in the experience table as a series of board positions with a strong winning probability. The program would do nothing really original but would copy the best moves of the best players.

If chess is thought to be too complex a game for attempting extension of the learning procedure described here, then checkers may be a little less daunting. The game has considerably fewer legal states than chess and there is also a considerable body of recorded games by the world's best players from which to derive an experience table. In fact Samuel (Ref. 5) has reported that he has stored on magnetic tape approximately 250,000 board situations from master checkers players.

References

1. Gardner, M., *"A Matchbox Game Learning Machine", Ch 8 in "The Unexpected Hanging and Other Mathematical Diversions", Simon and Schuster, 1969.*
2. Weir, R.R., *"Hexpawn: A Beginning Project in Artificial Intelligence" BYTE, Vol.1, No.3, Nov. 1975, p36.*
3. Yost, R.R.Jr, *"Eighteen with a Die: A Learning Game Player", BYTE, Vol.5, No.1, Jan. 1980, p212.*
4. Samuel, A.L., *"Some Studies in Machine Learning Using the Game of Checkers", IBM Journal of Research and Development, Vol.3, 1959, pp210-229.*
5. Samuel, A.L., *"Some Studies in Machine Learning Using the Game of Checkers II: Recent Progress" IBM Journal of Research and Development, Vol.11, 1967, pp601-617.*
6. Griffith, A.K., *"A Comparison and Evaluation of Three Machine Learning Procedures as Applied to the Game of Checkers", Artificial Intelligence, Vol.5, 1974, pp137-148.*
7. *"23 Matches" in "What to Do After You Hit Return", Peoples Computer Company, 1975, p24.*
8. Frey, P.W. and Atkin, L.R., *"Creating a Chess Player: Essay on Human and Computer Chess Skill", BYTE, Vol.3, No.10, Oct. 1978, p182.*

'Good God, Miss Hargreaves — it's a blackmail note!'

HIGH DENSITY VDU CARD

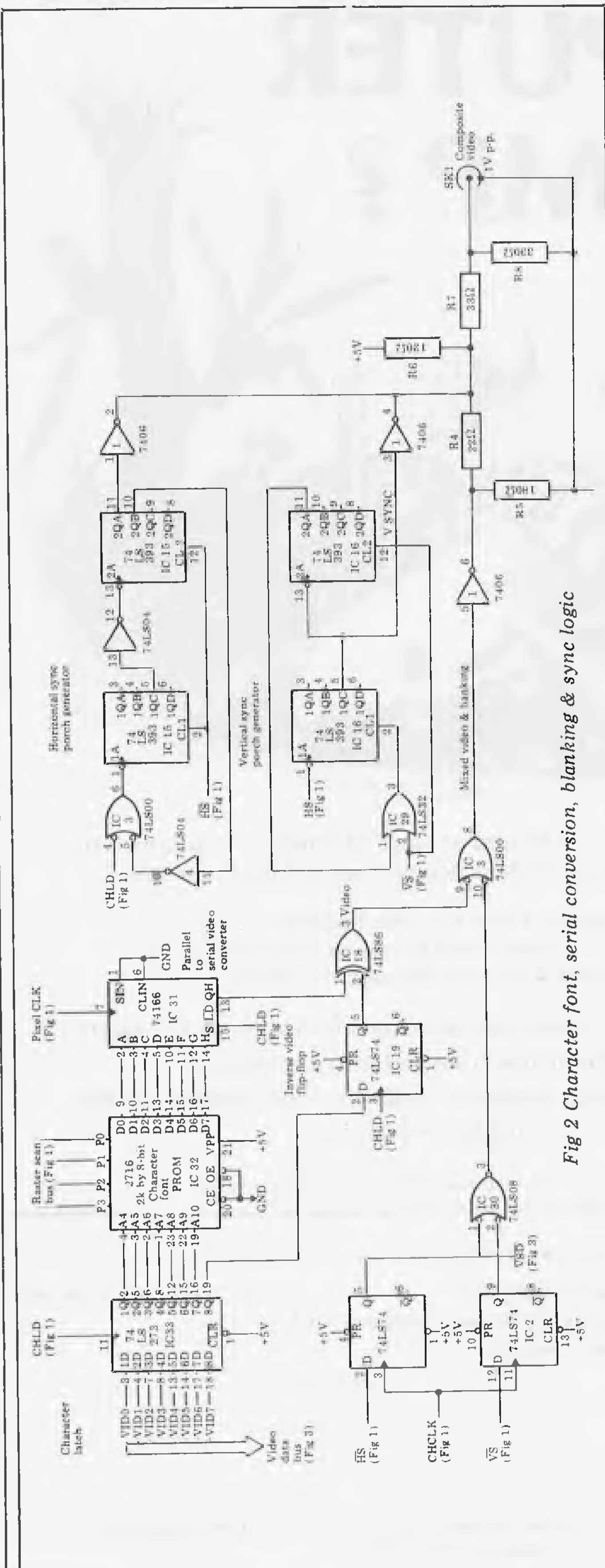
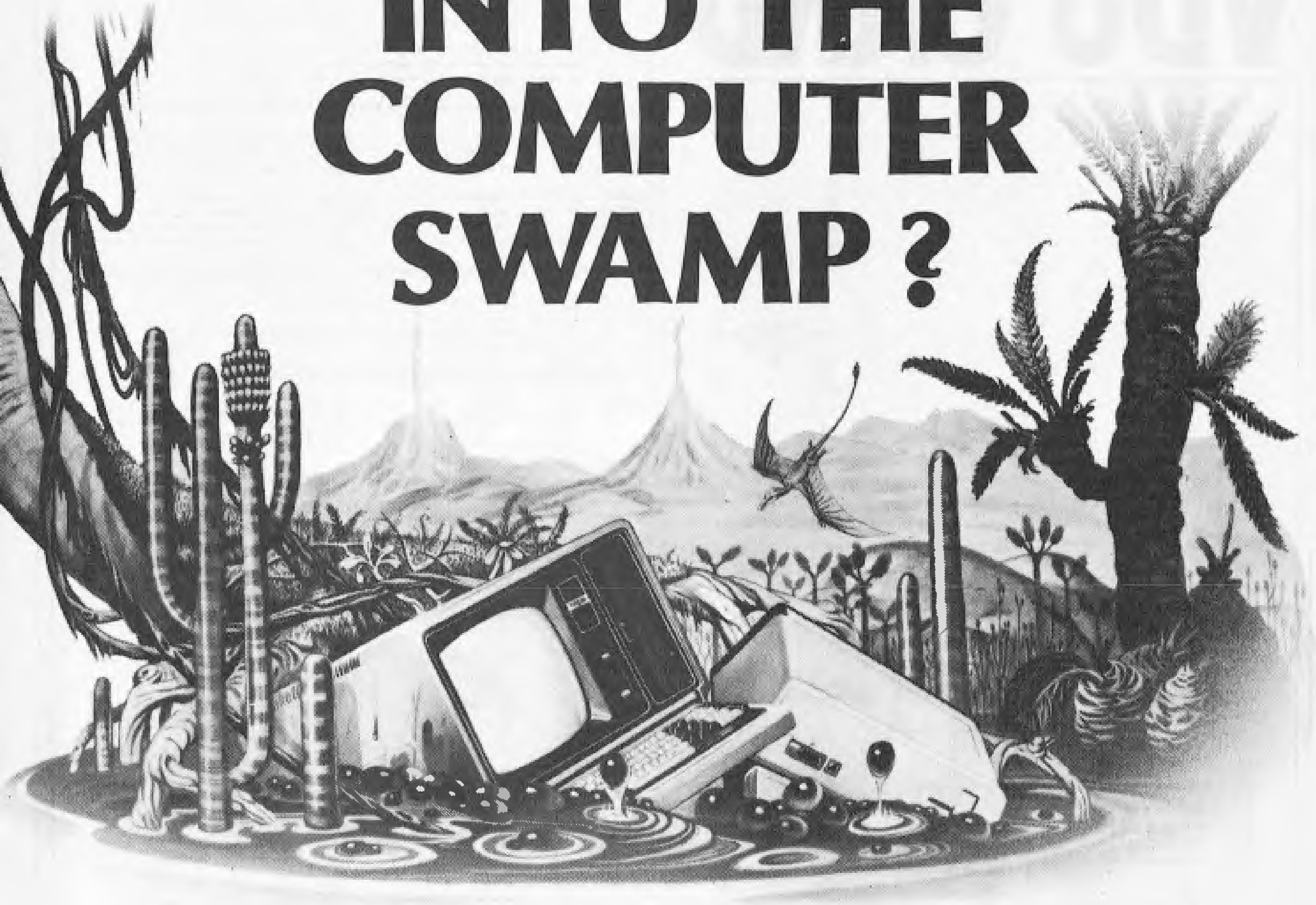


Fig 2 Character font, serial conversion, blanking & sync logic

```
;*************************************************************
;
;       VDU SCREEN HANDLER ROUTINES
;
;*************************************************************
;
;       Screen is memory-mapped from F000H.  The map is
;       not complete, ie:
;               the first line is from F000H to F04FH
;               the second line is from F080H to F0CFH
;               etc
;               the last line is from FC80H to FCCFH.
;
;       All memory not used by the screen, ie XX50H to XX7FH
;       and XXCFH to XXFFH in the range F000H to FDFFH is free
;       to be used for pointers, variables, etc.
;
;*************************************************************
;
SCRSRT: EQU     0F000H          ; Screen memory start address
SCNLIN: EQU     0F080H          ; 2nd line start address
LOFSET: EQU     00030H          ; Line offset value
;
LINENO: EQU     0F060H          ; Line number counter
CHARNO: EQU     0F061H          ; Character number counter
SLINEN: EQU     0F0D0H          ; Scroll line number counter
SCHARN: EQU     0F0D1H          ; Scroll character number counter
;
;*************************************************************
;
;       INITIALISE SCREEN SUBROUTINE
;
;       Initialises screen pointers, clears screen & sets
;       cursor position
;
;*************************************************************
;
INSCRN: CALL    CSCRN           ; Call clear screen subroutine
        CALL    INVCLC          ; Set cursor to first character
;                               ;   of first line
        RET
;
;*************************************************************
;
;       FIND SCREEN MEMORY LOCATION SUBROUTINE
;
;       Produces the address in H & L corresponding to the
;       current line & character numbers.
;
;*************************************************************
;
FNDLOC: PUSH    PSW             ; Save status, accumulator and
        PUSH    D               ;  D reg pair on stack
;
        LXI     D,SCRSRT        ; Load DE with screen start addr
        LDA     LINENO          ; Fetch line number
        MOV     L,A             ; Move into L
        MVI     H,00H           ; Clear H
        DAD     H               ; Multiply
        DAD     H               ;  HL
        DAD     H               ;  by
        DAD     H               ;  80H
        DAD     H
        DAD     H
        DAD     H
;
        DAD     D               ; Add screen memory start addr
        LDA     CHARNO          ; Fetch character number
        ADD     L               ; Add L
        MOV     L,A             ;  and store in L
;                                 H & L now contain a screen
;                                  memory location corresponding
;                                  to the current line & character
;                                  number.
;
        POP     D               ; Restore D reg pair &
        POP     PSW             ;  status & accumulator
        RET
;
;*************************************************************
;
;       INVERT VIDEO AT CURRENT LINE & CHARACTER POSITION
;       SUBROUTINE
;
;*************************************************************
;
INVCLC: CALL    FNDLOC          ; Call find location to find
;                               ;  addr of cursor
        MOV     A,M             ; Pick up code at cursor pos'n
        XRI     80H             ; Invert video by inverting top bit
        MOV     M,A             ; Return it to memory
        RET
;
;*************************************************************
;
;       SINGLE CHARACTER OUTPUT SUBROUTINE
;
;       Takes an ASCII code in acc and writes it to the
;       current cursor position.
;
;*************************************************************
;
CHAROP: PUSH    PSW             ; Save registers onto stack
        PUSH    B
        PUSH    D
        PUSH    H
;
        CPI     08H             ; Test for backspace code
        JZ      BACKSP          ;  & jump to backspace if so
        CPI     09H             ; Test for cursor right code
        JZ      CURADV          ;  & jump to cursor advance if so
        CPI     0DH             ; Test for carriage return
        JZ      CARRET          ;  & jump to carriage ret if so
        CPI     0AH             ; Test for lfeed (cursor down) code
        JZ      LFEED           ;  & jump to line feed if so
        CPI     1AH             ; Test for cursor up code
        JZ      CUP             ;  & jump to cursor up if so
        CPI     0CH             ; Test for clear screen code
        JZ      CLRSCN          ;  & jump to screen clear if so
;
        MOV     B,A             ; Store code in B
        CALL    INVCLC          ; Remove cursor from screen
        MOV     M,B             ; Write to current line & char no
```

ARE YOU SINKING INTO THE COMPUTER SWAMP?



Bogged down by sales literature, inflated claims, an enormous range of products from which to choose, and intimidating jargon, who can blame business people from sinking in the mire?

But business computing need not be a trap for your business. One Australian magazine — and only one — devotes itself purely to a down-to-earth coverage of business computing to help guide you through the pitfalls.

Australian Business Computer provides concise, timely and practical information on all aspects of business computing. We provide information from authoritative sources, both local and overseas, to help improve Australian businesses. ABC is not interested in novelty or amusement; our business is to help *your* business.

If you do not wish to cut out this coupon, simply send your remittance, with your name and address, to the address below.

I would like to subscribe to Australian Business Computer for 1 year (10 issues)

☐ $35 within Australia ☐ Aust $45 overseas surface mail ☐ Aust $100 overseas air mail

☐ I enclose my cheque/money order, made payable to Australian Business Computer for $

☐ Please invoice my company ☐ Please charge my Bankcard

Bankcard No . . . . .Expiry Date . . . . Signature . . . .

Name . . . .

Address. . . . .

. . . . Postcode . . . .

Please send this order form with your remittance to Australian Business Computer, Subscriptions Department, Freepost 10, 462 Burwood Road, Hawthorn, Australia 3122.

Listing in Basic of 23 Matches game

```
1000  REM *** 23 MATCHES GAME

      -LEARNING PROGRAM - D.L. CRAIG

1010  REM ***DIMENSION EXPERIENCE AND GAME RECORD TABLES
1020  DIM E(23)
1030  DIM G(23)

1050  REM *** INITIALIZE EXPERIENCE TABLE
1060  FOR I = 1 TO 23
1070  E(I) = 0
1080  NEXT I
1090  E(1) = 100

1120  REM *** PRINT THE INSTRUCTIONS
1130  PRINT
1140  PRINT "WE START WITH 23 MATCHES."
1150  PRINT
1160  PRINT "WE TAKE TURNS TO REMOVE 1,2 OR"
1170  PRINT "3 MATCHES FROM THE PILE."
1180  PRINT
1190  PRINT "THE ONE WHO HAS TO TAKE THE"
1200  PRINT "LAST MATCH LOSES."
1210  PRINT
1220  PRINT "YOU GO FIRST."
1230  PRINT
1240  PRINT

1260  REM *** INITIALIZE GAME RECORD TABLE
1270  FOR I = 1 TO 23
1280  G(I) = 0
1290  NEXT I

1310  REM *** INITIALIZE NO. OF MATCHES
1320  M = 23
1330  PRINT "THERE ARE N W 23 MATCHES."

1350  REM *** INPUT HUMAN MOVE
1360  PRINT "HOW MANY DO YOU TAKE";
1370  INPUT H
1380  PRINT
1390  REM *** CHECK IF INPUT IS 1, 2 OR 3
1400  IF H<1 GO TO 1780
1410  IF H>3 GO TO 1780

1430  REM *** CALCULATE NO. OF MATCHES LEFT
1440  M = M - H

1460  REM *** TEST IF HUMAN LOST
1470  IF M <= 0 GO TO 1850

1490  REM *** UPDATE GAME RECORD TABLE FOR HUMAN MOVE
1500  G(M) = -1

1520  REM *** TEST IF COMPUTER LOST
1530  IF M = 1 GO TO 1960

1550  REM *** CALCULATE COMPUTER'S NUMBER
1555  IF M = 2 GO TO 1630
1560  IF M = 3 GO TO 1600
1565  IF E(M - 3) < E(M - 2) GO TO 1600
1570  IF E(M - 3) < E(M - 1) GO TO 1630
1580  C = 3
1590  GO TO 1650
1600  IF E(M - 2) < E(M - 1) GO TO 1630
1610  C = 2
1620  GO TO 1650
1630  C = 1

1650  PRINT "I TOOK"; C

1670  REM *** CALCULATE NO. OF MATCHES LEFT
1680  M = M - C
1690  PRINT "...THERE ARE NOW"; M; "MATCHES."
1700  PRINT

1720  REM *** UPDATE GAME RECORD TABLE FOR COMPUTER MOVE
1730  G(M) = 1
1740  GO TO 1360

1770  REM *** HUMAN CHEATED
1780  PRINT "YOU CHEATED, BUT I'LL GIVE"
1790  PRINT "YOU ANOTHER CHANCE."
1800  PRINT
1810  GO TO 1360

1840  REM *** COMPUTER WINS
1850  PRINT "I WON! BETTER LUCK NEXT TIME."

1880  REM *** UPDATE EXPERIENCE TABLE FOR COMPUTER WIN
1890  FOR I = 1 TO 23
1900  E(I) = E(I) + G(I)
1910  NEXT I
1920  GO TO 2041

1950  REM *** HUMAN WINS
1960  PRINT "YOU WON, BUT YOU WERE LUCKY"

1990  REM *** UPDATE EXPERIENCE TABLE
2000  FOR I = 1 TO 23
2010  E(I) = E(I) - G(I)
2020  NEXT I
2040  REM *** PRINT EXPERIENCE TABLE
2041  PRINT
2042  FOR I = 1 TO 23
2043  PRINT E(I);
2044  NEXT I
2045  PRINT
2049  REM *** ANOTHER GAME?
2050  PRINT
2060  PRINT "DO YOU WANT TO PLAY AGAIN?"
2070  PRINT "TYPE 0 FOR NO, 1 FOR YES."

2100  INPUT A
2110  IF A = 1 GO TO 1230

2160  END
```

FMS SOFTWARE AND HARDWARE FOR CP/M BASED SYSTEMS

Agent for: **Lifeboat Associates, Sigma International, Compuview, Westico, Memtech, Anderson Digital Equipment, AWA.**

Software Available

New products appear in red.

System Tools:
BUG and uBUG
DESPOOL
DISILOG
DISTEL
EDIT
EDIT-80
FILETRAN
IBM/CPM
MAC
MACRO-80
MINCE
PASM
PLINK
PLINK II
PMATE
RAID
Reclaim
SID
TRS-80 Model II Customization Disk
Unlock
WordMaster
XASM: 05, 09, 18, 48, 51, 65, 68, F8, 400
XMACRO-86
ZAP80
ZDT
Z80 Development Package
ZSID

Telecommunications:
BSTAM
BSTMS
MicroLink-80
RBTE-80

Languages:
ALGOL-60
APL/V80
BASIC Compiler
BASIC-80
baZic II
BD Software C Compiler
CBASIC-2
CIS COBOL (Standard)
CIS COBOL (Compact)
COBOL-80
FORTRAN-80
KBASIC
muLISP/muSTAR-80
Nevada COBOL
JRT Pascal
Pascal/M
Pascal/MT
Pascal/M +
Pascal/Z
PL/I-80
STIFF UPPER LISP
S-BASIC
Tiny-C
Tiny-C Two
UCSD Pascal
Whitesmiths' C Compiler
XYBASIC

Language and Applications Tools:
BASIC Utility Disk
DataStar
FABS
FABS II
Form 1 for CIS COBOL
Form 2 for CIS COBOL
MAGSAM III
MAGSAM IV
M/SORT for COBOL 80
PSORT
QSORT
STRING/80
STRING BIT
SUPERSORT
ULTRASORT II
VISAM

Word Processing Systems and Aids:
Benchmark
MicroSpell
Letteright
Magic Wand
Spellguard
TEX
Textwriter III
WordIndex
WordStar
WordStar Customization Notes

Data Management Systems:
CONDOR
HDBS
MDBS
MDBS:DRS,.QRS,.RTL
dBASE II
PRISM/LMS
PRISM/IMS
PRISM/ADS

General Purpose Applications:
CBS
Selector III-C2
Selector IV

Mailing List Systems:
Benchmark Mailing List
Postmaster
Mailing Address
MailMerge for WordStar
NAD

Financial Accounting Packages:
BOSS Financial Accounting System
Peachtree Financial Packages
Structured Systems Group Financial Packages
GLector

Numerical Problem-Solving Tools:
T/MAKER II
Ipl
PLAN80
Analyst
Microstat
muSIMP/muMATH
Statpak

Professional And Office Aids:
Angel
American Software Property Management Package
Cornwall Apartment Management
Datebook
ESQ-1
Guardian
Professional Time Accounting
Property Management
PAS 3 Medical
PAS 3 DENTAL
Sales Pro
Univair 9000 Series Family Medical Management
Univair 9000 Series Family Dental Management
Univair 9000 Series Insurance Agency Management
Univair 8000 Medical Management
Univair 8000 Dental Management
Wiremaster

Books, Periodicals, Accessories
APL—An Interactive Approach
Accounts Payable and Accounts Receivable-CBASIC
The CP/M Handbook (with MP/M)
The C Programming Language
8080/Z80 Assembly Language Techniques For Improved Programming
Fifty BASIC Exercises
General Ledger-CBASIC
H.W.Sams Crash Course in Microcomputing
Introduction to Pascal
Lifelines
Pascal User Manual and Report
The Pascal Handbook
The Pascal Primer
Payroll with Cost Accounting —CBASIC
Structured Microprocessor Programming
Using CP/M—A Self-Teaching Guide
Smartmodem
DC Data Cartridges
Flippy Disk Kit
Floppy Saver
Diskette Drive Head Cleaning Kits
Vari Clean Cleaning Kit

Disk Operating Systems
Software Bus Family
SB-80
CP/M-80
MP/M

Hard Disk Integration Modules

Media and Formats

These are diskette, cartridge disk and cartridge tape format codes, to be specified when ordering software for listed computer or disk systems. All software products have specific requirements in terms of hardware or software support, such as MPU type, memory size, support operating system, or language.
New formats appear in **red.**

ADDS Multivision	RT
Altair 8800	B1
Altos	A1
Apple CP/M-80 13 Sector	RG
Apple CP/M-80 16 Sector	RR
BASF System 7100	RD
Blackhawk Micropolis Mod II	Q2
California Computer Sys 8 in	A1
CDS Versatile 3B	Q1
CDS Versatile 4	Q2
Columbia Data Products 8 in.	A1
Columbia Data Products 5.25 in.	S4
COMPAL-80	Q2
Computer Ops N.C. HQ	S2
CPT 8000	A1
Cromemco System 3	A1
Cromemco System 2 SD/SS	R6
Cromemco System 2 DD/SS	RX
Cromemco System 2 DD/DS	RY
CSSN Backup	T1
Datapoint 1550/2150	A1
DEC VT18X	SD
Delta Systems	A1
Digi-Log Microterm II	RD
DTC Micro 210A	SC
Durango F-85	RL
Dynabyte DB8/2	R1
Dynabyte DB8/4	A1
Exidy Sorcerer + Lifeboat CP/M-80	Q2
Exidy Sorcerer + Exidy CP/M-80 5.25 in.	RW
Exidy Sorcerer + Exidy CP/M-80 8 in.	A1
EXO	A1
Findex	P6
Heath H8 + H47	A1
Heath H89 + Magnolia CP/M-80	P7
Heath H89 + Heath CP/M-80	P7
Helios II	B2
Hewlett-Packard 125, 5.25 in.	SB
Hewlett-Packard 125, 8 in.	A1
IBEX 7100	RQ
iCOM 2411 Micro Floppy	R3
iCOM 3712	A1
iCOM 3812	A1
IMSAI VDP-40	R4
IMSAI VDP-42	R4
IMSAI VDP-44	R5
IMSAI VDP-80	A1
Industrial Microsystems 5000	RA
Industrial Microsystems 8000	A1
Intel MDS SD	A1
Intertec Superbrain DOS 0.1	R7
Intertec Superbrain DOS 0.5-2.x	RJ
Intertec Superbrain DOS 3.x	RK
Intertec Superbrain QD	RS
ISC Intecolor 8063/8360/8963	A1
Lexitron VT1303 DSDD	S8
Lexor Alphasprint Model S1	S1
Meca Delta-1 5.25 in.	P6
Micromation	A1
MicroMega 85	SC
Micropolis Mod I	Q1
Micropolis Mod II	Q2
MITS 3200-3202	B1
Morrow Discus	A1
Mostek	A1
MSD 5.25 in.	RC
MULTI-TECH-I	Q2
MULTI-TECH-II	Q2
Nascom (Gemini drives)	R3
NCR 8140/9010	A1
NEC PC-8001	RV
NNC-80	A1
NNC-80W	A1
North Star SD	P1
North Star DD	P2
North Star QD	P3
Nylac Micropolis Mod II	Q2
Ohio Scientific C3	A3
OKI IF-800	RZ
Osborne-1	SA
Pertec PCC 2000	A1
Processor Technology Helios II	B2
Quay 500	RQ
Quay 520	RP
RAIR DD	RE
Research Machines 5.25 in	RH
Research Machines 8 in	A1
Sanco 7000 5.25 in	RQ
SD Systems 5.25 in	R3
SD Systems 8 in	A1
Spacebyte	A1
Tarbell 8 in	A1
TEI 5.25 in.	R3
TEI 8 in	A1
Televideo DD/DS	S5
T.I.P. (Alloy Engineering, Inc.)	T3
Toshiba T200	SF
TRS-80 Model I Standard	M-R2
TRS-80 Model I + Shuffleboard 8 in	A1
TRS-80 Model II	A1
Vector MZ	Q2
Vector System 2800	A1
Vector System B	Q2
Vector VIP	Q2
Vista V-80 5.25 in. SD	R8
Vista V200 5.25 in. DD	P6
Wangwriter	SE
XEROX 820, 5.25 in.	S6
XEROX 820, 8 in.	A1
Zenith Z89 + Magnolia CP/M-80	P7
Zenith Z89 + Zenith CP/M-80	P7

COMING SOON!
ARCHIVES 1
AVL Eagle II
Commodore CBM/Pet 2000/3000/4000 Series
Commodore CBM/Pet 8000 Series
Dysys ESC 4500
EXO
Heurikon
Heuristics
MULTI-TECH III, IV
Nascom/Lucas
NCR 2950
Northern Telecom 503
Pet/CBM 2000/3000/4000 Series
Pet/CBM 8000 Series
Philips P2000 with MMU
Solid State Technology
Toshiba T250
TRS-80 Model III
Zeda 580

Single-sided single-density disks are supplied for use with double-density and double-sided 8" soft sector systems.
IMSAI formats are single-density with directory offset of zero.
A media surcharge will be added to orders for the following: Tape formats T1 and T3
The list of available formats is subject to change without notice. In case of uncertainty, call to confirm the format code for any particular equipment.

Note that Lifeboat programs have specific hardware and memory requirements but will run on most CP/M machines with 48 KByte available.

SEND FOR FULL LIFE BOAT CATALOGUE $10

FMS 95 CANTERBURY ROAD, MIDDLE PARK, VICTORIA, 3206. (03) 699 9899

HIGH DENSITY VDU CARD

Concluding his series on building a VDU, Brian Hawkins presents the full circuit diagrams and the software to control it.

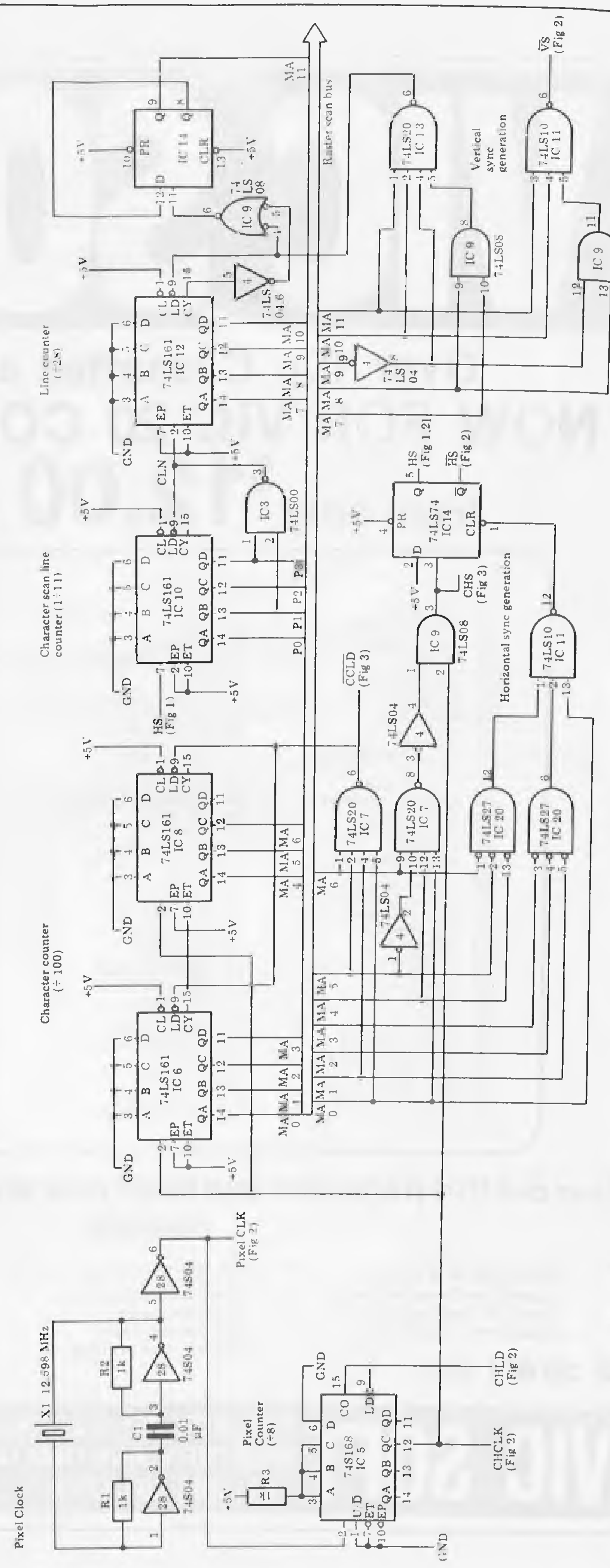


Fig 1 Main timing generation logic

VIC SOFT

Over 100 Cassettes available
NOW FOR VIC 20 COMPUTERS
from only $12.00 each

Ref No.	Title	Ref No.	Title	Ref No.	Title
001	DIG & BURY	101	HESCCOUNT	403	BREAKOUT
002	U.F.O. SHOOTING	102	DRAGON MAZE	404	INVADER FALLS
003	SUB-ATTACK	103	HESPLOT	405	VICTERM
004	MOORSE	104	MAZE OF MIKOR	406	VICAT
005	RAINBOW	105	PINBALL	407	VICHECK
006	CRAZY BALLOON	106	SKIER	408	SPACE DIVISION
007	MOLE ATTACK	107	SIMPLE SIMON	409	LONG DIVISION
008	SUBMARINE	108	TANK WARS	410	SUPER HANGMAN
009	NAVAL BATTLE	109	VICTREK (8K)	411	GLOBBLER
010	FIRE TREK	110	FUEL PIRATES	412	THE ALIEN
011	TREASURE CARRY IN	111	PAK BOMBER	413	3D MAZE
012	HEAD ON	112	LASER BLITZ	414	CAROM
013	I.C.B. MISSION	201	HOUSEHOLD FINANCE 1 & 2	415	RACEWAY
014	BALLOON BOMBER	202	HOUSEHOLD FINANCE 3 & 4	416	LASER WAR
015	TEN PIN BOWLS	203	VIC TRAP	417	DRAGON MAZE
016	HI-RES. DEMO	204	SEAWOLF	501	JUNGLE
017	VIC CUBE	205	BOUNCE OUT	502	DUNE BUGGY
018	MONSTER CHASE	206	MONSTER MAZE	503	SKI-RUN
019	LASER FIGHT	207	HOME INVENTORY	504	THE WORM
020	EUREKA STOCKADE	208	MATHS HURDLER	505	SUPER WORM
021	LUNAR LANDER	209	LOAN ANALYSER	506	COSMIC BATTLES
022	ROCKET COMMAND	210	CODE MAKER	507	SPACE DEFENSE
023	CITY BOMBER	211	CODE BREAKER	508	FROGGER
024	MINEFIELD	212	CAR COSTS	509	CAR RACE
025	HANGMAN	301	METEORS	601	A-MAZ-ING (6K)
026	PACKMAN	302	DEPTH CHARGE	602	AMOK
027	SUPERLANDER	303	GUNFIGHT	603	MASTER WIT
028	TARGET	304	ASTEROIDS	604	ALIEN BLITZ
029	MAZE	305	VICVADER	605	KIDDIE CHECKERS
030	PIANO/DRAGON	306	UFO ATTACK	606	SIMON
031	SOUND EFFECTS	307	SLOT MACHINE	607	WALL STREET
032	CHEQUE BALANCER	308	SLALOM	608	VICALC
033	AUNT SALLY	309	INDY 500	701	SPACE INTRUDERS
034	SEMAPHORE (3K)	310	OTHELLO	801	CHECKER KING
035	SEMAPHORE (6K)	401	STAR WARS	802	TIME TREK
036	CHARACTER CREATOR	402	SKY MATHS (6K)	901	MIN-VIC

Tear out this page now and send your order in to the following dealers:

WESTERN AUSTRALIA
Victoria Park Computer Centre,
Suite 1, 38 Teddington Street,
Victoria Park (09 361 1355

SA, NT, VIC & TAS
Computer Imports Colour
Computer Centre,
220 Morphett Street,
Adelaide (08) 211 8146

NSW, QLD, ACT & NZ
Computer Cellar Pty. Ltd.,
136 Maitland Road,
Mayfield,
Newcastle (049) 67 5700

or direct to...

