Australian Personal Computer

REGISTERED FOR POSTING AS A PUBLICATION – CATEGORY B.
REGISTRATION NO. VBP 3691. ISSN-4415 NZ $3.00 DECEMBER 1982

$2·50*

NEW! EPSON PORTABLE BENCHTESTED

AUSTRALIA'S TOP SELLING COMPUTER MAGAZINE

ET ON A CHIP

We review Atari's Extra Terrestrial game

IT HAD TO HAPPEN...

The new HITACHI PEACH out-performs any small business/personal system you may now be considering, but only costs $1495 plus sales tax if applicable

Includes numeric key pads, ten programmable function keys (5 with shift), clock, speaker, the power of 6809 CPU, (RS232, centronics, light pen, audio, video) connections, six expansion slots for peripherals, 32k to 96k RAM, 24k ROM, advanced MICROSOFT BASIC, machine language, complete documentation, main frame TERMINAL capabilities, 640 by 200 dot byte – addressable graphics, eight colours, upper/lower case, 40/80 columns and much more. Monitors, diskette drives, hard disks, CIS-COBOL, PASCAL, FORTRAN, etc. are also available to complement the high standards and unequalled quality of this, the most powerful small computer in the world.

Delta – Australian Distributors for Hitachi Personal Computers
4th Floor, 520 Collins Street, Melbourne. 03-62 2008.

HITACHI

Dealer enquiries welcome.

ADVERTISERS INDEX

inside APC

Volume 3 No. 12 1982

Cover courtesy of Futuretronics Australia Pty Ltd.

REGULARS

FEATURES AND SERIES

BENCHTESTS & REVIEWS

22

Editor	Technical Editor	Advertising	Printed by	Produced under
Sean Howard	Ian Davies	Gerard Kohne (03) 544 8855	Lewis Printing	licence from Felden Productions

Published by Howard Productions,
500 Clayton Road, Clayton, Vic 3168. Telephone (03) 544 8855.

SORCERER BASIC COMPILER $69

The BASIC COMPILER is truly an amazing product! Now for the first time, Sorcerer users can join the many thousands of users who develop programs using a BASIC compiler.

WHAT DOES THE BASIC COMPILER DO?
The BASIC COMPILER is an assembly language program which translates a BASIC Source program into object code which may then be run at a speed comparable to machine code. At the time of compilation, the compiler converts constants to binary, enters variables into a symbol table, checks the syntax of each statement, resolves line number references and generates object code. When the object code is executed there is no further conversion of constants or searching for variables or line numbers. The object code is fully processed for immediate execution. An interpreter (such as the BASIC ROM PAC) is at a severe speed disadvantage because it must analyse each statement **every time** it executes the statement. It must convert constants to binary, search for variables and line numbers and analyse the syntax. Thus an interpreter does the two operations of analysis and execution every time a statement is executed. The BASIC COMPILER carries out the analysis only once (during compilation) and the generated object code may be immediately executed as many times as required or may be stored on cassette or disk for subsequent use.

BASIC COMPILER OVERVIEW
This compiler is a direct enhancement of the Sorcerer BASIC ROM PAC language. With the ROM PAC inserted, the user may load, RUN and debug his BASIC program then simply compile his code. The compiled program may be run and tested, then, if an error is found, the user may re-enter BASIC and edit the original source. All this may be done without having to save anything on cassette or disk!

DOCUMENTATION
The BASIC COMPILER comes with a full set of professional quality documentation, including:
- Specially made and printed quality ring binder.
- Over 55 pages of text.
- Full table of contents.
- Full explanation of all statement and functions including extremely clear description of constants, variables, operators and expression valuation.
- Page headings include chapter, section and subsection number for quick reference.

SUPPORT
SYSTEM SOFTWARE is one of the leading international developers and suppliers of software for the Sorcerer Computer. SYSTEM SOFTWARE is committed to continuing research and development of new and better products for Sorcerer users. Suggestions for new products or enhancements to the BASIC COMPILER are always welcome. The BASIC COMPILER is a reliable, solid product and SYSTEM SOFTWARE is committed to its continuing exellence.

FEATURES
- Complete easy to read documentation.
- Full software support and back-up.
- Extremely compact code and data structure.
- Full boundary and overflow checking.
- Easy to use.
- Fast program development
- Byte, integer and floating point variables.
- 3 to 50 times faster than ROM PAC BASIC.
- The worlds least expensive compiler.
- Disks are not required.
- Simplified programming techniques.
- String cassette Input/Output.
- Special Sorcerer high resolution graphics commands.
- Compatible with the ROM PAC.
- Advanced string handeling techniques used.

SPEED
The BASIC COMPILER is designed for fast compilation and execution of small or large programs with hundreds of lines and variables.
Programs will typically execute 3 to 20 times faster than ROM PAC BASIC. By optimizing the code the user may obtain speed advantages in excess of 50 times, particularly with long programs.

COMPACTNESS
The BASIC COMPILER highly optimizes the generated object code to reduce its size to a minimum. Compared with other compilers, the object code is ½ to ⅓ the size. Together with optional byte and integer constants and variables (Occupying only 1 or 2 bytes of memory each), significant space savings are made and thus allow the compilation and execution of large programs.

RELIABILITY
Unlike some compilers, the BASIC COMPILER checks that all arrays subscripts are within bounds and checks for integer and real overflow. Hence there is no chance of a program producing erroneous and unpreditcable results if a bounds or overflow error occurs.

OPERATIONAL EASE OF USE
The BASIC COMPILER is easy to use. A few simple steps are all that is required to edit, debug, compile, execute and save a program. In addition, all this can be done without having to save anything on cassette or disk. There is no complicated linking and loading process. All BASIC COMPILER options are menu driven for user friendliness.

ENHANCED DATA TYPES
The BASIC COMPILER supports bytes, integers, reals and strings. These data types provide programming flexibility, compact code and maximum execution speed.

ENHANCED CASSETTE INPUT/OUTPUT
The BASIC COMPILER allows any type of array to be saved or loaded. Thus byte, integer, real and string arrays may be used in the CLOAD* and CSAVE* statements. The ability to save string arrays opens the opportunity for a whole new range of applications for home, family and business. Text and numeric values (using the new numeric to string conversion BASIC function) can be stored in string arrays, saved and reloaded. Thus, files containing names and addresses together with numeric values can be created, saved, re-read, updated and resaved with maximum efficiency.

GRAPHIC ENHANCEMENT
The BASIC COMPILER includes extra graphics statements (PRINT &, SET and RESET). These facilities, combined with the very fast processing of integers, means that graphics application programs (plots, games, etc.) may be more easily programmed and give much faster screen animation. A 1/6 of a character size dot may be turned on or off anywhere on the screen.

COMPATABILITY
The BASIC COMPILER language is a super set of ROM PAC BASIC with many useful enhancements and few restrictions (which should not seriously inconvenience any user). Any ROM PAC BASIC program should be able to be processed by the BASIC COMPILER. Because integer and byte variables are specified with REM/BYTE and REM/INTEGER statements, which are ignored by ROM PAC BASIC, these programs may be RUN with either ROM PAC BASIC or the BASIC COMPILER.

ADVANCED STRING HANDLING
The BASIC COMPILER has advanced string handling capabilities. The unique method of dynamic string allocation provides full flexibility and also enables string compaction to be avoided completely. Additional string statements and functions (left hand MID$, SPC$, CVI, MKI$ etc) simplify programming and increase execution speed.

PROGRAMMING EASE
Additional features of the BASIC COMPILER may be used to simplify programming. These features include the IF — THEN — ELSE statement, graphics commands, cursor control and additional string functions and others.

PRICE
The BASIC COMPILER is a full compiler with advanced extensions to ROM PAC BASIC and was specifically designed for a wide range of Sorcerer Users. The documentation sets new standards for BASIC language documentation in terms of completeness, logical arrangement and ease of understanding.
The BASIC COMPILER is priced to sell to as wide a market as possible. It is a fraction the cost of any other compiler on the world market today. Considering the development time (in excess of 4000 man hours) and hardware costs we believe that the BASIC COMPILER is unbeatable 'value for money'.

HARDWARE REQUIREMENTS
The BASIC COMPILER has been designed to run on an Exidy Sorcerer computer with at least 32K of RAM.

BASIC COMPILER LANGUAGE

Constants:
Byte, Integer, Real, String.
Scalar Variables:
Integer, Real, String.
Arrays:
Byte, Integer, Real, String.
Operators:
Arithmetic -, +, *, \, /, =
String Concatenation +
Relational =, <, >, <>, <=, >=
Logical NOT, AND, OR, XOR

Specification Statements			
REM/OPTION		DIM	REM/BYTE
REM/INTEGER		CLEAR	REM
Assignment Statements			
LET		MID$	
Flow Control Statements			
END	ON	THEN	STOP
GOSUB	FOR	RETURN	ELSE
IF	GOTO	NEXT	STEP
Input/Output Statements			
INPUT	WAIT	RESET	PRINT&
RESTORE	READ	POKE	CLOAD*
SET	PRINT	DATE	OUT
CSAVE*			
User Routines			
DEF		FN	USR
Numeric Functions			
ABS		EXP	INT
LOG		SGN	SQR
Trigonometric Functions			
ATN	TAN	COS	SIN
String Functions			
ASC	CHR$	CVI	CVS
INSTR	LEFT$	LEN	MID$
MKI$	MKS$	RIGHT$	
SPC		STR$	VAL
Input/Output Functions			
POS		SPC	TAB
PEEK		INP	INKEY

ORDERING INFORMATION

Airmail to:
Name: ....
Company: ....
Address: ....
City: .... Country: ....

Payment Method
Money Order Cheque
Credit Card
☐ Australian Bankcard ☐ American Express ☐ Visa ☐ Diners Club (Please tick one)
Name on credit card ....
Account number .... Expiry date ....
Authorised signature .... Date ....

N.B. Orders may be phoned through during office hours if a credit card is being used on (09) 330 7336.
Phone also during office hours for additional information.

Cost:
Cost of compiler including manual in ring binder file: $69
Airmail postage and packaging:
- Within Australia $3.00
- Overseas $8.00

SYSTEM SOFTWARE
1 KENT STREET, BICTON,
W.A., 6157, AUSTRALIA.

PRINTOUT

APC reports on the latest news from the world micro scene.

The average store

What constitutes your average micro store in Australia today? Certainly three years ago it wasn't terribly impressive, usually had only one or at the most two makes of microcomputer and catered for a dedicated buyer. Since then there has been a trend to flashier stores each carrying a range of micros which has completely turned the tables on the distributor/retailer relationship. Today the retail store has no "political" difficulty putting one micro next to another on a shop bench as evidenced by Robs Computer Centre which carries the VIC-20, Atari 400 and 800, Hitachi Peach and Success, Kaypro, Columbia and ICL micros. Robs Computer Store even boasts one of Australia's biggest "computers" on its roof.

New database

Infostar is a new data base management system apparently designed for "non-programmers". It actually consists of two programs, Datastar 1.4 and Reportstar 1.0.

Datastar 1.4 has additional features over version 1.101 including increased file limit to 8 Mb, new user-friendly training guide, the ability to name individual fields in addition to numbering them, more memory available for data files by having the ability to define any field as an intermediate field, and not surprisingly the ability to work with Reportstar.

It sells for $610 and more details can be obtained from Imagineering on (02) 358 3011.

More than games

VIC Education has been quietly developing software for Commodore's VIC-20 over the last half year. It's aimed at young school children and provides a variety of "lessons" on such subjects as maths and spelling.

Dennis Argall, MD of VIC Education, says that the software is extensively "tested" by children in the appropriate age groups to assure that bugs and unsuitable material is com-

PRINTOUT

pletely expelled before the software is distributed to the marketplace. In keeping with the inexpensive machine for which the programs are designed, the range of ten cassettes retail for around $14 each.

VIC Education is keen to promote education software at this level, so if you have suitable material and are interested in having it marketed write to Dennis Argall at P.O. Box E230, Canberra, 2600.

Apple III Advanced Visicalc

The Visicalc Advanced Version program protects any areas of a worksheet you don't want changed. You need not worry about someone accidentally changing a formula, title or any other important feature. It also provides comprehensive on-screen help, with the "?" key, to assist users in completing their worksheets.

And the worksheets you've already developed with your Visicalc III program on your Apple III are upward compatible with the Visicalc Advanced Version program.

Perfect?

The Kaypro computer now comes with a new suit of software in the basic price of $2950. It's comprised of five modules each prefixed with the somewhat smug name "Perfect". Perfect-Writer and Perfect-Speller constitute a wordprocessing/dictionary; Perfect-Calc is a spreadsheet program; Perfect-File a database and Profit-Plan is a cash flow planner. This software wasn't available for the review last month so you'll have to take this into account if you're about to buy. More details from President Computers.

Teaching Tandy

Closely following Apple's special offer to schools, Tandy has announced discounts to schools for the TRS-80 and associated peripherals. Tandy has developed a number of products with specific applications to schools, the most notable being the "Network 2 Controller" which enables teachers to set up an interactive classroom computer system with up to 16 non-disk Model III "student stations".

Until the special offer expires at the end of February, Tandy is offering to match the number of student stations purchased with an equivalent TRS-80 Pocket Computer. More information can be obtained from Michael Lehman on (02) 638 6633.

Bludners extra

Due to rushed printing and production schedules to get this December issue out earlier than usual, a gremlin has crept into The Technical Bookshop's advertisement on page 24. There's no provision for the purchaser's name or address, so if you're ordering please, please include your name and address. And also please do cut up your copy of *APC* by using the somewhat lacking order form so The Technical Bookshop know which of their advertisements is bringing in the best response.

Programmer

A multi-function card has been released in Sydney for the Apple II. It's a combined EPROM programmer, memory and I/O device. The memory facility is comprised of up to 24k of EPROM in two 12k banks (equivalent to two Apple ROM boards) and 2k of CMOS RAM with optional battery backup.

For more information contact the New Generation Computer Store on (02) 427 4780.

PAUG

A users' group has been formed apparently because professional and business users are fed-up and they're not going to take it anymore: "The Professional Apple Users' Group was born out of frustration with the present level of support business and professional users are receiving. Too many times we have heard stories of users being sold substandard software, software that doesn't suit Australian conditions, software which is incompatible with hardware, peripherals that won't work with critical pieces of software, and just plain lies or half truths about product capabilities. We're fed up with this situation and we intend to do something about it", says the press release.

This group, called the PAUG is publishing a magazine titled PRO (first issue October '82) which is a twenty four page presentation of reviews and articles on aspects of the Apple II. A free sample of PRO can be obtained by sending a large self addressed envelope to PAUG, GPO Box 969G, Melbourne, 3001.

Smaller, faster, cheaper plotting from Watanabe

Watanabe, the leading Japanese plotter manufacturer, has finally broken the Y100,000 barrier. The Miplot Junior DA6000 is slated to sell for $390 in Japan and is fully command compatible with its predecessor, the Miplot.

Plotting speed is said to be 20cm per second, twice as fast as the Miplot. It's half the size and a third the weight too. Step size is 0.1mm.

An optional ROM allows circles, arcs and kana (Japanese phonetic "letters") to be plotted. The Miplot Junior interfaces through an eight-bit Centronics type interface.

Watanabe expects to sell 10,000 units over the next year.

4 colour printers sell for $155

A new 4-colour printer plotter – the PC-6022 – has been jointly announced by NEC and sister company Shin Nippon Denki.

The PC-6022 can print or plot in four colours (black, red, green and blue) and has an average print speed of 12

COMPAK
COMPUTER SHOP

BRIGHTON
592 6285
592 8343
44 THE ESPLANADE,
BRIGHTON BEACH
VIC. 3186

ALL COMPAK PRODUCTS CARRY A SIX MONTH WARRANTY

DANDENONG
793 5701
793 4335
81A FOSTER STREET,
DANDENONG
VIC. 3175

COMPUTERS FOR CHRISTMAS

TAKE ADVANTAGE OF THE HOLIDAYS TO GET TO KNOW YOUR NEW COMPUTER

COMPAK
HAVE GOT EXTRA STOCKS AND SPECIAL DEALS FOR CHRISTMAS

* PRINTERS
* DRIVES
* ACCESSORIES
* HOME & BUSINESS SYSTEMS
* SOFTWARE PACKAGES

TO SUIT MOST POPULAR BRANDS
COME AND SEE US

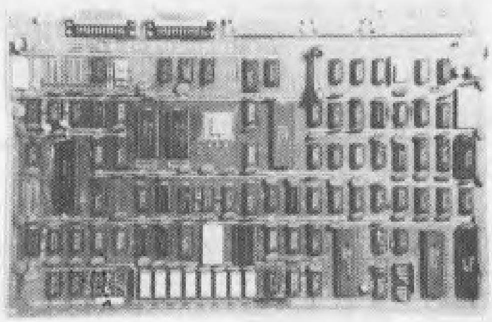
WE ARE OPEN OVER THE CHRISTMAS PERIOD SO THAT YOU WONT MISS OUT OVER THE HOLIDAYS

SUPER CHRISTMAS SALE

V–10 SINGLE BOARD COMPUTER

WAS ~~$945~~ **NOW $625***

Build your own CP/M system in less than 1 hour!



FEATURES:
* 4MHz Z80 CPU
* 64K RAM
* 8" & 5" F.D.C.
* 80 x 25 Video
* 2 x RS232 Ports
* Centronics Printer Port

OR

V–10 COMPUTER KIT including:
V–10 SINGLE BOARD COMPUTER CP/M 2.2
POWER SUPPLY & CABLES & FULL MANUAL

ONLY $895* Fully Tested

DISK DRIVE SPECIALS FOR V–10

8" MITSUBISHI DRIVES & POWER SUPPLY ONLY $795*
5" MITSUBISHI DRIVES & POWER SUPPLY (80 Track DS.DD) only $550*
* + Sales Tax

VIC-20

HAS ARRIVED AT COMPAK

MAIL ORDER OR CALL FOR INSTANT SERVICE

$299 TAX INCL.
ALSO ALL VIC-20 ACCESS.

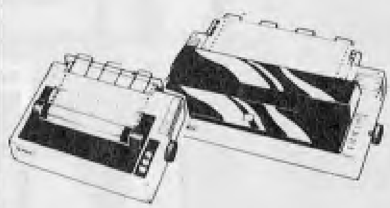
MAIL OR CALL !