VIC SOFT
P.O. BOX 251 GOSNELLS 6110
WESTERN AUSTRALIA
TEL. (09) 361 1355

EXPANDER BOARD FOR VIC 20

AVAILABLE NOW!

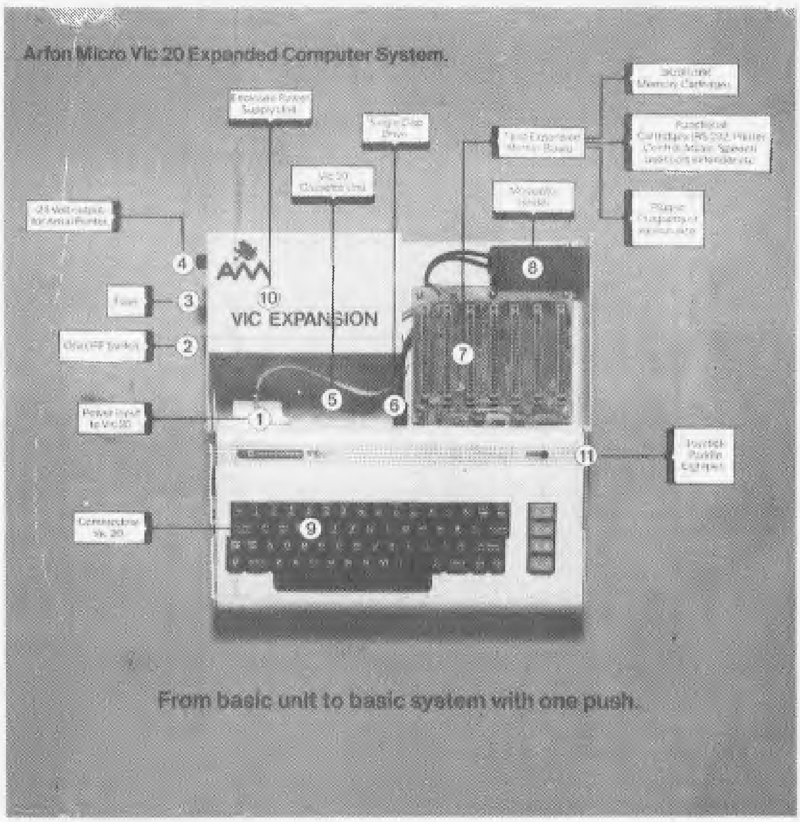
It is still under wraps but we will tell you WHAT IT CAN DO!!

WESTERN AUSTRALIA
Victoria Park Computer Centre,
Suite 1,
38 Teddington Street,
Victoria Park (09) 361 1355

SA, NT, VIC & TAS
Computer Imports Colour
Computer Centre,
220 Morphett Street,
Adelaide (08) 211 8146

NSW, QLD, ACT & NZ
Computer Cellar Pty. Ltd.,
136 Maitland Road,
Mayfield,
Newcastle (049) 67 5700

By now you must be delighted with your Vic 20 and are starting to realise more fully the enormous potential that this product offers.
The value of a computer that uses the same micro chip and operating system as a world wide success gives it a range that will be hard to match in a totally new computer. The addition of a colour display makes it a very up-market product right from the start.
We have taken this microcomputer with its power unit, its modulator, decided upon the expansion you would need and condensed it all into an expandable computer system. To allow you sufficient power we have replaced the small power supply with a specially designed power pack. We are giving you a mother board with 7 cartridge expansion sockets.
The cassette port and disc port are still totally usable as are the games paddles etc. Your modulator will now be held at the rear of the expansion board.
We have housed your new computer system including the Vic 20 itself in an aluminium shell. The front of the housing follows the contour of your Vic 20 while the rear is raised to give protection to your cartridges.
As an option we offer an aluminium cover to give greater security and protection, at the same time providing a base for your television. Educational users will no doubt secure this cover to the main shell preventing the removal of cartridges.
The power pack has an extra 24 volt rail and socket at the side of the shell that will power the new Arfon Micro Printer which is being especially designed for our Vic 20 range.
We wish you continued success with your new Vic 20 system and look forward to any correspondence you may like to send to our users group.



APPROVED FOR USE WITH THE VIC 20 COLOUR COMPUTER

From basic unit to basic system with one push.

MAIN DISTRIBUTOR:

VIC SOFT

P.O. Box 251
Gosnells 6110 W.A.
Tel. (09) 361 1355

DINERS CLUB

AMERICAN EXPRESS

VIC 20

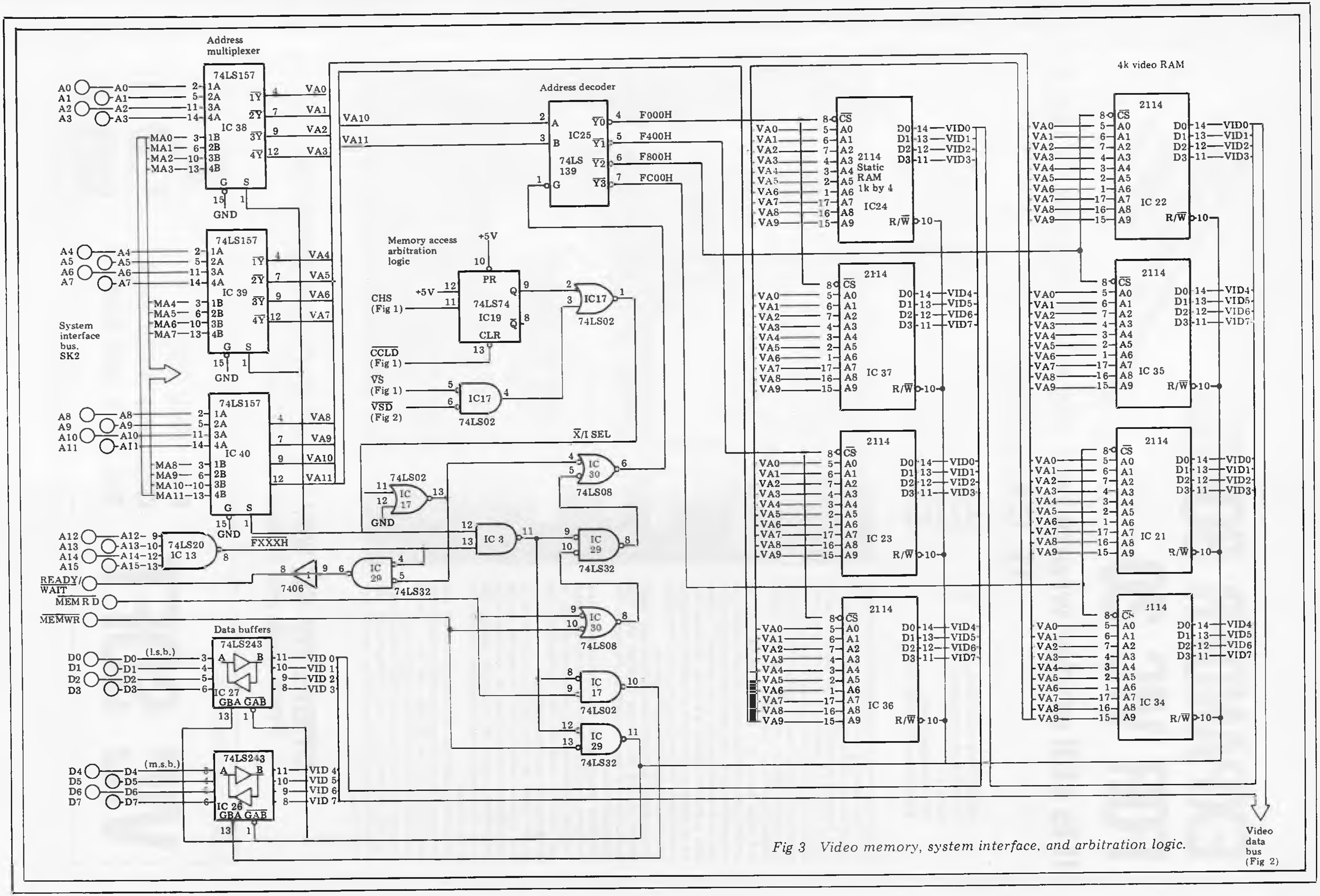


Fig 3 Video memory, system interface, and arbitration logic.

```
        CALL    CURINC          ; Call increment cursor
        CALL    INVCLC          ; Replace cursor
;
RETCH:  POP     H               ; Restore registers
        POP     D
        CALL    INVCLC          ; Replace cursor
;
RETCH:  POP     H               ; Restore registers
        POP     D
        POP     B
        POP     PSW
        RET
;
;+++++++++++++++++++++++CARRIAGE RETURN++++++++++++++++++++++
;
CARRET: CALL    INVCLC          ; Remove cursor from screen
CAR1:   CALL    FNDLOC          ; Calculate location
        MVI     A,20H           ; Load acc with ASCII space code
        MOV     M,A             ; Load screem memory
        CALL    CURINC          ; Incr cursor position
;
        LDA     CHARNO          ; Test for 1st loc in next line
        CPI     00H
        JNZ     CAR1            ; If not, jump to CAR1
;
        CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;+++++++++++++++++++++++CURSOR ADVANCE++++++++++++++++++++++
;
CURADV: CALL    INVCLC          ; Remove cursor
        LDA     CHARNO          ; Pick up char no
        CPI     4FH             ; Test for end of line
        JZ      CURAD1          ; If so leave CHARNO alone
        INR     A               ; If not, incr it
        STA     CHARNO          ; Store CHARNO away
CURAD1: CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;+++++++++++++++++++++++BACKSPACE CURSOR++++++++++++++++++++++
;
BACKSP: CALL    INVCLC          ; Remove cursor
        LDA     CHARNO          ; Pick up'char number
        CPI     00H             ; Test for start of line
        JZ      BACK1           ; If so, leave char no alone
        DCR     A               ; If not, decrement it
        STA     CHARNO          ; Store it away
BACK1:  CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;+++++++++++++++++++++++LINE FEED (CURSOR DOWN)++++++++++++++
;
LFEED:  CALL    INVCLC          ; Remove cursor
        LDA     LINENO          ; Pick up line number
        CPI     25              ; Test for last line
        JZ      LF1             ; If so leave line no alone
        INR     A               ; Else incr it
        STA     LINENO          ; Store it
LF1:    CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;+++++++++++++++++++++++CURSOR UP++++++++++++++++++++++++++
;
CUP:    CALL    INVCLC          ; Remove cursor
        LDA     LINENO          ; Pick up line no
        CPI     00H             ; Test for first line
        JZ      CUPI            ;  & leave alone if so
        DCR     A               ; If not decr it
        STA     LINENO          ; Restore it
CUP1:   CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;+++++++++++++++++++++++CLEAR SCREEN+++++++++++++++++++++++
;
CLRSCN: CALL    CSCRN           ; Call clear screen subroutine
        CALL    INVCLC          ; Replace cursor
        JMP     RETCH
;
;**********************************************************
;
;       CURSOR INCREMENT SUBROUTINE
;
;       Increments cursor, adjusts pointers for new line and
;       scrolls screen when bottom line reached
;
;**********************************************************
;
CURINC: LDA     CHARNO          ; Increment character no pointer
        INR     A
        STA     CHARNO          ; Test for end of line
        CPI     50H
        RNZ                     ;  and return if not at line end
;
        MVI     A,00H           ; Reset character no to zero
        STA     CHARNO
;
        LDA     LINENO          ; Incr line no
        INR     A
        STA     LINENO
        CPI     26              ; Test for end of page
        RNZ                     ; Return if not at page end
;
        LDA     LINENO          ; Decr line no
        DCR     A
        STA     LINENO
        CALL    SCROLL          ; Call scroll subroutine
        RET
;
;**********************************************************
;
;       SCROLL SCREEN SUBROUTINE
;
;       Moves line 1 to line 0, line 2 to line 1, etc, line
;       25 to line 24 and clears line 25
;
;**********************************************************
;
SCROLL: PUSH    B               ; Save regs
        PUSH    D
        PUSH    H
;
        MVI     A,00H
        STA     SCHARN          ; Clear scroll char number
```

```
        STA     SLINEN          ; Clear scroll line number
;
        LXI     B,SCNLIN        ; Load B with start of line 1
        LXI     H,SCRSRT        ; Load H with start of line 0
        LXI     D,LOFSET        ; Load D with line adjust value
;
SCNEXT: LDAX    B               ; Load from location in B into acc
;                                  ie, pick up code from line N+1
        MOV     M,A             ; Store in A, ie, move to line N
        INX     B               ; Incr pointer in B
        INX     H               ; Incr pointer in H
;
        LDA     SCHARN          ; Incr scroll char no
        INR     A
        STA     SCHARN
;
        CPI     50H             ; Test for end of line
        JNZ     SCNEXT          ; If not, jump to move next char
        DAD     D               ; Adjust H & L to start of next line
        MOV     A,C             ; Adjust B & C to start of next line
        ADD     E               ;  by adding DE to HL
        MOV     C,A
        MOV     A,B
        ADC     D
        MOV     B,A
;
        MVI     A,00H           ; Clear scroll char counter
        STA     SCHARN
;
        LDA     SLINEN          ; Incr scroll line no
        INR     A
        STA     SLINEN
;
        CPI     25              ; Test for last line
        JNZ     SCNEXT          ;  & jump if not
;
CLRLIN: MVI     A,20H           ; Set acc to ASCII space
        MOV     M,A             ; Store in screen memory
        INX     H
        LDA     SCHARN          ; Incr scroll char no
        INR     A
        STA     SCHARN
;
        CPI     50H             ; Test for end of last line
        JNZ     CLRLIN          ;  & jump if not
;
        POP     H               ; Restore registers
        POP     D
        POP     B
        RET
;
;**********************************************************
;
;       FILL SCREEN SUBROUTINE
;
;       Enter with code in acc
;
;**********************************************************
;
FSCRN:  PUSH    B               ; Save registers
        PUSH    D
        PUSH    H
;
        LXI     H,SCRSRT        ; Load HL with start of screen map
        LXI     D,LOFSET        ; Load DE with incr to next line
;                                  start value
        MOV     B,A             ; Store code to fill with in B
        MVI     A,00H
        STA     LINENO          ; Set line no to 0
        STA     CHARNO          ; Set char no to 0
FSMEM:  MOV     M,B             ; Store code in screen memory
        INX     H               ; Incr memory pointer
        LDA     CHARNO          ; Incr char counter
        INR     A
        STA     CHARNO
        CPI     50H             ; Test for end of line
        JNZ     FSMEM           ; Jump back if not to fill next
;                                  char position
        DAD     D               ; Adjust HL to start of next line
        MVI     A,00H           ; Reset char counter
        STA     CHARNO
        LDA     LINENO          , Incr line number counter
        INR     A
        STA     LINENO
        CPI     26              ; Test for end of page
        JNZ     FSMEM           ; Jump back if not to fill next
;                                  char position
        MVI     A,00H           ; Reset line number
        STA     LINENO
;
        POP     H               ; Restore registers
        POP     D
        POP     B
        RET
;
;**********************************************************
;
;       CLEAR SCREEN SUBROUTINE
;
;       Clears the screen
;
;**********************************************************
;
CSCRN:  MVI     A,20H           ; Load acc with ASCII space char
        CALL    FSCRN           ; Call fill screen subroutine
        RET
;
;----------------------------------------------------------
```

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. Organisers are requested to notify APC of forthcoming events well in advance to allow time for inclusion in 'Diary Data'.

Sydney	HP (Aust) Productivity Course, North Ryde, Enquiries: Ms Nankervis (02) 887 1611	June 14–18, 1982
Sydney	Data 82, Centrepoint, Contact: Graphic Directions	August 10–12, 1982
Canberra	National Microcomputer Conference, Contact: MICSIG, ACS	August 18–20, 1982
Melbourne	Data 82, Melbourne Showgrounds	November 9–11, 1982
Sydney	The 1st Australian Personal Computer Show, Centrepoint, Contact: Australian Exhibition Services	March 10–12, 1983

USER GROUPS INDEX

Below is a list of alterations and additions to the list of user groups published in the March issue. Updates have also been published in the April and May issues of APC.

ACTAPPLE
The 'ACT Apple Users' Group' meets on the second Thursday of each month. For more information contact Jeff Brock, Secretary/Editor, P.O. Box 1231, Canberra City, 2601. (062) 313630.

ILLAWARRA SUPER 80 USERS GROUP
The group was formed in January of this year and meets on the first Monday of each month (usually!?) commencing at 5.30pm at the Australian Offices, 86 Market Street, Wollongong. The postal address of the group is P.O. Box 1775, Wollongong 2500.

APPLE USERS GROUP OF WA
The new Secretary/Treasurer, John Currie, can be contacted at 8 Solomon Street, Fremantle 6160. (09) 7257.

NETWORK NEWS

Here is a list of all Australian personal computer networks. As more networks appear – and as more facilities are added to existing ones – we'll report them in this section, which appears monthly.

The Australian Beginning. Operator: The Australian Beginning Pty. Ltd. 364 La Trobe Street, Melbourne. Tel: (03) 329 7998. Facilities: Information service, electronic mail, software storage, and software downloading. Hours: 24 hours/day, 7 days/week.

INFONET. Operator: Network Services Division of Computer Sciences of Australia Pty. Ltd., 460 Pacific Highway, St Leonards, NSW. Tel: (02) 439 0033. Facilities: Access to databases produced by the Australian Bureau of Statistics and the Institute of Economic and Social Research. Hours (E.S.T.): Monday to Friday (7am to 9pm), Saturday (8am to 5pm) and Sunday (8am to 11.30am).

AUSINET. Operator: ACI Computer Services, P.O. Box 42, Clayton, Victoria. Tel: (03) 544 8433. Facilities: Medium to databases whose subject coverage includes agriculture, education, energy, industry, public affairs, science and technology and an online Australian database directory. Hours: 8.30am to 9.00pm E.S.T. Monday to Friday.

CSIRONET. Operator: CSIRO Division of Computing Research, P.O. Box 89, East Melbourne. Tel: (03) 419 1333. Facilities: Carries a broad range of scientific and technological databases compiled in Australia and overseas, including general scientific research, renewable energy resources, agriculture, water resources, mining and geology, and thermodynamics.

IP Sharp Associates Network. Operator: IP Sharp Associates Pty. Ltd., 13th Floor, 175 Pitt Street, Sydney. Tel: (02) 232 6366. Facilities: The network is an international time sharing data processing network, the host computers being located in Toronto, Canada. Hours: 24 hours/day, 7 days/week.

TJ's WORKSHOP

Our monthly pot-pourri of hardware and software tips for the popular micros. If you have a favourite tip to pass on, send it to 'TJ's Workshop', APC, P.O. Box 115, Carlton, 3053. Please keep your contributions as concise as possible. We will pay $10 for any tips we publish.

'PROAID' PROGRAMMING AID

Ian Davies' APC-80 is an excellent machine-language program made up of a number of modules. Some modules are designed to speed up the writing and editing of Basic Programs (single key commands, key debounce, etc) while others provide new execution routines (BEEP, MOVE, etc). The Program has become rather large, and resides in high memory where it can conflict with other high memory machine language routines. We have therefore shortened and modified it to serve purely as an aid to programming. It is suitable for TRS-80 (Model I) or System-80, of any capacity.

PROAID consists of single commands, key debounce, blinking cursor, and key repeats (except shift @). The accompanying listing shows that it is located at the start of the Basic program area (42E9H), and automatically protects itself by relocating the floor of the Basic Program area above itself. Once in place, it is therefore safe to write or CLOAD a Basic program, and edit or run it; it is also safe to input and use high memory machine language routines (e.g. sort or beep). Since PROAID only occupies 218 bytes (the final 18 bytes being an unprotected initialisation routine), it will not cramp the style of those who write very large Basic programs.

Because of its location in low memory, the listed hex code must be input and saved on tape using a high memory monitor program, such as BMON or ZMONH (Micro-80) or MON-3 (Howe Software). The required addresses are:

START = 42E9
END = 43D4
ENTRY = 43C3

Thereafter it can be loaded. Before writing or CLOADing a Basic program, by entering SYSTEM, P or PROAID, and /. Successful loading is shown merely by the appearance of the blinking small-block cursor, which remains as a constant reminder that PROAID is there. When running graphics programs, you should reinstate the normal stationary cursor thus:

POKE 17153, 0

To bring the blinking back again, use:

POKE 17153, 211

The single key commands are activated by "shift letter-keys", and in some cases the commands differ from this in APC-80. We have tried to make them easy to remember, either by using the first letter of the command (e.g. A for AUTO) or using a unique letter of the command (e.g. Q for SQR), or at least adjacent to a meaningful letter (e.g. T for TABC, Y for THEN). There is only one maverick, J for LPRINT, if you don't have a printer, change address 4378H from AF to 90 for RESTORE, or to 99 for DEFINT, or to 9F for RESUME, but do not use 95 for ELSE (as this causes the system to crash and loses your program, for some strange reason).

Development of a counterpart program consisting of the main execution modules of APC-80 is well under way, and should be reported soon.

Dr R. Langley,
Dr D. Hewett.

```
42E9  00 00 05 00 05 CD F4 42 C3 89 43 2A EA 42 2B 7C
42F9  A7 20 0A 2A 20 40 7E EE D3 77 21 00 05 22 EA 42
4309  C3 31 43 21 36 40 01 01 38 16 00 0A 5F AE 73 A3
4319  20 08 14 2C CB 01 F2 14 43 C9 C5 01 DF 04 CD 60
4329  00 C1 0A A3 C2 FB 03 C9 2A 63 43 7E A7 20 22 CD
4339  0C 43 FE 61 D8 DD 21 0E 43 32 47 43 DD 7E 00 21
4349  65 43 77 01 67 43 CD 8C 2B AF 02 21 67 43 22 63
4359  43 2A 63 43 7E 23 22 63 43 C9 67 43 00 00 00 20
4369  20 20 20 20 20 20 B7 92 F7 88 9D F8 8D 91 89 AF
4379  C9 B4 FA 87 B1 E5 DD F9 F4 BC BF C0 C4 BA CA B9
4389  21 00 05 A7 28 05 32 E9 42 18 27 21 01 38 B6 CB
4399  25 F2 97 43 A7 28 1B 2A EC 42 2B 7C A7 3E 00 20
43A9  11 3E 03 32 EA 42 21 60 01 3A E9 42 FE 60 20 02
43B9  3E 00 22 EC 42 C9 00 00 00 00 21 EE 42 22 16 40
43C9  21 C1 43 22 A4 40 CD 49 1B C3 CC 06
```

MBASIC TIP

In an earlier issue P.H. Elliott points out how to duplicate MBasic lines with control-A. There is another very useful technique akin to this. When testing a program, a syntax error can be corrected and the program continued without restarting as follows:

1. The program is run and halts with an error in, say, line 550.
2. Edit mode is entered automatically.
3. Enter Q. Edit mode is exited, without altering the statements in error or losing variable values.
4. Enter control-A. Statement 550 will appear in direct mode, without a line number.
5. Edit the line to correct it, without re-entering a line number.
6. Press RETURN, and the statement will execute in direct mode.
7. Enter GOTO 560 (assuming this is the next statement in sequence), and the program will continue.

Note that it will still be necessary to correct the statement in error before saving the program, but this technique can save debugging time.

Brian Hebbes.

PRINTING TO 3 DECIMAL PLACES

Basic normally prints numerical values less than 0.01 in the exponent form 1.23E–03. This can be annoying and untidy if you are listing a table of figures where the third decimal place is needed (e.g. dimensions in mm or thou or prices).

To overcome this I developed the short subroutine below, which can be inserted into the program where the number is to be printed. A sample run is shown to demonstrate the improvement to the format.

Sample program

(Line 20 is for print enhancement)

```
10  INPUTN
20  IFN<.01THENPRINT
    ".00";CHR$(48+INT
    (N*1000+.5)):GOTO
    40
30  PRINTN
40  GOTO 10
```

Example of formats

Normal	Improved
.123	.123
.012	.012
1E-03	.001
5E.03	.005
.054	.054
.543	.543

D. Gayler.

SYSTEM 80 SEARCH

This program allows the user to determine which lines of his program contain any Basic keyword or any other form of text string. Also an indication will be given (by line number repetition) of how many times the search key was found on each line.

```
2CLS: DEFINTI.L:
L–OEEJ(16634)*256+
PEEK (16633)–4: K–
17133: FORI-K 5 TOL:
IFPEEK (I)=0: N–PEEK
(1+4)*256+ PEEK(I+3):
I–I+4: NEXTI: ELSEFOR
J–K TOK+20: IFPEEK(J)
–38: NEXTJ: ELSEI
FPEEK (J)–0: PRINTN; :
NEXTI: ENDELSE IF
PEEK(J) <> PEEK(I+J–K):
NEXTI: ENDELSENEXTJ:
NEXTI: END
```

When typing the program use ? instead of PRINT otherwise the ND of the final END will have to be inserted using the Edit mode. This is because the keywords are not tokenised after the line has been entered and the line is too long until this has taken place. Using the ? means that less initial text is entered.

Line number 1 is also required by this program to create the search key. This is used in the following manner:

The search key may be any basic keyword or any text not including a keyword, e.g. 1 PRINT
1 SPIDERS 1PRINT "SPIDERS".

But note the problem if the search key is 1 TARGET

GET is a Basic keyword, hence the word TARGET will not be found. To overcome this problem the text may be shortened or the special character (&) used, e.g. 1 TARGE or 1 TARG&T. Keywords split by blanks are the same as those not, e.g. GO TO is the same as GOTO but GOTO24 is not the same as GOTO 24.

Note: & is reserved for use as follows:

1. Immediately after the line number of line 1, to separate a numeric search key from the line number. This *must* be used for a numeric search key to avoid accidental deletion of a line in the user program – e.g. 1&24 will create a search key for 24, while 1 24 will delete line 124 if it exists.
2. Following an extra blank after the line number 1 to enable a blank to be used as the search key e.g. 1 &.
3. In any other position the character in a search key occupied by the & will not be checked. Therefore, & can be used in place of any character to disallow a keyword if the spelling of a word is not known exactly.

Line 1 may contain any data for the search key and does not have to be in correct Basic syntax. This line is not intended to be run.

There are some restrictions with this routine:

1. The user program must start at a line number greater than 2.
2. The search key must not exceed 20 characters (although keywords may be counted as only 1).
3. The search programs and search key line will occupy up to 190 bytes.

After the search key line is entered simply enter RUN 2. To run the user program either RUN number, where number is the first line number of the user program. Alternatively, replace line 1 by 1 GOTO number then RUN will access the user program directly.

Each time the search key is found, the line number of the line containing the search key will be printed starting at the top left corner of the screen. The line number will be printed each time the search key is found, therefore the number of times the line number appears indicates how many times the search key occurred on this line. Approximately 12 line numbers are printed for each line of display.

M. Hale.

APPLE SCEEN BUG

With most TV/monitors the Apple produces half-intensity dots at certain points on the screen. This occurs with colours 5, 6 and 7 and is apparent in both b/w and colour. To assess whether this problem occurs with your system try program 1 which produces three half-intensity dots and then changes them to full intensity. If the dots remain unchanged then your system has no problems and you need read no further. To see how annoying this problem is run programs 2 and 3. The problem in plotting the point X, Y only occurs in a limited number of situations. Namely:

1. HCOLOR = 5, X odd and X+1 divisible by 7.
2. HCOLOR = 6, X even andX+1 divisible by 7.
3. HCOLOR = 7 and X+1 divisible by 7.

Normally the remedy is to set HCOLOUR = 4 (i.e. black) and then to plot the point X+1, Y in black. Thus program 2 can be corrected by adding the lines:

```
35 HCOLOR = 4: HPLOT
14,1 TO 14,150: HCOLOR
– 5 65 HCOLOR= 4:
HPLOT 49,1 TO 49,150:
HCOLOR – 6 105
HCOLOR = 4: HPLOT
98,1 TO 98,150: HCOLOR
= 7
```

The remedy for program 3 is similar. Add the lines

```
42 HCOLOR – 4: HPLOT
245,0 TO 245,100
44 HPLOT 49,0 TO 49,100
46 HPLOT 231,20 TO
231,80.
```

Note that if HCOLOR = 7 is used in program 3 instead of HCOLOR – 6 then the given correction produces two small black dots at 49,0 and 49,100. This can be corrected by changing line 44 to:
44 HPLOT 49,1 TO 49,99.

This problem is caused by a fault in the Apple's plotting routines. Since this can be corrected in Basic then the machine code routines could also be corrected. Let's hope that Apple will correct this most annoying fault.

G. Manson.

Program 1

```
10  HOME : TEXT : HGR
20  HCOLOR– 5
30  HPLOT 13,100
```

Cont. on Page 82

BACK ISSUES SERVICE

Here is a complete guide to available back issues of APC. A quick guide to their contents is shown here. Check the coupon for the issues you require.

Volume 1 Nos. 1, 2, 3, 5 & 12 and Volume 2 Nos. 1 & 2 are no longer obtainable, but photocopies of any article published in these issues are available on request at a cost of $2.00 each. Please state clearly volume and issue number.

Volume 1 No. 3
Benchtest: Tandy TRS-80 Model II. Animistics. M68000-Motorola's Sweet 16. Computer Games: Big trees and the Alpha-Beta algorithm. Plotting in Three Dimensions. The Complete Pascal: Loops. Checkout: Exatron Stringy Floppy. Z80's control signals. Programming the Simple Approach. Z80 Instruction Set. Programs: Byte Saver (TRS-80), TRS-80 Graphics, Word Processor (Sorcerer), Space Slalom (Apple), Robot Nim (PET).

Volume 1, No. 4
Benchtest: Hewlett Packard's HP85. Benchtest: Texas TI 99/4 Home Computer. Gateways to Logic. Adding a different family of chip to your micro. Computer Games: Speeding up the Search The Complete Pascal: Data Structures. Programs: 3D Noughts & Crosses (TRS-80), PET Backgammon

Volume 1 No. 5
Limited supplies only available
Benchtest; Sinclair ZX80. Marketing Strategies of Micro Manufacturers. Computer Games. Gateways to Logic: Boolean Algebra. Micro Chess: A brief history of chess programming. The Complete Pascal: Control Structures II. Whistling up Colour: Apple "hands off" control. Programs: Showpiece (Apple), Dots & Boxes (PET).

Volume 1 No. 6
Benchtest: CBM Model 8032 'SuperPET'. Microchess. Basic Basic. Benchtest: Super-Brain. Programs: Lunar Lander (TRS-80), PET Fighter Pilot, Apple Plotting, Lprint to Print Utility (TRS-80), ZX80 Breakout, Graph (TRS-TRS-80).

Volume 1 No. 7
Robotics. Micro Chess. Computer Games: Guessing the odds. APC-80. The Complete Pascal: Records and files. Calculator Corner: Speed up your TRS-80 pocket computer programs. Programs: TRS-80 Tarot, PET Bloobers, PEEK & POKE for Apple Pascal, PET Demolition, Superboard Bug Bypass, String Function, Several Sorts.

Volume 1 No. 8
Benchtest: Atari 400 & 800. The Complete Pascal. Gateways to Logic. APC-80. Helping the Handicapped: A look at a US company specialising in this field. Programs: Reading 'System' Tapes into the System 80, Monster Multiplier (Apple II), Read-write Routines (PET), Program Formatter.

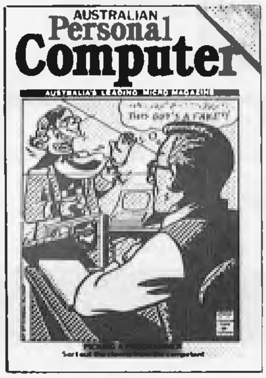
Volume 1 No. 9
The Last One: The program which writes programs. Multi-user Systems. Computer Games. Disk Basics: What makes them click? Get Well Soon: How to recover from a data tape disaster. APC-80. Gateways to Logic. Ultra-fast Tape Storage: Data-transfer rates of up to 9600 baud for the Ohio Scientific Superboard. The Complete Pascal. Forth: A review of the language. Programs: TRS-80 Target Practice, TRS-80 Four in a Row, PET Anagram, PET Obstacle Course, Minefield (TRS-80).

Volume 1 No. 10
Slow Scan TV: How micros are putting themselves in the picture. Disk Basics: More on disk drive interfacing. APC-80. On The Level: Cure those tape troubles. Computer Games: Draughts. The Complete Pascal. Choosing a Computer System: Hints on buying your first business computer. Word Processing: How to define your needs. Profile: Rudi Hoess, the man behind Apple in Australia. Multi-user Systems. Systems Development: Program development and implementation. Searching String Arrays. Turning USR(X) Routines into Data Statements. Pack It In: Squeeze more data onto your disks or tapes. Programs: TRS-80 Rocket Attacks, Superboard Mines, PET Brick Stop.

Volume 1 No. 11
Benchtest: Acorn Atom. Switch in Time: For when 'Dallas' is too important to miss . . . Gateways to Logic. Another Dimension: Three-dimensional plotting. The Complete Pascal. APC-80. Computer Games: Chess. Wordprocessor Benchmarks. WP Benchtest: Word Pro 4 Plus. Multiuser Test: MP/M. Systems Development: Profile: Dick Smith. Program Packing. Single Key Keyword

Entry: Superboard. Search/Change in Applesoft. Put Your Hooks into OSI Basic. Programs: Sorcerer Simple Word Processor, Compucolor II Function Plotting, TRS-80 Auto Fuel Consumption, TRS-80 Towers of Hanoi, PET Aircraft Landing, TRS-80 Giant Trap, PET Wire, PET Convoy.

Volume 1 No. 12
Inside Printers. Computer Games. Benchtest: Bigboard, Gateways to Logic. TRS-80 Compiler. OSI Basic-in-ROM. WP Benchtest: Magic Wand. Profile: Nigel Shepherd. Parliament House in Micro: The role of Crome co. Multi-user Test: MVT/Famos. Solar System Simulation. Drawing Lines on PET's 80 x 50 Grid. Single Key Key Development: Abbreviating commands. Programs: PET Replace, TRS-80 Demon Hunts, PET Chords, ZX80 Sliding Letters.



Volume 2 No. 1
Printer Review: Dick Smith's offering. Gateways to Logic. Joystick Interfacing: Control on the TRS-80. Classroom Zest: A computer aided instruction package described. TRS-80 Compiler. APC-80. The End of Flowcharting. Wordprocessor Comparison: Spellbinder vs Wordstar. Multiuser Test: Onyx C8002. Choosing a Programmer. Profile: The Tandy Corporation. Unix: A Review. Define Your Own Function Key on PET. Superboard II Edit: Extra function key. Multiplexing PET: Increasing I/O data handling. Programs: Program Packing (6502), Cosmic Wars '80 (TRS-80), TRS-80 Worm, System 80 Dodgem.

Volume 2 No. 2
Apples for the Handicapped. TRS-80 Tiny Compiler. Profile: Bill Anderson. Facing the Future: Barry Jones. Print Using for Applesoft. SBII Plus: Improve the Super board II. Multiple File Tape Backups. Relocating OSI Basic-in-ROM. Programs: Bigboard Real Time Clock, APC-80 Alien Invasion, PET Radio Technician Course.

Volume 2 No. 3
Benchtests: Sinclair ZX81. APC-80. Building a Bigboard. Keystroke Reduction: TRS-80. WP: The Sanders Printer. Profile: Bill and Niki Saunders. CP/M by Rodnay Zaks. Rapid Bubble Sort. Encryption with RND and USR. The 6502 Dream Machine. Vectors and the Challenger IP. Programs: TRS-80 Input Routine, Treasure Hunt (PET).

Volume 2 No. 4
Benchtest: Vic 20. Software Review: MCMON. Computer Games. Tree Access Routines. Gateways to Logic. Is it a Bird? What is a personal computer. How Computers Communicate. Benchtest: Tandy III. Profile: Gary Blom. Program Planning. 6502 Assembler in Basic. Increase Your Word Power. Programs: PET Arithmetic Test. Apple Mondrian.

Volume 2 No. 5
Benchtests: IBM, NEC PC-8000, Spellbinder. The Gates are Wide Open. Winchester Hard Disks. Computer Games: Poker. Gateways to Logic. Hal Towards 2000. Applesoft Mystery Parameter. Pet Symbolic Disassembler. Programs: TRS-80 Sailing Simulation, ZX80 Eldorado, PET Gomoku.

Volume 3 No. 1
Newsbrief. TD's Workshop. Hitachi 'Peach' Review. Benchtest: Tandy TRS-80 Color. Profile: Rodnay Zaks. The BBC Proton. Disk Sorting Routine. Getting More out of your system 80. Computer Games: Go-Moku. Patterns. Go Forth (and Multiply). Stext Graphics. Checkout: Le Sharp Microtranslator. Man, Machine and Mouth. Rubik Cube Simulation. Polygons. Programs: Apple II Get Simulation, Bug Bug, Superboard Cryptography.

Volume 3 No. 2
Apple III Review. Cubic Spline Fitting. Man, Machine and Mouth. Computer Games: Bridge. Relocatable Assembly Language Code. Binary Sort. March to a Different Drummer. Number Crunching. Inside the Interpreter. ZX81 Printer Checkout. APC-80 Rides Again. Putting Arrays on Tape. Frames of Reference. How Computers Communicate. Microcomputer Databases. Programs: Alien Seabattle (TRS-80/System 80).

Volume 3 No. 3
Checkout: The Australian Beginning. Videotext – An Overview. Benchtest: Hewlett Packard HP-125. Frames of Reference. The Man Behind the West Coast Faire. How Computers Communicate. Software Benchtest: Scripsit 2.0. User Group Index – a complete listing. TJ's Workshop. Checkout: Dick Smith's Votrax Type 'N Talk. Hi Res for the TRS-80. Computer Games. Programs: Galacti-Cube (Apple II), PET Fantasy, ZX80 Labyrinth, PET Jungle.

Volume 3 No. 4
A Machine For All Seasons: The Commodore 64. Screenplay: Atari 400. Adam Osborne: A Profile. ANS Basic. Patterns: Frames of Reference. High Density VDU Card. How Computers Communicate. APC-80: Recovery after Syntax Errors. TJ's Workshop. Calculator Corner: HP's networking system. Benchtests: Osborne 01, Applied Technology's Micro Bee. Programs: TRS-80 Reaction Timing, ZX81 Graphplot, PET Cheese, Superboard II Spin-Fighter, TRS-80 Extra.

Volume 3 No. 5
Calculator Corner: Casio's printer. TJ's Workshop. Screenplay: Tandy TRS-80. Frames of Reference. How Computers Communicate. 3D Made Easy. Getting to the Roots. High Density VDU Card. Logo – An Overview. Printer Survey. Benchtests: Texas Instruments TI 99/4A, Rank Xerox 820. Software Benchtest: FMS-80 Database. Programs: TRS-80 Double Precision Maths and Trig, Apple 3D Maze, Atari Sums for Kids, Apple Air Fight.

Please send me the following back issues of
Australian Personal Computer

Vol. 1 No.* ☐4 ☐6 ☐7 ☐8 ☐9 ☐10 ☐11
Vol. 2 No.* ☐3 ☐4 ☐5
Vol. 3 No.* ☐1 ☐2 ☐3 ☐4 ☐5

* Vol. 1 Nos. 1, 2, 3, 5 & 12 and Vol. 2 Nos. 1 & 2 are unavailable.

Any one issue $3.00; any two issues $5.70; any three issues $8.40; any four issues $11.00; further issues $2.50 each. All prices include post and packing. Cheque or P.O. payable to Australian Personal Computer, P.O. Box 115, Carlton, 3053. Please allow up to 4 weeks for delivery.

Name . . . .
Address . . . .
. . . . Postcode . . . . . .
Please charge my Bankcard.
Bankcard Number . . . .
Expiry Date . . . .
Signature . . . .

TJ's WORKSHOP

```
 40  HCOLOR– 6
 50  HPLOT 48,100
 60  HCOLOR– 7
 70  HPLOT 90, 100 TO 97, 100
 80  VTAB 21 : PRINT "WATCH THE LEFT HAND DOT"
 90  PRINT "AND PRESS THE 'SPACE BAR' " : GET A$
100  HCOLOR 4 : REM BLACK
110  HPLOT 16,100 : REM SHOULD BE NO CHANGE
115  HOME
120  VTAB 21 : PRINT "NOW WATCH THE NEXT DOT"
130  PRINT "AND PRESS THE 'SPACE BAR' " : GET A$
140  HPLOT 52,100: REM AGAIN THERE SHOULD BE
     NO CHANGE SINCE PLOTTING IN BLACK
145  HOME
150  VTAB 21 : PRINT "NOW WATCH THE LAST DOT"
160  PRINT "AND PRESS THE 'SPACE BAR' " : GET A$
170  HPLOT 100,100 : REM THERE SHOULD BE NO
     CHANGE SINCE PLOTTING IN BLACK"
180  END
```

Program 2

```
 10  HOME : TEXT : HGR
 20  HCOLOR– 5
 30  HPLOT 13,1 TO 13,150
 40  HPLOT 17,50 TO 17,150
 50  HCOLOR= 6
 60  HPLOT 48,1 TO 48,150
 70  HPLOT 52,50 TO 57,150
 80  HCOLOR= 7
 90  HPLOT 96,1 TO 96,150
100  HPLOT 97,1 TO 97,150
110  HPLOT 101,50 TO 101,150
170  HPLOT 102,50 TO 102,150
130  END
```

Program 3

```
10  HOME : TEXT : HGR
20  HCOLOR= 6
30  HPLOT 48,0 TO 244,0 TO 244,100 TO 48,100 TO 48,0
40  HPLOT 52,20 TO 230,20 TO 230,80 TO 52,80 TO 52,20
50  END
```

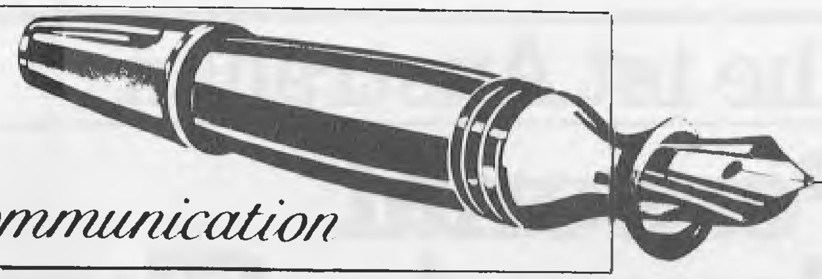
ZX81 POKE

While PEEKing and POKEing on my ZX81 recently I discovered an easy way to obtain a 34 column display (as opposed to the usual 32). The effect is obtained by POKEing 16441 and 16442 with the values 20. The following program illustrates this. When the program ends or is interrupted by BREAK, to return to normal mode press NEWLINE.

```
10 POKE 16441 , 20
20 POKE 16442 , 20
30 GOTO 10
```

Other interesting effects can be obtained by POKEing the two addresses.

R. Gosling.



Communication

APC welcomes correspondence from its readers but we must warn that it tends to be one way! Please be as brief as possible and add 'not for publication' if your letter is to be kept private. Address letters to: 'Communications', Australian Personal Computer, P.O. Box 115, Carlton 3053.

APC-80 transfer

As a real newcomer to micro-computers I have commenced buying your magazine. I found the February 1982 issue most interesting.

I, like many others, have purchased and constructed 'Super 80' Computers from D.S.E. and like those others, am a real novice, and know little about programming.

Is the APC-80 program adaptable to the 'Super-80' version of the Z80? If so, would you have an interest in publishing, perhaps, the necessary changes to the assembler (monitor) program so that APC-80 would run on the 'Super-80'.

I am sure the numbers of new dealers attracted by such a move would be surprising especially since I believe nearly all with a 'Super-80' have this computer as their first!

Thanks for a most interesting magazine.
P. Thursby.

We'd be delighted for any reader to embark on the project of converting APC-80 to other machines. However, as the program relies heavily on 'ROM-calls' converting the code to run even on another Z80 based machine would be quite an undertaking. Readers interested in converting to other machines could write to Ian Davies, C/- APC for suggestions. Editor.

Blunders

I am writing to alert you to the fact that an error (Gasp!) has been made in the March issue of *APC*. I refer to the letter in the 'Communications' column entitled "Sort or Search" written by myself.

On reading the letter I discovered two mistakes which were obviously made by the typesetter (note the diplomacy!) In the subroutine published with the letter, line 1008 should read:

```
1008 REM P > O – SEARCH
     SUCCESSFUL: P POINTS
```

and line 1040 should read:

```
1040 IF S > I (P) THEN
     L=P+1 : GOTO 1010
```

It is imperative that the latter correction be made or the subroutine will not work.
Earl Chew.

Micro Bee revisited

Some comments on the review of Applied Technology's "Microbee" in the *APC* of April '82.

Ian Davies was not able to use the exponential function in "LET" statements, and so could not run Bench-mask 8 test.

In fact, the exponential function requires a *real* variable assigned to it. BM-8, as published in an early edition of *"APC"* uses an *integer* variable. Substituting a real for an integer variable, BM-8 runs at approximately 57 secs on my machine.

Substring manipulation is provided for; in fact, its implementation is very close to that mentioned in the article on ANS BASIC in the same issue of *APC*, except that the qualifier requires a semi-colon e.g. A0$(;4,7)

The final issue of the manuals – three small booklets in bright orange covers – deals with substituting segmentation and manipulation very thoroughly, and is a great improvement on the draft photo-copy set that presumably came with Mr Davies machine.
D. Kenney.

Zaks' reply

I read with pleasure the "profile" that you ran in a recent issue of Australian Personal Computer about myself. I believe that Ms Cosic did a superb job and I am naturally flattered by the attention and pleased by the exposure. As you know, I had already contributed some editorial material to your magazine such as the article on CP/M in your October 81 issue. In the future I will make sure that our authors or myself send you articles of interest for your readers.
Dr Rodnay Zaks.

PET Get

I have been writing some games programs on my PET and have been having some trouble with the keyboard input. The GET function cannot detect a key that is being held down. How can I get the PET to recognise this?
D. Harrison.

Fortunately for games writers, the keyboard on Commodore computers is controlled by the operating system. It is possible to bypass these functions and directly find out the state of the keys. This is best done in machine code programs but the Basic operating system has some useful locations which may be examined. As an example look at location 152, this will normally be 0, but while the shift key is pressed it will be 1. If this is not enough, then location 151 shows which key is pressed down. You can use a GET to detect a key being held down. Try including in your GET loop the statement: POKE 151,255. This will confuse the machine into thinking that the key was previously released. This can be used to construct a simple screen editor with auto-repeat keys in Basic. I'll leave you to work out how to start the repeat after a preset time, at a defined rate. Editor.

Loading problems

I have had three ZX81s that have given me tape loading problems. I want a more reliable micro and have used a CBM PET that has a VERIFY command which seems a very useful feature because it reduces the risk of tape load errors. Is it available on other micros and what are the problems with cassette loading?
B. Moore.

You are quite right that tape loading problems are both common and annoying. Most of the problems can be traced back to a mismatch between the signal from the micro or back to it, or to what is called the 'azimuth' adjustment of the recording head in the tape deck. The read and record heads each have a very narrow gap between the poles of the magnet that records or the soft iron core that senses the recording on playback. If these gaps are not exactly at right angles to the line of movement of the tape, the signal is blurred unless the recording machine was similarly out of adjustment. This blurring may be bad enough to cause errors on reading the tape even if the signal levels are perfect. I have sometimes only been able to read tapes produced on other machines by forcing down one corner of the cassette or putting a cardboard wedge under one side (a procedure to adopt with caution).

Unfortunately there is nothing I can do to help you to tell the difference between a satisfactory recording and a duff one, except to advise you to find a friendly computer club where you can listen to good recordings that sound clean and undistorted. Another approach is simply to try every recorder you can beg or borrow. It is not usually neccessary to use anything of high quality. A micro club may also have someone with a meter who will help adjust any circuits if your machine has adjustments. Editor.

LAZING AROUND

by J J Clessa

At last I have received enough entries for our "fair" draw and Brad Merrick of Frenchs Forest was the first out of the hat. The correct answer was that the sheep is in Pen C (refer to diagram on page 102 of the March issue). Brad will be receiving his next twelve months of *APC* courtesy of the *APC* Subscription Department.

Quickie

No answers — no prizes. Two women sell 60kg of tomatoes in the market. The first woman sells half of them at 3kg for 50c, making $5 in all. The second sells the other half at 2kg for 50c, making $7.50, giving a total of $12.50 between them.

The following week they decide to sell their produce jointly and so they fix the price of the tomatoes at 5kg for $1. Once again they have 60kg to sell. However, when all the tomatoes have been sold they find they only have $12, and a quarrel ensues as to where the extra 50c went. Where did it go?

Prize puzzle

A number cruncher very solvable by micro — or any other method. We want you to find a six digit number which, when multiplied by an integer between 2 and 9 inclusive gives the original six digit number with its digits reversed.

Thus, if the original number is 123456 and the required integer is 8, then 123456 x 8 should equal 654321. Of course, it doesn't, but it is possible to find more than one solution to this problem. We'll accept any as eligible for this month's prize which, again, is a twelve month subscription to *APC*.

Answers, on postcards only, to June Prize Puzzle, APC, 462 Burwood Road, Hawthorn, Vic 3122, to arrive not later than last post 31 July 1982.

Learn how to use a computer for $199

and keep the computer.

sinclair ZX81

The new Sinclair ZX81 is the most powerful, yet easy-to-use computer ever offered for anywhere near the price: only $199* completely assembled.

Don't let the price fool you. The ZX81 has just about everything you could ask for in a personal computer.

*Price includes connectors for TV and cassette, FREE manual, sales tax, post and packing (but excludes mains adaptor).

★ **SINCLAIR ZX81 SALES PASS 250,000***

*Figures supplied by Sinclair Research

NEW SOFTWARE: Sinclair has published pre-recorded programs on cassettes for your ZX81, or ZX80 with 8K BASIC.

ZX PRINTER: The Sinclair ZX Printer for your ZX81. Costs $190.

ZX81 MANUAL: The ZX81 comes with a comprehensive 164-page programming guide and operating manual. Free with ZX81.

A breakthrough in personal computers

The ZX81 is a major advance over the original Sinclair ZX80 — the world's largest selling personal computer and the first for under $300.

In fact, the ZX81's new 8K Extended BASIC offers features found only on computers costing two to three times as much. Just look at what you get:
- continuous display, including moving graphics
- multi-dimensional string and numerical arrays
- mathematical and scientific functions accurate to 8 decimal places • unique one-touch entry of key words like PRINT, RUN and LIST • automatic syntax error detection and easy editing • randomize function useful for both games and serious applications • built-in interface for ZX Printer • 1K of memory expandable to 16K.

The ZX81 is also very convenient to use. It hooks up to any television set to produce a clear 32-column by 24-line display. And you can use a regular cassette recorder to store and recall programs by name.

Warranty and Service Program

The Sinclair ZX81 is covered by a 10-day money-back guarantee and a limited 90-day warranty that includes free parts and labor through our national service-by-mail facilities.

16K MEMORY MODULE: The ZX81 is expandable. Sinclair's 16K memory module plugs right onto the back of your ZX81. Costs $150.

SINCLAIR DEALERS VIC: B.B.J. Computer Shop P/L; C. & G. Computers P/L; Computerland of Melbourne, Camberwell; Dick Smith Electronics; Logic Shop, Burwood, Prahran; Mindbenders of Croydon; Minit Computer Service; Myer (Melb. stores); Radio Parts Group; Rod Irving Electronics; Strachan Office Suppliers. **ACT:** Dick Smith Electronics; Steve's Communication Centre. **TAS:** Birchalls; The Logic Shop; Quantum Computers; J. Walsh & Sons P/L; **NT:** Ascom Electronics; Eyles Computer Services. **NSW:** Acorn Electronics; A.E.D. Microcomputer Products; B.B.J. Computer Shop P/L; CISA Microcomputers P/L; Computer Factory; Computer Gallerie; Computerland — Chatswood, East Sydney; Computers Galore; Computer Wave P/L; David Reid Electronics P/L; Dick Smith Electronics (all stores); Direct Computer Sales; L. & B. Television Services P/L; The Logic Shop; L.S.T. Electronics; Micla Sales and Service; Minit Computer Service; Paul Derez Radio T.V. Services; Trinity Computing. **WA:** Dick Smith Electronics; Microbase. **SA:** Acuis Aust. P/L; Dick Smith Electronics (all stores); Myer (all stores); Omni Computing. **QLD:** Alliance Computer Products; Computer City; Custom Computer Service; Datacom; Dick Smith Electronics; Myer (all stores); The Logic Shop; Purely Electronics P/L; Scott's Audio-Optical; Software 80; TEC Electronic Imports.

ORDER FORM: SINCLAIR EQUIPMENT (A'SIA) P/L
86-88 Nicholson St., Abbottsford, Vic. 3067. Ph. (03) 419 3033.

QUANTITY	ITEM	ITEM PRICE	TOTAL
	Ready assembled ZX81 Sinclair Personal Computer including mains adaptor, leads, BASIC manual	$199	
	16K-BYTE RAM pack (optional extra)	$150	
	1.2 Amp Adaptor	$17.50	
	ZX Printer	$190	

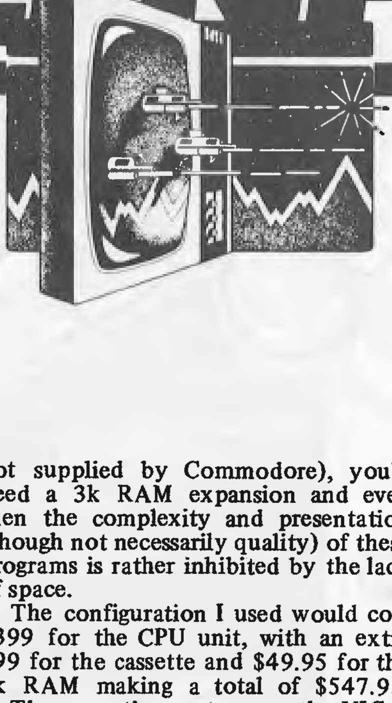
I enclose cheque/Bankcard/Diners Club/Amex
☐☐☐☐☐☐☐☐☐☐☐☐☐ Total ______

Name ______
Address ______ Postcode ______
Signature ______

TIME2

SCREENPLAY

This month we review games for Commodore's VIC-20.

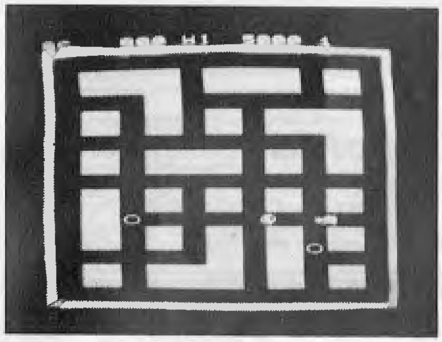


Hardware

The VIC-20 is part of Commodore's marketing strategy of gearing its products to more specific markets and attempting to exorcise the spectre of the PET (CBM is, it seems, upset that this rather endearing brand name has become so integral to its 'corporate image'). It is very much a home machine, offering colour graphics and sound through a domestic TV (see Benchtest in November '81 *APC)*. To keep the price as low as possible, however, the basic machine carries only 3k of user RAM. This means that for most of the cassette-based games (i.e. those not supplied by Commodore), you'll need a 3k RAM expansion and even then the complexity and presentation (though not necessarily quality) of these programs is rather inhibited by the lack of space.

The configuration I used would cost $399 for the CPU unit, with an extra $99 for the cassette and $49.95 for the 3k RAM making a total of $547.95.

The operating system on the VIC is refreshingly friendly, giving clear messages as to what it is doing when loading in from cassette, which makes the machine very easy to use. Overall I have only two complaints regarding the hardware. Firstly, the lead going to the television could be a bit longer, though of course it would be easy to extend it yourself. The second criticism is rather more important and regards the ROM packs on which nearly all the Commodore games are supplied. The edge connector on these is very exposed and quite difficult to guide into the socket on the back of the CPU unit. This makes them highly susceptible to damage, particularly if they are to be handled by impatient children.

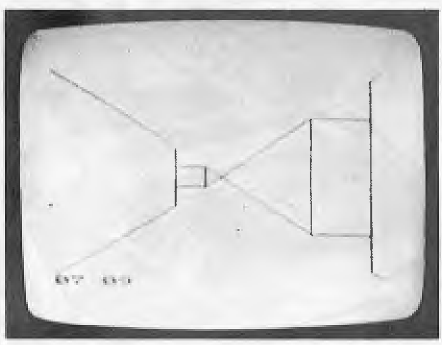


Game: Alien

Another descendant from the arcades, this game takes its general principles from a machine called 'Space Panic' which I have always felt was rather understated. You are being hunted by aliens in a maze and the only way to destroy them is to dig a hole, wait until one falls in and then bury him. Space Panic showed a vertical section with several floors and the creatures would actually drop through the holes but in Aliens you are given an aerial view and thus there is only one level. The enemy are quite fast but very dumb so they won't necessarily find you – but if your time runs out all hell breaks loose with hordes of super-fast aliens appearing from nowhere.

Movement can be controlled by either the keyboard or the joystick. Digging and filling, however, are initiated only through the keyboard (the 'A' and 'D' keys) so if you use the joystick you will find you need three hands to operate the game properly (can this mean that Commodore plans to bring out an add-on limb interface?) I overcame the problem by grasping the joystick between my knees and I'm sure there are other possibilities! Despite a good response time and reasonable graphics Alien is not nearly as good as its predecessor but it still has a certain unique quality that makes it attractive.

Value for money: ***
Use of graphics: ****
Addictive quality: ****
Response speed: *****

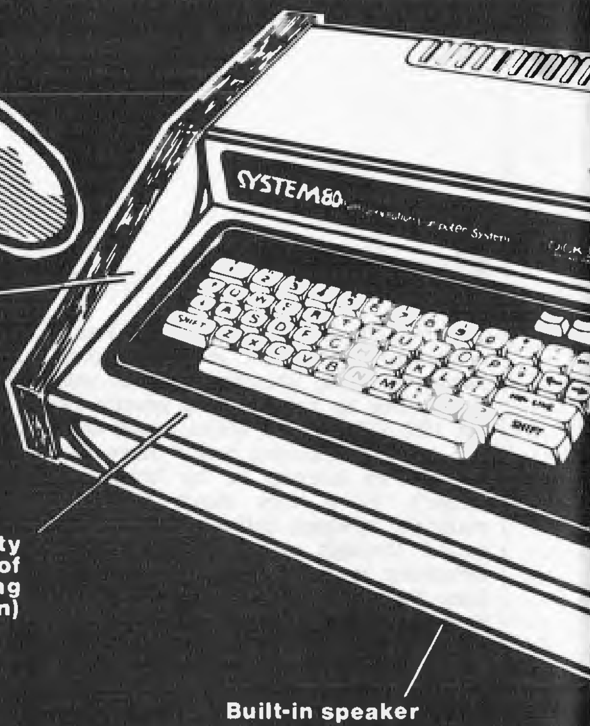


Game: 3D Maze

The name of this game is perhaps slightly misleading. One might be led to think that your task was to find your way around a three-dimensional maze but in fact – from a playing point of view – the maze in question quite definitely has a mere two dimensions. The 3D refers to the line perspective representations of the walls and entrances displayed while you are inside it. You can look at an overall map at any time, so the trick is to find a position compatible with the short range view you are given of the corridor. Movement and mapping are controlled by either the joystick or the keyboard. This must have been an interesting graphics project but as a mind teaser it has severe limitations. I'm afraid the distributor's proud proclamation that 3D Maze provides 'hours of endless frustration' is highly exaggerated (though I'm not sure how inviting that sounds anyway!)

Value for money: **
Presentation: **
Use of graphics: *****
Addictive quality: **

Australia's most popular computer now even better!



Full upper and lower case video display capability

Screen print facility (obtain a print-out of any 'page' appearing on the screen)

Built-in speaker

OVER 6,000 SOLD!

Cat. X-4005

WAS $750

All new Expansion Unit — you save!

A brand new model — now with extra value! The first row of 16K RAM is fitted as standard in this unit, with provision for an extra 16K (which gives the System 80/Expansion Unit 48K total memory). And it's $100 cheaper than the previous model! It provides as standard, a floppy disk controller capable of controlling up to 4 single density disk drives. Although the exterior of the unit is similar to the previous model, there are significant changes internally; the Centronics printer port is still standard, but the RS 232C interface and connector are now only available as an option. The S-100 motherboard has been removed, and is available as a second option for those who require S-100 expansion capability.

$100 less than previous model!

Cat. X-4020

$399

Colour Graphics

Cat. X-3275

Now you can have colour with your System 80 computer! This kit gives you 8 different colours AND it has a PAL encoded modulated RF output which allows it to be used with any standard colour TV receiver.

Only experienced hobbyists or technicians should attempt this kit

$129.00

NEW LOW PRICES ON PRINTERS!

UNDER $500 PRINTER!

Superb dot-matrix printer that uses ordinary fan-fold sprocketed paper up to 204mm wide. Its most unusual feature is the single hammer print-head — this means very few moving parts for increased reliability. Fitted with a standard Centronics type parallel data interface making it compatible with most modern computers. Cat. X-3252

WAS $495!

$399

ITOH 8300P DOT MATRIX PRINTER

The ITOH 8300P features high speed, bi-directional printing (125 characters per second), with full upper and lower case character set. It accepts standard fan-fold sprocketed paper up to 240 mm wide. This means you can do 80, 40 or 132 column printing. Fitted with standard Centronics type parallel port. A great seller with great features. Cat X3255

WAS $920!