OPEN DURING CHRISTMAS HOLIDAYS

COME IN AND LOOK OVER OUR RANGE OF MICROCOMPUTERS DURING YOUR HOLIDAYS

WE ARE OPEN FROM JAN 4TH AND WILL BE PLEASED TO DISCUSS YOUR MICRO REQUIREMENTS

EPSON SPECIALS!!!



EPSON PRINTERS
NEW TYPE 3 NOW AVAILABLE

MX80 FRIC/TRACTOR	$850 + Tax
	$999 incl. Tax
MX100 FRIC/TRACTOR	$1100 + Tax
	$1250 incl Tax

COMPAK
COMPUTERS FOR AUSTRALIA

COMPAK SELLS, SUPPORTS, IMPORTS, MANUFACTURES, WHOLESALES →

SORCERER MICRO
SORCERER DISKS
APPLE MICRO
APPLE ADD-ONS
TRS-80 MOD III

VII MICRO
ARCHIVES II
DISCOVERY
EPSON PRINTERS

ARCHIVES III
MITSUBISHI DISKS
C.ITOH PRINTERS
NATIONAL DISKS
OLIVETTI TWRS

ADLER TWRS
TEAC DISKS
AMUST V-10 MICRO
COMPAK MICROS
SVA APPLE PROD.

COMPAK
MAIL ORDER

BUY DIRECT FROM IMPORTER/MANUFACTURER

*FREE DELIVERY *MONEY BACK GUARANTEE

DEALER ENQUIRIES PHONE RON HARRIS 03 592 6285

CHRISTMAS SPECIAL

C.ITOH 1550 PRINTER

$999 + TAX

SPECIAL UNTIL XMAS ONLY

XXX SCOOP XXX PURCHASE

NEW APPLE™ COMPATIBLE DISK DRIVES

NEW

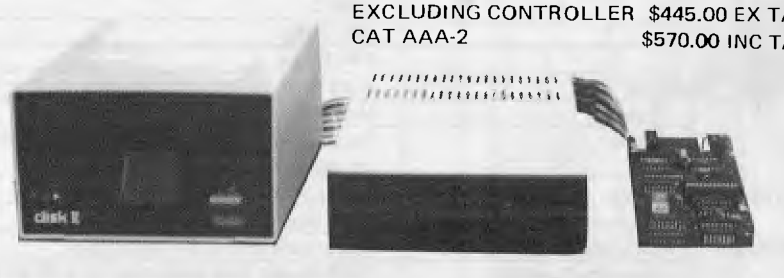
SLIMLINE — FAST ACCESS
NEW JAPANESE DRIVE AND CONTROLLER
IMPORTED DIRECT BY COMPAK

$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 $714.00 INC TAX CAT AAA-1

EXCLUDING CONTROLLER $445.00 EX TAX $570.00 INC TAX CAT AAA-2



APPLE FDD COMPAK FDD CONTROLLER

ALSO COMPATIBLE WITH COMPAK VII SYSTEMS

APPLE DOS DISKETTE & MANUALS SUPPLIED WITH AAA-1

EIGHT INCH FLOPPIES ON APPLE

SVA CONTROLLER AND OUR 8" DISK UNITS NOW RUNNING CP/M AND APPLESOFT
SEE OPERATING AT OUR SHOPS

	Ex Tax	Inc Tax
* Single 8" 1Mbyte unit with Cabinet, Cont., Power Supply, Ready to run	$1490	$1750
* Dual 8" 2Mb unit	$2390	$2790
* Controller only	$ 690	$ 810

LOW PRICES!
C. ITOH PRINTERS

STAR with Apple I/F	$495.00	ex
	$595.00	inc
8510 120 CPS 80 COL	$815.00	ex
	$999.00	inc
F-10 40 CPS Daisywheel	$1740.00	ex
	$1990.00	inc

"THIS MONTH'S SPECIAL"
C. ITOH 1550 PRINTER

* 120 C.P.S.
* 15" Carriage
* Bit Image Graphics
* Bi Direct. with Logic Seeking Head
* Tractor/Friction Feed
* Serial/Centronics Parallel I/F
* 3 K Byte Buffer

MITSUBISHI 8" FDD
NATIONAL 8" FDD

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

STANDARD 8	SLIMLINE 8
$595 ex Tax	$690 ex Tax
$699 inc Tax	$795 inc Tax
CAT MIT-8A	CAT NAT 8

FLOPPY DISK DRIVES, DISK CABINETS AND POWER SUPPLIES

COMPAK IMPORTS A LARGE RANGE OF DISK DRIVES AND LOCALLY MANUFACTURERS DISK CABINETS TO PROVIDE COMPLETE DISK UNITS

CAT APL-8A APPLE 8IN DISK UNIT & CONT.	1mBYTE $1590 ex Tax
CAT SOR-8A SORCERER 8IN DISK UNIT & CONT.	1mBYTE $1490 ex Tax
CAT SOR-5A SORCERER 5IN DISK UNIT & CONT.	.7mBYTE $1190 ex Tax
CAT MIT-8U ADD-ON 8IN DISK UNIT & CONT.	1mBYTE $ 845 ex Tax

ALSO MANY MORE COMBINATIONS. ADD 17½% Sales Tax

MAIL ORDER WORKS WITH COMPAK!

IN AUSTRALIA COMPUTER MAIL ORDER HAS NOW COME OF AGE.
WITH ALL OF OUR OTHER ORDERS WE PROCESS MAIL ORDERS ON OUR COMPUTER SYSTEMS TO GIVE YOU
WE ARE CONSTANTLY SENDING OUT COMPLETE SYSTEMS AND WILL GIVE PERSONAL SERVICE TO YOUR MAIL ORDER! READ MONTHLY APC FOR NEW PRODUCTS!

SORCERER IS A TRADE MARK OF EXIDY CORPORATION; TRS-80 IS A TRADE MARK OF TANDY CORPORATION

COMPLETE SATISFACTION GUARANTEED OR RETURN WITHIN 7 DAYS FOR FULL REFUND

Cut out here

Emona launches a New

Micro-Professor MPF-II

A VERSATILE PERSONAL COMPUTER FOR PEOPLE FROM THE AGE OF 7 TO 70.

For only $499 +S Tax

The Microprocessor MPF—II is the perfect computer for modern homes.

All the features that a home/personal computer can have are packaged in the compact, portable MPF—II. Yet, the booksize computer has some other unique features that you cannot find on other computers.

SPECIFICATIONS OF THE MPF—II

CPU		R6502
ROM		16K Bytes
RAM		64K Bytes
BASIC Totemeter		More than 90 instructions stronger than those for Apple II
Video Display	Type	Memory mapped into system RAM.
	Mode	Text, low-resolution graphics, high-resolution graphics (three modes are mixed).
	Screen	960 characters (24 lines, 40 columns).
	Character Type	5 x 7 dot matrix.
	Character Set	Upper case ASCII, 64 characters.
	Graphics Capacity	1920 blocks (low resolution) in a 40 by 48 array. 53760 dots (high resolution) in a 280 by 192 array.
	Numbers of Colours	6 colours.
Keyboard		49 alphanumeric and function keys.
Cassette Interface Software Cartridge Interface		Use various cassette tapes and cartridges as data storage units.
Printer Interface		Connects to printers with Centronics interface.
Display Interface		Connects to colour home TVs or video display.
Remote Control Paddle		Used for education & entertainment.
Speaker		8 Ω 2¼ inches, 0.25W.
Power		A switching power supply is provided to convert AC power to required power supply.
Dimensions		9.84 x 7.16 x 1.24 inches.

- YOU CAN USE THE MPF—II IN THE HOME, OFFICE, SCHOOL, ENGINEERING APPLICATIONS OR JUST FOR FUN.
- THE MPF—II CAN BE CONNECTED TO ANY COLOUR TV SET AND CASSETTE RECORDER.
- OPTIONS INCLUDE VIDEO MONITOR, THERMAL PRINTER, FULL SIZE KEYBOARD, REMOTE CONTROL PAD, RS 232C INTERFACE BOARD, FLOPPY DISK DRIVE, SPEECH SYNTHESISER BOARD, SOUND GENERATOR BOARD, CHINESE CHARACTERS CONTROLLER AND A HOST OF SOFTWARE CASSETTES AND CARTRIDGES.

FOR FURTHER INFORMATION PLEASE CONTACT:

SOLE AUSTRALIAN AGENT

EMONA ENTERPRISES PTY. LTD.

CBC Bank Building, 661 George Street, Haymarket, Sydney (02) 212-4815, 211 3038.

TRADE ENQUIRIES WELCOME

ALSO AVAILABLE FROM:

NSW: David Reid Electronics, 29 6601. VIC: Radio Parts, 329 7888. SA: Int Communication Systems P/L, 47 3688, WA: Hinco Engineering P/L, 381 4477, ACT: Electronic Components P/L, 80 4654, TAS: D & I Agencies, 23 2842.

PRINTOUT

characters per second. Continuous 24.5mm (10 inch) roll paper is used.

The printer plotter can be operated under program or manual control. Software or manual switching allows font and character size to be changed.

The PC-6022 is intended primarily to produce diagrams, graphs and text for the PC-6000 series of computers but can also be connected to other computers. The input is 8-bit parallel Centronics type.

The PC-6022 will sell for Y39,800 ($155) and sales commence in Japan on 21 November.

IBM compatible portable micros

Two companies that believe IBM will dominate the micro market recently introduced portable microcomputers in New York, which they claim are compatible with the IBM Personal Computer.

The Compaq computer, from Compaq Computer Corporation, a Houston, Texas based company, is a tightly packaged 28lb box that contains up to two 5¼ inch floppy disk drives, a detachable keyboard identical to the IBM PC's and a nine inch diagonal display.

The Dot computer, from Computer Devices Inc, a Burlington, Mass. based company making portable terminals, weighs slightly less at 26lb. and uses 3½ inch Sony disk drives. It uses a 9 by 5 inch display, and in some configurations comes with a built-in dot matrix thermal printer and a direct connection 300/1200 baud modem.

Both systems, like the IBM PC, use the Intel 8088 microprocessor and run Microsoft MSDOS operating software. The Dot comes with a separate Z80 processor for running CP/M software. Compaq claims that any software designed for the IBM PC will run on its computer without modification.

"We had to have compatibility. We never traded compatibility for another characteristic," said Rod Canion, president of Compaq. "To run just MSDOS software does not necessarily mean that a computer will be IBM compatible," he added.

The Dot computer runs MSDOS software that is converted to Sony 3½ inch diskette format. Computer Devices supplies a "dealer kit" for US$8,000, which performs this conversion. "We think the media world is going to remain inconsistent and incompatible, but that this is going to be compensated for through networking and communications," said Seaforth Lyle, president of Computer Devices.

He said IBM's PC software could be downloaded to Dot because the 3½ inch floppies could be formatted in the same way as the IBM 5¼ inch diskettes.

IBM PC hardware compatibility poses different problems for Dot. Lyle said PC add-on boards longer than 10 inches will not fit on Dot. Compaq, with a bigger box, does not face this restriction. Although most boards are 10 inches, colour graphics boards for the IBM PC tend to be longer.

A standard 128 Kbyte Compaq system with one drive, RGB video and composite video interface, parallel printer interface, two hardware expansion slots, MSDOS software and a socket for an Intel 8087 microprocessor will retail at US$2,995. A similar configured IBM PC costs US$600 more, according to Compaq president Rod Canion.

Compaq will start shipping its computer in January 1983, says Canion. A stripped down version of Dot, aimed at the OEM market with 32 Kbytes of memory, no printer, no modem and no disks, is available for US$2,995.

Since most of the software which Computer Devices plans to offer with the system occupies a minimum of between 64 Kbytes and 128 Kbytes, this model of Dot is unlikely to appeal to end users.

A more useful configuration, however, which will retail for about US$4,200 has 128 Kbytes RAM, 24 Kbytes of user programmable ROM memory, one disk drive, a built-in printer, 300 baud modem, asynchronous communications, two hardware phantom slots, and MSDOS software will retail for about US$4,200. The 300/1200 baud modem costs US$950.

Small volume shipments start this month.

Pro Digital

Prospero Software recently announced its decision to make the Pro Pascal compiler available on the DEC range of personal computers, including the Rainbow. An agreement to this effect has recently been signed which will run initially for five years giving DEC worldwide marketing rights for Pro Pascal on its own computers. Pro Pascal will be included in DEC's classified software directory.

Maggie Burton

Business Brain for Sale

BUSINESS PACK 1
Peach CPU — extra memory Hi-quality green screen. Dual 5″ drives 80 CPS printer
$4200 incl. tax

BUSINESS PACK 2
Peach CPU — extra memory Hi-quality green screen. Dual 8″ drives 80 CPS printer

With 64K CPM system

Business Program Modules $310 each

With Hitachi Peach buying a business computer is made simple. Now Robs Computer Centre brings you 4 Hitachi Peach Business Packs — you can be sure there is one to suit your needs.
These unique Business Packs come with incredible features built into each system to give you total control of your business at a fraction of cost of any other system.

DELUXE COLOUR BUSINESS PACK 3
Peach CPU — extra memory Hi-quality colour screen. Dual 8″ drives — 2.2 Meg. 80 CPS printer

With full colour CPM

HI-WRITER WORD PROCESSOR PACK 4
Inc. Hi-writer program Peach CPU — extra memory Hi-quality green screen. Dual 5″ drives, printer

For as little as $4200 Hitachi Peach will lift the performance of your business

ICL
XEROX
THE COMMODORE
VIC 20
KAYPRO II
ATARI®
OSBORNE

robs COMPUTER CENTER

We'll put you in touch with the best brain in the business

295 Thomas Street,
DANDENONG, 3175 Phone: 791 2900

PRINTOUT

Peach roots in down under

Management Sciences America Inc, which claims to be the world's largest independent software company, has announced the setting up of a Peachtree Software International sales and support operation in Australia on 1st January '83.

Peachtree Software International was established in 1975 and acquired by MSA in July 1981. Peachtree began international operations in January 1982 when offices were opened in the UK.

Commenting on the decision to establish the Peachtree operation in Australia, Robert Fisher, general manager designate, said, "We realise the importance of the Australian market place and believe it has an enormous potential.

. . . Although we may be seen to be making a late entry to the microsoftware market in Australia we know that our pedigree and the stability of our software will do much to open the door for our products. We will initially concentrate on supplying manufacturers and distributors of hardware in Australia and New Zealand, with a comprehensive range of proven accounting and office systems for their existing and future machines."

Peachtree Software has already had considerable success in making agreements with manufacturers in the UK and in the USA. Peachtree Software is widely sold under manufacturer's own labels; such distributors include IBM, Hewlett Packard, Zenith and Apple.

For further information contact Barry Nash, Management Science America (Aust) on (02) 929 0711.

Blowing up a storm

Anybody who has written a Basic program longer than 20 lines will have found, one day, that there are two instructions to GOTO 200, and they can't remember where the other one is, or that there are three GOSUB 3000s and one of them is line 450, but what of the other two?

On the Apple these people can now do a simple search for a string, using new software called Gale, from MicroSparc.

Gale 'gives Applesoft programmers the ability to perform global searches and replacement of any specified character or characters throughout a program,' says the supplier in Lincoln, Massachusetts. 'Line by line editing allows you to make fast changes without recopying an entire line – you can delete characters, insert characters, compact a line by replacing PRINT statements with the equivalent (a question mark), enter lower case characters, and so on.'

Details P.O. Box 325, Lincoln, Mass 01773, or try phoning (617) 259 9710.

The biggest little disks in Japan

Toshiba is claiming the highest data density on an eight-inch disk in Japan – up to 94.4 megabytes. Four machines have been announced and sample quantities will be available from next January.

The four machines (with sample Japanese prices) are: MK184F ($3,500) for 26.1 Mbytes; MK184FL ($3,900) for 94.4 Mb; MK182FL ($3,500) for 84 Mb; MK 182 FL ($3,500) for 67.4 Mb.

All four models have internal microprocessors to simplify interfacing with the host mini or microcomputer. The carriage has been designed to have a high resonant point and specially developed positioning devices enable the track density to be three times higher than previous designs – 900 tracks per inch.

CP/M Fair

The idea of a CP/M Fair must be absolutely unique in the history of computing. It is going to happen, nonetheless, in January in San Francisco.

Previous gatherings of likeminded user groups have always centered on a language or a machine – and, mostly, the languages on one machine have been different enough from those on other machines to keep IBM users from discussing

Cobol with Honeywell or ICL people.

CP/M, however, is an operating system which is not native to any one machine more than any other. The problems which CP/M software designers have vary from machine to machine, but still have more in common with each other than they have to distinguish them.

The scope of the show is enormous. Over 200 exhibition stands are planned, and these 'will show the full spectrum of application packages, development aids, microcomputers (how did they get in there?), peripherals, accessories, publications and services available to CP/M users.'

The strength of the show is at the same time its weakness.

The strength lies in the fact that this is an 'official' CP/M show, with Gary Kildall himself (the inventor and designer of CP/M and boss of Digital Research which sells it) plus people like Tony Gold, founder of the CP/M Users Group and of Lifeboat Associates, and people like Adam Osborne, who swears by the system and sells a machine which runs it.

In other words, the people who can answer your questions will be there.

But the weakness lies in the fact that the dissidents will not be invited to speak. The organisers will claim that there is no point in having dissidents at workshops, where people are trying to get things done, not question the basis of why they need doing.

The organisers will therefore fail to consider ways of moving CP/M away from its dreadful limitations.

Limitations of CP/M are all derived from paper. CP/M is a standard, and it assumes that you may want to talk to a CP/M machine from a remote teleprinter.

The effect of this is a catastrophe on a comparable level to what would happen to air transport if laws were passed insisting that aircraft should be capable of pulling trains through tunnels.

You can't say this, however, to a CP/M believer. To such a one, the fact that CP/M is a standard is all, and if the standard implies a lowest common denominator, so be it.

Yet without this assumption, we would have had concurrent screens, because there would be nothing to stop them. We would have global, relational data structures, because you could look at them. We would have flexible screen formatting, we would eliminate dual terminals, we would have screen editors, and we would have soft keyboards and screens.

Anyway, that isn't what you will get at CP/M-83, which is the name of the exhibition and conference.