$799

NEW DAISY WHEEL WORD PROCESSOR PRINTER

This new Dick Smith daisy Wheel printer delivers ultra sharp copy which is a definite prerequisite for word processing. It will accept standard office stationery or continuous stationery up to 400mm wide. The cartridge ribbon and economical Diablo print wheels are freely available making this a very versatile printer with an unbelievable speed of 40 ch/second. Cat X-3265

$1995

DICK SMITH SYSTEM 80

NOW ONLY $690

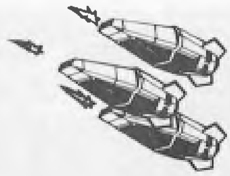
and look at this great new software!

KILLER BEETLES

Pit your skills against the killer beetles. You dig traps and when the beetle falls in you bury him. Problem, they don't stay buried! Cassette based, req. 16K.
Cat. X-3598 **$19.95**

GALAXY INVASION

Real time graphics and sound, the idea is to destroy the invaders and save Earth, but beware of the Flagships!
Cat. X-3693 **$19.50**



GHOST HUNTER

The aim is to eat the energy dots in the maze before the ghost can get you. Random power pills' will assist you in chasing them.
Cassette based, req. 16K.
Cat. X-3597 **$19.95**

SUPERMAZE

The maze game to end all maze games. It can generate mazes up to 100 x 100 elements — it can take you many hours to find the way out!
Cassette based, req. 16K.
Cat. X-3672 **$17.95**

DEPTH CHARGE

The latest version of the arcade favourite. Your mission is to destroy the enemy submarine pack. Good graphics and includes sound.
Cassette based, req. 16K.
Cat. X-3686 **$14.95**

RAIN ANALYZER

A most useful program for the man on the land, the geography student, etc. By keying rain pattems for your area in the past (these figures normally available at your local post office, etc.) you can predict rain fall patterns, volumes, etc.
Cat. X-3767 **$19.95**

TREK 78

The classic computer game based on the ever popular TV series 'Star Trek'. This game utilises good graphics and provides continuous 'status' reports. Your aim is to destroy the enemy and save the Galaxy.
Cassette based.
Cat. X-3644 **$12.95**

New Exciting Games ...

PENETRATOR

You are the sole survivor of a fighter squadron, whose mission is to invade enemy defence forces and destroy their illegal neutron bomb caches. But the bomb stores have individual defence circles, each with its own special dangers. Once you get through and destroy the bomb deposits, you have to return through the defence rings. Cat X-3673 .... **$34.50**

STOCKMARKET

For one to four players, accumulate wealth by buying and selling shares. Stay ahead of the market index by as big a margin as possible, and watch out for the other players too! The market index starts at 100 and is updated each time a share value changes. Financial fun for the whole family. Cat X-3768 .... **$19.50**

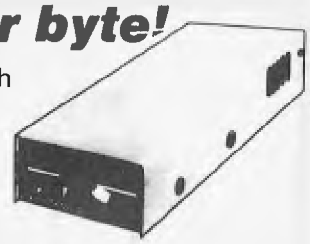
WIN A DATSUN STANZA

DATSUN

When you purchase a VIC-20, System 80 or Sorcerer computer from Dick Smith Electronics, you become eligible to win this superb
DATSUN STANZA
GL four door sedan
So hurry in for your fabulous System 80
NOW!
Not valid in states where this contravenes State Laws.

Permit No. TC82/278 Issued under the Lotteries and Art Unions Act

Still cheapest per byte!



These superb disk drives cost so much less than the Tandy disk drives. Each drive has a high storage capacity 40 tracks and is suitable for both single and double density recording. With double density recording (using X-3540 Double Density Adaptor), you can store more than 180K bytes per diskette! The disk drives are suitable for use with most computers and disk controllers using the standard 34 way connection bus — such as the System 80, TRS 80 etc. (Don't pay their prices for disk storage.)
Cat X-4060 Drive 0
Cat X-4061 Drive 1

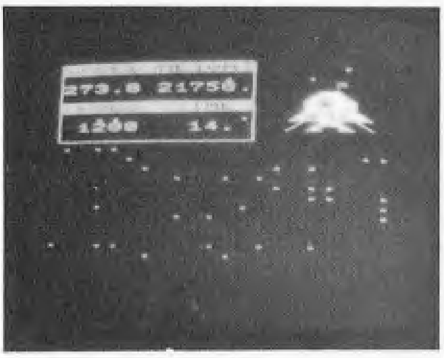
ONLY $419

Stock available and prices correct at press time

DICK SMITH ELECTRONICS

NSW: 145 Parramatta Rd AUBURN 648 0558 ● T55 Terrace Level BANKSTOWN SQ 707 4888 ● 613 Princes Hwy BLAKEHURST 546 7744 ● 552 Oxford St BONDI JUNCTION 387 1444 ● 818 George St BROADWAY 211 3777 ● 531 Pittwater Rd BROOKVALE 93 0441 ● 147 Hume Hwy CHULLORA 642 8922 ● 162 Pacific Hwy GORE HILL 439 5311 ● 396 Lane Cove Rd NORTH RYDE 888 3200 ● 30 Grose St PARRAMATTA 683 1133 ● 125 York St SYDNEY 290 3377 ● Cnr Tamworth Acde & Kable Ave TAMWORTH 66 1961 ● 173 Maitland Rd TIGHES HILL (Newcastle) 61 1896 ● 263 Keira St WOLLONGONG 28 3800 **VIC:** 399 Lonsdale St MELBOURNE 67 9834 ● 260 Sydney Rd COBURG 383 4455 ● 656 Bridge Rd RICHMOND 428 1614 ● Cnr Springvale & Dandenong Rds SPRINGVALE 547 0522 ● 205 Melbourne Rd GEELONG 78 6363 ● Cnr Ross Smith Ave & Nepean Hwy FRANKSTON 783 9144 **ACT:** 96 Gladstone Rd FYSHWICK 80 4944 **QLD:** 166 Logan Rd BURANDA 391 6233 ● 842 Gympie Rd CHERMSIDE 59 6255 **SA:** 60 Wright St ADELAIDE 212 1962 ● 435 Main North Rd ENFIELD 260 6088 ● Cnr Main South & Flagstaff Rds DARLINGTON 298 8977 **WA:** 414 William St PERTH 328 6944 ● Cnr Wharf St & Albany Hwy CANNINGTON 451 8666 **TAS:** 25 Barrack St HOBART 31 0800

DSE/A230M/PAI



Game: Lunar Lander

This game is written in Basic but gives a good imitation of a real time game. The object is to use as little fuel as possible to guide your 'Lunar Lander' onto the surface of the planet at a sufficiently slow speed to avoid crashing.

The craft which you're guiding unfortunately doesn't descend during the course of the game, so you have to rely completely on the digital readout (in the top left hand corner of the display) for information on fuel, time elapsed, velocity and height. The graphics are not good but the basic idea (which is not original) is intriguing and quite addictive.

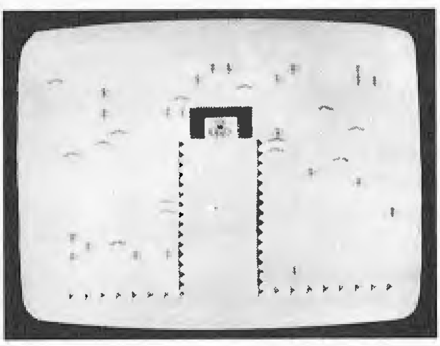
Fuel consumption is dictated by the thrust which is controlled by pressing a key from 0 – 9. A thrust of 5 will balance gravity (as the instructions put it) and maintain a steady gravity. It is not easy to master (I crashed the first seven times I played), despite its simplicity, but could do with more use of the VIC-20 graphics.

Value for money: ***
Use of graphics: *
Addictive quality: ****
Response speed: ****

Game: City Bomber (cassette)

There are a number of versions of this game around for various machines. The basic principles are extremely simple. The program builds a city (looking rather like New York) with 18 skyscrapers of various sizes (it changes in each game) and you use the space bar to bomb it from an airplane travelling across the top of the screen. The latter is gradually getting lower so you must destroy the buildings before you crash into the side of one. You can only have one bomb at a time on the screen and they drop quite slowly so you must be extremely selective about your targets. This game is much more difficult than one might think and demands quite a high degree of skill. The presentation, graphics and sound effects are all impeccable and the give-away price makes City Bomber a must for all VIC owners.

Value for money: *******
Use of graphics: *****
Addictive quality: *****



Game: Ski Run

One way to improve on the memory limitations posed by the VIC is to load the program in two parts, and Ski Run is one of the many games which use this technique. The first part displays the playing instructions and sets up the graphics, while the second actually runs the game, of which there are three variations. It basically involves guiding a cleverly depicted skier down a mountain, avoiding snowdrifts, trees and flags. In Slalom and Giant Slalom, the idea is to ski between evenly placed pairs of marker flags (in Giant Slalom the flags are more widely spaced) while downhill involves following a narrow fenced track. A skill level of one to nine determines the quantity and complexity of the hazards. The skier moves quite fast at a steady speed and you move it left or right using the 'A' and 'D' keys (there is one joystick option). If you manage to complete the course, you are given a time for the run.

The graphics, though hardly complex are remarkably effective – especially the sideways movement of the skier – as is the steady whooshing sound. Although Ski Run may not be particularly sophisticated, it is well designed and executed and is certainly worth looking at.

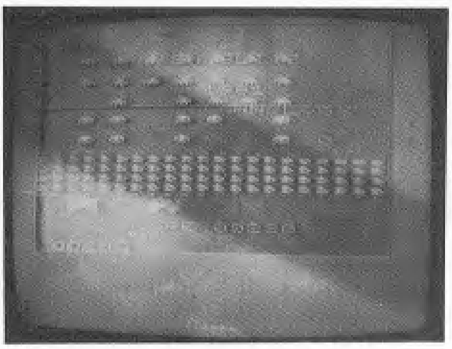
Value for money: ***
Use of graphics: *****
Addictive quality: ****
Response speed: ****

Game: Adventure Island, Pirate Cove, Mission Impossible, The Count, Voodoo Castle

This set of Scott Adams Adventure games is due out soon on Commodore's own ROM packs. The ones I was sent had apparently only just arrived and were each held on a pair of EPROMs (TI264s in fact) poking from the standard packaging. I can't claim to have fully completed any of them but I did have a good look at them all. Their format is completely standard with no concessions to the VIC's graphics capabilities except coloured text! All of them are imaginative and complex and should certainly provide hours of entertainment even for the most cunning adventurer.

Value for money: *****
Presentation: *****
Complexity: ******

SCREENPLAY



Game: Alien Blitz

Using the same old "invaders v's us" theme, Alien Blitz is a reasonably good implementation of Space Invaders. It takes a few minutes to locate the correct buttons for left and right horizontal control of the player's missile launcher and firing button as no documentation is supplied with the game. Ten levels of play are provided from 0 to 9 increasing the speed of the descending invaders at each successive level but (a little unfairly) not that of the player's missiles. I think it would be practically impossible to knock out all the invaders without any advanced weaponry such as an exocet missile or rapid fire machine gun.

There is no special explosion when either home base or invaders are hit, but appart from this grumble and the lack of documentation this would definitely be one of the best games for the VIC-20.

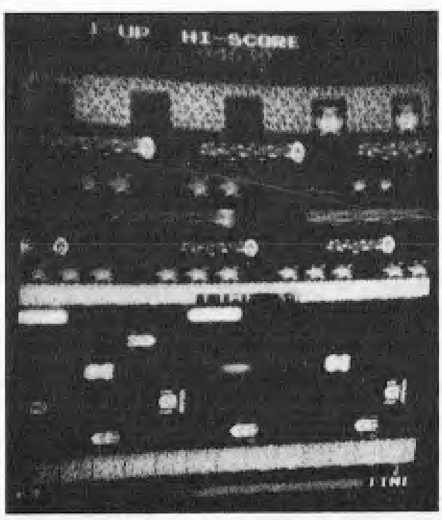
Value for money: *****
Use of graphics: ***
Addictive quality: ****
Response speed: ****

Conclusion

Except for the memory restrictions I mentioned at the beginning, the Commodore VIC-20 has all the makings of a superb games-orientated home computer and this has been reflected in its popularity. The general quality of the games is quite high and I was particularly impressed by the way some of the cassette software overcomes the lack of space. Commodore seems to be developing a very good selection of ROM packs, which is greatly enhanced by the introduction of the Adventure games. The graphics could be better but used properly they yield some good results. The price of this machine makes the VIC not only a strong contender in the home market but also a viable alternative to a non-programmable games console.

VIC-20 supplied by Computer Cellar, Newcastle, in conjunction with Computer Imports, Adelaide.

ARCADE ACE



Having so far concentrated on well-established classics the game I have chosen this month is rather more recent, although it already has a good following. It is called Frogger and is definitely in the Pac Man rather than the Space Invaders genre. You control a small green frog using a joystick to move and a button to jump. It has to cross a road followed by a river in order to reach its home on the other side of the screen. The road is crowded with vehicles of various types which speed up as you work through the frames and must be negotiated using the joystick. Since the frog is unable to swim the river due to a stream of logs (and the odd crocodile), he must jump from log to log, taking care not to be swept away. As you progress other creatures appear presenting further hazards.

I can't claim to be an expert at this, though the small boy who was playing when we took the photograph most certainly is. The graphics on Frogger are extremely good, however, with all sorts of nice touches which make it an interesting spectator sport. Timing is the essence of this game, which requires quick and careful planning if you're to avoid getting squashed. It may not be as hectic as a space battle but it demands just as much concentration. The concept behind Frogger is pretty sick in a harmless sort of way but the game itself is amusing and challenging.

END

CP/M SYSTEM CALLS

by Jeff Richards

Any implementation of Basic is likely to be lacking in certain features. Implementing these features in assembly language can be time consuming and complex, and will probably make the resulting program unique to a particular machine. If the operating system supports the function it is preferable to implement it by a call to the operating system. This article describes how to implement CP/M system calls from Microsoft Basic, and includes a number of examples to show how powerful this technique can be.

An operating system call is a procedure by which a user program can jump into a routine that exists within the operating system in order to execute a particular task. If the operating system supports a particular function then there will be a defined procedure by which the calling program can make the jump to the correct function, after setting up variables if needed. The operating system will execute the function and return to the calling program in a defined manner, again perhaps setting up certain variables.

The procedure for system calls in CP/M is to execute a subroutine jump to location 5 (CALL 0005). Variables are passed in the CPU registers and the system jumps back to the calling program with a return from subroutine instruction (RET).

To understand how to make operating system calls it is necessary to understand how variables are passed between the calling program and the system. The system calls fall into categories according to what data they handle. Some involve no data at all; some expect to receive data from the calling program; some return data to the calling program; and some both accept data and return it. All data is passed or returned in the CPU registers (notwithstanding the fact that, during the system call, CP/M may have read or updated a memory buffer).

The registers used are consistent. Data to be passed to the operating system is passed in register E if it is one byte in length, or in register pair DE if it is a double-byte item. Data is returned in register A if it is single-byte, and in register pair HL if it is double-byte.

The method of telling the operating system which particular function it is to perform is to pass a code number in register C. To actually perform the function the calling program loads the appropriate registers and jumps to memory location 0005. Here there is a jump to a special section of CP/M which decodes the command code in register C and performs the required task – perhaps altering the values in the registers, or accessing data in memory.

. . . assembly language can be time consuming and complex. . . . it is preferable to implement it by a call to the operating system.

For instance, the code for the READ CONSOLE function is 1. By loading a 1 into register C and executing a CALL 0005, the program will find that the next character struck at the keyboard is loaded into register A. While accepting the character CP/M gave the user the benefit of the edit keys (rubout, tab, etc) and, if appropriate, echoed the character on the screen. It also trapped a CONTROL/P or CONTROL/S because these have special meaning to CP/M.

Similarly, the code for a CONSOLE OUTPUT function is 2. By loading a 2 into register C and a character to be displayed into register E, then executing a CALL 0005 the calling program will find that character is displayed on the console device.

A third example involves passing a parameter that is a pointer to data, rather than passing the data itself. CP/M supports a function called PRINT STRING. To use this feature the calling program builds in memory a string of characters it wishes to display, finishing with a "$". It loads the address of the first byte of this data into register pair DE, it loads 9 (the PRINT STRING code number) into register C, and does a CALL 0005. The string, up to but not including the "$", is printed on the console, with control characters correctly interpreted. (If you want to print a "$" then bad luck).

It is obvious that these features of the operating system save a lot of work for the programmer. But, more importantly, they are the device whereby programs can be written that will run on any system that runs CP/M. This is made possible by the fact that when CP/M was installed it was customised to the peculiarities of the hardware. If the keyboard was at I/O port 8 then CP/M was told, and it would be correctly addressed. If the display is a memory mapped device that is driven by a jump to address FF00 then CP/M can be told, and console output will always be handled in the same way. So anyone who writes a program that has to run on a wide variety of machines, e.g. a Basic interpreter, can write it using operating system calls and he knows that it will run on any machine that the operating system is installed on.

How can these functions be accessed from Basic? Executing a CALL to location 0005 is not difficult, but Basic has no procedure for loading data into CPU registers or for getting it back. Microsoft Basic does support a CALL with parameters, and this can be used, with a little bit of juggling, to invoke operating system functions.

Since all CP/M functions require only two parameters we only have to be concerned with one version of the Basic CALL. This is the form CALL R(A,B). This instruction will execute a subroutine call to location R with the address of the variable A in register pair HL and the address of variable B in register pair DE. If A contains the function number and B contains the data item, then we can work out what we have to do before doing the jump to CP/M at location 0005. Firstly, we have to get the value of variable A into register C. Secondly we have to get the value of variable B into register pair DE, or perhaps register E. Remember that the values that Basic loaded into register pairs HL and DE were the addresses of the variables, not the values.

In order to extract the values of the variables we have to know how they are stored, and this in turn depends on what variable type they are. As CP/M requires a value greater than 64k all variables used in operating system calls will be integers. Basic stores integers as two contiguous bytes, low byte first. Although some system calls use register E and some use register pair DE there is no harm in loading the full 16 bits for each call. The routine that does the necessary conversion for a system call that passes one (or none) parameters and expects no reply is given in listing 1. A program that uses this routine to send a character to the console is given in listing 2. This is a pretty boring program, but becomes more interesting when the range of possible data calls is considered.

For instance, CP/M supports an output device called PUNCH. This may actually be a paper tape punch, but could be any output device at all. In some systems it is a second printer, or perhaps a modem transmit line. Without the facilities of the operating system the Basic program would have to use direct I/O (IN and OUT) and it would be very difficult to program. Using the above method a system call with code 4 in register A will send the character in register E to the "punch" device, whatever it may be.

A similar routine can be provided for functions that return a value to the calling program. An example of this is CONSOLE INPUT which functions in a similar way to CONSOLE OUTPUT. Again, console input is not very interesting, but "READER" input can be valuable. Like the PUNCH, the READER is supported by CP/M, and could be anything from a light pen to a modem input line.

Extending listing 1 to include procedures to put the returned variables back into Basic's data area gives a routine that can be used for system calls that both accept and return data. An example of this is DIRECT CONSOLE I/O (code 6). This function accepts data from the keyboard without echo and with no editing or trapping, or displays data on the console with no editing. It is the function used for the INPUT$(n) functions, but if your Basic does not support this function it can be implemented with system call 6. The assembly language routine to handle two-way communication is in listing 3.

Note that the small interface routines were needed because Basic does not automatically put the data into the registers that CP/M requires. But they were also needed because Basic passes the address of the variables, not their values. It is important to remember the difference between passing an address and a value, especially when CP/M expects an address. In this case the VARPTR function in Basic is used to get the address of a data item, which is then passed to the interface routine as a variable. The interface routine could have been modified to accept an address directly, but it is probably better to stick with one interface routine, and force the Basic program to conform to it. An example that passes an address instead of a value is given in listing 4.

What are the function calls that might be interesting to the Basic programmer? READER, PUNCH and DIRECT CONSOLE I/O have already been mentioned. LIST may be useful if you find Basic is interfering with control characters that the printer requires. RETURN VERSION NUMBER is an interesting call that could be used to support version dependant programming, for instance suppressing random disk I/O if version is less than 2. GET CONSOLE STATUS is equivalent to the Basic INKEY$ function, and could be used if this function is lacking.

It is in the area of disk functions that the use of system calls comes into its own. One very useful function is SELECT DISK. Careful use of this function means that it is possible to write software that can configure itself to the number of disk drives in a system. Software could be set up so that programs could be always referenced with a "A:" while data files would have no reference. By selecting the default drive to be disk 0 in a single drive system and disk 1 in a dual drive system maximum effective use would be made of the available disk space. It can be used in conjunction with RETURN CURRENT DISK to work out which drive the system was initiated from. Similarly, RETURN LOGIN VECTOR can provide information about which drives are currently active and available.

There is a call tc WRITE PROTECT DISK which could be used by a system master disk to prevent the possibility of damaging programs or data. And finally there are two calls that return the address of disk parameter areas – GET ADDR (ALLOC) and GET ADDR (PARMS). Listing 5 is a sample program that uses GET ADDR (PARMS) to display the information that the STAT DSK: command returns.

Naturally, any procedures that make use of system calls must be very carefully evaluated for unforseen effects. But using system calls in the manner described above is an easy and effective way to get around some of the limitations of Basic with a minimum of assembly language programming.

TABLE 1 – USEFUL CP/M SYSTEM CALLS

NAME	CODE	DATA IN 8/16	DATA OUT 8/16
Reader Input	3	–	Character 8
Punch Output	4	Character 8	–
Direct Console I/O	6	OFFh 8	Character 8
" " "	6	Character 8	–
Get Console Status	10	–	Status 8
Return Version Number	11	–	Version 8
Return Current Disk	25	–	Cur Dsk 8
Get Addr (Alloc)	27	–	Alloc Addr 16
Write Protect Disk	28	–	–
Get R/O Vector	29	–	R/O Vector 16
Get Addr (Parms)	31	–	Parm Addr 16

Note:
1. Code is always passed in Register C
2. 8 bit Data In is passed in Register E
3. 16 bit Data In is passed in Register Pair DE
4. 8 bit Data Out is returned in Register A
5. 16 bit Data Out is returned in Register Pair HL

LISTING 1 – SIMPLE OUTPUT INTERFACING ROUTINE

```
4E        MOV  C,M  ;REG C = VBL A
EB        EXCH      ;HL = ADDR V8L B
5E        MOV  E,M  ;REG E = VBL B LO
23        INX  H    ;BUMP ADDR
56        MOV  D,M  ;REG E = VBL B HI
C3 05 00  JMP  5    ;GOTO CP/M
```

LISTING 2 – EXAMPLE USING FUNCTION 2

```
10   DEFINT A-Z
20   GOSUB 1000
99   REM CONSOLE OUTPUT
100  A=2
110  INPUT A$
120  IF A$="N" THEN STOP
130  B=ASC(A$)+1
140  CALL R(A,B)
150  GOTO 100
1000 DATA &H4E,&HEB,&H5E,&H23
1001 DATA &H56,&HC3,&H05,&H00
1010 REM R= ANY AVAILABLE SPOT
1020 R=&HFA00
1030 FOR I = 0 TO 7
1040 READ A
1050 POKE R+I,A
1060 NEXT I
1070 RETURN
```

LISTING 3 – ROUTINE TO RECEIVE AND EXPORT VARIABLES

```
4E        MOV  C,M  ;REG C = VBL A
E5        PUSH H    ;SAVE ADDR VBL A
EB        EXCH      ;HL = ADDR VBL B
5E        MOV  E,M  ;REG E = VBL B LO
E5        PUSH H    ;SAVE ADDR VBL B
23        INX  H    ;BUMP ADDR
56        MOV  D,M  ;REG D = VBL B HI
CD 05 00  CALL 5    ;GOSUB CP/M
EB        EXCH      ;REG DE = REG HL
E1        POP  H    ;HL = ADDR VBL B
73        MOV  M,E  ;VBL B LO = REG E
23        INX  H    ;BUMP ADDR
72        MOV  M,D  ;VBL B HI = REG D
E1        POP  H    ;HL = ADDR VBL A
77        MOV  M,A  ;VBL A LO = REG A
23        INX  H    ;BUMP ADDR
36 00     MVI  M,0  ;VBL A HI = ZERO
C9        RET       ;RETURN
```

LISTING 4 – USING VARPTR TO PASS AN ADDRESS

```
10   DEFINT A-Z
20   GOSUB 1000
30   CRLF$=STR$(13)+STR$(10)
99   REM PRINT STRING FUNCTION
100  A=9
110  A$="THIS IS THE MESSAGE"
     +CRLF$+"$"
119  REM GET STRING DESCRIPTOR
120  B=VARPTR(A$)
129  REM GET STRING ADDRESS
130  B!=PEEK(B+1)+PEEK(B+2)*256
140  IF B!>32767 THEN B!=B!-65536!
150  B=B!
160  CALL R(A,B)
170  END
1000 DATA &H4E,&HE5,&HEB,&H5E,&HE5
1010 DATA &H23,&H56,&HCD,&H05,&H00
1020 DATA &HEB,&HE1,&H73,&H23,&H72
1030 CATA &HE1,&H77,&H23,&H36,&H00,&HC9
1050 R=&HFA00
1060 FOR I = 0 TO 20
1070 READ A
1080 POKE R+I,A
1090 NEXT I
1100 RETURN
```

LISTING 5 – EMUALATE "STST DSK:" COMMAND

```
10   DEFINT A-Z
20   GOSUB 1000
30   CLS$=CHR$(26)
40   PRINT CLS$
99   REM GET ADDRESSS (PARMS)
100  A=31
110  B=0
120  CALL R(A,B)
130  SPT=PEEK(B)+PEEK(B+1)*256
140  B=B+2
150  BLS=2^(PEEK(B)+7)
160  B=B+2
170  REX=PEEK(B)*128+128
180  B=B+1
190  DSM=PEEK(B)+PEEK(B+1)*256+1
200  B=B+2
210  DRM =PEEK(B)+PEEK(B+1)*256+1
220  B=B+4
230  DRS=DRM*32
:240  DRB=DRS/BLS
250  CKS=PEEK(B)+PEEK(B+1)*256
260  B=B+2
270  OFF=PEEK(B)+PEEK(B+1)*256
500  PRINT "","DRIVE CHARCTERISTICS
510  PRINT BLS*DSM/128,":
     128 Byte Record Capacity
520  PRINT DSM*BLS/1024,":
     Kilobyte Drive Capacity
530  PRINT DRM,":
     32  Byte Directory Entries
540  PRINT CKS*4,":
     Checked  Directory Entries
550  PRINT REX,": Records/ Extent
560  PRINT BLS/128,": Records/ Block
570  PRINT SPT,": Sectors/ Track
580  PRINT OFF,": Reserved Tracks
590  STOP
1000 REM INSERT SUBROUTINE FROM
1010 REM LISTING 4
1020 RETURN
```

POURING SCHOONERS INTO MIDIES

When you plan to buy your computer equipment you normally look very carefullyat the things that cost most money. You tend to think very hard about, for instance, what printer you need and can afford. People usually regard the purchase of disk units as inevitable, with no scope for savings. That is not necessarily so: this article discusses a technique which could save you money.

Have you thought carefully about how data is stored on your disk? For example, when you investigate the way numeric data is stored on the Apple Disk II using the normal facilities of Applesoft and DOS 3.3 you will discover it is rather like a sieve. Each numeric digit is put into a separate byte. This is very wasteful since each byte can hold numbers up to 255 and we can normally only put in values up to 9.

Figure 1 shows how the value 12,345,678 is stored in 9 bytes, using conventional methods. Although the number is held in binary form in memory, it gets expanded to base 10 on the disk. The accompanying program shows how the number can be compacted to three bytes expressed to base 256, as in Figure 2. This technique can make all the difference between needing bigger, or more disks, and being able to manage with what you've already got.

The example in the program is based on an imaginary requirement, in which an organisation needs to store information (sales figures perhaps) for 99 products and 29 areas over a period of 12 months. The program enables us to store numbers for all these combinations (34,452 in all), each with a value up to over 16 million. Using conventional methods this requires over 310,000 bytes. Using this method it requires only about 195,000 bytes, and can therefore be stored on a 5¼in disk II.

You can store numbers (option 1), add to them (option 2) and display them (option 3). Option 0 allows you to set the file up in the first place. Although this takes about 20 minutes it is necessary in order to avoid problems on reading fields which have not previously been addressed. Option 4 is required to close the file down, and to clear whatever happens to be in the buffer. Each option returns you straight to the Menu except option 3, which lets you read the value displayed and waits until you press any key before returning to the Menu.

The method used is by successive division to reduce numbers into three one-byte portions of values up to 255. The three byte fields are joined together in the buffer area and written to disk using the facilities of DOS Random Access storage. They are subsequently read from disk, unpacked and converted back to their original form. This process inevitably requires the use of PEEK and POKE, since Basic by itself does not allow us to address individual bytes.

Record length is given as 256 bytes — that is the whole of the buffer area. Bytes 0 and 1 contain 1 and Return respectively. Bytes 2 to 253 contain 84 fields of 3 bytes packed together with no separating characters between fields. Bytes 0 and 1 contain the values stated because a DOS "WRITE" needs a PRINT, and a DOS "READ" needs an INPUT. The 1 in byte 0 has no data significance, of course, but is there in order to take the 84 packed binary fields on to the disk by POKE and from the buffer by PEEK. Bytes 254 and 255 are wasted because there is no more room in the sector for a 3-byte field. (The loss of the first two and last two bytes in each sector is a minor but unavoidable loss of efficiency.)

A Write Indicator (WR) is used to keep the Buffer READs and WRITEs under control in subroutines 600 and 700. Without this additional check DOS will use other buffer areas some of the time, which would prevent this method from working correctly.

Extensions

The method can be extended in four different ways.

Firstly, although the program illustrates the method with a three-byte field, the number of bytes can be changed by altering the value of 1 in subroutines 400, 600, 800 and 900, and expanding the range of values for Z. The more bytes you pack together into one field, the bigger the saving of space.

Secondly, it is possible to add to the code by recording negative numbers. This can be done by allocating one binary position for negative numbers prior to coding and POKEing them and later PEEKing and de-coding them.

Thirdly, decimals can be coded by moving the decimal point before coding and after de-coding.