For what you will get, contact Digital Research, or Northeast Expositions Inc, which is contactable at 824 Boylston Street, Chestnut Hill, Mass 02167. Phone, if you like, (617) 739 2000.

The exhibition runs from January 21 to 23.

Guy Kewney

Minis losing out

The micro market has burst the minicomputer bubble, figures just released in America show.

Research company Frost and Sullivan predicts, in a report on the minicomputer hardware market, that the US minicomputer business "will advance from $4.7 billion in 1981 (last year) to $12.5 billion by 1986, in constant 1982 dollars.

This growth, says Frost and Sullivan, while "impressive", falls far short of many predictions made as recently as two years ago.

"This drastic slowing of growth can be attributed to a number of factors – primarily the current economic slump and concommitant high interest rates, but also the availability of microcomputer systems capable of competing with minicomputers at far lower price levels," says the report.

A minicomputer is a system costing around $90,000, according to the company and it predicts that this definition will remain valid even if technology advances.

"Capacities of microcomputers – the fastest growing segment – can be expected to continue to grow," says the report. "Systems with 32-bit architecture will soon be competing with the traditional eight-bit and 16-bit systems – providing more impressive price/performance ratios at the bottom of the market." And as micros get more powerful, the report says, they will undermine more and more mini computer applications.

The study added that user problems in minicomputers were approaching the level encountered in micro systems.

A Frost and Sullivan survey of end users indicated "a significant level of dissatisfaction with the quality of service, and especially support." This is a state of affairs "which one might more likely expect to find in the microcomputer industry," adds the company rather naively.

The figures put in context the urgency with which Digital Equipment must approach the microcomputer market, since it is, by an enormous margin, the largest minicomputer supplier.

Digital has around 30 per cent of the total US market, followed at some distance by Hewlett-Packard with only 11.6 per cent, Data General marginally behind that at 11.4 per cent, IBM some way after in fourth position (6.8 per cent) and Wang, nearly as big in this area, with a surprising 6.6 per cent.

New micro disk system from Sony

Sony's new 3½ inch micro disk system is now available to OEM customers. The Japanese company has already signed a $30 million deal with Hewlett Packard which will be incorporating the system in future products. It uses a 3.5 inch micro disk with a storage capacity said to be twice that of regular 8 inch and 5.25 inch mini disks.

The Sony system has a storage capacity of 437.5 Kbytes, and is a double density, single-sided disk. It has one touch disk load/unload, and fail-safe disk insertion.

"The 3.5 inch micro floppy disk is a growth area for Sony and it has already been seen as a major break-through by Hewlett

Packard who will be using the system as a future standard for some of its computer products," said Sony product manager Robin Allison.

National launches low cost home micros

Matsushita, better known by their National brand name, has announced a $318 addition to their range of domestic computers.

The new model is the JR-200 and the Y79,800 price tag makes it one of the cheaper colour computers available in Japan. As supplied, the JR-200 has 32 k of RAM and 16 k of ROM with Basic built-in. Eight colours (black, blue, red, purple, pale blue, yellow and white) are displayable using either a domestic TV set or a colour monitor. Video outputs provided are UHF, composite and RGB.

Also designed to appeal to the home user is a built-in five octave sound generator which can harmonise three voices at a time. The baud rate for the cassette interface can be switched between 2400 and 600 baud for faster program storage and retrieval.

The Computer Company expects to have stock here in the first quarter of next year.

Paasokon -terebi . . .

. . . is the word coined by Sharp for its latest marketing gimmick — a personal computer (paasokon) combined with a television receiver (terebi).

The new system, called the X1, was unveiled to the waiting masses on 19th October at the Electronics '82 Show in Tokyo. Sales will start in mid November at a retail price of Y268,000 ($1045). The X1 is based on the CZ-800C microcomputer and the CZ-800D 14 inch colour television. The unit allows computer and video (broadcast) signals to be combined on the screen. Possible applications are said to include video editing.

Programs galore

"Programs" freaks will delight in the release of "VIC Innovative Computing" published by Melbourne House. It's thirty Basic games listings and the odd full colour picture of the games in action. Most of the programs are also available on a series of three cassettes at $20.00 each. Both the cassettes and book ($17.95) are available from Compshop Australia at 4/75 Palmerston Crescent, South Melbourne.

Sirius hard disk

An integral 5¼ inch winchester hard disk is now available for the Sirius 1 providing 10.6 Mb of online storage.

Chuck Peddle, president of Sirius Systems Technology Inc., announced the new offering at the opening of the SICOB Trade Show in Paris on September 22nd, 1982.

Barson Computers introduced the Sirius 1 Winchester on October 21st at a Sirius dealer seminar at Melbourne University. "Already recognised as the most powerful 16-bit micro on the market, the addition of an integrated 10 Mb winchester reinforces this position, by increasing disk storage capacity and access speed, and by enhancing data integrity", said Julian Barson.

Mr Barson recently returned from the Paris Trade Show and a visit to A.C.T. in the United Kingdom and the U.S.A., where he studied computer marketing and microcomputer developments.

He said "Sirius is enormously successful in all its market areas. It has fulfilled all its early promises and is definitely the leader in technology. It has gained a competitive edge with new developments such as the 10.6 Mb winchester and networking to be introduced soon.

Further details from Greg Johnstone on (03) 419 3033.

SOFTWARE

DATABASE COMPARISON THE STORY SO FAR

Kathy Lang presents a comparison of the more popular databases for microcomputers on the market today.

In this series so far I've looked at several packages for managing files of information, ranging in complexity and price from Cardbox, a simple but powerful 'electronic card index' system running under CP/M and costing $298, to FMS-80, a complex and sophisticated system with its own programming language and costing (in its full version) $1100. In this article I shall try to put together some of the qualitative information in a way which will make it easier to compare packages and also show the results of some Benchmarks to try and get some feel for comparative speed.

Benchmarks

It is important to remember when looking at the timings that these are only of value in comparing one package with another — they may not give any indication of how quickly a package will run on your system. The machine I used for the tests was a Sirius with 128k of memory and a Z80 processor card; all the tests on CP/M packages were run on the Z80 card under CP/M-80 with the exception of DMS, for which I used the native 8088 processor. I wouldn't expect that to make any significant difference when comparing DMS with the rest, as by far the major limiting factor in data management is disk access time, where the hardware is of course the same for both processors. Silicon Office runs only on the PET so to run those tests I borrowed a 80-column PET with two 500k disk drives.

The tests themselves were conducted using a test file of plain text data created by a Basic program. The purpose of this was to ensure a completely determined set of data identical to all the packages and relatively easy to transport to each system. The data file consisted of 1000 records containing seven fields and 46 characters, which were then imported into a file, where each record contained 21 fields, totalling 152 characters. This arrangement made it possible to test the package's agility at importing information from elsewhere and the efficiency of the storage methods (since part of each of the larger records could either be null or blank, depending on the file organisation). The tests were designed to cover the major headings I've used in assessing each package. The results of each test are shown in a table under the relevant section, with an attempt at a qualitative assessment of the same facilities alongside. The purpose of the latter is to try to give some impression of the power and ease of use of the particular facility, as well as its speed.

Four of the packages I tried, Cardbox, Silicon Office, DBMSIII, and FMS-80, were able to read my file only if I wrote a special program (in any language, SO own, Basic, EFM respectivly) for the purpose. DMS, dBaseII and Condor all had commands or procedures for the purpose, so a program wasn't needed. To import 'stranger' files into Pearl requires a special one-off program which can be written only by its originators; apparently this can be done only in the US.

Data models

Most of the packages in this series call themselves 'data base management systems' but, unlike true DBMS, they are all basically single file systems. They thus do not vary tremendously in file organisation and methods of direct access to information. All hold their data either in linear files with indexes, or in relational files with or without indexes. (All that is meant by 'relational' here is that every field in the record is 'related' to every other, without any explicit hierarchy or network arrangement — in fact DBMS does use a kind of network to link records from different files together, but not within a single file.)

Five of the packages (Condor, dBaseII, DBMS, FMS-80 and Pearl) permit the linking of two or more files by keys, although the DBMS approach is rather unusual and requires more human intervention than the others. All the packages except Cardbox and Condor use fixed length records and, with very minor exceptions, fixed length fields. Figure 1 shows the limitations each package places on size of file, record, number of fields, number of records, etc.

Input and editing

All the packages require you to decide on at least an initial format for a data file before you can put information into it. I personally find the most satisfactory method of doing this is to 'paint' the format on the screen, in the way that most of the packages now permit (but not DBMS). Some are easier to use than others — for instance, I found Silicon Office tricky because the start and end of field markers were very similar, and Cardbox's three-tier naming system I found very confusing. Once designed, and data inserted, the packages vary substantially in the ease with which you can add new fields. In Cardbox this is very easy; in dBaseII and Condor straightforward and in most of the others possible but more difficult and often slow. Figure 2 shows the time taken for each of the packages tested.

Reporting

Only Cardbox uses the same mechanism for designing printed reports and for screen formats. This method of 'painting' the format on the screen is great provided you can keep within the single screen space. The other systems require you to specify row and column positions for each field, heading, etc, with calculations to be performed where this is permitted. dBaseII is rather limited in its ability to display report formats, though whether many users would ever get past the complexity of design to actually use all the features FMS-80 provides I rather doubt.

Selection

It is, of course, very common to want to select records according to particular criteria and display or process them separately. I've used two measures of selection time: one uses two selection criteria and a simple sequential search (ie, no index is used), while the other uses an index and shows indexing as well as access time. Times for both tests

are shown in Figure 4. When comparing indexing time you should bear in mind the difference between dBaseII and Cardbox, which keep their indexes up to date as they go, and the others which only keep the index to the main key field up to date. You should also remember that both FMS-80 and DMS require you to set up a selection file first — you can't just ask for the selection and display in a single statement, and then modify your request in the light of the display. Condor is also rather inflexible, in that it creates a separate data file each time you request selection, even if you only want one or two records. In Silicon Office, you have to write a program for the kind of selection used in the tests (except when selecting on the key field), since the selection command does a sequential search of every field in every record in the file; my tests used a more discriminating method in a program, but it was still very slow.

Sorting

Here the packages vary considerably in whether you can, or need to, sort the whole data file, or whether sorting the indexes is sufficient to enable you to display records in a particular order. DMS probably has the best approach here: the data file is never sorted, but individual indexes may be created and sorted so that the records are displayed in any required order. In FMS-80 you may sort either data files or indexes, but creation and sorting of indexes are two distinct processes. dBaseII allows you to sort the data file if you need to — and you will if you want to select using non-unique keys and then display in key order. Cardbox doesn't have a sort feature, a curious omission in a package so oriented towards bibliographic and free-text applications — surely libraries need their records in author or title order? Figure 5 shows sort times with one 20-character field as the key. The figures shown are for sorting index only, data only, or both, as appropriate.

Adding single records to the file can be done directly — that is by specifying a key value, displaying the record on the screen and editing it there — in all the packages except Condor, which requires the creation of a transaction file which is applied without intervention from the keyboard. This is the approach which is recommended in FMS-80, but not enforced. Personally I think much of the point of an interactive system is lost if you have to think up possible problems in advance rather than being able to check errors as you go. A decent file-handling system should be able to make on-line updating sufficiently secure for the batch process to be unnecessary except in special circumstances. Figures for batch and for on-line updating are shown in Figure 3.

Indexing

Getting all the information in of course begs the question of how to get it out. All the packages (except Condor in its Levels 1 and 2 versions) have some methods of accessing particular records quickly, but they vary a lot in their power and flexibility. Silicon Office, DMS and FMS-80 require you to choose a primary key field and every record is indexed by that field. DMS and FMS-80 also allow you to create other indexes to permit fast access by more than one field but these are not kept up to date when the file is updated — you have to do this for yourself. dBaseII can keep up to seven indexes per file up to date 'on the fly'. These factors should be born in mind when looking at Figure 4, which shows the times for direct searching, since the indexing times are only significant if indexes have to be re-organised explicitly rather than being included as a (very small per record) overhead in the updating process. Cardbox is the most flexible of all in some respects, in that it permits indexing of all or any items (words) within fields either automatically or on request from the person updating. However, once you have put data into a Cardbox file, if you decide later that you want to index a field which was previously not indexed, then this can be very tedious.

Screen display

Usually you can use the same features for designing a screen format for display and editing. Some packages, notably FMS-80, DMS, Cardbox and the later version of dBaseII, allow you to vary the formats so that some information is not displayed, or shown in different formats for different users. DBMS

Package	Cardbox	Condor	dBaseII	DBMSIII	DMS	FMS-80	Pearl	Silicon Office
Max file size	65500 recs †	32767 recs †	65535 recs †	32000 recs †	26214 recs †	CP/M limit	CP/M limit	3 files per disk
Max size record	1404 chars	1023 chars	1000 chars	1400 chars	1024 chars	64k chars	3080 chars	252 chars
Max no fields	26	127	32*	20**	60	255	250	no limit
Max field size	1404	127	254	70	80	255	80	78
Field	C	CDI£	CNL	CN	CN	CDN	CDN	CN

Notes: † = or CP/M limit (8 Mb), whichever greater * = unlimited no subfields
** = 200 subfields
Field types: C=Character/D=Date/I=Integer/N=real numbers

Fig 1 Constraints

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
Time for 1000 records to add 1 new field to each	Inst's: just mod definition	6m	10m	58m30s	36m30s	3h27m for 250 records
Star rating	****	***	***	**	*	*

Fig 2 Time to add one new field to each record

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
Interactive time to add 50 records +	Inst's	NA	Inst's	Inst's	Inst's	3 secs per record
Batch	35s	25s	1m30s	1m45s	6m30s	NT
Star rating	****	*	***	**	*	**

Notes: + excludes time to key in new data/NA=Not Available/NT=Not Tested

Fig 3 Time to add 50 records

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
Sequential Time for 1000 records	2m31s	2m + scroll	3m20s total	2m20s total	4m5s + scroll	4h25m for 263 records
Indexed access	8s	NA	Inst's	Inst's	Inst's	NA
Create Index (25 char field)	NA		(14m44s) if needed	4m45s	19m49s	(only one key field)
Star rating	***	***	***	**	**	*

Fig 4 Sequential and indexed read (two criteria, 50 records match)

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
Time to sort 1000 records on 5-char field	NA	26m30s	35m	5m30s (index only)	NT	1h10m for 250 records
Star rating	NA	***	**	****	NT	**

Fig 5 Sorting 1000 records

ANYTHING IBM CAN DO....

president proudly introduce

COLUMBIA

16 BIT IBM PC COMPATIBLE

It's 16 BIT. It's IBM PC compatible. And it's here now! President have a new business computer manufactured by Columbia Data Systems of the USA complete with its own software and hardware. It will also run all software and hardware produced for the IBM PC.

At first glance, this machine may bear a certain resemblance to its mentor.

That's before you switch it on.

PRICE INCLUDES FULL COLOR

16 BIT IBM PC COMPATIBLE

WHAT A PERFORMANCE!

Appearances can be deceptive as a look at the credentials of the Columbia MPC machine will tell you.

Operating system software is MS-DOS® or CP/M 86®.

It's 16 BIT 8088 hardware configuration provides 128 K RAM with parity, two RS-232 serial ports, Centronics parallel printer port, DMA controller, eight levels of interrupt, dual floppy disc system with one megabyte storage, Winchester interface and eight expansion slots.

Not bad for the standard version.

EXAMINE YOUR OPTIONS

Columbia like to leave your options wide open. So they've included 256K RAM boards, 8087 arithmetic co-processor for high speed math functions, dual RS-232/RS-422 boards, and a Winchester disc based system.

The Columbia MPC's expansion capability can easily accommodate just about any imaginable hardware configuration including one megabyte RAM and 10 megabyte disc.

PRESIDENT GROUP STAR SERVICE

The President Group offers the services of 290 staff and over 160 vehicles, branches in all states, dealers in country areas and a service network of highly experienced professionals that's second to none. We guarantee the product, the service and the customer's satisfaction.

President is the all-Australian company that specialises in Australian business requirements.

Standard Software includes Electronic spreadsheet and Word Processing with lesson diskette.

FOR FURTHER DETAILS, OR IF YOU WOULD LIKE TO BECOME A DEALER YOURSELF, CONTACT

Call one of the President's men for the name of your local dealer

N.S.W. DISTRIBUTORS:
PRESIDENT COMPUTERS NSW
100 George Street,
Hornsby. Tel: 476 2700
Telex: AA 30363

COMPUTERMAX
539 Pittwater Road,
Brookvale. Tel: 93 1383

DELTA COMPUTERS
12 Fetherstone Street,
Bankstown. Tel: 705 6636

DESKTOP COMPUTERS
10 Cowper Street,
Granville. Tel: 637 3063

VICTORIAN DISTRIBUTOR:
PRESIDENT COMPUTERS VIC
Suite 1, 609 St Kilda Road,
Melbourne. Tel: (03) 529 1788

SOUTH AUSTRALIA DISTRIBUTOR:
PRESIDENT COMPUTERS SA
100 Pirie Street,
Adelaide. Tel: (08) 223 6333

QUEENSLAND DISTRIBUTOR:
PRESIDENT COMPUTERS QLD
416 Logans Road,
Stones Corner, Brisbane
Tel: (07) 397 0888

W.A. DISTRIBUTOR:
W.J.MONCRIEFF PTY LTD
176 Wittenoom Street,
East Perth. Tel: (09) 325 5722

A.C.T. DISTRIBUTOR:
PRESIDENT COMPUTERS A.C.T.
Shop 2, Warrmanga Centre,
Warramanga. Tel: (062) 88 2000

TASMANIAN DISTRIBUTORS:
HOBART COMPUTER CORNER,
202 Elizabeth Street,
Hobart. Tel: (002) 344 044

LAUNCESTON NORTHERN BUSINESS MACHINES
177 Brisbane Street,
Launceston. Tel: (003) 318 388

president

(version III) allows you to 'tag' fields as protected for security reasons.

Calculations

Since Cardbox can handle only text data, it can't do calculations. All the others have some calculating facilities. Those in DMS and FMS-80 are quite powerful — except that you can't modify the order of execution of expressions with brackets! In terms of power and flexibility, dBaseII is easily the best of those I've looked at in this area. Figure 6 shows two tests on calculation. The first involves replacing one field in each record with a value derived from a calculation on another field in the same record. The second simply totals three accumulators to give the sum total of each of three variables.

Security

Important features are the ability to prevent unauthorised access to information, and to see what you've actually done in a session if something should go wrong. DMS and FMS-80 provide the most flexible approach to the first question, while DBMSIII has some comparable features. Looking at recovery from error and detection of possible deliberate fraud, DMS provides a paper audit trail, though I couldn't find a way to store this on disk; dBaseII allows you to record in a file everything which takes place on the screen. Cardbox has an explicit Repair function should the data become corrupted in any way.

Tailoring

All the packages except Silicon Office and DBMS allow you to tailor the terminal interface to your requirements; SO comes already set up for the PET, and DBMS for whatever machine you've specified when ordering. Though some of the packages try to be helpful, this is still a very thorny area for anyone, especially the novice. I'd especially like to see every package make it possible to specify things like cursor movement codes by actually pressing the key concerned, then there's no mistake. Another essential feature is some confirmation that the tailoring has worked — before you leave the terminal definition part of the package. Also, the ability to edit an existing terminal definition, rather than starting from scratch, is another necessity. None of these packages did all these things.

As for tailoring the presentation of the package itself to your needs, this involves a sharp divide between the menu-driven packages and the command-driven sustems. With the menu-driven approach, as used by DMS and Cardbox, if your application fits that method of working you're fine, otherwise you just have to fit yourself round the package. Cardbox, particularly, would be even better if it had the ability to store search sequences for re-use. In the command-driven packages like Condor, dBaseII and Silicon Office, you can set up files containing sequences of instructions to be executed as a group, including instructions for looping which give you the power of a programming language. You can use these features to construct menus of your own. Of the three, dBaseII is the most flexible, and the ability to call other 'programs' with parameter passing is very valuable.

FMS-80 combines these two approaches by providing menus for the simple things and a crude programming language for the more complex. However, unlike the other command-driven packages, the programming language is quite distinct in its syntax from the commands which execute FMS-80 procedures, and I found it quite difficult to use in anger — especially with the lack of comprehensible error messages. dBaseII is pretty unhelpful with its endless 'syntax error' messages, but FMS-80 often just stops and gives no clue as to why nothing has happened.

Relations with outside

Most packages allow you either to read files written by other programs, or to write files for other programs, or both. To do the Benchmarks I had to read an external file, and Figure 7 shows the timings for reading the file in, restructuring it to match my '21 field' format, and doing any necessary indexing.

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
I calculation on 1 field on each record	NA	4m	6m	12m30s	7m10s	25m for 250 records
Star rating	NA	***	****	**	*	**
Totals of 3 fields over 1000 records	NA	1m30s	1m30s	14m	NT	1 hr for 250 records
Star rating	NA	**	****	**	*	**

Fig 6 Calculations on 1000 records

Package	Cardbox	Condor	dBaseII	DMS	FMS-80	Silicon Office
Importing a file	23m30s	13m	13m	39m17s	16m40s	4h45m for 250 records
Star rating	**	****	****	***	*	*

Fig 7 Importing a file of 1000 records

Stability and reliability

One of the more cheering features of these reviews has been the level of reliability of most of the packages — there have been occasional glitches, but very few, and I haven't suffered any loss of data or programs during the running of the Benchmarks. In fact my complaints have been more the other way — that once you have started a process going, most packages prevent you from interrupting it in any way — so if you've made a mistake you've had it. FMS-80 takes the sensible line of asking for confirmation that you want to abort.

User image

Here I'm just going to touch on a few principles, because so much of this area is a matter of taste. Firstly, you need to know whether you prefer menu or command driven systems, and, if the former, whether your needs fit sufficiently one of the packages which only has menus, like DMS and Cardbox, or whether to go for a combined approach, like Condor, dBaseII and FMS-80. Whatever you decide, there is little substitute for giving the package a thorough road test yourself before you buy — though if the price is as low as $298, maybe you can afford to experiment anyway.

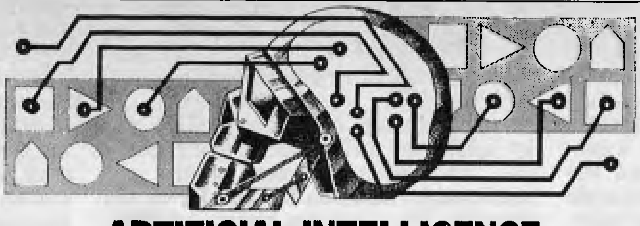
Consistency is, in my view, the most important element in the user image. You need to be able to generalise the way the package works, so that learning about one part makes it easier to use another. Personally, I found Cardbox, dBaseII and Condor the easiest in that respect. Clarity and 'naturalness' is another important virtue, and another pretty subjective one, but there are some rules. For instance, I found the DMS menu — which for example uses the letter 'C' as its prompt for Keyed access, and 'K' for another function — very confusing. I also found the Cardbox three-tier naming very confusing in an otherwise highly 'natural' package.

Finally, taking a look at the manual is always a good indicator of what the package is like. Every package should, I think, have a Tutorial Guide for beginners, a Reference Manual for experienced users and a Prompt Card for reminding people about particular command formats. The Cardbox manual is extremely good in all those areas, I thought; personally I also like the dBaseII approach of an initial manual written by an experienced user to complement the reference manual, with lots of examples — another must. The Silicon Office manual is also, I think very good.

Conclusions

I've tried throughout to view each package as having a niche in the market, and to think in terms of 'horses for courses' rather than plumping for just one or two. I still think that's the right approach, and I hope this article will help you to make up your own mind about what's best for you.

END



ARTIFICIAL INTELLIGENCE

INTELLIGENCE TESTS FOR COMPUTERS?

Bev Mason explains how computers can tackle a standard intelligence test by using the calculus of finite differences, including a program for System 80/TRS-80 computers.

We are all familiar with the tests used to gauge intelligence, particularly with children, where the subject is asked, among other things, to deduce the next term in a sequence of numbers. For example, the given sequence might be 1, 4,9,16, — or 3,5,8,12,—. In the first case the series is obviously a sequence of ascending squares, the next term being 25. In the second case, it is easily seen that the difference between successive terms increases by one at each step, and the next term is 17.

A less obvious sequence is −1,1,19, 65,—, where the next term is 151. Even when the answer, the next term, is known the relationship is by no means obvious. In this case, each term is derived by subtracting four times the square of the term *position* from twice the cube of the position and adding one, ie, the *third* term is equal to 2*3*3*3 — 4*3*3+1=19.

It is interesting to pose the question 'Can a computer be programmed to do these intelligence tests?' When one considers the virtual infinity of relationships which can exist between successive terms of a sequence it would seem that the answer must be in the negative. How can one disassemble the mental processes involved in the 'casting around' for a possible relationship, in order to write a program? Also, if a computer can be programmed to accomplish these tests when computers are undeniably *not* intelligent, do the tests really warrant the name 'intelligence tests'?

Before delving further into these questions, let us say at the outset that a computer *can* be programmed to derive the next term in a sequence, or indeed the nth term, forward or backward, and also display the relationship between the terms. The short program at the end of this discussion does just these things. We shall see some further anomalies!

A simple sequence

Let us suppose a child is given the first sequence above, ie, 1,4,9,— and is asked to write in the next term. This looks like, and is, a sequence of squares, the next term being 16. So when the child puts a 10 on his paper he gets no marks and is considered to be little better than an idiot — perhaps unjustly, as we shall see.

It is necessary to examine the question more closely. We have become so familiar with this type of test that usually only the minimum is stated. Much has to be inferred or 'understood'.

The fact that the next term in a sequence of terms is requested indicates a count. What is really being stated is

y= 1 4 9
n= 1 2 3 4 5

and it is obvious that $y=n^2$ and the term for n=4 is 16; that for n=5 being 25 and so on. We could have been asked to fill in the *preceding* terms — ie, those corresponding to 0,−1,−2,etc.

y = 4 1 0 1 4 9
n = -2 -1 0 1 2 3

The series extends to infinity both ways and the relationship $y=n^2$ holds true. However, n is nothing more than a count, simply marking the position of each term and there is nothing in the original question regarding the starting point of the count. In the above example y has a minimum value of 0 and it is convenient to zero the count at that point. Had the given terms been stated as 9,16,25,—, we could have written

y = 9 16 25 — —
n = 1 2 3 4 5

The next terms would still be 36 and 49 but y would certainly not equal n^2. Common sense, or 'intelligence' would have told us to start the count with n=3 rather than n=1. But suppose that the given sequence had been a series of numbers with no immediately recognisable relationship. The zero point, in the absence of any guidance, would have to be positioned arbitrarily. Let us start a new count 'p' one position further along:

y=1 4 9 —
n=1 2 3 4
p=0 1 2 3

Now n=p + 1. We can see that $y=n^2$ therefore $y = (p+1)^2$, which simplifies into $y=1+2p+p^2$. This is not as immediately obvious as $y=n^2$ but is just as valid. When p=3, y=16, which comes as no surprise.

The child in our example, far from being an idiot, is in fact a genius of rare talent. He had spotted that the relationship between y and p is not only by $y = 1+2p+p^2$ but also by $y = 1+4p^2 - p^3$. In the first case the term for p=3 would be 16 while in the second case it would be 10. In the absence of any guidance from the question he plumped for the 'cleverer' answer and wrote '10'. Had the question sought the preceding term, the child would have written 6 and not 0.

It would seem then that not only can this type of intelligence test be correctly answered by an unintelligent computer, but it can also give a damaging and misleading assessment of a child's intelligence. Perhaps it would be more meaningful if, in addition to requesting the next term, the relationship used in arriving at that term should also be stated.

Before leaving this brilliant, misjudged and probably unbearable child let us not utterly condemn this intelligence test. The computer arrives at the answer by mathematical (arithmetic) means which can, as will be shown, be achieved by humans fairly simply with pencil and paper but not without (except in the easiest or trivial cases). Instead the human uses intuitive mental processes which are undoubtedly facets of intelligence which can be tested.

We shall see, with the aid of our program, that it does not matter in practice where the count is started, whether we count in ones, twos or twenty-threes, or whether the given sequence is three numbers or ten numbers long. It does not even matter if those numbers are generated randomly by our computer! We can still find a relationship between them and state the next (or for that matter the next but sixteenth) number in the sequence.

The 'other' calculus

All of the above is by way of being a lead into a small facet of a fascinating branch of mathematics known as the calculus of finite differences. (The non-mathematical reader need not worry. The remainder of the discussion contains only arithmetic and very elementary algebra.)

The more usual infinitesimal calculus deals with relationships between *continuously varying* quantities. It assumes that one variable changes by an infinitesimal amount and studies the effect on related variables. An elementary application is in the calculation of maxima and minima. Thus, given a relationship such as $y=81/8x+x/2$, it can be shown that y will be at its lowest value when x=4.5. However, if y represents the number of hours needed for a job to be done by x men, this method is of no great value. We cannot employ 4.5 men and there is no indication whether 4 men would be more economical than 5. Since we cannot have fractional men, nor negative men, their number x cannot be continuously variable and the methods of the infinitesimal calculus are not strictly applicable. Hence the calculus of finite differences.

Finite differences

In the 'normal' calculus briefly described above, the variable x is assumed to change by an infinitesimal amount Dx to a value (x +Dx). By subtracting the initial value x from the increased value (x+Dx) we obtain the infinitesimal difference Dx. Since y depends on x it also will change, from y to (y+Dy), and by taking the difference we can arrive at Dy. The calculus then studies the relationship between Dx and Dy or the ratio Dx/Dy, a process known as differentiation. In the same way that the process was applied to x and y, it can be applied to the quantities Dx and Dy and to successive differences: the process is then known as successive differentiation.

When x can assume only integer values the difference between one value and the next can only be finite, and is invariably taken to be unity. The difference in x is therefore 1 and the corresponding difference in y is designated by the Greek capital letter Delta. We shall use the letter D. The process is known as differencing and, if continued, becomes successive differencing.

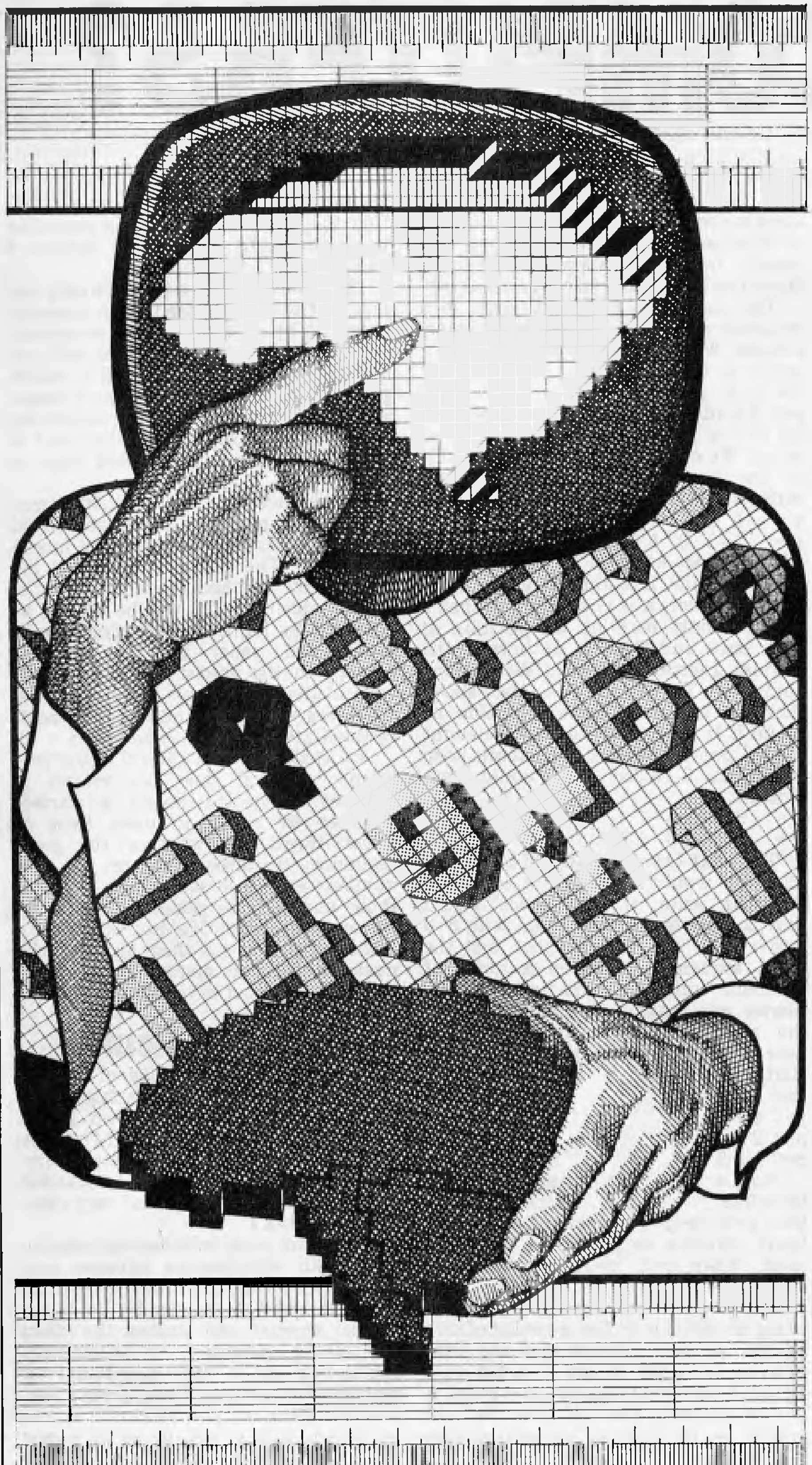


Illustration by Robin MacFarlan

A series of cubes

By way of an example, consider the sequence of cubes:-

y =	0	1	8	27	64
x =	0	1	2	3	4

To start the process of differencing, set the work out in tabular form:

x	y	D	D^2	D^3	D^4
0	0				
		1			
1	1		6		
		7		6	
2	8		12		0
		19		6	
3	27		18		
		37			
4	64				

The column headed D is obtained by subtracting each value of y from the next higher value. In a similar way, the column headed D^2 (second difference, *not* D*D) is obtained from the D values, and so on. The relationship between x and y is then given by the top figures in each column starting with the y column:

$$y = 0 + 1D + 6D^2 + 6D^3 \qquad (A)$$

where

$D = x/1$,

$D^2 = x(x-1)/(1*2)$,

$D^3 = x(x-1)(x-2)/(1*2*3)$,

$D^n = x(x-1)(x-2) \text{ --- } (x-(n-1))/(1*2*3 \text{ -----} *n)$.

Substituting these D values in equation (A) we get

$$y = x + 3x(x-1)+x(x-1)(x-2) \qquad (B)$$

By substituting in this equation values of x=0 to 4 the original values of y are obtained, as are also those for x=5 (y=125), and for x = −2 (y= −8), etc. We know of course that in this example the equation (B) reduces to $y=x^3$.

The theory requires only that x should vary in equal steps. However, differencing is done on the basis that the independent variable varies by unit steps starting at zero. If x does not start at zero, or if it increases or decreases by equal amounts other than unity, it is necessary to introduce a further variable, say p, which does fulfill the necessary conditions. It is a simple matter to establish the relationship between x and p, eg:

y	5	7	12	21
x	13	20	27	34
p	0	1	2	3

y will be obtained as a function of p:

$$y = f(p) \qquad (C)$$

where $p=(x-13)/7$. If the value of y is sought for (say) x=15 it is only necessary to calculate the value p=2/7 and substitute in equation (C).

But what use is it?

Suppose we have obtained experimentally a series of values of y in relation to values of x. If the values of x vary in equal steps we can use the method outlined above to forecast values of y for other values of x. If the values of x are not equally spaced we can plot the points, join them with a smooth curve,

The information race

In a world where the speed of conveying information is so vital, Ampec have the fast, quality printers to help you move with the times.

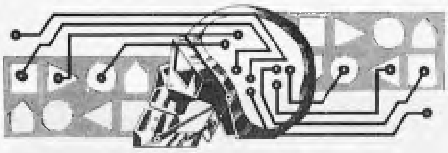
The Ampec Mercury 8510 provides 2" to 9½" paper widths and combines a print speed of 120 cps with high resolution graphics (144 x 144 positions per square inch maximum). It lets you switch select for unidirectional or bidirectional print and all switches for setup selection are placed together.