Fourthly, alphabetical characters can also be coded — although the saving is not as great. Since the range of alphanumeric characters can be accommodated within six bits (0-63) it will be possible to store four alphabetical characters in three bytes by using an extension of this method. END

Contents	188	97	78
Byte No	0	1	2

188 x 65536 = 12,320,768
97 x 256 = 24,832
78 x 1 = 78
12,345,678

Figure 2

Contents	1	2	3	4	5	6	7	8	R
Byte No	0	1	2	3	4	5	6	7	8

Figure 1

```
5 REM COMPACT
6 REM AUTHOR ERWIN SCHNEIDER
7 REM TRIAD COMPUTING SYSTEMS
8 REM TEL 01-631-7211
15 REM
16 REM THE PURPOSE OF THIS
17 REM PROGRAM IS TO COMPRESS
18 REM NUMERIC DATA ON DISK
19 REM
21 REM THIS IS ACHIEVED BY
22 REM DIVIDING NUMBERS INTO
23 REM SUCCESIVE BYTES ON DISK
24 REM EACH BYTE REPRESENTS AN
25 REM ELEMENT TO BASE 256.
26 REM THIS IS DONE WITH PEEKS
27 REM AND POKES IN THE BUFFER
28 REM AREA.
29 REM
50 Z(1) = 65536
51 Z(2) = 256
52 Z(3) = 1
70 GOTO 1000
100 REM ** INPUT **
110 HOME : VTAB 5: HTAB 18
120 PRINT "INPUT DATA"
140 VTAB 7
150 INPUT "AREA (1- 29)     ";A
160 IF A < 1 OR A > 29 THEN 140
170 VTAB 9
180 INPUT "PRODUCT (1 - 99) ";P
190 IF P < 1 OR P > 99 THEN 170
195 VTAB 11
200 INPUT "MONTH (1 - 12)   ";M
210 IF M < 1 OR M > 12 THEN 195
220 RETURN
250 REM ** GET VALUE **
260 VTAB 15: HTAB 5
270 PRINT "DATA INPUT"
280 INPUT "VALUE (1 TO 16777215)
    ";V
290 IF V < 0 OR V > 16777215 THEN
    268
295 RETURN
300 REM ** FORM KEY **
305 K2 = K1: REM PREV REC NO
310 K = (1188 * (A - 1)) + (12 *
    (P - 1)) + M - 1
320 K1 = INT (K / 84): REM REC
    NO
330 K = 3 * (K - (K1 * 84)): REM
    REMAINDER
340 K1 = K1 + 1
350 RETURN
400 REM ** TO BINARY **
410 FOR I = 1 TO 3
420 V(I) = INT (V / Z(I))
430 V = V - V(I) * Z(I)
440 NEXT I
450 RETURN
500 REM ** BUFFER CONTROL **
510 IF K1 = K2 THEN 540
520 GOSUB 700: REM OUTPUT
530 GOSUB 750: REM INPUT
540 RETURN
600 REM ** NEW ITEM IN **
610 FOR I = 1 TO 3
620 POKE (38401 + K + I),V(I)
630 NEXT I
650 WR = 1: REM WRITE INDIC
660 RETURN
700 REM ** OUTPUT **
707 IF WR = 0 THEN 740
710 PRINT D$;"WRITE COMPACT-B DA
    TA,R"K2
720 PRINT 1
730 WR = 0
740 RETURN
750 REM ** INPUT **
760 PRINT D$;"READ COMPACT-B DAT
    A,R"K1
770 INPUT A
780 RETURN
800 REM ** FROM BINARY **
810 A = 0
820 FOR I = 1 TO 3
830 A = (A * 256) + V(I)
840 NEXT I
850 RETURN
900 REM ** GET ITEM **
910 FOR I = 1 TO 3
920 V(I) = PEEK (38401 + K + I)
930 NEXT I
940 RETURN
1000 REM ***INITIALISATION***
1010 D$ = CHR$ (4)
1020 PRINT D$;"OPEN COMPACT-B DA
    TA,L256"
1100 PRINT D$
1105 HOME : VTAB 5: HTAB 10
1110 PRINT "MAIN MENU"
1120 VTAB 7: HTAB 5
1130 PRINT "0  INITIAL FILE SET-
    UP"
1140 PRINT "      (TO BE USED ONL
    Y ONCE"
1150 VTAB 10: HTAB 5
1160 PRINT "1  REPLACE A FIELD"
1170 VTAB 12: HTAB 5
1180 PRINT "2  ADD TO A FIELD"
1190 VTAB 14: HTAB 5
1200 PRINT "3  DISPLAY A FIELD"
1210 VTAB 16: HTAB 5
1220 PRINT "4  STOP"
1230 VTAB 18: HTAB 6
1250 INPUT "";N
1260 IF N < 0 OR N > 4 THEN 1230

1270 ON N + 1 GOTO 1300,1400,150
    0,1600,1700
1300 REM *** FORMAT SECTORS ***
1310 FOR K2 = 1 TO 411
1320 FOR I = 2 TO 255
1330 POKE (38400 + I),0
1340 NEXT I
1350 GOSUB 710: REM OUTPUT
1360 NEXT K2
1370 GOTO 1100
1400 REM ***** REPLACE *****
1410 GOSUB 100: REM INPUT
1420 GOSUB 250: REM GET VALUE
1430 GOSUB 300: REM FORM KEY
1435 GOSUB 500: REM BUFFER CTRL
1440 GOSUB 400: REM TO BINARY
1450 GOSUB 600: REM NEW ITEM IN
1460 GOTO 1100
1500 REM **** ADD TO FIELD ****
1510 GOSUB 100: REM INPUT
1520 GOSUB 250: REM GET VALUE
1530 GOSUB 300: REM FORM KEY
1540 GOSUB 500: REM BUFFER CTRL
1550 GOSUB 900: REM GET ITEM
1560 GOSUB 800: REM FROM BINARY
1570 V = V + A
1580 GOSUB 400: REM TO BINARY
1590 GOSUB 600: REM NEW ITEM IN
1595 GOTO 1100
1600 REM **** DISPLAY FIELD ****
1610 GOSUB 100: REM INPUT
1620 GOSUB 300: REM FORM KEY
1630 GOSUB 500: REM BUFFER CTRL
1640 GOSUB 900: REM GET ITEM
1650 GOSUB 800: REM FROM BINARY
1660 VTAB 15: HTAB 5: PRINT D$
1670 PRINT "VALUE IS ";A
1680 GET A$: REM TIME TO READ
1690 GOTO 1100
1700 REM ****** END ******
1710 K2 = K1
1720 GOSUB 700: REM OUTPUT
1730 PRINT D$;"CLOSE"
1740 END
```

LISP

Michael Liardet spent several years as a research worker in artificial intelligence and computer aided design and has worked for the last two years as a systems consultant for a well-known microcomputer systems house. Here he presents an overview of one of the most fascinating of computer languages.

In this article I shall present the background and a basic introduction to the programming language Lisp. As a motivation for the readers who doubt that Lisp can hold any advantages over their own favourite language, I shall wind up with an almost trivial Lisp program that would be a fairly daunting project in most other programming languages: the symbolic differentiation of an arbitrarily complex mathematical expression.

Lisp is a programming language widely used in artificial intelligence research but now available to the more general public through various implementations on micros. Lisp is an acronym for LISt Processing and, although not quite the earliest list processing language, it is certainly the earliest one that is still with us. Designed by John McCarthy and implemented by him and his students at Massachusetts Institute of Technology, the language has been around for over 20 years now, which certainly pre-dates the more familiar Basic and Pascal (both from the late '60s).

In these last 20 years, Lisp has been used on a wide variety of fascinating research projects such as computer chess, natural language understanding, automatic theorem-proving and even program correctness verification. In addition, it has spawned a number of other languages, most of them not widely known outside the AI community, including languages like Prolog and Pop-2.

The language has not been greatly used outside academic circles but this is due more to general unawareness of it rather than some intrinsic flaw. Certainly the language could be extremely useful in, for example, the implementation of a database system where complex relationships and interrogations need to be handled. It could be a highly relevant language in other applications, too, but it does not look as though the great mass of Fortran/Cobol/Basic programmers are going to be reaching for their Lisp manuals just yet!

Lisp completely outclasses the better-known languages when symbol processing and complex data structures are required. It's ideal when ideas need to be quickly coded up, tested and re-evaluated, etc. (Once the ideas have been shown to work, the solution can, if necessary, be re-coded in some other language which can execute them more efficiently — most of the super-fast machine-code chess programs are based on principles first explored by Lisp programmers.) Most Lisp systems, mainframe and micro alike, are fully interactive and generally interpretive rather than compiled, just like Basic in that respect at least! An interesting historical note: a lot of the pioneering work in interactive multi-user mainframes was based around Lisp systems.

Following its strong emphasis on symbol processing, Lisp tends to pay lip-service (Lisp service?) only to number-crunching facilities. So if you want to have some fun with matrix inversion (does anybody?) then you had better stick to Fortran and friends. If, however, you want to experiment and possibly break new ground with some AI project, or even if you are just interested in learning some new programming techniques and concepts, then there's a whole new world awaiting you!

Before embarking on a quick run-through of the Lisp features, a word of warning to the experienced programmer: Lisp is likely to have next to nothing in common with any previous language of your experience. Thus learning Lisp is almost like learning to program all over again! This is not going to be particularly discouraging to someone who cannot yet program anyway but does place a higher than expected learning curve in front of someone who is expecting it to be as easy as, say, a transfer from Fortran to Basic.

So much for the background; but what is the language like to use? The first sight of a Lisp program is, to put it mildly, likely to be slightly discouraging. With its vast numbers of parentheses and curious terminology, the language looks almost as unreadable as APL. (Incidentally, it has at least one other feature in common with APL — the fanaticism of its converts!) Anyway, bear in mind that beauty is only skin-deep and try not to be put off by superficialities. The language has great beauty and elegance underneath all those parentheses! For example: just about everything, from a simple arithmetic expression through to conditional expressions or even complex function definitions, is based on the same underlying structure: the list! As the data is also generally represented in lists, some very useful possibilities start to emerge — for example a program to modify a program can be written as simply as a program to modify a data structure. (If you are wondering why you should need to do that then ask yourself what a program (as opposed to *APC's*) editor is? (By the way the Lisp lesson has started (basic training in the use of '('s (and ')'s too!)))).)

To get some sort of feel for the language let's see how you might translate a relatively simple Basic statement into Lisp. We will break down the translation into simple stages. Of course the intermediate stages are for explanatory purposes only and don't resemble any programming language yet devised! Let's try:

```
IF A+B=2 OR C*D< 5 THEN E=C—D
```

If we now introduce brackets everywhere, to make the order evaluation absolutely explicit (in Basic, order of evaluation in expressions is implicitly determined by operator precedences):

```
(IF (((A+B)=2)OR((C*D)<5))
THEN (E=(C—D)))
```

Next we re-name most of the functions with rather more lengthy names. Note that IF... THEN translates as COND, '<' as LESSP and the assignment '=' as SETQ. We won't bother if we now have to write the expression on several lines, as Lisp always ignores the way a program is laid out, and just relies on the parentheses to determine when expressions begin and end:

```
((((A PLUS B) EQ 2) OR ((C TIMES D)
LESSP 5)) COND
(E SETQ (C MINUS D)))
```

Now there is only one step left to reach Lisp. All the functions, currently lying between their arguments, are moved in front of their arguments and one extra pair of brackets (just after COND) is introduced.

Lisp uses 'prefix' notation; to add 2 and 2 we say PLUS 2 2. So the actual Lisp translation looks like this:

```
(COND ((OR (EQ (PLUS A B) 2)
(LESSP (TIMES C D) 5))
(SETQ E (MINUS C D)))
```

The extra brackets (mentioned above) must be introduced to allow for the fact that the Lisp COND is more general than Basic's IF... THEN. Most Lisps support:

```
(COND (test1 do this)
      (test 2 do this)
      etc)
```

This is the IF... THEN... ELSEIF structure and thus (COND test1 do this) would be a correct translation.

Having seen what a fragment of Lisp code looks like, now let's look at a fragment of Lisp data. Let's suppose we need to represent details of various people's names, ages and sex. (Imagine we have written a program to organise the seating at a dinner party or some similar style of problem uncomplicated by disk accesses or validations, etc.) Under these circumstances, in Basic we would probably choose to set up the details in DATA statements and then initialise pre-dimensioned arrays with these details. The Lisp way of doing this would be:

```
(SETQ PEOPLE '((BASIL 32 M)
(CYNTHIA 30 F) (JOHN 28 M)))
```

The variable PEOPLE is assigned (SETQd) a list of length three (it could easily be much longer), each element in the list itself being a list of length three. A list is defined by enclosing its elements ('atoms') in parentheses. Notice the small quote '" just before the start of the list. This single character is really all that is needed to distinguish between program and data. It means that the list following is to be treated literally and not 'evaluated' — to use the Lisp jargon. In contrast, a list like (PLUS 2 NUM) would not generally be quoted as normally we would be expecting Lisp to evaluate it to the number two greater than NUM. Notice also that there is no closing quote — the parentheses determine the extent of the quotation.

Now, suppose we need to access details of the first of the PEOPLE. What is required is the front or head of the list held by the variable PEOPLE. The Lisp function CAR returns the head of a list and so (CAR PEOPLE) will return (BASIL 32 M). The sister function to CAR is CDR which will return the rest of a list without its head. These two functions used in combination allow any part or sub-part of a list to be accessed — eg, CAR (CAR (CDR PEOPLE))) returns CYNTHIA. The terminology CAR and CDR by the way, is a hangover from the original terminology where these functions were encoded using Contents Address Register and Contents Decrement Register machine instructions. The probable reason for them still being in use today is that it is absolutely trivial to set up alternate names — just SETQ CAR or CDR to name of your choice and use that instead!

Before moving on from the data structuring, there is only one other major Lisp list function to cover, CONS, which constructs lists, Thus if we want to add another person, say (JEAN 21 F, to PEOPLE then we use CONS to construct a new list whose CAR (ie, head) is (JEAN 21 F) and whose CDR (ie, tail) is the old list PEOPLE. The new list is simply SETQd (ie, assigned) to PEOPLE which will henceforth have a list of length four rather than three. In case you can't work this out for yourself the code to do this is:

(SETQ PEOPLE (CONS '
(JEAN 21 F) PEOPLE))

At this point we have covered enough ground to start on a simple Lisp function — a function to determine the length of a list. A Lisp function, by the way, is almost directly analogous to a Basic function, which is usually restricted to being an expression. In Basic this amounts to a very severe limitation, since conditionals, loops and so on are not counted as being expressions, but in Lisp things like COND, as we have already seen are not greatly distinguished from things like PLUS and so function definitions are not really limited in Lisp at all.

It is possible to determine the length of a list by setting a counter to zero and then repeatedly CDRing the list and incrementing the counter until there is no list left to CDR (Lisp calls such a list NIL — equivalent to a pair of parentheses with nothing between them). This sort of approach is an iterative solution but we shall consider an alternative recursive solution: to find the length of a list, find the length of its tail and add one. If it doesn't have a tail. (ie, it's NIL) then its length is zero. The essence of a recursive solution is that it doesn't 'try very hard' to solve the problem but just simplifies it enough to call upon itself again to solve the simpler problem and so on until the solution is transparently obvious (when the list is NIL the answer is 0). Without further ado, here is the code for LENGTH (DE is the function defining Function, called DEFUN in some dialects):

```
(DE LENGTH (LIST)
   (COND ((EQ LIST NIL) 0))
      (T (PLUS 1 (LENGTH
(CDR LIST))))))
```

You may wonder how a Lisp function returns a value. Well, in this particular instance, LENGTH will return either 0 (if (EQ LIST NIL)) or (PLUS 1 (LENGTH (CDR LIST))) (if it isn't, T means 'true' and will always satisfy a conditional).

Basic programmers who may not be familiar with recursion should try and master it. Most interesting applications of Lisp rely heavily on recursion and an iterative technique very rarely works as well. For example, many games programs use a variant of the recursive algorithm — to evaluate the current position generate, in turn, every possible move to modify this position and evaluate the new position (recursively). The value of the position is the value of the best possible continuation.

```
DIFF (A+B) = DIFF (A) + DIFF (B)
DIFF (A-B) = DIFF (A) - DIFF (B)
DIFF (A*B) = A*DIFF (B) + B*DIFF (A)
DIFF (A/B) = (B*DIFF (A) - A*DIFF (B))/B²
DIFF (A^n) = n*(A^(n-1))*DIFF (A)
DIFF (n) = 0
DIFF (X) = 1
where A and B are expressions in X and
n is a constant
```

Fig 1

Well at this point we have really come far enough to demonstrate the promised program to differentiate symbolically an arbitrarily complex mathematical expression. First a quick refresher course for those whose O-level maths has got a bit rusty (using DIFF to mean 'differential with respect to x') — see Figure 1.

The Lisp implementation of differentiation is almost solely a syntactic translation of the above, substituting Lisp symbols for mathematical ones and using the standard Lisp 'grammar' of prefix function operators, rather than the infix — see Figure 2.

```
(DE DIFF (X)
(COND
 ((EQ 'X X) 1)
 ((ATOM X) 0)
 ((EQ (CAR X) 'PLUS)
  (CONS 'PLUS (CONS (DIFF (CADR X)) (CONS (DIFF (CADDR X)) NIL))))
 ((EQ (CAR X) 'MINUS)
  (CONS 'MINUS (CONS (DIFF (CADR X)) (CONS (DIFF (CADDR X)) NIL))))
....
(T 'UNKNOWN)))
```

Fig 2

Not all of the definition of DIFF has been included here. The rules for *, / and powers should be inserted as '. . .'. Briefly the definition can be read as:

To define differential of x:
If
1 x is in fact 'x' then return 1.
2 x is not a list (and it can't now be 'x') then return 0.
3 x's head is 'plus' (and it must now be a list) then return a list of the form (plus (diff of 2nd element of list) (diff of 3rd element of list)). NB:CADR, etc, is the abbreviation of CAR(CDR.
4 x's head is minus then return a list of the form (minus (diff of 2nd element of list) (diff of third element of list)).
5, 6 & 7. An exercise for the reader!
8. If all else fails then return 'unknown'.

To use this function (assuming by now you have (a) rushed out to buy a Lisp system and (b) typed it in), just enter:

(DIFF '(PLUS (TIMES X 3)
(quotient 4 X)))

If everything is okay the system should immediately respond with the answer. Notice how, although the answer is mathematically correct, it is not in its simplest form, containing subexpressions like (TIMES 1 3) where 3 would do just as well, etc. Well, I didn't promise a simplification of a symbolic differentiation, did I? In any case it leaves you with a nice little follow-up exercise! Or alternatively (if your version of Lisp supports floating point arithmetic), you can use this routine without simplification in, for example, a numerical analysis program which would otherwise expect the differentiation to be done by the user (eg, Newton's method to find the roots of an equation) — a touch of irony for Lisp to be useful in numerical analysis!

Finally, for any reader who has been stimulated to the point of buying a Lisp system, here are the best known currently available (but not reviewed!):

Mu-Lisp — a Microsoft product for CP/M systems.

Mu-Simp — Ditto with simplification facilities.

END

APC SUBSET

by Ian Davies

This month sees the start of a new series for machine language programmers. It attempts to break through the difficulties presented by the ever swelling multitudes of machines on the market by presenting routines for processors rather than machines.

More and more micro enthusiasts are entering the world of assembly language programming. The purpose of this series is to present subroutines for the major processors using standard documentation and interfacing techniques. In this way, we will collect a library of routines which can be used by programmers as building blocks. The routines must be machine independant. In other words, the routines can call each other, but cannot call any subroutines provided in the ROM of a particular machine. Similarly, they cannot use system RAM or hardware addresses. By enforcing these rules, a routine written for a particular processor will run on *any* machine based on that chip, thereby providing coverage.

All the routines will be presented in a standard documentation format, to be described in this article. Contributions from readers in the form of new routines, improvements, criticisms and suggestions are very welcome as long as they are thoroughly tested and adhere to the standards.

RULES

The following are the programming rules to be maintained in the routines presented. Some of them are specifically for the Z80 as slightly more rigid conventions are required for this processor due to its many registers.

1. Registers not being used to convey data into or out of the routine will, if used by the routine, be saved on entry to and restored before exit from the routine.
2. A routine may call any other previously defined APC SUB-SET routine, but may not redefine that routine merely for its own convenience.
3. No routine may alter itself in any way. In other words, the routine should also be able to execute in ROM, as well as RAM.
4. RAM addresses, outside the general routines' library will never be explicitly specified in routines. All references should be made via parameters passed into the routine.
5. To avoid having areas of RAM that must be declared by the user, the stack may be used for local RAM.
6. The Z80 alternate register set will not be used by routines, to leave it available for processing interrupts.
7. Parameters may be passed to the routine either in the bytes immediately following the call or in registers loaded prior to the call. For the Z80, the following convention has been suggested:
   – A single parameter is passed in C (byte) or BC (word).
   – Two parameters are passed in BC (first) and DE (second), or C and E for bytes
   – A single result is returned in A (byte) or HL (word).
   – If more than two input parameters are passed, these are pushed onto the stack.

These conventions are suggested only, as there are many circumstances where a more customised parameter mechanism is appropriate.

DOCUMENTATION

We have agreed upon a standard system of documentation for all routines. An example of this is shown in the first listing, and it is arranged in the following way.

Section 1, prefixed by "; = " on the start of the line gives the name of the routine and a two or three word description. All other sections are prefixed by "; /".

Section 2 gives the class of the routine and, for class two routines (described later), can optionally give a brief explanation of why it is class two.

Section 3 specifies whether or not the routine is time critical. Routines that are time critical should also give the number of T states used. The T states of individual instructions can optionally be listed down the right hand side of the page.

Section 4 contains a brief description.

Section 5 gives the main actions carried out by the routine, preferably independantly of the program listing.

Section 6 specifies any subroutines called by the routine.

Section 7 describes any I/O interfaces, peripherals or local RAM areas needed.

Section 8 specifies flags, registers, parameters, stack or other areas assumed to have meaningful values when the routine is called.

Section 9 specifies flags, registers and other areas containing results when the routine returns.

Section 10 gives the register distributed by the use of the routine, including any routines it may call.

Section 11 gives the maximum number of bytes that could be added to the stack. This includes growth from the routine calling other routines but excludes the two bytes used by the main program call. Recursive routines may require some sort of formula here.

Section 12 specifies the memory size of the routine.

Section 13 is optional for routines not time critical, and gives the exact total or maximum number of T states.

Section 14 gives the processor or processors that will run the machine code.

The final part of the documentation is a complete list of the routine, with assembler mnemonics, commands (including T states if applicable) and machine code.

RE-ENTRANT CODE

Having an area of RAM reserved exclusively for a particular routine wouldn't do if the routine were to be interrupted by some other code that called the same routine. In this case, the second time the routine was called it would also use the reserved RAM area and corrupt it for when execution of the original routine was resumed. Using the stacks as local RAM avoids this problem.

Many of you might not be at the stage yet of switching control between programs running, in different time slots, over the same period. But some are doing this now and everyone is likely to want to do this some time in the future. So general purpose routines should allow for re-entrant code where possible.

RELOCATABLE CODE

How much we want code that will function in any location without re-assembly, depends on whether or not we have an assembler.

With an assembler that requires a displacement with a relative jump, there's a case for absolute jumps. They are less error prone and the labels they use are unaffected by the deletion and insertion of code.

Even so, there's something very satisfactory about code that functions regardless of where it's placed and, because it fits everybody's circumstances, must be preferred for general purpose routines.

RECOMMENDATIONS

As described in the documentation, the routines are separated into two classes, depending on their adherence to the recommendations.

Class 1 routines follow the recommendations completely and are re-entrant, relocatable and will not be self modifying.

The purpose of such recommendations is to enable programmers to use package subroutines 'blind' or modify with least trouble for their own purposes. How about:

a) Promable code:
Routines should not alter their own op-codes.
b) Re-entrant code:
No explicit RAM addresses – supply data following subroutine call or in registers – use stack as scratch pad.
c) Blind use:
Save and restore registers not used for data transfer.
d) Interruption:
Addresses below the stack pointer (SP) must be unused. Do not use alternate register set.
e) Position independant code:
No absolute jumps or calls except to general routine area.

All 'rules' are breakable with justification, understanding and documentation. For example, routines that don't use the alternate register set may not be interruptable through time restrictions and may be degraded without the alternate register set.

Routines not conforming to these recommendations will be in Class 2. They can either be incorporated with care into one's own system or after conversion into Class 1 routines.

LOCATING THE ROUTINES

We can't give addresses for the general purpose routines, as different users will have already committed different areas of memory for various purposes.

It's worth adopting the suggestion of one reader that other routines in the library will be called by their labels in the mnemonic listing and the memory address in the machine code will be represented by the dummy symbols 'XX XX'. Should it be necessary (against standard practice) to make an absolute jump or call within the current routine, the address in the machine code listing will be represented by the symbols 'YY YY'.

HOW SMALL?

Are very small routines worth making into Datasheets? Yes, if you are going to call them often enough. With only four bytes of working code in a routine plus three bytes to store and restore a register and return and three more bytes for the call, you have broken even when you have called it seven times. With eight bytes of working code and the same overheads you are wining on the third call.

READY, SET

Well, that's our documentation and coding conventions for APC SUB-SET. They may undergo a small amount of shuffling and alteration as we get reader feed-back, but that will be their general format. So much for the formalities, now we can get to business.

TEXT HANDLING

Our first routine for this month is for the Z80 processor and performs a highly useful function in text compression.

It can both pack characters in the range 20H to 5FH into 6 bit portions of contiguous memory and expand such blocks back into 8 bit bytes. This is very handy for all forms of data storage applications which contain text fields such as name or description. If the text is stored one character per byte, each byte may contain one of 256 different values, but in practice, probably only 64 different values actually are used. Packing the text fields when they are stored results in a 25% saving of space.

Datasheet

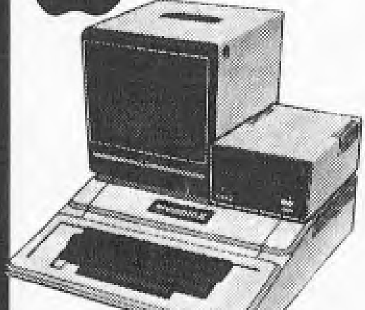
```
;= PAC/UNPAC — Compress and expand characters
;/ CLASS: 1
;/ TIME CRITICAL? No
;/ DESCRIPTION:  When entered at PAC, converts a string of 8-bit
;/               characters, in the range 20H to 5FH, to a string
;/               of 6-bit characters. When entered at UNPAC,
;/               converts 6- into 8-bit characters.
;/ ACTION: Not given
;/ SUBr DEPENDENCE: None
;/ INPUT:   HL holds the address of the 1st byte of source data
;/          DE holds the address of the 1st byte of the destination
;/          buffer
;/          BC = Number of characters to be packed/unpacked
;/ OUTPUT:  HL holds end of string + 1 address
;/          DE holds end of destination + 1 address
;/          BC = zero
;/ REGs USED: AF HL DE BC
;/ STACK USE: 4
;/ LENGTH: 64
;/ PROCESSOR: Z80

PAC:     LD    A,80H     ; clear A & set PAC flag.              3E 80
         JR    BINC      ;                                      18 01
UNPAC:   XOR   A         ; clear A for byte indicator.          AF
BINC:    INC   B         ; adjust BC for queer decrmnt.         04
LOOP:    PUSH  BC        ;                                      C5
         PUSH  HL        ;                                      E5
         INC   A         ; get next byte sequence no.           3C
         LD    C,A       ; and save in C.                       4F
         ADD   A,A       ; PAC or UNPAC flagged by carry        87
         LD    B,A       ; & no of bits to shift in B.          47
         LD    A,(HL)    ; get next source byte and             7E
         JR    C,DOPAC   ; jump if PAC.                         38 0D
         DEC   HL        ; else get other part of               2B
         LD    H,(HL)    ; packed byte in H and                 66
RISH:    RR    H         ; shift right until                    CB 1C
         RRA             ; in bits 0-5,A.                       1F
         DJNZ  RISH      ;                                      10 FB
         AND   3FH       ; mask out unwanted bits and           E6 3F
         ADD   A,20H     ; convert to ASCII.                    C6 20
         JR    LODE      ; jump to common part.                 18 0C
DOPAC:   SUB   20H       ; adjust ASCII to bits 0-5,A           D6 20
         LD    L,A       ; and copy into                        6F
         LD    H,+0      ; HL.                                  26 00
LESH:    ADD   HL,HL     ; shift left by twice the byte         29
         DJNZ  LESH      ; sequence number.                     10 FD
         LD    B,L       ; move lowest bits into B.             45
         DEC   DE        ; point to lowest avail dest.          1B
         LD    A,(DE)    ; byte, get in A, merge any bits       1A
         OR    H         ; from current source byte.            B4
LODE:    LD    (DE),A    ; converted byte to destination.       12
         POP   HL        ; restore source pointer.              E1
         LD    A,C       ; byte indicator to A.                 79
         AND   83H       ; do a "mod 4" on it.                  E6 83
         JR    Z,DINC    ; skip if UNPAC, byte 4.               28 09
         CP    81H       ; test to split PAC, bytes 1—3         FE 81
         JR    C,HINC    ; off & skip if UNPAC 1—3/PAC 4.       38 04
         INC   DE        ; lowest bits of PAC 1—3               13
```

The Logic Shop
Computer Systems.

has
Cheap, Clever, Computers.

apple



The Logic Shop has the best Apple deal in Australia:

First Class service at very competitive prices! A full range of Apple hardware, accessories and software are on display, to satisfy your personal or business needs.

NorthStar™

Northstar have a reputation for outstanding performance and reliability. The Northstar "Advantage" is true to name, and provides a very powerful desk top computer package:

Z80A Processor, twin floppies, (or optional 5MB Winchester), green phosphor screen, superb graphics and CP/M.

An ideal word processor and financial planner for around $5,000 (plus tax).

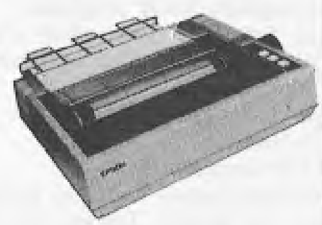
sinclair ZX81

Value!

The Sinclair ZX81 provides our lowest cost entry to computing. The ZX81 includes the "Basic" programming language and has a place in every home and school. Give your children the head start they deserve.

NEW!

EPSON



The outstanding Epson MX 80 III and MX 100 III have arrived! Superb new features and selling for the *old* prices! Hurry while stocks of these quality printers last!

Also in Stock:

IMAGINEERING
MICROPRO
CENTRONICS
MICROLINE
COMPUCOLOR
DIGITAL
TEXAS INSTRUMENTS
OLYMPIA
SENDATA
HOUSTON
SHARP
DISKETTES
SUPPLIES

The Logic Shop

For nearly three years The Logic Shop has combined a range of the best computer systems available, with a professional level of customer service and support. We carry stocks of the products we sell, so you can take delivery immediately.

For personal, business or educational needs, call into your nearest Logic Shop today for advice on the best computer solution.

WHERE:

PRAHRAN, 212 High Street, (03) 511950, 5292771
KMART, Cnr Blackburn Road & Burwood Highway, (03) 232 2011
CHIPPENDALE, 91 Regent Street, (02) 6994910 6994919
BRISBANE, Civic Arcade, Adelaide Street, (07) 312330
HOBART, 120 Murray Street, (002) 310818

Qume®

Quality!

The Qume Sprint 9 series daisy wheel printers have broken the price performance barrier to bring word processing letter quality printing to small computer users.

Qume printers provide superb quality with great reliability. An ideal combination for business and educational applications.

commodore

VIC-20

The Computer for everyone.

Just $399

Why buy just a video game when you can get a full colour computer for this price.

A computer like this would have been fiction a few years ago. Now it's a reality. It's the new Commodore VIC-20, a full-fledged, expandable colour computer that costs little more than the video games.

Everybody loves video games and the VIC-20 has some of the best. But the Commodore VIC-20 can also help the kids with their homework and mum with her home budgeting. Dad can even take the light, portable VIC-20 to the office for financial and business applications. And Commodore has many more applications on the way. With full capability for:

- Education programmes
- Recreational programmes
- Personal computing
- Includes Microsoft, PET BASIC
- Connects to any TV set
- 4 sound generators ● 16 colours
- 66 graphic characters
- Full-size typewriter-style keyboard
- Easy-to-follow instruction manual
- 25K total memory ● 5K RAM
- Memory expansion to 32K RAM

The VIC-20 is the friendliest way we know to learn computing. It has a full computer keyboard even a small child can operate. It plays music, has exciting graphics and lets you create pictures. It even tells you when you've made a mistake and how to correct it. The VIC-20 can take your children from pre-school through post-graduate studies.

Why get just another game that could end up in the closet? Get an honest-to-goodness computer for just $399. Get the Commodore VIC-20.