For greater capacity and performance at a surprisingly low price you can choose the Ampec Mercury 1550. This high-quality matrix printer accommodates up to 15½" paper width and can print up to 231 columns at a fast 120 cps.

Quality printing, quietly and quickly produced, is now within your reach, through the Ampec Mercury matrix printer range.

AMPEC

The absolute in peripheral thinking.



and read off y at standard intervals. The method will then forecast or extrapolate *possible* y values outside the graph. But remember, we get only a forecast, not a certainty.

Suppose our first four results were 0, 1,8 and 27 as in the cube series above. If these were RUN in the program, the computer would correctly forecast 64 and 125 for x=4 and 5 respectively and print the relationship as $y=x^3$. However, suppose the value of y for x=4 were subsequently determined experimentally as 40 and not 64. On feeding the computer the sequence 0,1,8,27,40 it will reply correctly with y=5 for x=5 instead of y=125. The relationship will now be printed as $y=6x-11x^2+7x^3-x^4$. It can easily be seen that the sequence 0,1,8,27 can be followed by an infinity of different numbers of which 64 and 40 are but two.

Our child (I thought we had lost him) could have put any number he liked, positive or negative, after the sequence in his intelligence test, and proved that it was logically related to the earlier numbers.

In the 'conventional' calculus, the opposite operation to differentiation is integration. One method of finding the integral of a mathematical expression is to plot the graph of the equation and measure, or calculate, the area between the curve and the x axis; Simpson's Rule is a method for calculating irregular areas. It is applied by dividing the area into equally spaced strips, measuring or calculating the length of each strip, and applying a simple formula. It can be programmed into home computers to perform the function of integration. The division of the area into finite steps or strips suggests the calculus of finite differences, and in fact Simpson's Rule and similar methods are derived from that calculus. However, the derivation of Simpson's Rule is too complex to be described here, but its application in computers to calculate definite integrals is simple.

The universe operates in a continuous and smooth fashion — eg, time and planetary motion — and the infinitesimal calculus has been universally applied. However, there are those who maintain that the calculus of finite differences offers a more practical picture since the world operates in a digital manner: a view which should be popular with computer people who can more easily appreciate things operating a 'bit' at a time. It is obvious that the number of men needed to do a job, or the number of bricks required to build a house, must be integers. It is perhaps not so obvious that the 'smooth' speed of a car is related to the integer number of engine firing strokes per second: one cannot have a fraction of an explosion. Even the classic problem of water running into and out of a bath could be stated in terms of whole numbers of molecules.

When is a random number not?

When I asked my computer to generate four random numbers between 1 and 20 it replied with 8,2,5,17. I set them as a series:

y = 8 2 5 17
x = 0 1 2 3

and fed them into the program. The computer showed that the numbers are related to the expression $y = 8-10.5x + 4.5x^2$ and that the next number in the 'series' would be 38!

The program

The program is written in Microsoft Basic for use with the Z80 based System 80 computer. It occupies about 1.2k of memory, which could be reduced considerably by compression. In the absence of DIMension statements the size of the arrays D(N), E(N), X(N), Y(N) is automatically set to 10 and that of the array C(N,M) to 10*10. The program will therefore accept a sequence of up to ten numbers, as printed, and will require a further 0.8k of memory to RUN. This requirement can be reduced considerably by restricting the sizes of the sequence and of the arrays by including correspondingly suitable DIMension statements.

Lines 10-70 are self-explanatory
Line 80 determines the interval of finite differences between the x values.
Lines 120-170 calculate the differences between the successive y values, setting the differences as a new series of y values and repeating the process. The intial values of y and of the successive difference series are set into the array D(O) — D(N).
Line 180 establishes where x falls on the p scale, zero p falling on the first y value.
Lines 210-270 evaluate the successive terms of the relationship y=f(x) and add them together to give the value of y.

As described above, the program deduces and evaluates expressions such as: $y=x+6x(x-1)/1*2)+6x(x-1)(x-2)/(1*2*3)$

This expression is equivalent to the much more convenient and understandable form $y=x^3$. In writing a program to convert from one form to the other it is necessary to overcome the computer's inbuilt tendency to substitute numerical values for algebraic symbols. After all, the computer was designed to do arithmetic and not algebra.

Lines 300-560 are therefore a rather tedious way of reducing the first untidy algebraic forms into ascending power forms, and printing the result. If preferred, these lines could be replaced with a very much shorter version to print the relationship in the original untidy form. Note that System 80 language P[3 means P cubed.

END

```
10 INPUT "HOW MANY VALUES ARE KNOWN":V1:V=V1-1
20 INPUT "LOWEST VALUE OF X"; X(0)
30 INPUT "CORRESPONDING VALUE OF Y"; Y(0)
40 FOR N = 1 to V
50 INPUT "NEXT VALUE OF X"; X(N)
60 INPUT "NEXT VALUE OF Y"; Y(N): NEXT N
70 INPUT "FOR WHICH VALUE OF X IS Y REQUIRED"; X
80 DX = X(1) - X(0)
90 IF X(V) <>X(0) + V*DX PRINT "INCORRECT DATA": GOTO 10
100 IF X - INT(X)<>0 PRINT "INCORRECT DATA": GOTO 10
120 D(0) = Y(0): N = 1
130 IF N = V + 1 GOTO 180
140 FOR N1 = 0 TO V - N
150 Y(N1) = Y(N1 + 1) - Y(N1): NEXT N1
160 D(N) = Y(0)
170 N = N + 1: GOTO 130
180 P = (X - X(0))/DX
210 Y = D(0)
220 N2 = 1
230 Q = 1
240 FOR N1 = 1 TO V
250 N2 = N2*N1
260 Q = Q*(P - N1 + 1)
270 Y = Y + D(N1)*Q/N2: NEXT N1
280 CLS: PRINT:PRINT:PRINT "WHEN X="X"Y="Y
290 REM 'EVALUATE ALGEBRAIC RELATIONSHIP'
300 C(1,0) = 0: C(1,1) = 1
310 FOR N = 1 TO V
320 C(N,N+1) = 0: NEXT N
330 FOR N = 2 TO V
340 FOR R = 1 TO N
350 C(N,R) = C(N - 1,R - 1) - (N - 1)*C(N - 1,R)
360 NEXT R: NEXT N
370 N2 = 1: E(0) = D(0)
380 FOR N = 1 TO V
390 N3 = N2
400 FOR N1 = N TO V
410 E(N) = E(N) + D(N1)*C(N1,N)/N3
420 N3 = N3*(N1 + 1): NEXT N1
430 N2 = N2*(N + 1): NEXT N
440 PRINT "Y = ";
450 IF E(0) = 0 GOTO 470
460 PRINT E(0);
470 IF E (1)<0 GOTO 500
480 IF E(1) = 0 GOTO 510
490 PRINT "+" E(1) "P";: GOTO 510
500 PRINT E(1) "P";: GOTO 510
510 FOR N = 2 TO V
520 IF E(N)<0 GOTO 550
530 IF E(N) = 0 GOTO 560
540 PRINT "+" E(N) "P["N;: GOTO 560
550 PRINT E(N) "P[" N;
560 NEXT N
570 PRINT:PRINT:PRINT "WHERE P =(X-" X(0)")/"DX
```

Listing

MORE THAN GAMES FOR VIC*20

VICeducation

Dear Reader,

We don't sell the VIC-20 computer but we think it is the best machine for families that want to learn about the new technology while having a lot of fun.

The VIC is a powerful computer. We have now used it to produce educational programs that make VIC much more useful at home or school. Ask your VIC retailer to show you VICeducation. Compare the computer and the VICeducation programs with the rest of the market.

VIC + VICeducation = the best available. You pay far less and get much more. Remember when you buy a computer to check the programs available. Check the rest (where it exists) against our features:

- maths and language development from kindergarten to teenage;
- programs made in Australia with Australian spelling and teaching;
- programs with colour, sound and graphics, that are fun to use, that kids want to play over and over;
- programs that cost about $14.

Yours sincerely

Margaret Gray and Dennis Argall

P.S. If you can't get to a VIC retailer order from us by phone day or night. A thirty second message to our answering machine after 9pm will cost no more than postage from anywhere in Australia. Quickly state your name, telephone number, address, bankcard number and expiry date, then item numbers from this list.
Price $14.40 per item including postage.

M-01 Young Maths - ages 5 to 10
M-02 Multiplication and series - 6 to teens
M-03 Addition, subtraction and numberline - 6 to teens
M-04 Addition and subtraction tutor - follows M-03
M-05 Division tutor - follows M-02
M-06 Multiplication tutor - follows M-02
L-01 Spellstart 1 — ages 6 to 9
L-02 Spellstart 2 — one thousand words in programs
L-03 Spellstart 3 — for vocabulary and spelling
L-04 Spellstart 4 — development.

© ALL RUN ON UNEXPANDED VIC20

Telephone (062) 310494 or write to

VICeducation

P.O. Box E230 Canberra 2600

BENCHTEST PERSONAL COMPUTER

EPSON HX20

THE ALL-IN-ONE PORTABLE

Dick Pountain Benchtests a remarkable new portable computer from one of Japan's most famous manufacturers

I said my piece on the future of hand-held computers in last month's review of the Hewlett-Packard HP75C and so will not take up space repeating it here. Suffice to say that the subject of this review is the second general purpose fully portable microcomputer to reach the Australian market; by general purpose I mean that it has the speed, memory and other facilities to take on tasks hitherto reserved for mains-powered machines and is not limited to mathematically oriented problems.

The HX-20 is also the first computer to be seen here from Epson, the Japanese firm which has achieved a remarkable dominance of the micro printer market in the space of three years. Epson, which is owned by the giant Seiko watch firm, has sold computers in Japan for some time but this is the first to be exported, and it will be followed soon by a series of desk-top CP/M machines.

Hardware

As delivered, the HX-20 is a self-contained battery-operated computer with 16k of non-volatile memory, a 32k

Side and rear views show various peripheral sockets and controls

Photography by John Mason

Basic in ROM and built in four-line display and printer.

The outward presentation of the HX-20 is very neat and very Japanese. The case is moulded in that metallised silver plastic which has become a trademark of Japanese consumer electrical goods; the moulding is robust and intricate with many embossed legends and removable panels. All the screws which hold the assembly together go into moulded-in brass collets rather than naked plastic.

The case is exactly A4 sized (the size of this page) and the depth tapers from an inch and a half at the front to two inches at the rear. As you will quickly grasp, this is not a pocket-sized machine nor, at 1.6kg, is it even 'hand-held' in the traditional sense; you would need to support it in a crooked arm. Epson is deliberately calling it 'portable' not 'hand-held', but I suspect that 'lap-held' is the way most people will use it in the absence of a table; it will, however, slip into an ordinary briefcase without trouble.

The display is a 20-character by 4-line LCD unit (Epson leads the world in production of liquid crystal displays) which has no calculator type annunciators; it is treated exactly like a VDU screen of 120x32 dots. This screen is in fact only a 'window' onto a virtual screen whose size is user definable up to 255 characters by 255 lines and so can be scrolled in all four directions. A most unusual feature of the display is that the angle of view may be changed by adjusting a control on the side of the case; you sit in the position in which you want to use the machine and turn the knob until maximum contrast is achieved. Most previous LCDs can only really be viewed from square-on.

The keyboard is of standard ASCII layout and is not just good for a portable computer but good, full-stop. The keys are of low cushion shaped profile with recessed tops and have full-sized typewriter pitch, travel and a satisfying feel and feedback. The keyboard is fully debounced and appears to have a type-ahead buffer, which is useful as liquid crystal displays have a noticeable lag compared to a VDU. All the main keys have auto-repeat. TAB, CTRL and CAPS LOCK keys are provided, though the latter regrettably has no indicator to show that it's engaged. There is no numeric keypad but a shift key called NUM will turn a block of the main board into one for data entry. Another shift key called GRPH produces block graphics and symbols. Editing keys include CLR/HOME, INS/DEL and four-way cursor movement, but irritatingly the latter is done by two keys, cursor up and down being shifted. Along the top of the main keyboard are five programmable function keys, the PAUSE and BREAK keys and a key called MENU of which more below. In short this is a keyboard, for the first time on a portable, upon which I'd be happy to type all day. My only real criticism is that the keys are mounted directly onto a PCB which bends slightly under pressure (like the early Acorn Atoms); it would feel more robust if more ribs were cast into the case bottom to support it.

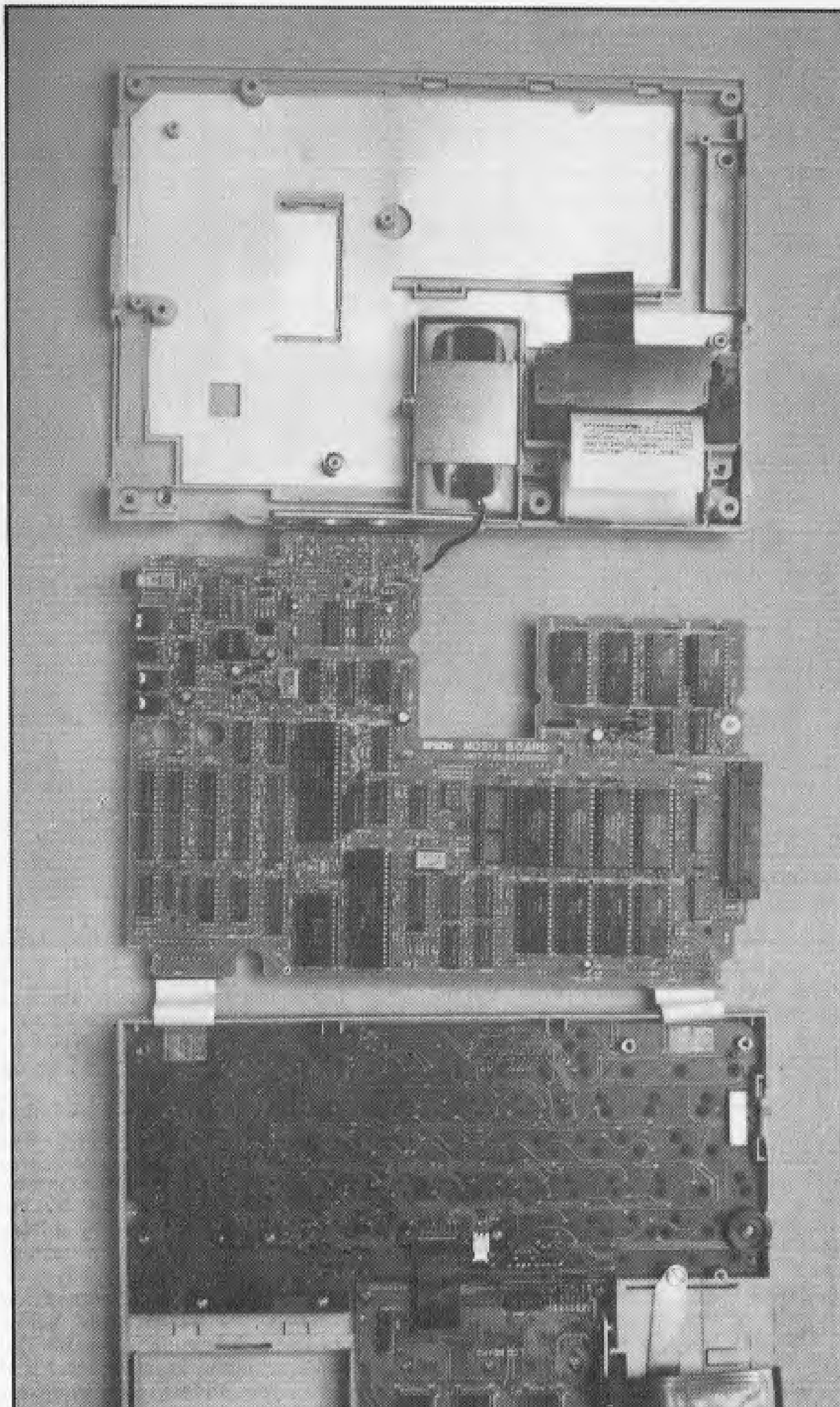
To the left of the display sits the built-in miniature dot-matrix printer. This unit uses rolls of plain paper and ribbon cartridges which are perfect

EPSON HX20

Lilliputian copies of the MX-80 ones; the review machine had purple ink in it. The 2¼in paper accommodates 24 characters on a line and it can print the whole HX-20 set including the block graphics characters, as well as performing dot graphics. Operation is either under program control using LPRINT or manual, screen dumps being possible using CTRL and one of the function keys. The printer has its own on/off switch so that programs which have printer output can be debugged without wasting paper; it also has a manual linefeed button.

Printing speed is not fast at around 40 lines/minute but is comparable to other units of this size, while the print quality is much better than electrostatic 'silver paper' printers. Paper rolls are spindle-less and merely drop into a cavity under a hinged trapdoor, while changing cartridges is a cinch. The plain paper rolls are a standard size and so many office calculator rolls should work. They think of everything.

A third type of output is revealed by a grille just below the display. This conceals a piezoelectric beeper which is programmable for pitch and duration.



Not sure what the warranty department would think of this treatment!

Around the sides of the case are various sockets for connecting peripherals. The back panel holds two RS232C ports which have different DIN plug connectors. One is for communication with terminals, printers, other computers and modems at up to 4800 baud and has an 8-pin DIN socket. The other is for communicating with a TV or monitor via an adaptor, or to disk drives; it has a 5-pin DIN socket and a maximum transfer rate of 38,400 baud. At the left side is a parallel connector for the expansion RAM box, while at the right are mini-jack sockets for an external cassette recorder with remote control, and a bar-code reader. Finally, a socket next to the RS232 port allows an AC/DC power supply to be plugged in to recharge the NiCad batteries. These have a capacity of around 50 hours (they didn't need charging during this test) and take eight hours to charge; the computer can be used on the mains unit while they are charging but Epson warns not to use it this way once they are recharged as overcharging can shorten their life. I was rather suprised that the HX-20 doesn't auto-power down after a set interval as do most other battery-operated machines; best remember to switch it off.

The on/off switch is at the right hand side, as is the reset button; the latter is required only when a run-away program does not respond to the BREAK key or the on/off switch, which is only likely with machine-code programs. It does not perform a true cold start (which is done through software) and preserves all memory contents.

On a standard HX-20 the area to the right of the display is blank. It contains a dummy module which can be removed (using a very ingenious lever at the back) and replaced by either the optional microcassette drive or by ROM modules containing software. None of the latter were available for the test but the cassette drive was. This is a digital, not audio, device and uses standard microcassettes. It reads and writes at a fast 1300 baud and gets around 50k onto a 30 min tape. Although it has a tape speed of 2.4cm/s, which is the same as most pocket dictaphone recorders, Epson does not recommend swapping tapes between recorders. A really nice feature is the software tape counter which allows fast winding to the exact location of a program before loading; the tape position at any time can be read by a program. Using this feature cuts the time wasted searching tapes enormously and approaches the convenience of disk. The tape drive can be operated manually as well as under program control by pressing CTRL and the first function key. This turns all the function keys into tape deck controls, including one for zeroing the tape counter (which is displayed on the screen in this mode).

The ROM packs are unusual in that that they are accessed by a high-speed serial link and are treated, like other serial mass-storage, as a sequence of files rather than as part of the memory-map. They will be available in sizes from 8 to 32k.

Getting inside the case is quite easy and reveals some surpises. It is no surprise that the manufacturing techniques are 'state-of-the-art'. The design is highly modular (the printer comes away as a sub-unit by undoing two screws) and uses two circuit boards, one for the logic and one for the keyboard switches. These are interleaved with sheets of a curious flexible metallised card which doubles as insulation and RF shielding. All intercard connections are by FCPs (flexible printed connectors) which are held by ingenious sliding grippers rather than those precarious sockets used on some units; they can be refitted over and again without damage. An interesting little anecdote; after I took the HX-20 to pieces for that 'squashed hedgehog' photograph, I reassembled it and the RAM contents were still intact! The 16k of on-board CMOS memory is in eight 16k chips while the ROM is in four 64kbit chips which are socketed; there is a spare socket for 8k of ROM expansion.

The architecture of the HX-20 is unorthodox, to say the least. It has two CPUs, which are related as master and slave. They are both 6301 CMOS devices of Epson's own manufacture and have 4k of mask ROM and 128 bytes of RAM on the chip; they are reputed to be code-compatible with the Motorola 6800. The master CPU controls memory, keyboard, display, clock and bar-code reader using the external ROM while the slave CPU controls the printer, cassette recorders, RS232 and high-speed serial ports and low-battery power-off function using its on-chip ROM. Most unusually, the two CPUs talk to each other by the 38,400 baud serial link rather than parallel. One consequence of this is that the memory map is, to say the least, complicated.

There are plenty of other oddities which space prevents me from describing; one I must mention is that both processors go into a 'sleep' state to save power when not being used and must be woken up when required. A glance at the Benchmark timings will show that they are by no means fast (50 percent slower than the HP75C) but are no disgrace.

The overall impression given by the HX-20 hardware is that here is a level of professionalism in design and manufacture which would give me nightmares were I an occidental computer manufacturer.

Firmware

The HX-20 operating system sits in ROM along with the Basic. Since all file handling for the cassette drives is done through Basic the operating system has very little to do with the user; it is responsible for the Menu selection function, managing the partitioned memory, cold and warm starting Basic (which includes setting the system clock) and responding to requests for manual control of the printer and cassette drive. Entry of machine code programs is done through a separate monitor which runs under the Menu function.

When you switch on the HX-20 the first reponse is always to present the menu on the display. The very first time you do it, it reads:

CTRL @ Initialize
1 MONITOR
2 BASIC

What this means is that to perform a cold start (which clears all memory contents and allows setting of the system clock) you must press CTRL @. Cold start also sets default values for user RAM size, amount of RAM file space and assigns 10 Basic keywords to the function keys. If you do this you will be prompted for the time and date; once you've entered these you will be returned to the above menu. You now have a choice of pressing the 2 key to enter Basic or 1 to enter the monitor.

Later on, when you have written some programs, up to five of these may be added to the menu so that they can be run immediately upon switching on the machine. The user RAM is divided into five partitions, called P1-P5, and each one may contain a separate program (or none). The partitioning is totally transparent to the user and you will only know it's happening if the total size of all your programs exceeds the total user RAM.

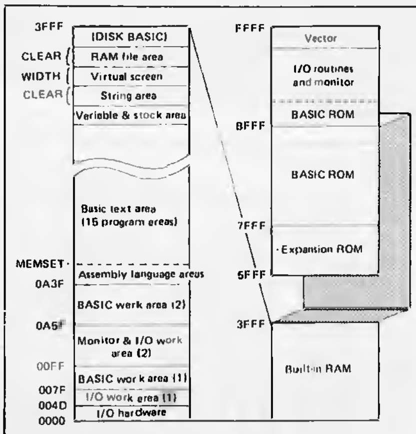
Unlike the Sharp and HP machines, there is no calculation mode; to perform calculations you must enter Basic and use PRINT 2+2, etc. Time and date can be displayed by pressing SHIFT and function key 1 from Basic but there is as far as I can discover no alarm function nor any of the sophisticated calendar functions of the HP75C; you can of course write your own alarm routine using the beeper.

Basic can be warm-started at any time (you must BREAK out of any running program) by pressing the Menu key; this returns the menu to the screen, and resets the default values for various system parameters such as the size of the screen but doesn't wipe out your programs. When you re-enter Basic, though, it is always into partition P1 no matter what you were in before warm start. In case you're wondering, no, you don't have to warm start to change program areas; that can be done from within Basic or even from within a program by the LOGIN x command which puts you into Px.

The monitor is quite a powerful program which has the ability to set breakpoints and change the contents of processor registers as well as the usual functions of dumping and changing blocks of memory and loading and saving binary files to tape. Coupled with the Basic MEMSET command which allows you to reserve space below Basic for machine code programs (the manual continually refers to them as 'Assembly Language' programs) this would be a very useful facility if only the instruction set were to be found anywhere in the documentation, which it isn't. If it is true that the processors are 6800-compatible this may not be a problem; I didn't have the time or the 6800 experience to find out.

One very neat feature of the monitor I should mention is the K command which lets you set up a 'boot' program which is automatically executed when power is switched on; if you wish it could be Basic itself so that you bypass the menu.

The amount of memory available to the user on the HX-20 is a whole story in itself. It depends not only on the setting of MEMSET but on space allocated for RAM files (see later) for strings, and on the size of the screen which you've defined. Of the 16k standard memory, the bottom 2624 bytes are used by the system (so this is the minimum value for MEMSET) and the most which can be allocated to Basic programs is 13,398 bytes with a 20x4 screen and no string or RAM file space. On warm start the default memory size is 12,891 bytes; adding the expansion memory pack would give 29,275 free bytes so the HX's advantage over the HP785C is not quite so great as it first appears in this respect.



Master CPU Memory Map

Although the business of controlling screen sizes and graphics modes is done from Basic I propose to deal with it here since the default values are set by the operating system.

There are two different screens available on the HX-20; a text screen and a dot-addressable graphics screen. There are two devices on which these screens can be displayed, namely the built-in LCD and an external TV or monitor. The text screen can be a 'virtual screen' bigger than the device, so that the physical device is a moveable window onto it. On the LCD both screens can be displayed superimposed on one another; on the TV you can show only one or the other but the graphics can be in colour. When you have picked the bones out of that lot you will realise that there is much to learn about the HX-20 screen, and I don't propose to teach you all of it here.

The size of the text screen is set by the WIDTH statement; for the LCD it can be up to 255x255 depending on available memory. For the external TV the limit is 40x37, of which the actual screen shows 32x6 lines. Since the TV adaptor was not available for test I'll concentrate on the LCD. When the cursor reaches the edge of the actual screen it 'drags' the window with it to the edge of the virtual screen. Alternatively you can move the window itself using control characters, in which case the cursor stays where it is. The cursor or the window can be sent to any point on th virtual screen using LOCATE or LOCATES x,y. It is possible to define margins round the actual screen so that that cursor starts to 'drag' the window before it reaches the edge; this makes life much easier as you can see if you're going to overwrite anything so the system defaults to a margin of four characters unless you tamper with it. In practice you soon get used to moving this window in two dimensions over a bigger screen and it becomes second nature. In Basic an excellent full screen editor is provided (with insertion and

Choose the System the Experts Are Raving About — Tandy's TRS-80 Color Computer!

"The Color Computer is THE best buy in the market today. It's jam-packed with hardware capabilities and the graphics commands of Extended Color BASIC are excellent. Look at it!"
— William Barden, Jr., Author

"Tandy's Extended Color BASIC is a breakthrough in colour graphics for personal computers. It's fast, easy-to-use, and capable of producing striking graphics."
— Stan Miastkowski
BYTE Magazine

"Anyone seeking a machine that will serve well as a teacher and a base for a home communication and entertainment centre will find the TRS-80 Color Computer a strong contender."
— Carl Warren
Popular Electronics

"The TRS-80 Color Computer has the most potential of any computer on the market today; more versatile, more expandable, easily interfaced, and easier to use — bar none!"
— H.C. Pennington, Author

The experts agree ... the Tandy TRS-80® Color Computer lets you create spectacular high-resolution colour graphics at an amazingly low price!

A Programmer's Dream at Only $849. Our 16K Color Computer with Extended Color BASIC uses one-line commands to make the creation of sophisticated graphics easy. Now you can produce colour diagrams, business and engineering charts, even simple animation with a screen resolution of up to 49,152 programmable pixel points (256x192). You get eight vivid colours, sound, a 32x16 screen format, editing, user-definable keys, error messages, PEEK, POKE, a built-in RS-232 serial interface, and much more.

We Don't Skimp on Documentation. Our easy-to-understand tutorial manuals on our Standard and Extended Color BASIC will quickly have you using the full potential of your TRS-80 Color Computer.

Tandy
COMPUTER CENTRE

The biggest name in little computers™
A DIVISION OF TANDY CORPORATION
Retail prices may vary at individual dealer stores.

Instant Fun and Games! Just plug-in a Program Pak™ for exciting game action! Our TRS-80 Color Computer connects to any standard TV set.

Expand At Any Time. Add a disk drive for only $699, and get 156K-bytes of on-line storage. Add more memory, a printer, and other accessories, too.

See It Today! Come in for a demonstration at a Tandy Computer Centre, store or dealer near you.

Tell me more! Please send me your free TRS-80 Computer Catalogue.
Tandy Electronics, Advertising Dept,
PO Box 229, Rydalmere 2116. 324070

Name: ____________________

Company: ____________________

Address: ____________________

____

City: __________ State: __________

Postcode: __________ Tel: __________

EPSON HX20

deletion) so if you define a big enough virtual screen you can scroll upward through a listing or re-use commands previously entered. In doing this it pays to be careful that nothing else is lurking on the same line but off the screen!

The graphics screen is 120x32 dots for the LCD and 128x96 mono or 128x64 in four colours for the TV. If you use colour, however, a bizarre addressing mode prevails where there are 64 physical dots vertically but 96 addresses, so that either 0,0 or 0,1 will light the dot at 0,0 but only 0,2 will light 0,2 and so on alternately. Personally I never cared much for colour anyway, sniff...

Software

Epson's Basic (originally called MFBasic but now EBasic) is a pretty substantial implementation of the language. It is much more orthodox and Microsoft-like than HP's 75C Basic but lacks the latter's modular programming facilities.

To write a program, select Basic from the menu and then either write it in P1 which comes up automatically or LOGIN to one of the other areas. AUTO line numbering, DELETE and RENUMber are all provided. To copy a program to another area use PCOPY x. If you wish your program to be added to the menu then TITLE it and it will automatically become the next item; the menu number is not necessarily the same as the P number but making a menu selection puts you into the right partition anyway. A program with a title cannot be NEWed or over-written until you give it a null title with TITLE "", which affords very valuable protection. STAT gives the title and size of a program and STAT ALL gives this for the whole lot and throws in the value of MEMSET and the RAMFILE space; a nice touch for homesick CP/M users. The function keys are easily set up by 'KEY number, "string" ' and the assignments can be inspected with KEY LIST or KEY LLIST for hard copy.

Error messages are rather more terse than I would like, but at least they have some mnemonic value and are not mere numbers; eg, SN ERROR (syntax), PP ERROR (protected program). There are over 60 of them including the intriguing 'Unprintable error'!

The HX can handle decimal, hex and, surprisingly, octal integers plus single and double precision (16 digit) reals. Variable names are up to 255 chars with a whacking 16 significant which should be enough even for the purist. Full type conversion functions are provided, type being indicated by the suffixes %, !, # and $. Arrays may have more than two dimensions.

The Boolean operators are remarkable for the inclusion of IMP(implication) and EQV(equivalence) in addition to the normal AND, OR, NOT and XOR.

Most of the rest is standard Microsoft, a single line DEF FN and IF... THEN... ELSE and the normal string functions including INSTR. INPUT is supplemented by LINE INPUT, which takes 255 characters without delimiters, and the unusual INPUT$ which reads a specified number of characters from the keyboard or a file and waits until they're all delivered. A powerful PRINT USING is supported to format numeric output. Error trapping by ON ERROR GOTO/GOSUB is included, too.

File handling is fairly orthodox, with a couple of nice flourishes such as LOF, a function which returns the byte length of an open file. The latest thinking on device independence has been implemented; files have a 'descriptor' in the format 'devicename: filename' (filename optional) and can be directed to another device by changing the name. Recognised devices are keyboard, screen, tapes, ROM packs, RS232 ports, and internal printer; floppy drives are not recognised as a new Disk Basic will be required to run them.

Program files may be saved in two forms — compressed binary or ASCII. All the files so far mentioned are sequential. Random files are supported in the form of RAM files. These sit in memory and can be written and read randomly by a program from any of the partitions as they occupy their own protected area. This area is set aside by issuing a CLEAR statement; as well as clearing numeric variables this can take parameters which set the amount of string and RAM file storage. Once allocated, individual files can be defined in this area using DEFFIL, which sets the record length and the offset in bytes from the beginning of the RAM file area of the first record. Data of any type may be mixed in one record. As RAM files do not have names it's up to you to keep track of what starts where; there are no absolute addresses to remember, only offsets. They are very handy for permanently storing look-up tables of much used constants or conversion factors as a means of passing parameters between programs.

Finally, while on the subject of storage, the cassette control statements are exceptional. TAPCNT is a system variable which keeps the tape counter reading. WIND xxx fast winds the tape to any given place xxx and MOTOR switches on or off an external cassette recorder. FILES will give a directory of the names and types of files on a tape, or any other storage device including a ROM pack.

The graphics commands are confined to PSET to plot a point, PRESET to unplot it and LINE to draw lines, plus a function POINT which tells if a given point is lit.

All considered, this is a very powerful Basic, biased if anything towards business use. It will be immediately familiar to programmers used to MBASIC, which is no bad thing.

No applications software was supplied with the machine, which was one of the first final production models with export ROMs in this country.

Documentation

I was supplied with a Xeroxed copy of what looks like a fairly finalised user's manual and the technical manual. Both are excellent, somewhat to my disappointment as I have relied for several years on Japanese manuals to inject a little humour into otherwise turgid Benchtests. These really are different. For a start they're in English with only the very occasional hint they've been translated.

The user's manual starts with a straightforward and comprehensible account of the machine's features and how to get it running. This leads on to the main section on EBasic which is laid out in the same format as Microsoft's MBasic manual and is clear and comprehensive. There follows a good chapter on machine code programming (apart from the aforementioned absence of instruction set) and file handling. It ends with numerous appendices of character codes, control codes, error messages and an index by function of the Basic reserved words and functions. There is no alphabetic index to the whole manual, though.

The technical manual is remarkably detailed, going through all the hardware down to the signal and timing level and ending with full maintenance and repair instructions which are intended for the dealer and repair shop. If you're a hardware junkie make sure you get this manual as well as the user one.

To summarise, a quantum leap forward from the stereotype of inscrutable Japanese documentation.

Expansion

There is room for 8k more of ROM on the main board, which may be used for alternative language character sets. An expansion module which fits externally onto the left side of the machine contains 16k of RAM and a further 16k ROM.

ROM modules which fit in the place of the microcassette drive are somewhat different from ordinary ROM in that they are treated as file-oriented read-only mass storage.

A TV/monitor adaptor should be available by the time of the launch in February, and a bar code reader for stock control applications is also pro-

Technical data

Processors	Twin 6301 (CMOS) 0.6MHz
Dimensions	290mmx215.5mmx44mm
Weight	1.6kg
Power	Rechargeable Ni/Cad cells 4.5-6.0V
RAM	16k CMOS standard. 32k expanded
ROM	32k
Display	20 col x 4 row LCD; 120x32 dot graphics
Keyboard	ASCII standard 68 keys, including 5 programmable function keys
Printer	Micro dot-matrix 24 chars/line, 144 dots/line, 150ms/dot line.
Ports	RS232C up to 4800 baud, High speed serial up to 38,400 baud
Language	EBasic

mised which will be HP compatible.

The RS232 interface can be used to attach a full-sized printer such as Epson's own MX-100 for which the protocols are built-in (though the interface is programmable to accommodate other devices). An acoustic coupler, the CX-20 will be available next year though Warburton Franki, Epson's Australian distributor, assures us that most commonly available couplers would work with the HX-20 making it a powerful roving terminal for sales teams and the like.

Floppy disk drives are to be produced; twin 5¼in drives with 328k per drive. They will be controlled by the TV adaptor, but don't expect to see them before 1983.

An obvious question given the excellent keyboard is 'Can I word-process on it?' The answer is probably. Warburton Franki are talking to both Epson USA and UK who are both developing applications software. More than that I cannot tell you.

Other software products which are in preparation include a spreadsheet program called 'Epsoncalc', a Sales Order package which takes orders, gives receipts and can communicate to the base computer, and 'Personal Office', which is a calendar/diary/alarm program with card file style data storage facilities; this should close the gap with the HP75C referred to above.

Conclusion

I feel rather shell shocked having reviewed the HX-20 so soon after the excellent HP75C. Both machines advance the art of portable computing beyond recognition – if in subtly different directions. The Epson is rather more like a conventional desk-top micro in its functioning than the HP is. It is aimed either at the business user who will buy ready made software on tape or in ROM packs, or at the owner-programmer who will write most of his/her own software. There is not much you can do with the naked machine – unlike the HP which has lots of 'super-calculator' firmware for the busy executive. On the other hand, the Epson is at least the equal of the HP in power and leaves it standing in the quality of its keyboard and the convenience of its built-in printer and tape drive. It is also cheaper.