So much brain for so little.

commodore
COMPUTER

MIVI1461

```
        EX     DE,HL     ; with trailing reset bits    EB
        LD     (HL),B    ; into next empty             70
        EX     DE,HL     ; destination byte..          EB
HINC:   INC    HL        ; point at next new bytes.    23
DINC:   INC    DE        ;                             13
        POP    BC        ; restore count               C1
        DEC    C         ; and test for                0D
        JR     NZ,LOOP   ; last byte and               20  C9
        DJNZ   LOOP      ; repeat if not.              10  C7
        RET              ; else return.                C9
```

BLOCK MOVE

Our second routine is for the 6800 processor, and translates a block of binary data to ASCII-hexadecimal characters, which are stored at a specified location with a checksum appended.

HEXMV is used to prepare for the serial transmission of ASCII data from one micro to another. Several systems put data to backing storage in this form. Storing data in ASCII-hexadecimal has the advantages of allowing for the 176 possible non-ASCII bytes to be used for control purposes or for redundant bits to be used in the detection and correction of transmission errors. It has the disadvantages of occupying twice as much space as the original data and therefore taking twice as long to transmit and allowing twice as many opportunities for errors in transmissions to occur.

There are other situations, apart from data storage, where it is useful to convert and hold data in ASCII-hexadecimal, in order to distinguish it from control information.

CONTINUING

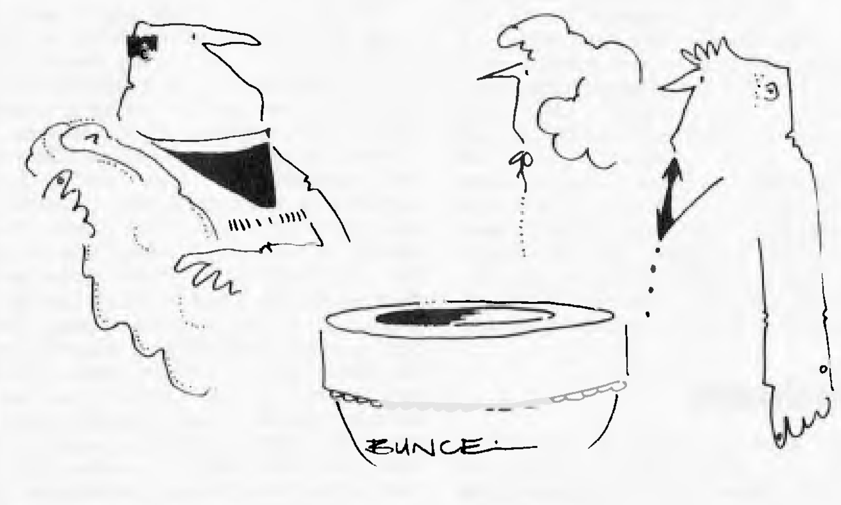
APC SUB-SET will continue next month with some handy ASCII to binary and binary to ASCII conversion routines, as well as a number cruncher. We are hoping to receive some contributions from readers in the near future. You can either send in original routines, or just translate the SUB-SET routines over to another processor. We will give appropriate credit for everything we use. See you next month.

Datasheet

```
;= HEXMV — Block move.
;/ CLASS: 1
;/ TIME CRITICAL ?: No.
;/ DESCRIPTION: Takes a block of data up to 256 bytes long,
;/              translates it into ASCII-hexadecimal and
;/              deposits the result in a destination table with
;/              a one-byte (two ASCII-hex digits) checksum
;/              appended.
;/ ACTION: 1. Initialise checksum to zero.
;/         2. Subtract 2 from destination address.
;/         3. Read byte pointed to by source address.
;/         4. Unpack low & high nibbles and add both to checksum.
;/         5. Convert to 2 ASCII-hexadecimal digits.
;/         6. Increment source address by 1.
;/         7. Increment destination address by 2.
;/         8. Store MS-digit of ASCII-hex at dest. address
;/            and LS-digit at Dest. address + 1.
;/         9. Decrement byte count and so to step 3 if not zero.
;/        10. Read checksum then repeat steps 4 thru 8.
;/        11. Return to calling program.
;/ SUBr DEPENDENCE: None.
;/ INTERFACES: None.
;/ INPUT: Parameters supplied on stack by calling program:
;/        SOURCE ADDRESS at SP+1 and SP+2
;/        DESTINATION ADDRESS at SP+3 and SP+4
;/        BYTE COUNT at SP+5
;/ OUTPUT: 2n+2 (where n = byte count) bytes of ASCII-hex data
;/         (including checksum) stored at destination.
;/         Parameters still on stack:
;/             SOURCE ADDRESS = input value + n
;/         DESTINATION ADDRESS = input value + 2n
;/                               (points to checksum)
;/                    BYTE COUNT = zero
;/ REGs USED: XR, ACCA, ACCB and CONDITION CODES
;/ STACK USE: 8
;/ LENGTH: 89
;/ PROCESSOR: M6800

HEXMV:  CLR A          ; initialise checksum          4F
        PSH A          ; and put on stack.            36
        TSX            ; point XR to stack workspace. 30
        LDA A 6,X      ; initialise                   A6 06
        SUB A #2       ; destination                  80 02
        STA A 6,X      ; pointer to                   A7 06
        BCC   A7       ; DEST-2 for                   24 02
```

```
        DEC   5,X      ; pre-incrementing.                6A 05
A7:     LDX   3,X      ; fetch next byte XR←(XR+3—4)      EE 03
        LDA A 0,X      ; of binary source.                A6 00
        TSX            ; convert to ASCII, store          30
        BSR  HEXMV1    ; at dest. and update              8D 0B
        TSX            ; pointers & checksum.             30
        DEC   7,X      ; decrement byte count &           6A 07
        BNE   A7       ; repeat until zero.               26 F4
        LDA A 0,X      ; append checksum                  A6 00
        BSR   HEXMV1   ; to ASCII data.                   8D 02
        PUL  A         ; clear stack of checksum.         32
        RTS            ; return.                          39
; convert byte from binary to hex, move it to
; destination and update pointers and checksum.
HEXMV1: BSR   BINASC   ; convert byte to ASCII-hex        8D 19
        INC   4,X      ; update                           6C 04
        BNE   A8       ; source                           26 02
        INC   3,X      ; pointer                          6C 03
A8:     INC   6,X      ; update                           6C 06
        BNE   B8       ; destination                      26 02
        INC   5,X      ; pointer.                         6C 05
B8:     INC   6,X      ;                                  6C 06
        BNE   C8       ;                                  26 02
        INC   5,X      ;                                  6C 05
C8:     LDX   5,X      ; XR←(XR+5—6)                      EE 05
        STA A 0,X      ; store ASCII at                   A7 00
        STA B 1,X      ; destination.                     E7 01
        RTS                                               39
; convert byte to two ASCII-hex bytes in A & B
; and update checksum.
BINASC: TAB            ; B←A.                             16
        AND A #0FH     ; separate byte into two           84 0F
        AND B #F0H     ; hex. digits in A and B.          C4 F0
        LSR B          ; logical shift 1 bit right.       54
        LSR B          ;                                  54
        LSR B          ;                                  54
        LSR B          ;                                  54
        PSH A          ; save A.                          36
        ADD A 0,X      ; add checksum to A.               AB 00
        ABA            ; add B to A.                      1B
        STA A 0,X      ; put updated checksum on stk.     A7 00
        PUL A          ; restore A.                       32
        BSR   HEXASC   ; convert hex digit to ASCII.      8D 06
        PSH A          ; save A.                          36
        TBA            ; A — B.                           17
        BSR   HEXASC   ; convert hex digit.               8D 02
        PUL B          ; restore A to B.                  33
        RTS                                               39
; hexadecimal to ASCII conversion subroutine.
HEXASC: ADD A 90H      ; so A—F will give carry on        8B 90
        DAA            ; decimal adjust.                  19
        ADC A 40H      ; add any carry & 40H.             89 40
        DAA            ; decimal adjust.                  19
        RTS                                               39
```



'We want to call him GOTO. That way he'll fit into the system better when he gets older.'

BENCHTEST
BUSINESS SYSTEM

SIRIUS BUSINESS

Peter Rodwell tests the Sirus 1–16 bits for the price of eight

Chuck Peddle is widely regarded as the man who started the personal computer industry. He designed the 6502 processor — now the second most widely-used microprocessor in the personal computer scene — and then designed the Commodore PET around it, thereby launching the first generation of true microcomputers.

Chuck has now left Commodore (and at the time of writing was involved in a complex, two-way lawsuit with CBM) and has started his own company, Sirius, the first product of which is the subject of this Benchtest.

The Sirius I is Chuck's idea of a 'third generation' microcomputer. The first generation, he says, comprised machines such as the PET and Apple; the second consists of the sort of machines now being offered to the business market — 64k, twin-disk micros with 8-bit processors. The third generation he defines as being based around 16-bit processors and plenty of memory and offering advanced features such as very high resolution graphics and good ergonomics; above all, he says, they should be designed for the end user rather than for the programmer, and should be regarded by the user not so much as a computer but as a piece of business equipment.

The Sirius I reflects these criteria in that it has been designed for the end user rather than for the programmers or computer freaks. In this country it is being sold through Barson Computers (formerly Consolidated Marketing Corporation) and is known as the Sirius I; in the States it is called the Victor 9000 and some of the software I saw displays the Victor 9000 as an introduction. The basic Sirius comes with 128k of RAM, twin minifloppy disk drives, a VDU and keyboard and software which includes CP/M-86, Microsoft's MSDOS operating system.

Hardware

Three modules make up the Sirius system; a main cabinet housing the processor, RAM, power supply and disk drives; a display unit which sits on top of the main cabinet on a turntable which allows you to rotate and tilt it and a low-profile keyboard unit.

Opening up the main cabinet immediately reveals one important aspect of the hardware design — it's easy to get at and service, which should reduce maintenance costs. The back panel is secured by two screws and, with this removed, the unit's lid simply unslots to allow easy access to the internals. At the back there's a heavily shielded power supply module, which can be removed simply by slackening half a dozen screws, under a PCB containing their control circuitry, all of which are similarly easy to remove.

The processor, 128k of RAM (Hitachi 64k chips) and all the other electronics are sited on one large PCB at the bottom of the cabinet; this slides out completely, allowing field maintenance staff to simply replace the entire board in the customer's office within minutes; faults can then be corrected back at the workshop while the client carries on computing. This is just as well, for only five chips in the whole system are socketted.

The CPU is an Intel 8088, running at 5 MHz. This is a 16-bit micro internally but looks like an 8-bit processor to the rest of the system, thus offering the power of a 16-bit micro with the lower system cost of an 8-bit engine. It's the same processor which IBM chose for its Personal Computer and which will be appearing in a number of new machines, including several from Japan, during the course of 1982. Cost-effectiveness apart, its big advantage is that it is code-compatible with the Intel 8086, a full 16-bit micro, and can thus run software developed for the '86, including, most importantly, the CP/M-86 operating system from Digital Research.

Four sockets are provided on the main board for add-in cards. Three of these will be used by people wanting to expand the system to its full 896k internal RAM capacity, by adding 128k or 384k expansion cards. As can be seen from the memory map, things aren't quite as simple as with an 8-bit machine — various parts of the memory are reserved for system use (screen, interrupt vectors, character dot patterns, I/O ports, etc) and the expansion memory in fact slots into the middle of the map, with the operating system then being configured to sit at the top of it.

Disk drives are usually pretty boring things but the twin 5¼in drives on the Sirius aren't — they each cram 600 Kbytes onto *one side* of a disk! Double-sided drives are available as an optional extra to give an incredible 2.4 Mbytes; I know of no other system which achieves this capacity on 5¼in drives. It's achieved by some pretty clever circuitry and software which, first, varies the number of sectors per track from 19 at the outer edge to 12 at the centre and then varies the speed at which the disk rotates according to which track the head is over, from 250rpm when it's at the outer edge to 350rpm when it's in the centre, in eight steps. You can actually hear the drives changing speed, especially when copying a number of files from one disk to another; they hum at different pitches. sometimes in harmony — it can only be a matter of time before some bright spark writes a program to play tunes on them. The drives are actually quite noisy and when the heads are seeking back and forth across the disk's surface, the machine emits weird clucking noises not unlike a demented chicken. Quite what the effect on disk life will be as a result of being spun at 350rpm (normal speed is 300rpm) for some of the time, I'm not sure.

The Sirius has three ports with which to communicate with the outside world. There's a parallel printer port which doubles as both Centronics standard and IEEE-488 and an asynchronous RS232 printer port with programmable baud rates from 75 to 9600 baud. A second serial port is provided for communication (i.e. to other computers under a number of protocols) at baud rates between 1200 and 9600.

One very unusual feature is an on-board Codec speech digitiser which allows you to store speech on disk and play it back through a built-in loudspeaker! The quality of the replayed speech is quite good – not hi-fi but much better than synthesised speech.

A surprising omission is a clock/calendar with Nicad battery back-up, but an add-in clock card should be available before long.

Two areas of memory are reserved for the display, one of the machine's outstanding features. In normal mode, the display is 80 characters by 25 lines. The 12in green screen monitor gives a clear, steady display, of which the only criticism I have is that the screen has a fairly long-persistance phosphor – the image takes a few seconds to die away, which is rather confusing when listing a long program or scrolling through text. I understand that the current green display will eventually be replaced with an amber-on-bronze screen when one can be found with sufficiently high resolution, and a full colour screen is also promised in the future.

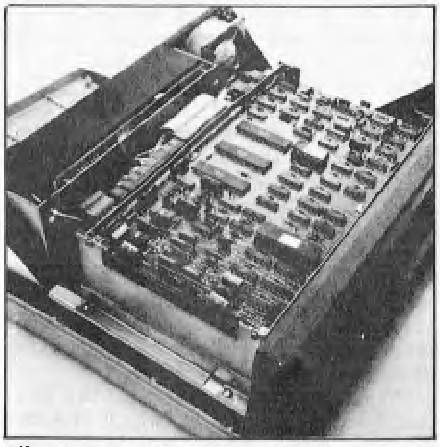
Like just about everything else on the machine, the screen is under full software control. A 4k area up in RAM holds the characters to be displayed while lower down there's another area containing the dot patterns which make up the characters. Characters on the 80 x 25 standard screen are built up on a 10 x 16 matrix, which makes for very clear, elegant text indeed. But because the dot patterns are there in RAM, you can access them and change them using a utility program called Edot, of which more later. Up to 2048 characters can be held in RAM at any one time and character sets can be stored on disk and called into memory under program control.

Naturally the Sirius has graphics capability, a bit-mapped 800 x 400 display, in fact, finer resolution than any other microcomputer of this price range. In the graphics mode it's possible not only to display some very spectacular graphics but also to display finer resolution text, up to 132 columns by 50 lines, all perfectly readable!

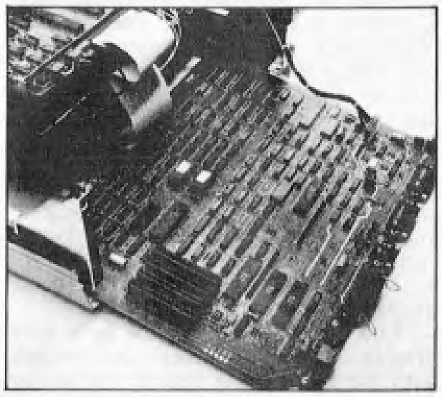
The display allows dual intensity, reverse video and proper underlining, either under program control or from the keyboard. And, of course, brightness and contrast are adjustable (through eight levels), again under software control or from the keyboard – there are no controls at all on the monitor itself. The loudspeaker's volume is similarly controlled, by the way.

The monitor is mounted on a turntable which allows rotation 42 degrees in either direction and it can be tilted upwards up to 11 degrees from horizontal. It has an anti-reflective coating on the screen.

The keyboard is wider than the main unit and has a firm, pleasant feel; it's also relatively heavy (1.5kg) so it doesn't slide around on a desk top. There are a total of 95 keys, divided into several groups. The main group is, naturally, a full qwerty keyboard, laid out (according to Sirius documentation) in IBM Selectric format; IBM seems to have changed its layout recently, if our newly-acquired office typewriter is anything to go by, but the differences are minimal. At power-on the keyboard is in lower-case, but there's a shift lock key (with a funny padlock symbol on it), although no warning light to show



The disk controller PCB



Main board contains 8088 and 128k RAM

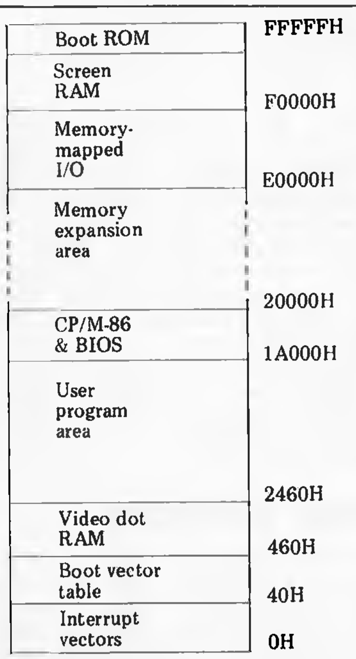


Fig 1 Memory map of basic 128k Sirius 1

Slimline remote keyboard is IBM Selectric format when you're in upper case. Two unusual keys sit at the ends of the space bar, right where my fingers expect to find the shift key. One is labelled 'ALT' while the other is called 'CONT'. The ALT key is, in fact, the control key, while CONT, confusingly, isn't control but suspends whatever the computer's doing and resumes the operation at a second touch, useful for stopping fast-scrolling text or listings while you read them.

Across the top of the keyboard are seven large programmable function keys. They are, of course, all user-programmable, although I would prefer to see more than just seven.

At the right-hand end of the keyboard is a numeric pad which includes percent, plus, minus, add and divide symbols but, strangely, no equals sign.

Between the numeric pad and the main block are two columns of keys controlling the cursor and, on the model I used, clearing the screen and deleting and inserting characters and lines. Pressing ALT with some of these keys gave control over screen brightness and contrast (eight levels) and loud speaker volume. On the far left of the board is another column containing the escape and repeat keys and controls for the dual intensity display, reverse video and underlining.

Two types of keyboards have been produced for the Sirius: an 'English' with the appropriate pound sign and an 'American' style which includes a number of characters not normally accessible from the keyboard.

(π <>.↑ ¬ ∧ ' { } ~ \ ).

The keyboard is fully debounced and has n-key rollover but it isn't encoded – it doesn't generate ASCII codes but 'logical key numbers' which system software converts to ASCII. This means that you can easily redefine any key to generate any character, a necessary feature when programming in Basic, as we'll see later.

Setting up the system is simplicity itself. All connectors are at the rear of the main cabinet (I would prefer the keyboard to plug in at the front) and all that's involved is for the video and keyboard plugs to be plugged in (they use different, keyed plugs, so it's impossible to get them mixed up or plug them in the wrong way up despite the lack of labels on any of the sockets, although there is a labelled drawing of them in the manual) and for the power to be turned on, again at the back of the main cabinet, where a reset button also lurks, sensibly out of harm's way. There is no power-on indicator, although the built-in fan makes so much noise that this isn't really needed. It's a pity about the fan – it *is* noisy and it forms my only real complaint about the hardware design. The disks spring to life and inserting a system disk into the left-hand drive immediately causes the machine to boot up CP/M-86 without any further action required from the operator.

BENCHMARK TIMINGS

BM1	2.0
NM2	7.4
BM3	17.0
BM4	17.5
BM5	19.8
BM6	35.4
BM7	55.9
BM8	42.5

All timings in seconds.

CP/M-86

The first interface a user will have with the Sirius (or most other micros, come to that) is via its operating system. At the moment, the Sirius comes with CP/M-86, the 8086 version of the industry-standard CP/M operating system, written by Digital Research. This was my first encounter with CP/M-86 and, frankly, I was deeply disappointed. If you're familiar with CP/M, you'll know what I mean when I say that, to the user, CP/M-86 looks and behaves exactly like the old CP/M. You can now skip the next two paragraphs while I explain that remark to those who have never tried to use CP/M.

Back in the early days of microcomputers, a Californian called Gary Kildall wrote a basic (not Basic) program which would take care of all the boring nitpicking things a computer has to do, such as receiving character typed in at the keyboard, displaying it on a screen or printer and, especially, carrying out all the intricate work involved in using floppy disks. The idea was to make life easier for the programmer; all these 'housekeeping' routines were supplied so that the programmer could use them easily while writing his (assembler-language) programs, without having the trouble of re-inventing half a dozen wheels by writing them all himself. It had the added, and very considerable, advantage that it was so designed that a program written under CP/M on one machine would run perfectly well on another, completely different machine, providing that it, too, had CP/M on it. And CP/M could easily be configured to fit any machine which had an 8080, 8085 or Z80 processor and a minimum of 16k or RAM.

This ease of transferring programs from one computer to another did much to encourage the growth of the microcomputer industry, both hardware and software. An enormous number of machines are now available which run CP/M and there is a correspondingly massive and ever-growing amount of software available, especially language compilers and interpreters and some applications packages, particularly word processors.

The problem is that CP/M was designed to make life easy for the *programmer* – it does little to make things easy for the *user* and in fact it's very unfriendly, with a nasty habit of displaying unintelligible error messages and stopping dead when something goes wrong.

In producing CP/M-86, Digital Research had the chance to make the system a lot friendlier and easier to use *for the user* but they didn't – they blew it. Various attempts have been made by other companies to produce CP/M-like operating systems which are more user-friendly; the one with which I am most familiar is Cromemco's CDOS, developed three or more years ago as an upwards-compatible version of the 8-bit CP/M (i.e. you can run CP/M programs under CDOS, usually, but CDOS contains extra facilities which, if you use them, prevent your program from running under CP/M).

I'll give you one example, comparing CP/M-86 with the older CDOS, to show what I mean about user-friendliness (and this doesn't mean that I regard CDOS as the ultimate in user-friendliness; it's just that it provides an indication of what can be done, quite easily, to make life a little easier for the user).

It's possible to write-protect a mini-floppy disk by placing a sticky label over a notch in its plastic envelope; this prevents the computer from altering the data on the disk or from adding new data to it. If you try to transfer a file from one disk to another which is write-protected, then CP/M-86 displays the following message: 'Drive=0, Track=7, Sector=0: Error =32 Bdos Err on A: Bad Sector' and stops dead. I defy any end user who isn't a programmer to make sense of that. CDOS, under the same circumstances, displays: 'Diskette in drive A is write-protected'.

There are those who would argue that a user shouldn't be trying to write to a write-protected disk; this is ridiculous, as it's one of those mistakes which *everyone*, even experienced users, make from time to time and an operating system should cater for it in a useful and helpful way, especially on a machine which is designed to be used by the rapidly-growing army of first-time end users.

In CP/M-86 and the Sirius, then, we have a curious combination of third-generation hardware running what is essentially still a first-generation operating system. Fortunately, the machine will also be sold with MSDOS (which is virtually identical to the operating system that company produced for the IBM Personal Computer as well as CP/M-86.)

From the programmer's point of view, CP/M-86 is very similar to the 8-bit CP/M, but with added facilities; this, coupled with the XLT-86 package from Digital Research (which translates 8080 source code into optimised 8088/8086 source code) means that many software houses may initially be inclined to produce CP/M-86 packages for the Sirius rather than tackle the unknown MSDOS. This is a pity, because, although I have no first-hand experience of MSDOS, from all accounts it does sound far friendlier and easier to use than CP/M-86.

Other software

The first package I looked at was Microsoft's Basic-86, an 8088/8086 implementation of that company's MBasic V5.0. It was pretty standard, with one or two minor alterations but when I ran the standard Benchmark tests I found it appallingly slow. Quite why it was so slow I could not figure out, but it could be that this version was a straight translation from the original 8-bit Basic and took no advantage of the 8088's superior computing power.

The slowness of the Basic will be overcome, though, as an 'extended business Basic compiler' is on its way.

Other software which will be available for the launch of the Sirius in Sydney on June 21 (about two weeks away from the time of writing) includes a fully configured Version 3 of Wordstar which incorporates SpellStar, and the standard, familiar line of IMS Software (Debtors, Creditors, General Ledger, Payroll etc.).

Additionally, a new Basic called GW Basic (it's rumoured that this stands for 'Graphic Word' Basic) was completed only a week ago and should also be here for the Australian launch. This will incorporate a full set of graphic commands, such as PLOT, DRAW, LINE and BAR. I was pleased to hear this as, although many end users will simply be running pre-written packages (according to Chuck Peddle's philosophy for third generation micros), there will still be many users particularly in engineering, science and education, who will want to do their own programming in a high-level language without having to dive into assembler to do anything fancy. Making the machine's facilities available through high-level languages will also speed up software development times, and consequently lower software costs, too.

Earlier I mentioned a package called EDOT, which enables the user to define or alter character sets. This I found a fascinating program indeed. Figure 1 shows the screen when Edot is running; the character dot matrix is displayed, together with the full character set currently in memory — you can read in other sets from disk and write your own-developed set to disk. By using the numeric pad, you can manouvre a 'blob' (it's up in the top left-hand corner in Figure 1) to any pixel you want displayed and a keystroke then lights up at the cell at the blob's position. Simultaneously the character you are building is displayed in normal, underlined, dual-intensity and reversed modes and is updated as you build it up, as are any other occurrences of that character on the screen if you happen to be working on a character in the set currently in use by the system.

Using Edot, you can call in any character set from disk and use it as the system's 'normal' character set. The demonstration disk contained several sets, including a handwriting-style script set, which looked decidedly funny when used with Wordstar.

Other packages which will be available for the Sirius — some immediately, others fairly soon — include an Assembler, Fortran, Cobol, Pascal and PL-1 for the programmers and products from other companies including Micro-Modeller, Decision Modeller, Supercalc and Micro Planner.

Documentation

The machine comes supplied with an introductory users' manual, and a hardware technical manual is also available. Sirius in the US is currently producing its own CBasic manual which, again, should be here in time for the launch.

Microsoft's Basic manual came with an updating leaflet describing the differences between MBasic and Basic-86, which was clear and readable as all Microsoft documentation seems to be; and Digital Research's CP/M-86 manuals, one entitled 'System Guide', the other called 'Programmer's Guide'. In fact, both seemed to be aimed totally at the programmer and would give an end user an even bigger shock than CP/M-86 itself. Some of the Sirius documentation mentioned a 'User Guide' for CP/M-86; this wasn't supplied but I hope it's a lot better than previous attempts by Digital Research to communicate with end users.

Users and potential

Chuck Peddle designed the Sirius as a business tool, primarily one which would sit on a desk and run applications packages. As such, with tailored software, it will be a big success, especially at its low cost.

The machine is very pleasant to use, CP/M-86 and the noisy fan excepted; I feel that any business user, either manager or secretary, will feel at home with the Sirius as, once an application package is up and running and the operating system is left behind, it reveals itself as a machine with many attractive features and into which a lot of very careful thought — especially on the ergonomics side — has been put.

I think, too, that the Sirius will prove useful and popular among other users, especially engineers, scientists and teachers. For these people, a much easier interface to the machine's facilities is essential and will hopefully be provided by the enhanced high-level languages which have been promised. In particular, the very high resolution graphics will be extremely attractive — there's nothing available to compare with the Sirius in this price range and indeed, given that the Sirius will have communications packages available to facilitate links to other computers, it would make a very cost effective alternative to an ordinary graphics terminal.

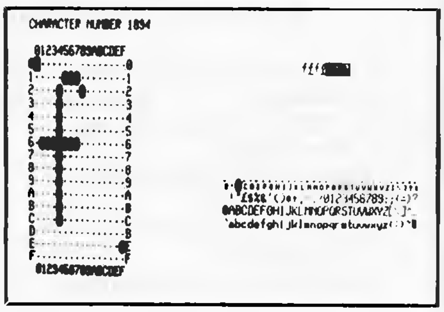


Fig 1 Edot in action.

Programmers will love the Sirius, too, as there's so much which is directly under software control and so many interesting features to liven up otherwise rather boring applications packages. I sincerely hope that programmers will make the effort to produce packages for MSDOS, if it's as good as it's said to be, instead of taking the easy way out and converting 8-bit CP/M packages to run under CP/M-86. And I'll give a free year's subscription to *APC* to the first person to write a program which plays a recognisable tune of more than 20 seconds' duration on the disk drives!

Expansion

As I've already mentioned, users will be able to expand the RAM up to 896k internally by plugging in 128k or 384k RAM cards. If you want more than that you'll have to wait for an 8086 machine but most people should find the basic 128k machine enough for many applications.

Double-sided floppy drives are provided as an option but many may prefer to opt for the 5Mb or 10Mb winchesters. Unfortunately, as you've probably already gathered, it's been a bit of a rush at Barson Computers to have everything ready for the launch and the winchesters appear to be another item which 'will be here for the release in Sydney on June 21'. So I couldn't test them.

Two hard disks will be produced, both 5¼in and both being housed in a structure external to the main cabinet. Only $1800 separates the 5Mb and the 10Mb drives so I expect the larger capacity drive will be far more popular. A tape streamer for backing up the hard disks is also mentioned in the preliminary Sirius documentation.

Two add-ons are already in the pipeline: a clock card and a thing called a Mouse, apparently a little box which you push around on a table top to move the cursor around the screen. A graphics tablet will also be developed and should be available in the second half of this year.

Other hardware add-ons mentioned in the Sirius documentation but as yet unscheduled include a modem card, a colour display and a network interface card, although at the time of writing no decision had been taken as to which network will be used.

On the software side, all the major language and applications packages should soon be available including a package to read CP/M-86 files from MSDOS and a package to allow the transfer of files, by direct hook-up, between the Sirius and the IBM Personal Computer — the Sirius is 'upwards compatible' from the IBM, says Chuck. Various communications packages, including TTY and Telex emulation and assorted big machine protocols (all right, IBM 3276 and 307B) are also on their way.

Conclusions

Chuck Peddle set out to design an office tool which would be easy to use, ergonomically satisfactory and provide 16-bit computing power at a low cost. He has succeeded; the Sirius is one of the few micros which has been designed completely with the user in mind and its astonishingly low cost (achieved by mass production) makes it about the best value for money on the micro market today.

Chuck perceives its market slot as slightly above the IBM Personal Computer, which, in the States, is being aimed at the very top end of the home

BENCHTEST
BUSINESS SYSTEM

market and the middle-to-low end of the business area. It's likely that the machines which will really feel the pinch as a result of the Sirius are those such as the Apple III and the Tandy Model II-type configurations, most of which are either more expensive or less powerful or offer fewer facilities, or all three.

In summary, the Sirius is a neat, well-made machine with considerable potential and great appeal, especially in view of its very competitive price. If I were looking for a general purpose business micro, I'd be hard put to find an excuse for not putting this machine at the top of the list.

Prices

Sirius 1 with 128k RAM, 1.2 Mbytes disk capacity, CP/M-86, MSDOS, utilities, documentation	$5,295
As above but with 2.4 Mbyte disk capacity	$6,295
128k RAM upgrade	$1,395
384k RAM upgrade	$3,500
5 Mbyte winchester with controller and software	$4,800
10 Mbyte winchester with controller and software	$6,600
Audio input package including PCB, microphone, software and documentation	$375

All prices exclude sales tax.

TECHNICAL SPECIFICATIONS

CPU:	8088, 5 MHz
Memory	128k (64k dynamic RAM chips) internally expandable to 512 kbytes; external module to expand to 1 Mbyte
Keyboard	95 keys inc 7 prog function keys, numeric pad, cursor control, editing, screen & loudspeaker control, all software re-definable.
Screen	80 char x 25 lines; hi-res graphics, 800 x 400 bit-mapped, user-definable character sets.
Disk drives	2 x 5¼in single-sided, high density, 600 kbytes per unit.
Ports	1 Centronics/IEEE-488 printer port, 1 RS232 printer port, 1 RS232 communications port.
System Software	CP/M-86 and MSDOS
Languages	Basic 86; Graphics & character set packages. Assembler, Fortran, Cobol, Pascal, PL-1, Extended business Basic compiler, various applications software.

APC is interested in Basic or Pascal Programs for any popular micro – please tell us which one you wrote your program on and how much memory it uses. Make sure your programs are fully debugged before you send them in on cassette (although we will accept disks) with a clear listing on plain paper. Documentation would be welcome, and if you want it returned please label everything with your name and address and include an SAE. Send contributions to APC Programs, P.O. Box 115, Carlton, Vic. 3053.

Invader

by R. Fuller

Invader is a two player game written for a 16k System 80/TRS-80. Simply set memory size to 28500, CLOAD and RUN the program. The controls for top base are the 'ADF' keys and for the bottom base, the '123' keys.

If you don't have a numeric keypad this arrangement may be too cramped for the two players. If so, insert the line below and use the ' , . / ' keys as controls for the bottom base.

```
105 POKE 31765,32 POKE31768,16:
   POKE31805,32:POKE31808,64:
   POKE32019,32POKE32022,128
```

```
5 CLEAR150
10 CLS:PRINT"LOADING INVADER MACHINE CODE":A=31744
20 READ A$
30 D=D+1:PRINT@56,D
40 IFA$="XX"THEN100
50 B=ASC(LEFT$(A$,1))-48
60 C=ASC(RIGHT$(A$,1))-48
70 IFB>9THENB=B-7
80 IFC>9THENC=C-7
90 POKEA,B*16+C:A=A+1:GOTO20
100 S$=STRING$(3,191):T$="   ":D$=S$+T$+S$+T$+S$+T$:D$=D$+D$+D$+S$+T$+S$
110 POKE16526,0:POKE16527,124
120 CLS:PRINT@24,"INVADER":PRINT
130 POKE28672,193:POKE28673,60:POKE28674,122:POKE28675,63:FORI=28676TO28690:POKE
I,0:NEXTI
140 INPUT"WHO IS PLAYER WITH TOP BASE";A$:A=65+LEN(A$):POKE28682,A:POKE28683,60
150 INPUT"WHO IS PLAYER WITH BOTTOM BASE";B$:B=193+LEN(B$):POKE28684,B:POKE28685
,63
160 INPUT"WHAT LEVEL OF PLAY (1-3)";C:POKE32303,76:POKE32304,29
170 IFC=3THENPOKE32303,166:POKE32304,14
180 IFC=1THENPOKE32303,152:POKE32304,58
190 INPUT"INVISIBLE GAME";C$:POKE31831,143:POKE31834,191:POKE31837,143:POKE31848
,188:POKE31851,191:POKE31854,188
200 IFLEFT$(C$,1)="Y"THENPOKE31831,32:POKE31834,32:POKE31837,32:POKE31848,32:POK
E31851,32:POKE31854,32
210 PRINT@64,CHR$(31):PRINT@128,STRING$(64,131);:PRINT@896,STRING$(64,176);:FORI
=15488TO16319STEP64:POKEI,191:POKEI+63,191:NEXTI
220 PRINT@64,A$;:PRINT@960,B$;
230 PRINT@384,D$;:PRINT@512,D$;
240 X=USR(0)
250 FORI=1TO750:NEXTI
260 S1=PEEK(28686):S2=PEEK(28687)
270 IFS1>S2THENPRINT@464,A$;" WON, CONGRATULATIONS!";
280 IFS2>S1THENPRINT@464,B$;" WON, CONGRATULATIONS!";
290 IFS1=S2THENPRINT@474,"TIED GAME!";
300 FORI=1TO1000:NEXTI
310 PRINT@64,CHR$(31)
320 INPUT"PLAY AGAIN";E$:IFLEFT$(E$,1)="Y"THEN120
325 PRINT:PRINT"HOPE YOU HAD AN ENJOYABLE GAME,
    PLAY AGAIN SOON...":PRINT
330 DATA 3A,01,38,FE,02,C2,14,7C,3A,00,70,FE,C1,CA,14,7C,3D,32,00,70
340 DATA 3A,10,38,FE,02,C2,28,7C,3A,02,70,FE,41,CA,28,7C,3D,32,02,70
350 DATA 3A,01,38,FE,10,C2,3C,7C,3A,00,70,FE,FA,CA,3C,7C,3C,32,00,70
360 DATA 3A,10,38,FE,04,C2,50,7C,3A,02,70,FE,7A,CA,50,7C,3C,32,02,70
370 DATA 2A,00,70,36,20,23,36,8F,23,36,BF,23,36,8F,23,36,20
380 DATA 2A,02,70,36,20,23,36,BC,23,36,BF,23,36,BC,23,36,20
390 DATA 3A,01,38,FE,40,C2,96,7C,3A,04,70,FE,01,CA,9E,7C,2A,00,70,11,42,00,19,22
,06,70,36,24,3E,01,32,04,70,C3,12,7D,3A,04,70,FE,01,C2,12,7D,2A,06,70,36,20,11,8
0,00,19,22,06,70,3A,07,70,FE,3F,C2,EA,7C,3A,06,70,47,3A,02,70,3C,C6,40,B8,CA,CA,
```

```
7C
400 DATA 3C,B8,CA,CA,7C,3C,B8,C2,EA,7C,2A,02,70,23,36,48,23,36,49,23,36,54,3E,01
,32,10,70,3A,0E,70,3C,32,0E,70,3E,00,32,04,70,C3,12,7D,2A,06,70,7E,FE,B0,C2,FB,7
C,3E,00,32,04,70,C3,12,7D,FE,BF,C2,0D,7D,2A,06,70,36,20,3E,00,32,06,70,C3,F3,7C,
2A
410 DATA 06,70,36,24
420 DATA 3A,10,38,FE,08,C2,36,7D,3A,05,70,FE,01,CA,3E,7D,2A,02,70,11,C2,FF,19,22
,08,70,36,23,3E,01,32,05,70,C3,AE,7D,3A,05,70,FE,01,C2,E0,7D,2A,08,70,36,20,11,C
0,FF,19,19,22,08,70,3A,09,70,FE,3C,C2,8B,7D,3A,08,70,47,3A,00,70,3C,C6,C0,B8,CA,
6B
430 DATA 7D,3C,B8,CA,6B,7D,3C,B8,C2,8B,7D,2A,00,70,23,36,48,23,36,49,23,36,54,3E
,01,32,10,70,3A,0F,70,3C,32,0F,70,3E,00,32,05,70,C3,E0,7D,2A,08,70,7E,FE,83,C2,9
C,7D,3E,00,32,05,70,C3,E0,7D,FE,BF,C2,A9,7D,2A,08,70,36,20,C3,94,7D,2A,08,70,36,
23
440 DATA 2A,06,70,ED,5B,08,70,B0,ED,52,7D,B4,C2,E0,7D,3E,2A,12,01,10,27,0B,79,B0
,C2,C3,7D,3E,20,12,3E,00,32,04,70,32,05,70,32,06,70,32,07,70,32,08,70,32,09,70
450 DATA 3A,10,70,FE,01,C2,F6,7D,01,10,27,0B,79,B0,C2,EB,7D,3E,00,32,10,70
460 DATA 2A,0A,70,3A,0E,70,FE,0A,C2,09,7E,36,31,23,36,30,C3,0C,7E,C6,30,77,2A,0C
,70,3A,0F,70,FE,0A,C2,1F,7E,36,31,23,36,30,C3,22,7E,C6,30,77,3A,0E,70,FE,0A,C8,3
A,0F,70,FE,0A,C8,01,4C,1D,0B,79,B0,C2,31,7E,3A,40,38,FE,04,C8,C3,00,7C,XX
```


PROGRAMS

PET Mini-animate

by M Whitworth

Mini-Animate will enable users of 'new ROM' PETs to produce animated graphics sequences of a good length. Instructions are not included in the program to save memory space. What the program actually does is to allow frames to be displayed in rapid sequence on the screen, thereby giving an animated effect. The author has included comprehensive editing facilities to allow easy handling of the frames. These are as follows:

Adjust: allows the current frame to be edited using the cursor controls.
Up/down/left/right: moves the current frame in the specified direction.
Clear: clears the frame on the screen.
Save: saves the displayed frame in memory.
Frame: recalls a specified frame.
Speed: determines the speed of animation.
Animate: animates the sequence.
Fetch/Dump: saves/loads an animated sequence.
New: 'NEWs' the sequence.

The chequered graphics character should not be used in any of the frames as it is used as a control character. Control mode is activated by hitting 'return' which also used to exit the Adjust mode. The control mode is advanced by pressing the space bar.

It is advisable to save the two programs in succession as the first automatically loads the second and also sets the memory pointers for it. Mini-Animate will need to be adjusted for 'old ROM' PETS by changing the POKE locations in line 130 of the first part of the program.

Part 1:

```
100 A=2815
110 A=A+1 POKEA,32:IFA<8192THEN110
120 FORA=826TO1010:READS:POKEA,S:NEXT
130 CLR:POKE42,168:POKE43,9:POKE44,168:POKE45,9:POKE46,168:POKE47,9
140 LOAD"MINI-ANIMATE",1
200 DATA169,81,141,108,3,169
210 DATA128,141,109,3,169,0
220 DATA141,142,3,169,11,141
230 DATA143,3,174,247,3,240
240 DATA22,173,142,3,24,109
250 DATA248,3,141,142,3,173
260 DATA143,3,109,249,3,141
270 DATA143,3,202,208,234,160
280 DATA0,185,201,128,201,102
290 DATA208,27,192,0,208,4
300 DATA238,247,3,96,173,108
310 DATA3,24,105,40,141,108
320 DATA3,173,109,3,105,0
330 DATA141,109,3,208,220,141
340 DATA108,11,238,142,3,208
350 DATA3,238,143,3,200,208
360 DATA208,169,81,133,0,169
370 DATA128,133,1,169,0,141
380 DATA203,3,169,11,141,204
390 DATA3,174,247,3,240,22
400 DATA173,203,3,24,109,248
410 DATA3,141,203,3,173,204
420 DATA3,109,249,3,141,204
430 DATA3,202,208,234,160,0
440 DATA173,9,11,238,203,3
450 DATA208,3,238,204,3,145
460 DATA0,200,204,246,3,208
470 DATA237,165,0,24,105,40
480 DATA133,0,165,1,105,0
490 DATA133,1,160,0,177,0
500 DATA201,102,208,216,96
```

Part2:

```
100 INPUT"JFRAME SIZE";V,H:IFV>20ORH>20THEN 100
110 POKE1017,V*H/256:POKE1016,V*H-PEEK(1017)*256:POKE1014,H:POKE1015,0
120 FORA=1TOH+2:PRINT"▒";:NEXT:PRINT:FORA=1TOV:PRINT"▒";SPC(H);"▒":NEXT
130 FORA=1TOH+2:PRINT"▒";:NEXT
135 A=0:PRINT"S■";TAB(23);"FRAME #";PEEK(1015);"▮  "
140 PRINT"SQQ";TAB(23);R$;"   "
145 GETR$:IFR$<>" "ANDR$<>CHR$(13)THEN145
150 IFR$=" "THENREADR$:A=A+1:IFR$="*"THEN RESTORE:READR$:A=1
155 IFR$=CHR$(13)THEN165
160 GOTO140
165 ON AGOSUB240,450,270,280,290,300,390,410,490,400,220,200,480
170 R$="            ":RESTORE:IFA=-1THENPRINT"J":GOTO110
175 GOTO135
180 DATAADJUST,SAVE,UP,DOWN,LEFT,RIGHT,CLEAR,FRAME,SPEED,ANIMATE,FETCH
190 DATADUMP,NEW,*
200 INPUT"JPROG. NAME";R$:INPUT"DEVICE #";D:OPEN 5,D,1,R$:PRINT#5,V
205 PRINT#5,H:PRINT#5,MF
210 FORA=2816TOMF*V*H+2816:PRINT#5,PEEK(A):NEXT:CLOSE 5:A=-1:RETURN
220 INPUT"JPROG. NAME";R$:INPUT"DEVICE #";D:OPEN 5,D,0,R$:INPUT#5,V
225 INPUT#5,H:INPUT#5,MF
230 FORA=2816TOMF*V*H+2816:INPUT#5,D:POKEA,D:NEXT:CLOSE5:A=-1:RETURN
240 PRINT"SQQ▮";:B=32849:A=PEEK(B)
245 GETR$:IFR$=CHR$(148)ORR$=CHR$(20)ORR$="J"ORR$="S"THEN245
250 IFR$=CHR$(13)THENPOKEB,A:RETURN
255 POKEB,A:PRINTR$;:B=PEEK(196)+PEEK(198)+PEEK(197)*256:A=PEEK(B)
260 IFA=102THENPOKEB,A:PRINT:PRINT"▮";:IFPEEK(B+1)=102THEN240
265 POKEB,86:GOTO245
270 A=32849:B=40:D=41-V*40:GOTO310
280 A=32809+V*40:B=-40:D=V*40-39:GOTO310
290 A=32849:B=1:D=41-H:GOTO310
300 A=32848+H:B=-1:D=39+H
310 IFPEEK(A+B)=102THENPOKEA,32:A=A+D:IFPEEK(A)=102THENRETURN
320 C=PEEK(A+B):POKEA,C:A=A+B:GOTO310
390 FORA=1TOH:FORD=1TOV:POKE32808+D*40+A,32:NEXT:NEXT:RETURN
400 POKE1015,0
405 FORA=0TOMF:SYS(923):B=PEEK(1015):FORD=1TOSP:NEXTD:POKE1015,B+1:NEXTA
407 POKE1015,0:RETURN
410 PRINT"SQQ";TAB(28);:INPUTA:PRINT"J";TAB(28);"
420 IFA>-1ANDA<=MFTHENPOKE1015,A:SYS(923)
430 RETURN
450 IFPEEK(1015)=MF+1THENMF=MF+1
460 SYS(826):RETURN
480 MF=0:POKE1015,0:RETURN
490 PRINT"SQQ";TAB(28);:INPUTSP:PRINT"J";TAB(28);"               ":RETURN
```

VIC-20 Trailblazer

by Bob Chappell

This is a colourful, fast-moving game in which you and the computer both draw different-coloured trails about the screen for as long as possible without colliding with part of either trail or going back on your own trail. The game is played in rounds, the first (either VIC or user) to win 10 rounds being the overall winner.

Trailblazer is a game which needs fast thinking and quick reactions. The computer makes a worthy adversary even at the slowest trail speed (the program includes speed options). Full playing instructions are presented at the beginning of the game.

```
2 GOTO300
10 PRINT"TRAILBLAZER"
20 H=7940:V=7958:GOSUB80:GOSUB90
25 A=FNA(4):GOSUB99:H=H+M:LH=A
27 A=FNA(4):GOSUB99:V=V+M:GOSUB80:GOSUB90:LV=A
30 FORJ=1TO4:A(J)=0:NEXT:GETA$:IFA$=""THENA=LH:GOTO32
31 A=ASC(A$)-132:IFA<1ORA>4THENA=LH
32 GOSUB99:IFH+M<L1GOTO35
33 H=H-L3:IFPEEK(H+M)=46GOTO45
34 GOTO200
35 IFH+M>L2GOTO40
38 H=H+L3:IFPEEK(H+M)=46GOTO45
39 GOTO200
40 IFPEEK(H+M)<>46GOTO200
45 LH=A:H=H+M:GOSUB80
46 A=LV:IFA=2ORA=3THENFV=A
47 J=FNA(24):IFJ>4GOTO51
48 A=J:IFFV=2ANDA=3THENA=FV:GOTO51
50 IFFV=3ANDA=2THENA=FV
51 GOSUB99:IFV+M<L1GOTO54
52 IFPEEK(V-L3+M)=46THENV=V-L3:GOTO62
53 GOTO61
54 IFV+M>L2GOTO57
55 IFPEEK(V+L3+M)=46THENV=V+L3:GOTO62
56 GOTO61
57 J=V+M+M:IFJ>L2ANDJ<L1ANDPEEK(J)<>46GOTO63
61 IFPEEK(V+M)<>46GOTO63
62 LV=A:V=V+M:GOSUB90:GOTO30
63 FORJ=1TO4
64 A=FNA(4):IFA(A)=1GOTO64
65 A(A)=1:GOSUB99:IFPEEK(V+M)=46GOTO62
70 NEXT:GOTO205
80 POKEH,160:POKEH+P,6
85 IFY3=1THENFORJ=1TO2:K=FNA(L3)+L2:POKEK,46:POKEK+P,2:NEXT
87 GOTO95
90 POKEV,102:POKEV+P,5
95 POKEN,NN:FORJ=1TOY/10:NEXT:POKEN,0:FORJ=1TOY:NEXT:IFY1=1THENY=Y-5
97 RETURN
```

```
99 ONAGOTO101,102,103,104
101 M=U:RETURN
102 M=L:RETURN
103 M=R:RETURN
104 M=D:RETURN
200 PRINT"■■":A=H+M:K1=160:K2=6:GOSUB280:PRINT"■■■■YOU ■";:VS=VS+1:GOTO210
205 PRINT"■■":A=V+M:K1=102:K2=5:GOSUB280:PRINT"■■■■VIC ■";:HS=HS+1
210 PRINT" CRASHED!":PRINT"■■VIC";VS;"■■YOU";HS
215 IFVS=10THENPRINT"■■■■■■VIC WINS!■■":GOTO500
220 IFHS=10THENPRINT"■■■■■■YOU WIN!■■":GOTO500
230 PRINT"■■■PRESS F,S,M G OR R
260 POKE198,0:PRINT"■":GOTO430
280 IFA>L1-1THENA=A-22
282 IFA<L2+1THENA=A+22
285 FORJ=1TO30:POKEA,42:POKEA+P,1:POKEN+2,J+190:FORK=1TO30:NEXT
287 POKEA,K1:POKEA+P,K2:POKEN+2,0
290 FORK=1TO30:NEXTK,J:RETURN
300 N=36874:POKEN+5,29:POKEN+4,15
301 DEFFNA(A)=INT(A*RND(1)+1):L=-1:R=1:U=-22:D=22:P=30720:A=RND(TI)
305 L1=8186:L2=7679:L3=506
310 PRINT"■■■■■TRAILBLAZER
320 PRINT"■ YOU ARE ■  ■":PRINT"■ VIC IS  ■■■■"
330 PRINT"■ YOU MUST NOT CROSS":PRINT" YOUR OWN OR VIC'S":PRINT" TRAIL.
340 PRINT"■ TO CHANGE DIRECTION,":PRINT" HIT■    F1■■■↑":PRINT"  F3<   >F5
350 PRINT"          V■■F7":PRINT"■ FIRST ONE TO WIN 10   ROUNDS IS THE WINNER.
360 PRINT"  ■PRESS ANY KEY NOW■
370 GETA$:IFA$=""THEN370
380 PRINT"■■ TO START A NEW":PRINT" ROUND PRESS:-
400 PRINT"■■ F FOR FAST SPEED
410 PRINT"■ M FOR MEDIUM SPEED
420 PRINT"■ S FOR SLOW SPEED
425 PRINT"■ R FOR RISING SPEED
426 PRINT"■ G FOR FAST + GAPS
427 PRINT"■■ ■PLEASE PRESS ONE OF":PRINT" ■■THESE KEYS NOW.
430 NN=150:Y=0:Y1=0:Y3=0:T1=TI
440 IFTI-T1>999GOTO10
450 GETA$:IFA$=""GOTO440
460 IFA$="M"THENY=100:NN=170
470 IFA$="S"THENY=300:NN=190
480 IFA$="R"THENY1=1:Y=300
485 IFA$="G"THENY3=1
490 GOTO10
500 FORJ=1TO40:CP=PEEK(N+5):CR=CPAND8:POKE36876,(CR=0)*30+160
502 IFCR=0THENCP=CP+8:GOTO506
504 CP=CP-8
506 POKEN+5,CP:FORK=1TO50:NEXTK:POKE36876,0:NEXTJ:FORJ=1TO4000:NEXT
510 POKE198,0:POKEN+5,27:PRINT"■■■■■■ PLAY AGAIN (Y OR N)
520 GETA$:IFA$=""THEN520
530 IFLEFT$(A$,1)="Y"THENRUN
540 END                    LIST
```

ZX81 Book Index.

by Ian Andrews

Useful applications for the ZX81 seem to be cropping up more and more in the programs mailbag. This one needs 16k and enables you to make an alphabetical index of book titles and their authors — although it could be used for indexing anything from the rest of your software to your record collection. Naturally you need a cassette recorder (unless you plan on leaving your ZX81 switched on 24 hours a day!) and a printer helps but is not necessary.

You have to specify at the beginning how many entries you wish to make and the maximum length for an entry. If you exceed either specification the computer lets you know and it will also tell you if it has insufficient memory to cope with the amount of data you wish to store. It is a pretty friendly program on the whole so using it is straightforward. Memory space could be saved by changing the PRINT ATs to plain PRINTs as they are purely cosmetic.

```
 30 CLS
 40 CLEAR
 50 FAST
 60 POKE 16510,0
 70 PRINT TAB 10;"I N D E X"
 80 PRINT
 90 PRINT "ENTER TITLE"
100 INPUT A$
110 CLS
120 PRINT "ENTER AUTHOR"
130 INPUT B$
140 CLS
150 PRINT "ENTER AN ESTIMATE
    OF THE NUMBER OF ENTRIES
    YOU WILL MAKE."
160 PRINT AT 15,0;"NB-BE
    GENEROUS AS YOU CANNOT
    EXTEND THE NUMBER LATER
    ON."
170 INPUT N
180 CLS
190 PRINT "ENTER MAX LENGTH
    OF ENTRY."
200 INPUT M
210 DIM L$(N+1,M+3)
220 CLS
230 LET D=10000-3*N
240 IF D<M*N THEN PRINT AT
    18,0;"RE-ENTER. MAXIMUM
    NUMBER OF ENTRIES WITH"
    ;M;"CHARACTERS IS";D/M
250 IF D<M*N THEN GOTO 170
260 PRINT "GET READY TO
    ENTER WORDS ONE BY ONE
    (UP TO**";M;"** CHARAC
    TERS IN LENGTH)."
270 PAUSE 100
280 PRINT
290 CLS
300 FOR I=1 TO N
310 PRINT "ENTRY";I;"TEXT"
320 IF I=N-10 THEN PRINT AT
    15,0;"WARNING-MEMORYSHORT"
330 INPUT X$
340 IF LEN X$>M THEN PRINT
    "ABBREVIATE ENTRY"
350 FOR F=1 TO M
360 PRINT AT 4,5;X$;AT 5,4
    +F;"-"
370 NEXT F
```

PROGRAMS

```
380 IF LEN X$>M THEN GOTO
    330
390 LET L$(I)( TO M)=X$
400 LET X$=" "
410 IF CODE L$(I)=227 THEN
    LET N=I
420 IF CODE L$(I)=227 THEN
    GOTO 510
430 CLS
440 PRINT "ENTRY";I;TAB 12;
    L$(I)( TO M);AT 15,0;
    "PAGE NUMBER?"
450 INPUT L$(I)(M+1 TO )
460 IF CODE L$(I)(M+1 TO )
    >0 AND CODE L$(I)(M+1 TO
    )<28 OR CODE L$(I)(M+1
    TO )>37 THEN PRINT AT 18,
    5;"ERROR"
470 IF CODE L$(I)(M+1 TO )>0
    AND CODE L$(I)(M+1 TO )<
    28 OR CODE L$(I)(M+1 TO )
    >37 THEN GOTO 450
480 CLS
490 NEXT I
500 REM ORDER
510 LET L$(I)=L$(I)( TO M)+
    L$(I)(M+1 TO M+3)
520 FOR K=1 TO N-1
530 FOR J=1 TO N-K
540 IF L$(J)<=L$(J+1) THEN
    GOTO 580
550 LET T$=L$(J)
560 LET L$(J)=L$(J+1)
570 LET L$(J+1)=T$
580 NEXT J
590 NEXT K
600 CLS
610 REM PRINT
620 SLOW
630 PRINT AT 15,0;"""";A$;
    """","BY;B$
640 PAUSE 100
650 FOR I=1 TO N
660 SCROLL
670 PRINT L$(I)( TO M);TAB M
    +7;L$(I)(M+1 TO )
680 NEXT I
690 PRINT
700 PRINT "INDEX ENDS"
710 PRINT
720 PRINT "DO YOU WANT TO SAVE
    IT (S),PRINT IT (P), NEW
    INDEX (A) OR END IT (E)?"
730 INPUT Q$
740 IF Q$="S" THEN GOTO 780
750 IF Q$="P" THEN GOTO 820
760 IF Q$="A" THEN GOTO 30
770 STOP
780 SAVE "BOOK INDEX"
790 CLS
800 GOTO 720
810 CLS
820 LPRINT """";A$;"""",BY";B$
830 LPRINT
840 LPRINT
850 LPRINT TAB 10;"I N D E X"
860 LPRINT
870 LPRINT
880 FOR I=1 TO N
890 LPRINT L$(I)( TO M); TAB
    M+7; L$(I)(M+1 TO )
900 NEXT I
910 LPRINT
920 LPRINT "INDEX ENDS"
930 LPRINT
940 GOTO 720
```

Weebug monitor

by B. Lavery

This "WEEBUG" monitor for the Level 2 TRS-80 and System 80 microcomputers will enable you to perform the following functions:

1. Record a System tape.
2. View or edit any byte in memory.
3. Jump to any memory address and start executing the code there.

WEEBUG occupies less than 500 bytes, and loads from tape in about 8 seconds. WEEBUG calls many routines present in the ROM memory of the TRS-80 or System 80, which accounts for its very short length.

LOADING WEEBUG FROM SYSTEM TAPE:

1. Type "SYSTEM" and (ENTER).
2. Prepare cassette to play.
3. Answer the *? prompt with "WEEBUG" and (ENTER).
4. When cassette stops, type "/" and (ENTER).

You should now be in WEEBUG.

USING WEEBUG

The command mode prompt in WEEBUG is the sign "≠." When the "≠" prompt is showing, the following commands will be accepted:

(CLEAR) on the TRS-80: Clear the screen.

CONTROL C, i.e. (SHIFT) (CTRL) C, on the System 80: Clear the screen.

M : Look at the contents of a particular memory location.

P : Punch (record) a system program onto cassette tape.

J : Jump to a particular memory location and execute.

While WEEBUG is in M (Memory) or P (Punch) or J (Jump) mode, it will only accept the proper type of keyboard input, as described below.

In M, P and J modes, you will encounter another prompt, the ":" character, which is the decision prompt. You can break back to command mode only at the decision prompt.

HEXADECIMAL NOTATION

The WEEBUG monitor uses hex notation addresses and memory contents. If you are unsure how to convert between hex and decimal numbers, refer to the information given in the manuals for the TRS-80 Level 2 or System 80.

MEMORY EDIT MODE

The format for entering M mode is:

M xxxx:

where the xxxx is the hexadecimal address you want to see. WEEBUG immediately supplies the contents (in hex) of xxxx, and gives you a decision prompt:

#M 4179 C3 :

The contents of address 4179H (the H meaning hex) is C3H. Translated into decimal numbers, the contents of address 16761 is 195.

At the ":" prompt, the following choices are available:

(BREAK) : Go back to command mode.
(ENTER) : Advance to next address.
(BACKSPACE) : Go to previous (lower) address.
(SPACE BAR) : Prepare to alter the contents of this address.

To alter the contents of the current address, press (SPACE BAR) and type two hex letters of the new contents:

```
#M 4179 C3 : You type: (ENTER)
417A 2D : 00           (SPACE
                        (BAR)  00
417B 01 : 7C           (SPACE
                        (BAR)  7C
417C C3 :              (BREAK)
#
```

The example just given allows the computer to acknowledge the command "OPEN" in Basic, not with L3 ERROR, but with a jump into WEEBUG!

JUMP MODE

The format for J MODE is:

J xxxx

where xxxx is the hex address you want to go to. After the address is entered, you again get the decision prompt:

(BREAK) : Abort the jump. Go back to WEEBUG command mode.
(ENTER) : Jump.

Be sure that you know where you are jumping to. In particular, if you are experimenting with a bit of new machine language code, make sure it finishes with a jump to somewhere suitable, e.g. back to WEEBUG at address 7C00H.

PUNCH MODE

The format for using P MODE is:

#P xxxx yyyy zzzz NNNNNN

where xxxx is the start address in hex of the machine language code that you want to record to tape, yyyy is the end address of the program, and zzzz is the entry address. The entry address is where the program starts execution, after the tape loads and you answer with " / " (ENTER). NNNNNN is a 6 character name that you assign to the program.

For example, WEEBUG itself resides in memory from 7C00H to 7DD5H. It begins execution at 7C00H. (Note that some programs enter at a different point from the start or first byte of the program.) To make a backup copy of WEEBUG under the new name of MINMON, type:

#P 7C00 7DD5 7C00 MINMON :
You type: (ENTER)

Note the decision prompt again. It gives you two choices:

(BREAK) : Return immediately to command mode, i.e. abort.
(ENTER) : Record program to tape. Don't forget to have the tape ready.

The program returns you to command mode when the recording is finished.

Sometimes it can be a problem identifying the correct start, end and entry addresses for a program. If you start the computer up from cold, most RAM memory will contain either 00H or FFH. Load in WEEBUG but don't run it. (BREAK) back to Basic and find PEEK (16607) + PEEK(16608) * 256. This is the entry address of a system tape, in this case of WEEBUG. For WEEBUG this entry address should be 31744 (decimal). Keep a note of it. Now load in the wanted program. Find its entry address the same way. Now, using the address for entry to WEEBUG, go to WEEBUG as follows:

```
SYSTEM
#? / 31744          (WEEBUG's entry)
```

Search around using M MODE to try to find where your program starts and ends. You should now have the desired start, end and entry addresses.

EXAMPLE 1 : CHANGE WEEBUG PROMPT

WEEBUG's M MODE can be used to change itself! The M character in M MODE is at memory address 7C1BH. The ASCII code for the M is 4DH. Let's change it to E for EDIT. The ASCII for E is 45H:

```
#M 7C1B 4D : 45
7C1D CA :
```

Now, WEEBUG will respond only to E to edit memory. Let's change the command mode prompt to a question mark:

```
#E 7C13 23 : 3F
7C14 CD
?
```

If you like, you can punch yourself a copy of the revised WEEBUG.

EXAMPLE 2 : RETURN TO BASIC

To return from WEEBUG to Basic, use the J instruction to go to address 06CCH. This is a routine in the ROM memory that returns you properly from SYSTEM mode on the computer to Basic mode:

```
#J 06CC :
READY
>
```

You are now back in Basic. If you want to go right back to "MEMORY SIZE?" jump to address 0000H.

EXAMPLE 3 : RESERVE MEMORY BELOW BASIC

Sometimes you may want to use a machine language program that is in low memory, and still put in a Basic program as well. In this case, you must reserve some low memory, and load in the Basic above it. Level 2 Basic programs usually start at 42E9H. There are several pointers in reserved RAM that tell Basic to start there. The main one is at memory location 40A4H-40A5H. The "least significant byte" (lsb) of the bottom address of Basic is in 40A4H, and the "most significant byte" (msb) is in 40A5H. Also, the byte just before where Basic starts must be a zero. You can change all these using the M mode.

For example, to set the bottom of Basic at 44D0H, change the pointer at 40A4H using WEEBUG, zero the byte before new start of Basic, return to Basic, and then type "NEW". "NEW" corrects all the other pointers associated with the bottom of Basic, relative to the one at 40A4H:

```
#M 40A4 E9 : DO
40A5 42 : 44
40A6 0B :
#M 44CF FF : 00
44 D0 FF :
#J 06CC :
READY
>NEW
READY
>
```

METHOD 1:

If you have an assembler program (for example, Microsoft Editor Assembler Plus), then type the assembler source code given in listing 2. You may choose another start address rather than the 7C00H given in the listing. Assemble and produce your object tape, as described in the instructions for the assembler program.

METHOD 2:

If you already have another monitor/debug program (for example, T-BUG, then type in the object code given at the left of listing 2, starting at address 7C00H. Then jump to address 7C00H (31744 in decimal) and use WEEBUG to punch a copy of itself:

P 7C00 7DD5 7C00 WEEBUG

METHOD 3:

Type in the Basic program given in listing 1. This will create WEEBUG for you, and will jump to WEEBUG. Again, use WEEBUG to make a copy of itself.