When a decent word processing package can be had for it then writing on the train or aeroplane can become an affordable reality at last, even if you do have to take it home to print it. For journalists and for business letters 32k is quite enough memory for away-from-base use to be a practical proposition.

For tasks such as stock-control, sports result calculations, and sales records as well as general engineering and scientific work it is an excellent tool and an impressively well made one at that. Given Epson's market penetration (how many computer shops do you know who don't sell Epson printers) I expect to see a lot of them about in 12 months' time and this also bodes well for a supply of third party software; from previous experience of Japanese firms I wouldn't expect a lot of software support from the manufacturer but Epson does seem to be trying to do things differently. The choice of a 6800 look-alike chip is rather unfortunate as it will definitely hinder the conversion of many existing programs but I expect it to open up whole new areas of applications where this is not so relevant.

I wouldn't mind finding one in my stocking, anyway.

Benchmark timings

BM1	2.7
BM2	15.3
BM3	33.1
BM4	32.8
BM5	35.3
BM6	59.1
BM7	100.6
BM8*	133.3

All timings in seconds. For a full explanation of Benchmarks see APC Vol 3, No 9, Sept. '82.

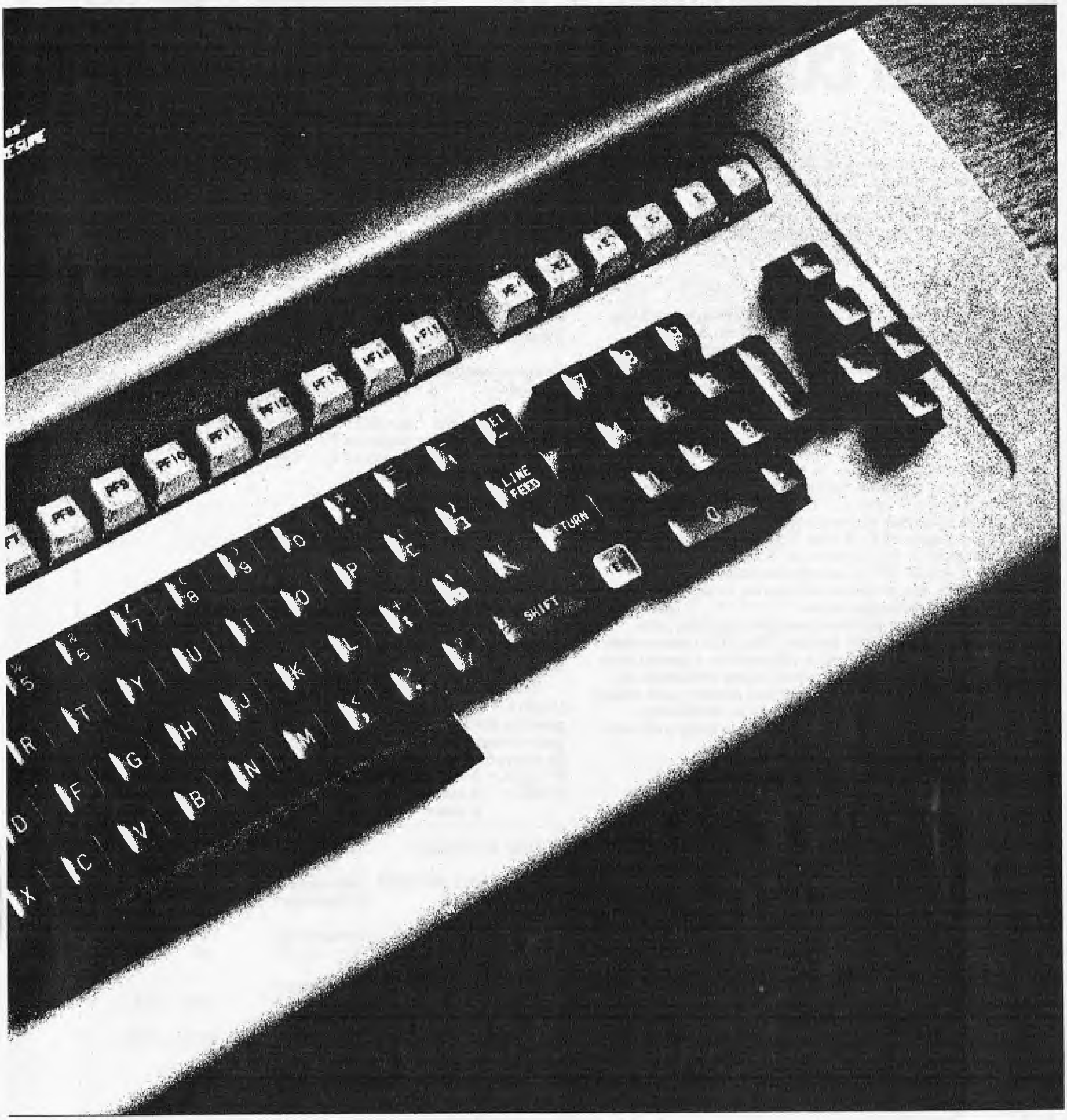
** Note: BM8 now counts 1000 loops.*

Prices

Epson HX-20 (including travelling case)	$1296.00
Microcassette Drive	$200.00
Expansion Memory Unit	$214.00
ROM cartridges	$74.00
TV/Monitor Adapter	na
Barcode Reader	$229.00
Acoustic Coupler CX-20	na
Rolls of printed paper	$3.37
Ribbon cartridge	$3.10
A/C Adaptor	$17.00

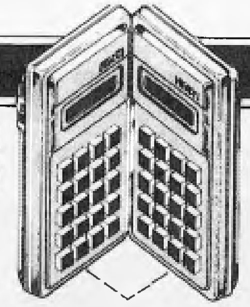
All prices include Sales Tax.

END



For further visitor and exhibitor information about The 1st Australian Personal Computer Show, please contact Rachel Tetley, Australian Exhibition Services Pty Ltd, 1250 Malvern Road, Malvern, Victoria 3144. Telephone: (03) 20 1208. Telex: AA 39329.

PC-1211 COMPLEX ARITHMETIC



The new HP-15C, reviewed in this column in September, has built-in complex number arithmetic — probably for the first time on a programmable. It is quite possible, however, to program this feature on most machines, and the following is a particularly nice implementation for the Sharp PC1211 by John Kerr.
Dick Pountain

This program converts the 1211 into a calculator operating on complex numbers. It is run in the DEF mode. 'shft Z' creates storage space for up to 18 complex variables, labelled 'A' through 'R'. The assignment statements allowed have the general form

```
variable name (A. . . R) shft = operand  enter  operator  enter
                                operand  enter  operator  enter
                                   :       :       :        :
                                operand  enter  enter
```

Operands are of two types. A type 1 operand is a variable name, which may be preceded by either or both of the monadic operators '—' and 'C'. These denote complex negation and conjugate respectively; eg, '—CF' represents minus the conjugate of F. A type 2 operand is a numerical input. Examples: 3—j4 is entered as '3 shft J—4'; 5 ∠—53° is entered as '5 shft K—53' (although the latter assumes DEGREE mode, the program will operate in any angle mode). Legal dyadic operators are '+', '—', '*', '/' and '∧'. The first four of these represent complex addition, subtraction, multiplication and division. The fifth raises a complex number to a real power; it is followed by a special type of operand (a REAL operand), which is any arithmetic expression. Dyadic operators all have equal priority, and evaluation proceeds from left to right (there is no bracketing facility). Monadic operators have a higher priority and must not be followed by 'enter'.

Several types of prompt are used. These are:

- > Begin entry of calculation
- # For entry of an operand
- ? For entry of an operator, or final 'enter'
- J Rectangular notation; entry of second component
- < Polar notation; entry of second component

The calculation buffer has 32 stages. Type 1 operands and REAL operands occupy one stage; type 2 operands require two, as does the final operand in a calculation, regardless of its type. If the calculation is not terminated (by final 'enter') before this buffer is saturated, the computer automatically suspends input and calculates and displays an intermediate result. To continue the calculation, press 'enter' then input the next operator symbol. This continuation is also permitted when the partial result has been ordered manually.

Memory recall uses the defined program 'L' (for 'look'). For example, 'F shft L' brings the content of memory F to the display. Results are always expressed in rectangular notation, but a result displayed by a memory recall or assignment statement can be converted to polar form by 'shft SPC' (the internal representation of the number is unchanged by this operation). Press 'enter' to return to rectangular form.

Program 'L' also provides the effect of *assignment operators* as found in Algol: 'F shft L enter / enter G enter enter' performs the assignment F=F/G and, in general, 'F shft L enter' is equivalent to 'F shft = F enter'.

Any of the 18 labels A. . . R may be used in the ordinary way, during calculations, to store REAL numbers. The corresponding complex space will no longer be accessible by its label, but its numerical pointer can be used instead; these pointers are 27 for A, 29 for B, etc. Thus '37 shft L', for example, will bring the value of complex memory F to the display. The calculation register (memories X and Y) is referred to by pointer value 24. On the LHS of an assignment statement it is used to display a calculation result without a destination; while on the RHS it can be used as any other type 1 operand, and has the interpretation 'result so far' — eg, used after '* enter', it squares the current value in the calculation register. NB: memory C should not be used for REAL storage as the operation of complex conjugation would be lost.

Four operations involving REAL storage are:

Assign the real part of F to R:	R=A(F) enter
Assign the imaginary part to Q:	Q=A(F+1) enter
Assign the modulus of F to M:	F shft L shft SPC M = U enter
Assign the argument of F to A:	A = V enter

Lexicon

Variable names:		A, B, C, . . . R 27, 29, 31, . . . 61
Complex assignment symbol:		shft =
(Literals)	x + jy:	x shft J y
	r ∠ s:	2 shft K s
(Other operands)	minus F:	—F
	conjugate of F:	CF
	— conjugate of F:	—CF
(Complex dyadic operators)		
	plus:	+
	minus:	—
	times:	*
	divide:	/
	power:	∧
Memory recall procedure:		shft L
Rectangular to polar:		shft SPC
Real part of F:		A(F)
Imaginary part of F:		A(F+1)
Modulus of F:		U (after 'F shft L shft SPC')
Argument of F:		V (after 'F shft L shft SPC')

Evaluate the polynomial $P = A^4 + (2 + j3)A^3 - 4A^2 - (5+j6)A + 7-j8$, for the present value of A(—1—j8). The expression for P is rewritten in 'nested multiplication' form prior to entering the calculation, since the operators can then be obeyed in the order in which they appear:

$P = (((A + 2 + j3)A - 4)A - (5 + j6))A + 7 - j8.$

```
P shft =  A enter + enter 2 shft J 3 enter * enter
          A enter —enter 4 shft J 0 enter * enter
```

CODE ENTERED	COMMENTS	DISPLAY
(DEF mode) DEGREE enter shft Z	Sets angle mode & pointers	
A shft = —1 shft J —8 enter enter	Assignment: A = —1—j8	—1 —8
B shft = 10 shft K 45 enter enter	Assignment: B = 10 ∠ 45°	7.07... 7.07...
C shft = A enter + enter B enter enter	The sum of A and B is stored in C	6.07...—0.92...
D shft = A enter * enter B enter enter	Product in D	49.49...—63.63...
E shft = A enter / enter B enter enter	This calculation assigns √(A/B+CD) to E; note it has the form	—6.36...E—01 —6.36...E—01 —4.94...E—01
E shft = C enter * D enter + enter E enter ∧ enter ½ enter enter	temp = A/B: E = (C*D+temp)^{½}	19.18...—11.28...
F shft = —CA enter + enter CC enter * enter 1 shft J 2 enter enter	(—Ā + C̄) * (1+j2) where Ā = conjugate of A, etc, is assigned to F	21.21...7.07...

Sample calculations

A enter —enter 5 shft J 6 enter * enter
A enter + enter 7 shft J —8 enter enter

This calculation occupies 12 stages of the buffer: one each for the four occurrences of 'A', and two for each complex constant ('shft J'). The correct answer is P = 2847 — j493.

Memory use

A to R	Pointers to complex variables
S to Z	Program working space
A(27) to A(62)	Complex variables (storage area)
A(63) to A(126)	32-stage calculation buffer

Remaining memory is entirely filled by the 624 steps of program.

Defined programs

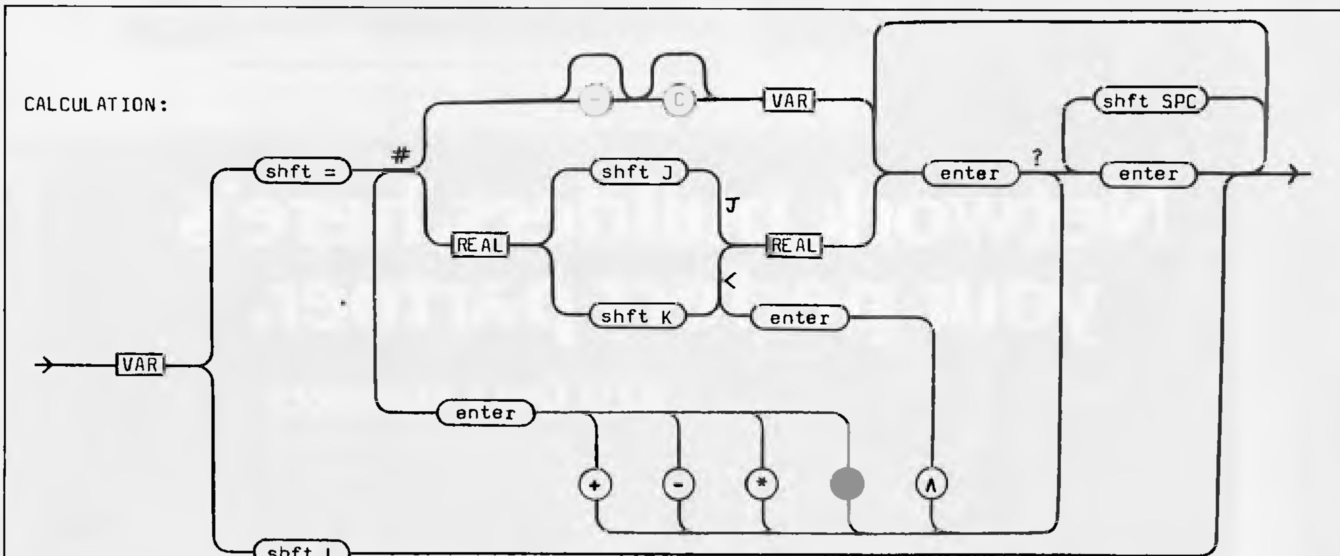
shft Z	Pointer initialisation routine
shft =	Used as assignment symbol, complex arithmetic
shft J	Numerical input, rectangular notation
shft K	Numerical input, polar notation
shft L	Data recall
shft SPC	Rectangular to polar conversion

NB: Program length is critical; add so much as one colon after a label and it will not work at all. Any addition to the program (such as making the 'complex calculator' programmable) will entail a reduction in the size of the calculation buffer (reduce 'T=126' at lines 80 and 6). Conversely, T can be increased by two (and the buffer by one stage) for every group of 16 program steps deleted.

John Kerr

```
  1: "+" U=-U: V=-V
  2: "-" X=X-U: Y=Y-V: RETURN
  3: "*" S=1: GOTO 5
  4: "/" V=-V: S=UU+VV: IF S=Ø BEEP 2: PRINT "DIV. BY ZERØ": END
  5: T=UY+VX: X=(UX-VY)/S: Y=T/S: RETURN

  6: "=" AREAD Z: S=64: T=126: INPUT "#", A(S): GOTO 1Ø
  7: "J" AREAD A(T-1): INPUT "J"; A(T): GOTO 9
  8: "K" AREAD U: INPUT "<"; V: A(T-1)=U*COS V: A(T)=U*SIN V
  9: A(S)=T-1: T=T-2
 1Ø: S=S+2: IF S<T INPUT AØ(S-1), "#", A(S): GOTO 1Ø

 2Ø: X=Ø: Y=Ø
 3Ø: FOR W=63 TO S-3 STEP 2: IF AØ(W)="∧" THEN 9Ø
 4Ø:   V=A(W+1): S=SGN (2ØØ-ABS V): IF -S LET V=V/C
 5Ø:   U=A(ABS V)*SGN V: V=A(ABS V+1)S*SGN V: GOSUB AØ(W)
 6Ø: NEXT W
 7Ø: A(Z)=X: A(Z+1)=Y: BEEP 1

 8Ø: "L" AREAD Z: PRINT A(Z), A(Z+1): S=64: T=126: A(S)=Z: GOTO 1Ø

 9Ø: GOSUB 1ØØ: T=A(W+1): U=U∧T: X=U*COS TV: Y=U*SIN TV: GOTO 6Ø

1ØØ: U=√(XX+YY): IF X=Ø LET V=ASN SGN Y: RETURN
11Ø: V=ATN (Y/X): IF -X LET V=V+(SGN Y+(Y=Ø))*ACS -1
12Ø: RETURN

15Ø: " " X=A(Z): Y=A(Z+1): GOSUB 1ØØ: PRINT U, V: GOTO 8Ø

2ØØ: "Z" FOR W=1 TO 18: A(W)=25+2W: NEXT W: AØ(63)="+": BEEP 2: END
```

Listing



Syntax chart REAL: any arithmetical expression. VAR: name of a complex variable, A ... R.

Network builders, here's your perfect partner.

Once you get to know the new B20 range of microcomputers you'll want to say "I do".

The B20 is ideal for networking. Local or remote. B20 microcomputers can be linked together – up to 17 in a cluster – and each microcomputer can have up to 512K bytes of memory. Each microcomputer can execute its own interactive application program using its own disk, printer or data communications. Each microcomputer can also share the resources of the master B20 in the network.

B20 Highlights

The B20 offers a wide range of communications protocols which provide it with exceptional versatility for its size and price.

Its disk capacity can expand from 600Kb floppy disk to 60Mb Winchester

The B20 offers a wide range of software, including Word Processing and Multiplan,* a new, advanced electronic worksheet that's one of the most powerful modelling and planning tools available.

B20 Data Manager

But perhaps the most outstanding software available on the B20 is its Data Manager application generator. Data Manager is designed to simplify and accelerate the process of creating programs so reducing the time taken to implement new systems. With Data Manager the advanced hardware of the B20 has a perfect software partner.