Listing 1

```
1 '*************************************************************
5 CLS : PRINT"WEEBUG LOADING"
10 DEFINTJ-K
20 J=31744
30 READ K : IF K<0 THEN GOTO 80
40 POKE J,K
50 J=J+1
60 GOTO30
80 POKE16526,0 : POKE16527,124
90 X=USR(X)
100 DATA  205,201,1,33,203,124,175,50,156,64,205,167,40,62,13
110 DATA  205,51,0,62,35,205,51,0,205,73,0,254,77,202,228
120 DATA  124,254,74,40,15,254,80,202,45,125,254,31,40,212,254
130 DATA  3,40,208,24,219,205,51,0,205,162,124,205,118,124,205
140 DATA  168,124,205,51,0,233,241,205,43,0,245,214,48,250,66
150 DATA  124,254,10,250,92,124,254,17,250,66,124,214,7,185,242
160 DATA  66,124,71,241,205,51,0,201,14,16,205,67,124,203,32
170 DATA  203,32,203,32,203,32,197,205,67,124,241,176,201,205,98
180 DATA  124,103,205,98,124,111,205,162,124,201,198,48,254,58,250
190 DATA  139,124,198,7,205,51,0,201,245,203,63,203,63,203,63
200 DATA  203,63,205,130,124,241,230,15,205,130,124,201,62,32,205
210 DATA  51,0,201,213,62,58,205,51,0,205,73,0,209,254,1
220 DATA  40,5,254,13,32,238,201,241,195,13,124,213,205,118,124
230 DATA  125,209,18,19,124,18,19,201,87,69,69,66,85,71,13
240 DATA  40,67,41,32,83,79,70,84,87,85,82,88,32,49,57
250 DATA  56,49,0,205,51,0,205,162,124,205,118,124,126,205,143
260 DATA  124,205,162,124,62,58,205,51,0,205,73,0,254,1,202
270 DATA  13,124,254,13,40,21,254,8,40,15,254,32,40,2,24
280 DATA  234,205,162,124,205,98,124,119,24,2,43,43,35,62,13
290 DATA  205,51,0,124,205,143,124,125,205,143,124,205,162,124,24
300 DATA  192,205,51,0,205,162,124,17,210,124,6,3,197,205,191
310 DATA  124,193,16,249,14,36,6,6,197,213,205,67,124,209,18
320 DATA  193,19,16,245,205,162,124,205,168,124,42,212,124,237,91
330 DATA  210,124,175,50,209,124,237,82,218,13,124,35,213,229,205
340 DATA  134,125,209,225,122,183,40,8,21,14,0,205,159,125,24
350 DATA  244,123,183,40,4,79,205,159,125,205,194,125,195,13,124
360 DATA  175,205,18,2,205,135,2,62,85,205,100,2,33,216,124
370 DATA  6,6,126,205,100,2,35,16,249,201,213,62,60,205,100
380 DATA  2,121,205,100,2,125,87,205,100,2,124,205,100,2,130
390 DATA  87,126,205,100,2,35,13,32,246,130,205,100,2,209,201
400 DATA  62,120,205,100,2,42,214,124,125,205,100,2,124,205,100
410 DATA  2,205,248,1,201,-100
420 '*************************************************************
430 'WEEBUG: (C) SOFTWURX 1981, CANBERRA CITY, AUSTRALIA.
440 '*************************************************************
```

Listing 2

```
               00100 ;    **********************************************
               00110 ;    **                 WEEBUG                   **
               00120 ;    **********************************************
               00130 ;    **         ASSEMBLER SOURCE CODE            **
               00140 ;    **********************************************
               00150 ;
               00160 ;    A SHORT TAPE-BASED MONITOR FOR
               00170 ;    TRS-80 LEVEL 2 AND PMC-80 (SYSTEM-80).
               00180 ;
               00190 ;    COMMANDS AVAILABLE ARE:
               00200 ;            M   EDIT MEMORY
               00210 ;            P   PUNCH A SYSTEM TAPE
               00220 ;            J   JUMP TO A MEMORY LOC'N & EXECUTE
               00230 ;
               00240 ;
               00250 ;    COPYRIGHT (C)   SOFTWURX      DEC 1981.
               00260 ;    AUTHOR: BRIAN LAVERY
               00270 ;            P.O. BOX 895, CANBERRA CITY, 2601
               00280 ;            AUSTRALIA.
               00290 ;
               00300 ;            # IS COMMAND MODE PROMPT.
               00310 ;            : IS DECISION PROMPT.
               00320 ;
7C00           00330          ORG     7C00H          ;EQUALS 31744 DECIMAL
               00340 ;
7C00 CDC901    00350 ENTRY    CALL    01C9H            ;CLS
7C03 21CB7C    00360          LD      HL,IDENT         ;POINT TO WEEBUG IDENT
7C06 AF        00370          XOR     A
7C07 329C40    00380          LD      (409CH),A
7C0A CDA728    00390          CALL    28A7H            ;OUTPUT IDENT STRING
7C0D 3E0D      00400 PROMPT   LD      A,0DH
7C0F CD3300    00410          CALL    33H              ;LINE FEED
7C12 3E23      00420          LD      A,23H            ;PROMPT CHARACTER (#)
7C14 CD3300    00430          CALL    33H              ;CHARACTER TO SCREEN
7C17 CD4900    00440          CALL    49H              ;GET CHR (ROM ROUTINE)
7C1A FE4D      00450          CP      'M'              ;
7C1C CAE47C    00460          JP      Z,MEMEDT         ;
7C1F FE4A      00470          CP      'J'              ;
7C21 280F      00480          JR      Z,JUMP           ;   CHOOSE M,J,P
7C23 FE50      00490          CP      'P'              ;
7C25 CA2D7D    00500          JP      Z,PUNCH          ;
7C28 FE1F      00510          CP      1FH              ;CLEAR SCREEN
7C2A 28D4      00520          JR      Z,ENTRY
7C2C FE03      00530          CP      3                ;(CTRL) C  ?
7C2E 28D0      00540          JR      Z,ENTRY          ;ALSO CLEAR SCREEN
7C30 18DB      00550          JR      PROMPT           ;TRY AGAIN
               00560 ;
               00570 ;##################################################
               00580 ;
               00590 ;JUMP ROUTINE
7C32 CD3300    00600 JUMP     CALL    33H              ;PRINT 'J'
7C35 CDA27C    00610          CALL    SPACE
7C38 CD767C    00620          CALL    HEXIN4           ;GET ADDRESS
7C3B CDA87C    00630          CALL    ACCREJ           ;BREAK?
7C3E CD3300    00640 GOTO     CALL    33H              ;LINE FEED TO LOOK NEAT
7C41 E9        00650          JP      (HL)             ;EXIT TO NEW ADDRESS
               00660 ;
               00670 ;##################################################
               00680 ;
               00690 ;AN ASSORTMENT OF SUBROUTINES
               00700 ;
               00710 ;SUBROUTINE TO INPUT 1 DIGIT (0-9,A-F) OF HEX
7C42 F1        00720 PRE      POP     AF               ;POP NEEDED FOR LOOPING
7C43 CD2B00    00730 CHARIN   CALL    2BH            ;KEYBRD INPUT (ROM ROUTINE)
7C46 F5        00740          PUSH    AF
7C47 D630      00750          SUB     30H              ;MOVE DOWN 30 HEX
7C49 FA427C    00760          JP      M,PRE            ;REJECT
7C4C FE0A      00770          CP      0AH              ;NUMERIC?
7C4E FA5C7C    00780          JP      M,OUT
7C51 FE11      00790          CP      11H
7C53 FA427C    00800          JP      M,PRE            ;REJECT
7C56 D607      00810          SUB     7                ;ALFA: MOVE DOWN 7 MORE
7C58 B9        00820          CP      C
7C59 F2427C    00830          JP      P,PRE            ;REJECT
7C5C 47        00840 OUT      LD      B,A
7C5D F1        00850          POP     AF
7C5E CD3300    00860          CALL    33H              ;PRINT CHR
7C61 C9        00870          RET                      ;LEAVE WITH CHR IN B REG
               00880 ;
               00890 ;SUBROUTINE TO INPUT 2 HEX DIGITS
7C62 0E10      00900 HEXIN2   LD      C,10H            ;ACCEPT ALFA ONLY UP TO F
7C64 CD437C    00910          CALL    CHARIN           ;FIRST DIGIT
7C67 CB20      00920          SLA     B                ;
7C69 CB20      00930          SLA     B                ;
7C6B CB20      00940          SLA     B                ;
7C6D CB20      00950          SLA     B                ;SHIFT INTO LEFT 4 BITS
7C6F C5        00960          PUSH    BC               ;AND SAVE
7C70 CD437C    00970          CALL    CHARIN           ;SECOND DIGIT
7C73 F1        00980          POP     AF
7C74 B0        00990          OR      B                ;2ND INTO RIGHT 4 BITS
7C75 C9        01000          RET                      ;1 BYTE HEX IN A REG
               01010 ;
               01020 ;SUBROUTINE TO INPUT 4 HEX DIGITS (2 BYTES)
7C76 CD627C    01030 HEXIN4   CALL    HEXIN2           ;FIRST 2
7C79 67        01040          LD      H,A
```

PROGRAMS

```
7C7A CD627C   01050          CALL    HEXIN2          ;NEXT 2
7C7D 6F       01060          LD      L,A
7C7E CDA27C   01070          CALL    SPACE
7C81 C9       01080          RET                     ;2 BYTES HEX IN HL REG
              01090 ;
              01100 ;SUBROUTINE TO PRINT 1 HEX DIGIT (0-9, A-F)
7C82 C630     01110 HXOUT1   ADD     A,30H
7C84 FE3A     01120          CP      3AH             ;ALFA OR NUM?
7C86 FA8B7C   01130          JP      M,PRINT         ;IF NUM, PRINT IT
7C89 C607     01140          ADD     A,7             ;ALFA
7C8B CD3300   01150 PRINT    CALL    33H
7C8E C9       01160          RET
              01170 ;
              01180 ;SUBROUTINE TO PRINT 2 HEX DIGITS
7C8F F5       01190 HXOUT2   PUSH    AF
7C90 CB3F     01200          SRL     A          ;
7C92 CB3F     01210          SRL     A          ;
7C94 CB3F     01220          SRL     A          ;PUT LEFT 4 BITS INTO
7C96 CB3F     01230          SRL     A          ;      RIGHT 4.
7C98 CD827C   01240          CALL    HXOUT1     ;PRINT 1ST CHR
7C9B F1       01250          POP     AF
7C9C E60F     01260          AND     OFH        ;NOW LOOK AT ONLY RIGHT 4
7C9E CDB27C   01270          CALL    HXOUT1     ;2ND CHR
7CA1 C9       01280          RET
              01290 ;
              01300 ;SUBROUTINE TO PRINT A BLANK SPACE
7CA2 3E20     01310 SPACE    LD      A,20H
7CA4 CD3300   01320          CALL    33H
7CA7 C9       01330          RET
              01340 ;
              01350 ;SUBROUTINE TO GIVE OPPORTUNITY TO "BREAK"
              01360 ;BACK TO WEEBUG COMMAND MODE PROMPT.
7CA8 D5       01370 ACCREJ   PUSH    DE
7CA9 3E3A     01380          LD      A,3AH      ;OECISION PROMPT (:)
7CAB CO3300   01390          CALL    33H
7CAE CD490D   01400          CALL    49H        ;WAIT FOR KEY PRESS
7CB1 D1       01410          POP     DE
7CB2 FE01     01420          CP      1          ;BREAK?
7CB4 2805     01430          JR      2,REJ      ;YES - GET READY TO GO
7CB6 FEOD     01440          CP      ODH        ;ACCEPT?
7CB8 20EE     01450          JR      NZ,ACCREJ
7CBA C9       01460 ACC      RET
7CBB F1       01470 REJ      POP     AF        ;DUMMY POP TO CORRECT STACK
7CBC C30D7C   01480          JP      PROMPT
              01490 :
              01500 ;SUBROUTINE TO GET 4 DIGIT ADDRESS & STORE IT
7CBF D5       01510 ADDRIN   PUSH    DE         ;ENTER WITH DE POINTING
              01520 ;                            TO STORAGE PLACE.
7CC0 CD767C   01530          CALL    HEXIN4     ;GET ADDRESS IN HL REG
7CC3 7D       01540          LD      A,L
7CC4 D1       01550          PDP     DE
7CC5 12       01560          LD      (DE),A     ;STORE LSB
7CC6 13       01570          INC     DE
7CC7 7C       01580          LD      A,H
7CC8 12       01590          LD      (DE),A     ;STORE MSB
7CC9 13       01600          INC     DE
7CCA C9       01610          RET
              01620 ;
              01630 ;WEEBUG IDENTIFICATION MESSAGE
7CCB 57 45 45 01640 IDENT    DEFM    'WEEBUG'
     42 55 47
7CD1 0D       01650 CHT      DEFB    ODH
7CD2 28 43 29 01660 COPYRT   DEFM    '(C) SOFTWURX 1981'
     20 53 4F 46 54 57 55 52 58 20 31 39 38 31
7CE3 00       01661          DEFB    0
              01670 ;
              01680 ;BUFFERS USED DURING PUNCH OPERATIONS:
7CD2          01690 ADBUFF   EQU     COPYRT
7CD8          01700 NAME     EQU     ADBUFF+6
              01710 ;
              01720 ;################################################
              01730 ;
              01740 ;MEMORY EDIT ROUTINE.
              01750 ;TYPE IN 4 DIGIT HEX MEMORY LOCATION.
              01760 ;CURRENT CONTENTS WILL BE DISPLAYED.
              01770 ;THEN DECIDE: ENTER    NO CHANGE. ADVANCE TO NEXT MEMORY
              01780 ;             BREAK    RETURN TO COMMAND PROMPT
              01790 :             BSP(←)   NO CHANGE. BACKSPACE MEMORY
              01800 ;             SPC BAR  ALLOWS MEMORY TO BE ALTEREO
              01810 ;TO ALTER MEMORY CONTENTS, HIT SPACE BAR,
              01820 ;AND ENTER NEW VALUE AS 2 HEX DIGITS.
              01830 ;
7CE4 CD3300   01840 MEMEDT   CALL    33H        ;PRINT 'M'
7CE7 CDA27C   01850          CALL    SPACE
7CEA CD767C   01860          CALL    HEXIN4     ;GET ADDRESS
7CED 7E       01870 REPEAT   LD      A,(HL)     ;WHAT IS AT THIS ADDRESS?
7CEE CD8F7C   01880          CALL    HXOUT2     ;PRINT THE CONTENTS
7CF1 CDA27C   0189D          CALL    SPACE
7CF4 3E3A     01900          LD      A,3AH      ;DECISION PROMPT (:)
7CF6 CD3300   01910          CALL    33H
7CF9 CD4900   01920 SELECT   CALL    49H
7CFC FE01     01930          CP      1          ;BREAK?
7CFE CADD7C   01940          JP      Z,PROMPT
7D01 FEOD     01950          CP      ODH        ;ACCEPT?
7D03 2815     01960          JR      Z,NEXT
7D05 FE08     01970          CP      8H         ;BACKSPACE MEMORY?
7D07 280F     01980          JR      Z,BACK1
7D09 FE20     01990          CP      20H        ;NEW VALUE?
7D0B 2802     02D00          JR      Z,NEW
7D0D 1BEA     02010          JR      SELECT     ;TRY AGAIN
7D0F CDA27C   02020 NEW      CALL    SPACE
7D12 CD627C   0203D          CALL    HEXIN2     ;GET NEW CONTENTS
7D15 77       02040          LD      (HL),A     ;PUT IT AT THAT LOCATION
7D16 1802     02050          JR      NEXT
7D1B 2B       02060 BACK1    DEC     HL         ;BACKSPACE MEM LOCATION
7D19 2B       02070          DEC     HL
7D1A 23       02080 NEXT     INC     HL         ;ADVANCE TO NEXT LOCATION
7D1B 3E0D     02090          LD      A,ODH
7D1D CD3300   D2100          CALL    33H        ;LINE FEED
7D20 7C       02110          LD      A,H        ;
7D21 CD8F7C   02120          CALL    HXOUT2     ;
7D24 7D       02130          LD      A,L        ;PRINT NEXT LOCATION
7D25 CO8F7C   02140          CALL    HXOUT2       ;
7D28 CDA27C   02150          CALL    SPACE
7D2B 18C0     02160          JR      REPEAT
              02170 ;
              02180 ;################################################
```

```
                02190 ;
                02200 ;ROUTINE TO PUNCH A SYSTEM TAPE.
                02210 ;REQUIRES TYPING IN THE (START),  (END),  AND (ENTRY)
                02220 ;MEMORY LOCATIONS OF THE CODE TO BE PUNCHED,
                02230 ;FOLLOWED BY A 6 CHARACTER (ALPHA-NUMERIC) FILENAME.
                02240 ;FILENAME FIRST CHARACTER SHOULD BE ALPHABETIC.
                02250 ;SHORTER NAMES OR PADDING WITH TRAILING BLANKS
                02260 ;NOT AVAILABLE WITH THIS SHORT MONITOR.
7D2D CD3300     02270 PUNCH   CALL    33H            ;PRINT THE 'P'
7D30 CDA27C     02280         CALL    SPACE
7D33 11D27C     02290         LD      DE,ADBUFF      ;PDINT TO BUFFER
7D36 0603       02300 STENEN  LD      B,3            ; (START END & ENTRY)
7D38 C5         D2310 LOOP3   PUSH    BC
7D39 CDBF7C     02320         CALL    ADDRIN         ;INPUT 3 ADDRESSES
7D3C C1         02330         POP     BC
7D30 10F9       02340         DJNZ    LOOP3
7D3F 0E24       02350 PNAME   LD      C,24H          ;ALLOW ALL ALFANUMERIC
7D41 0606       02360         LD      B,6            ;CHARACTERS FOR FILENAME
7D43 C5         02370 LOOP6   PUSH    BC
7D44 D5         02380         PUSH    DE
7D45 CD437C     02390         CALL    CHARIN         ;GET 6 CHARACTERS
7D48 D1         D2400         POP     DE
7D49 12         02410         LD      (DE),A         ;STORE THEM IN BUFFER
7D4A C1         02420         POP     BC
7D4B 13         02430         INC     DE
7D4C 10F5       D2440         DJNZ    LOOP6
7D4E CDA27C     02450         CALL    SPACE
7D51 CDA87C     02460         CALL    ACCREJ         ;BREAK?
7D54 2AD47C     02470         LD      HL,(ADBUFF+2)  ;END
7D57 ED5BD27C   02480         LD      DE,(ADBUFF)    ;START
7D5B AF         02490         XOR     A              ;MAKE SURE NO CARRY
7D5C 32D17C     02500         LD      (CUT),A      ;CUT IDENT: ROOM FOR BUFFER
7D5F ED52       02510         SBC     HL,DE
7D61 DA0D7C     02520         JP      C,PROMPT       ;END BEFORE START!!!
7D64 23         02530         INC     HL
7D65 D5         02540         PUSH    DE             ;START ADDRESS
7D66 E5         02550         PUSH    HL             ;NO OF BYTES
7D67 CD867D     D2560         CALL    PLEADR         ;PUNCH THE LEADER CODE
7D6A D1         02570         POP     DE             ;NO OF BYTES
7D6B E1         025B0         POP     HL             ;START
7D6C 7A         02590 MORBLK  LD      A,D            ;HIGH BYTE OF DE
7D6D B7         02600         DR      A              ;SET FLAGS
7D6E 2808       02610         JR      Z,SHORTB       ;SHORT BLOCK (<256)
7D70 15         02620         DEC     D
7D71 0E00       02630         LD      C,0            ;0=256 BYTES
7D73 CD9F7D     02640         CALL    PBLOCK         ;PUNCH 256 BYTE BLOCK
7D76 18F4       D2650         JR      MORBLK
7D78 7B         02660 SHORTB  LD      A,E
7079 B7         02670         OR      A
7D7A 2804       026B0         JR      Z,TAIL         ;0=0, NOT 256 THIS TIME!
7D7C 4F         02690         LD      C,A            ;C=NO OF BYTES ( <256 ).
7D7D CD9F7D     02700         CALL    PBLOCK
7DB0 CDC27D     02710 TAIL    CALL    PTAIL          ;PONCH TAIL CODE
7D83 C30D7C     D2720         JP      PROMPT
                02730 ;
                02740 ;SUBROUTINE TO START CASSETTE AND RECORD LEADER
7D86 AF         02750 PLEADR  XOR     A
7D87 CD1202     02760         CALL    212H           ;DEFINE CASSETTE 1
7D8A CD8702     02770         CALL    287H           ;PUNCH LEADER & SYNC
7D8D 3E55       02780         LD      A,55H          ;IDENTIFIES SYSTEM TAPE
7D8F CD6402     02790         CALL    264H           ;ROM SUBROUTINE TO
                02800 ;                                RECORD 1 BYTE TO TAPE
7D92 21D87C     D2810         LD      HL,NAME        ;POINT TO STORED NAME
7D95 0606       02820         LD      B,6
7D97 7E         02830 LOO106  LD      A,(HL)
7D98 CD6402     02840         CALL    264H           ;PUNCH NAME TO TAPE
709B 23         02850         INC     HL
7D9C 10F9       02860         DJNZ    LOOP06
7D9E C9         02870         RET
                02880 ;
                02890 ;SUBROUTINE TO PUNCH 1 BLOCK (MAX 256 BYTES) TO TAPE
7D9F D5         02900 PBLOCK  PUSH    DE
7DA0 3E3C       02910         LD      A,3CH          ;BLOCK HEADER BYTE
7DA2 CD6402     02920         CALL    264H
7DA5 79         02930         LD      A,C            ;BLOCK LENGTH
7DA6 CD6402     D2940         CALL    264H
7DA9 7D         02950         LD      A,L            ;HL BLOCK START ADDRESS
7DAA 57         02960         LD      0,A            ;START KEEPING CHECKSUM
7DAB CD6402     02970         CALL    264H
7DAE 7C         02980         LD      A,H
7DAF CD6402     D2990         CALL    264H
7DB2 82         03000 MORBYT  ADD     A,D
7DB3 57         03D10         LD      D,A
7DB4 7E         03020         LD      A,(HL)         ;ACTUAL CODE
7DB5 CD6402     03030         CALL    264H
7DB8 23         03040         INC     HL
7DB9 DD         03050         DEC     C              ;BLOCK LENGTH COUNT
7DBA 20F6       03060         JR      NZ,MORBYT
7DBC B2         03070         ADD     A,D            ;BLOCK CHECKSUM
7DBD CD6402     D3080         CALL    264H
7DC0 D1         0309D         POP     DE
7DC1 C9         03100         RET
                03110 ;
                03120 ;SUBROUTINE TO PUNCH THE LAST PART
7DC2 3E78       03130 PTAIL   LD      A,78H          ;END CHARACTER
7DC4 CD6402     03140         CALL    264H
7DC7 2AD67C     03150         LO      HL,(ADBUFF+4)  ; "/ ENTER" ADDRESS
7DCA 7D         0316D         LD      A,L
7DCB CD6402     03170         CALL    264H
7DCE 7C         03180         LD      A,H
7DCF CD6402     03190         CALL    264H
7DD2 CDF801     0320D         CALL    1F8H           ;TURN OFF CASSETTE
7DD5 C9         0321D LAST    RET
                03220 ;
7C00            032B0         END     ENTRY
00000 TOTAL ERRORS
25132  TEXT AREA BYTES LEFT
```

VIC-20 Large Characters

by M Ahmed

Using this program should provide VIC users with some insight into how to make full use of VIC's graphics. It will display chosen characters in magnified form on the screen. Characters available are the alphabet, all ten digits, &, *, £, $ and mathematical signs. Graphics characters themselves cannot be reproduced using this program.

Input is requested by a large flashing cursor, and characters are displayed simply by pressing the appropriate key. Deletion is done by backspacing.

For use on PET computers the following modifications are all that are needed; in line 106 change T=T-4 to T=T-5; in line 240 change T=T+4 to T=T+5; line 140 should be IF T>38 THEN PRINT "QQQ": T=Q (here 'Q' is in inverse video).

An inverse 'R' and '—' in this listing mean inverse on and off respectively. An inverse heart is the symbol for clear screen. 'Q' reversed is cursor down and a circle (inverse) is cursor up.

```
10 DIMB$(45,4),C$(45):L=45
20 FORI=1TOL:READC$(I):FORJ=1TO4:READB$(I,J):NEXTJ,I
30 U$="TTTT"
90 PRINT"J";
100 GETG$:IFG$=""THENGOSUB300
105 IFG$=CHR$(13)THENT=0:GOTO90
106 IFG$=CHR$(20)THENF=1:GOSUB300:T=T-4:IFT<1THENT=0:GOTO115
110 GOSUB200:IFX=0THENG$="":GOTO100
115 GOSUB220
120 PRINTU$;TAB(T);
140 IFT>18THENPRINT"QQQ":T=0
150 GOTO100
200 X=1:FORJ=1TOL
210 IFC$(J)=G$THENC=J:RETURN
215 NEXTJ:X=0:RETURN
220 FORI=1TO4
230 PRINTTAB(T)B$(C,I):NEXT
240 IFQ=0THENT=T+4
290 Q=0:RETURN
300 IFF=1THEN350
310 G$="CR":F=1:Q=1:RETURN
```

```
350 G$=" ":F=0:Q=1:RETURN
500 END
1000 DATA"A","[illegible]","[illegible]","[illegible]","[illegible]"
1010 DATA"B","[illegible]","[illegible]","[illegible]","[illegible]"
1020 DATA"C","[illegible]","[illegible]","[illegible]","[illegible]"
1030 DATA"D","[illegible]","[illegible]","[illegible]","[illegible]"
1040 DATA"E","[illegible]","[illegible]","[illegible]","[illegible]"
1050 DATA"F","[illegible]","[illegible]","[illegible]","[illegible]"
1060 DATA"G","[illegible]","[illegible]","[illegible]","[illegible]"
1070 DATA"H","[illegible]","[illegible]","[illegible]","[illegible]"
1080 DATA"I","[illegible]","[illegible]","[illegible]","[illegible]"
1090 DATA"J","[illegible]","[illegible]","[illegible]","[illegible]"
1100 DATA"K","[illegible]","[illegible]","[illegible]","[illegible]"
1110 DATA"L","[illegible]","[illegible]","[illegible]","[illegible]"
1120 DATA"M","[illegible]","[illegible]","[illegible]","[illegible]"
1130 DATA"N","[illegible]","[illegible]","[illegible]","[illegible]"
1140 DATA"O","[illegible]","[illegible]","[illegible]","[illegible]"
1150 DATA"P","[illegible]","[illegible]","[illegible]","[illegible]"
1160 DATA"Q","[illegible]","[illegible]","[illegible]","[illegible]"
1170 DATA"R","[illegible]","[illegible]","[illegible]","[illegible]"
1180 DATA"S","[illegible]","[illegible]","[illegible]","[illegible]"
1190 DATA"T","[illegible]","[illegible]","[illegible]","[illegible]"
1200 DATA"U","[illegible]","[illegible]","[illegible]","[illegible]"
1210 DATA"V","[illegible]","[illegible]","[illegible]","[illegible]"
1220 DATA"W","[illegible]","[illegible]","[illegible]","[illegible]"
1230 DATA"X","[illegible]","[illegible]","[illegible]","[illegible]"
1240 DATA"Y","[illegible]","[illegible]","[illegible]","[illegible]"
1250 DATA"Z","[illegible]","[illegible]","[illegible]","[illegible]"
1260 DATA"£","[illegible]","[illegible]","[illegible]","[illegible]"
1270 DATA"!","[illegible]","[illegible]","[illegible]","[illegible]"
1280 DATA"%","[illegible]","[illegible]","[illegible]","[illegible]"
1290 DATA"&","[illegible]","[illegible]","[illegible]","[illegible]"
1300 DATA"'","[illegible]","[illegible]","[illegible]","[illegible]"
1310 DATA"*","[illegible]","[illegible]","[illegible]","[illegible]"
1320 DATA" ","    ","    ","    ","    "
1330 DATA".","    ","    ","[illegible]","[illegible]"
1340 DATA"0","[illegible]","[illegible]","[illegible]","[illegible]"
1350 DATA"1","[illegible]","[illegible]","[illegible]","[illegible]"
1360 DATA"2","[illegible]","[illegible]","[illegible]","[illegible]"
1370 DATA"3","[illegible]","[illegible]","[illegible]","[illegible]"
1380 DATA"4","[illegible]","[illegible]","[illegible]","[illegible]"
1390 DATA"5","[illegible]","[illegible]","[illegible]","[illegible]"
1400 DATA"6","[illegible]","[illegible]","[illegible]","[illegible]"
1410 DATA"7","[illegible]","[illegible]","[illegible]","[illegible]"
1420 DATA"8","[illegible]","[illegible]","[illegible]","[illegible]"
1430 DATA"9","[illegible]","[illegible]","[illegible]","[illegible]"
1440 DATA"CR","[illegible]","[illegible]","[illegible]","[illegible]"
```

'I told you Miriam! In all probability he's being unfaithful every Tuesday!'

'This new system is selling like hot cakes. Now it's just up to you to design it'.'

ADVERTISERS INDEX

A successful sandwich shop means more than cutting great sandwiches. It means cutting costs to the bone without cutting corners.

The NEC PC8000 personal computer makes it possible for the small businessman with no computer training, to analyse his business in every aspect.

This will almost certainly produce ideas as to how to make the business more efficient and profitable.

The NEC PC8000 personal computer has many kinds of application programs available.

You can get them from the store at which you buy your machine. With these you can do anything from basic book-keeping to producing financial forecasts and trend analysis of your business.

The NEC PC8000 personal computer is designed and manufactured by NEC in Japan, where 10,000 a month are being manufactured and sold worldwide.

★★★★★

Application programs available include:-
2 Word Processors, Mailing List, Telecommunications, Target Business Planner, Debtors, Creditors, General Ledger, Inventory, Invoicing, Program Generator, Dbase II, Micro Modeller.

NEC
TOKYO JAPAN
PC 8000 series

"WHEN IT COMES TO CUTTING COSTS THIS COMPUTER IS AS KEEN AS HIS FAVOURITE KNIFE."

. . . describing the price performance leaders in table top computers.

Owning an SX from IMS International is a joy no matter which model you choose.

Both give you Winchester performance and a lot more.

Our 5000 SX table top computer features either a 5.5 or 11 MByte Winchester subsystem, plus one or two 5¼" floppy drives.

And our 8000 SX offers a 10, 20 or 40 megabyte Winchester, plus one or two 8" floppies, and a 17 megabyte incremental cartridge tape drive subsystem.

Both high performance Winchester subsystems deliver awesome speed — ten times faster than floppies, on the average. You can load a 20 KByte system program in less than a second.

More importantly, IMS systems are engineered for reliability, so downtime won't take away the gains Winchester technology brings. They're strictly business, from their metal-not-plastic cabinets to their test-don't-guess circuitry.

And with our full 2-year warranty, you have it in writing.

Software for the SXs includes either single user, or our new high performance multiuser/multiprocessor operating systems, plus BASIC, FORTRAN and COBOL languages, and a host of compatibility-tested application programs.

For complete information and specifications on the 5000 SX and 8000 SX, along with the location of your nearby IMS International dealer, just contact us.

We'll tell you everything you ever wanted to know about SX.

IMS
INTERNATIONAL COMPUTERS

S. I. MICROCOMPUTER PRODUCTS PTY LTD

H.Q. MANHATTAN CENTRE.
CNR WALKER & MOUNT STS,
NTH SYDNEY, 2060.(02)9223977
CANBERRA (062) 47-8064
MELBOURNE (03) 26-5522
BRISBANE (07) 397-7322