Partners in Success

Choosing any data processing system is very much like marriage. Because once you order it, you've formed a long-term relationship with its supplier

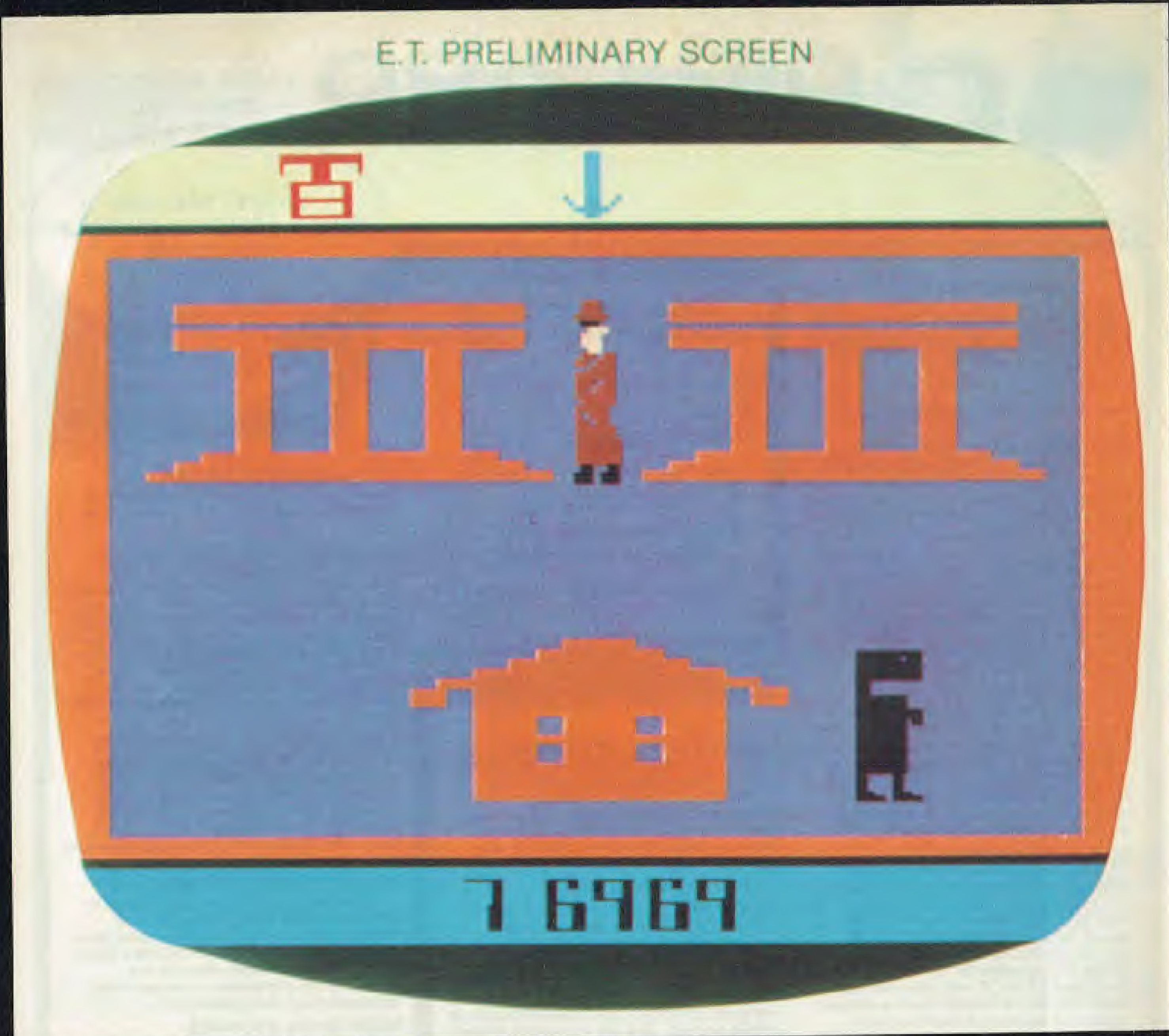
That's why one of the B20's biggest assets is the company behind it. Burroughs has 95 years experience in managing information for businesses – from the smallest to the largest. So you can depend on the B20 to come with strong, knowledgeable support. Worldwide service. And a total solution that can make networking more productive for you. For more information on the B20, call Jon Matthews on (02) 922 9300

*MULTIPLAN IS A REGISTERED TRADEMARK.

Burroughs

Building on strength

BUR0056



E.T. the film is the hit of the decade. But how good is the game?

CHECKOUT

E.T.

by Peter MacDonald

It might be an unusual way to begin a software review, but it will demystify the following E.T. Atari "Checkout" if I first briefly describe (to the uninitiated) the highest grossing motion picture ever.

E.T. (the film) is a clever play on human emotions to present the often maligned life forms existing elsewhere in the galaxy in a pathetically helpless and vulnerable position. The film portrays the adventures of the extra-terrestrial (E.T. for short) in his endeavours to rejoin his spaceship after being stranded on Earth. E.T. is befriended by a young Earthling by the name of Elliot who assists him in tracking down the fragmented components of his telephone, enabling him to contact his spaceship and so arrange his rescue. The film is directed by Steven Spielburg (he assisted in the development of the computer game and has to his credit such films as Jaws and Raiders of the Lost Ark) who capitalises on the "space anything" of current cinema audiences without going for overkill by setting this "space" adventure on Earth.

This review is actually of the Atari games machine E.T. cartridge. Versions for the Atari 400 and 800 micros are expected here next March which will be a big improvement over the games machine version if Atari's 'Star Raiders' is an indicator. Also, it should be borne in mind that comments regarding documentation could be misleading in that only a prototype cartridge and packaging etc were available for this review.

Atari's E.T. is an animated adventure type game; the aim being to guide E.T. over four sites in search of his interplanetary telephone (so he can call his ship) and then to the landing pad in time to be rescued. Each site is dotted with pits into which E.T. can fall and from which he has to use valuable energy to escape. But, pits also provide refuge from E.T.'s only two adversaries: the scientist and the FBI agent who pursue E.T. across the screen endeavouring to capture him. Apparently the scientist has developed levitation which he has conferred on his FBI associate as they are both able to pass freely over the pits. This makes it a bit unfair as E.T. has to circumvent these obstacles.

Each site is divided into zones which allow E.T. to use one of his powers which is indicated at the top of the screen. These include the ability to sense if a part of his telephone is in any of the pits on the screen, to send a pursuing FBI agent or scientist back to Washington, to eat a piece of candy (which increases E.T.'s energy level) and to move to a new site. One of the most annoying features of the game was that there seemed to be no relationship between sites. If you move E.T. to the next site on the right, then up, left and down, you find you're in a different position from which you started. Still, as there are only four sites in which the pieces of E.T. communicator can be hidden, and each site is quite distinctly different from the other, it is easy enough to jump from one to another and know if you've searched that site before.

The FBI agent will, if he makes contact with E.T., take one of the interplanetary telephone pieces and hide it in a pit or, if E.T. hasn't any phone pieces, will take all the candy pieces E.T. has managed to accumulate. The scientist, however, has more sinister intentions and will take E.T. back to Washington for "studying". But he's not interested in E.T.'s telephone or candy, so the scientist is really only a time waster.

If E.T. manages to gather up nine pieces of candy he can call Elliot (from the "Call Elliot" zone on the screen) who'll do some of the hard work and fetch a phone piece for E.T. Once E.T. has all three pieces he has to find a "Call Ship Zone" and move quickly to the landing zone and wait for his spaceship to return. Of course the FBI agent and scientist are sufficiently persistent to attempt to thwart E.T.'s rescue by appearing on the screen at the same time as the spaceship is returning to Earth. The spaceship will not land if any Earthlings are present on the screen, so before you plant E.T. on the landing zone it's a good idea to find the nearest "Send Humans Back Zone" to use as one or other Earthlings appears during the rescue.

Documentation supplied with the cartridge was, as I mentioned before, preliminary and a little difficult to follow, but I would assume that the final product will be of typical Atari quality – excellent.

My biggest criticism of the game is that once you've got E.T. in a pit, either by accident, to evade Earthlings, or to find a part of a telephone, it's occasionally very difficult to get him out. Time after time I'd use E.T.'s levitation power to float him to the surface, but he would simply fall back down again. The problem is mentioned in the documentation and the advice given was some help but didn't really solve this rather trying "feature".

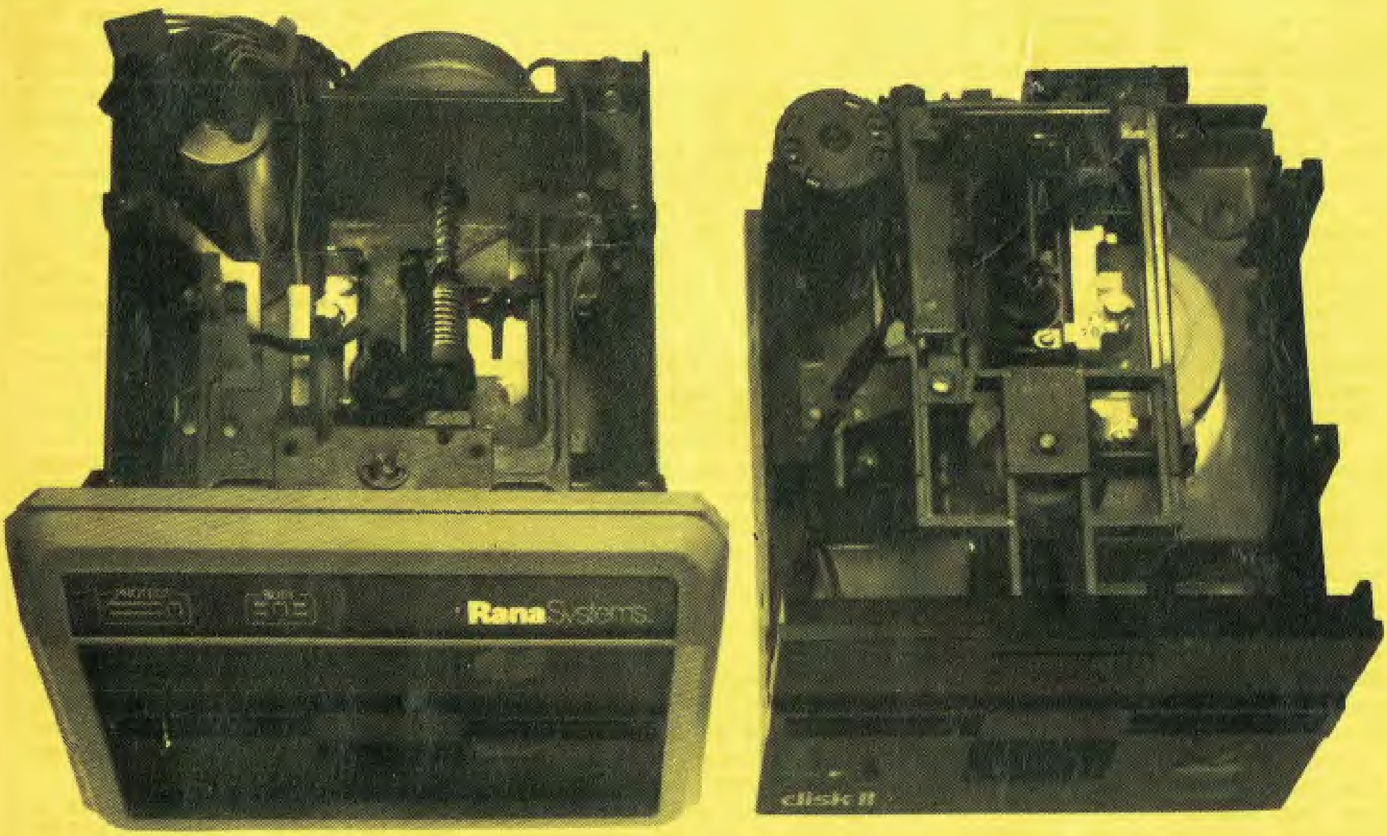
There are quite a few features I've not been able to fully describe for space reasons, such as Elliot's appearance should E.T. run out of energy. But 'three strikes and you're out' and Elliot will only help three times under most circumstances. If E.T. runs out of energy again he goes into hibernation. There are also a few strategies I've picked up after playing the game several times: at the beginning of each round it's a good idea to have a quick reconnaisance around the landing pad to locate a "Send Humans Back Zone" which you are likely to have to use several times while waiting for the spaceship to return.

Another important point to remember is that the humans appear from the edge of the screen without warning, so it is a good idea to keep to the centre of the screen wherever possible, while looking for a zone which will allow E.T. to "see" if any telephone pieces are in that site. And there's no use struggling once he's been caught by a human; I've tried and it only wastes valuable energy.

I suppose I missed out on an important point, but the documentation states that once E.T. has accumulated nine pieces of candy, he can call Elliot to fetch a telephone piece for him. Well, on one occasion I called and called and called while at the same time darted from one end of the screen to the other dodging Earthlings. Elliot simply wouldn't come.

The game should be available in retail stores this month for $69.95. Planned for next year is another Steven Spielburg creation for your Atari: "Raiders of the Lost Ark".

When you say your disk drive has more juice than Apple's, be prepared to cut one open.



Your Apple computer can grow from ordinary to awesome in a matter of minutes.

The Elite disk drive Series by Rana Systems *gives* you that kind of magic. Quickly, easily and cost-effectively.

This superb family of sophisticated floppy drives was designed and perfected by a team of industry-respected engineers expressly for Apple computer owners who have long been demanding more disk drive for their money.

Now they *have* a drive that makes an Apple perform the way it should. For example:

Elite One, the most economical Rana drive, delivers 15% more storage capacity than Apple's drive. The top-of-the-line Elite Three will provide an astonishing *four-times* more storage, approaching hard disk performance. It's done through exclusive high-density single and double-sided disks and heads.

Here's a tantalising taste of just how formidable your Apple can become.

Next to enormously increased disk space, Apple buyers are most excited about Rana's fingertip Write/Protect feature. A pushbutton on the LED-lit panel gives you a new measure of failsafe control.

While Elite drives are all plug-compatible with Apple's controller, Rana's vastly superior controller card is a *must* for those with growing data processing demands. With it, you control up to *four* floppy drives using only one slot . . . while still utilizing your Apple computer to achieve dramatic new heights of system flexibility and processing power.

A specially programmed Rana utility disk assures you of far-reaching compatibility, and extra dimensions of business, leisure and learning applications with your Rana-enhanced Apple computer.

There are many, many more operating refinements: Uncanny head-positioning accuracy, 300% faster track-to-track speed, better operating economy, and gentler clamping action through an all-new diskette centering cone, to name just a few.

A disk drive package like this doesn't grow on trees, so write or call today for more information about the Elite Series.

ELITE 1	$640	(163kb)	with Controller $790
ELITE 2	$830	(326kb)	with Controller $990
ELITE 3	$1090	(652kb)	with Controller $1250
CONTROLLER ONLY.	$160	(All prices include Sales Tax)	

CONTACT:

Computer Edge Pty. Ltd.

364 FERRARS STREET, ALBERT PARK, VIC 3206. Telephone: (03) 690 1477

DINERS CLUB

AMERICAN EXPRESS

★★WOMBAT★★

The world of computing is now available to everyone at very reasonable cost.
For parents who wish their children (and themselves) to learn programming,
to utilise educational programs, or just to make use of some of the hundreds of programs available
the WOMBAT is now released on the Australian market.

*** BONUS** –
For each WOMBAT sold before January 25th, a choice of any ten programs from the MECC Educational Library will be given free – valued at approx. $288!

Standard features:

* built-in floating-point BASIC;
* built-in disassembler;
* a ROM-based system control program;
* color graphics and sound capabilities;
* sockets for up to 48K bytes of user memory (RAM); 16K with the plug-in Language Card;
* cassette interface;
* game input/output connector, and two hand controls for games and other human-input applications;
* typewriter-style ASCII keyboard;
* high-efficiency switching power supply;
* eight accessory expansion slots;
* autostart monitor;
* upper/lower case function.

Inputs and Outputs:

* cassette interface (1500bps);
* game I/O: four analog to digital inputs, three TTL inputs and four TTL outputs.
* typewriter-style ASCII keyboard;
* 7 peripheral board connectors fully buffered, with interrupt and DMA priority structure;

NOTE: THE WOMBAT SOFTWARE IS COMPATIBLE WITH ALL EXISTING PROGRAMS FOR THE APPLE II

CONTACT:
Computer Edge Pty Ltd
364 FERRARS STREET, ALBERT PARK, VIC 3206 Telephone: (03) 690 1477

DINERS CLUB
AMERICAN

JUST $940

(plus S.T. – $120)
IF APPLICABLE

DEFOREST SOFTWARE

APPLE

AC= apple cassette
AD= apple disk

ADVENTURE

CLASSIC ADVENTURE -the original adventure	AC041-0106	$ 13.50

BUSINESS

FINANCE AND INVESTMENT -disk	AD162AD	$ 24.50

BOARD GAMES

SARGON II -perhaps the ultimate micro chess game	ACBUT007	$ 30.00

ELECTRONICS — HAM RADIO

HAM PACKAGE ONE -disk -contains ohms law, series and parallel calcs, voltage dropping and dividing	AD176AD	$ 17.50

HOME ENTERTAINMENT

BOTH BARRELS -2 games on the one disk	AD-SIR02	$ 25.50
COPTS AND ROBBERS -can you escape the tomb	AD-SIR13	$ 28.70
GAMMON GOBLINS -a real heart stopper	AD-SIR09	$ 25.50
GENETIC DRIFT -observations on modern man	ADBR16	$ 25.50
GOLF -play golf in the comfort of your own home	AC018A	$ 10.00
OIL TYCOON -drill wells, make a million	AC079A	$ 10.00
OLYMPIC DECATHLON -become an olympic champion	AD2210MIC	$ 30.00
SAHARA WARRIORS	AC080A	$ 10.00

FLIGHT SIMULATORS

AIR FLIGHT SIMULATOR -control your own aircraft	AC148A	$ 10.00

SPACE SIMULATIONS

CYBER STRIKE -3D space game	AD-SIR04	$ 28.70
MISSION INVASION -space invaders	AC041-0121	$ 13.50
ORBITRON -look out for the killer satellites	AD-SIR08	$ 25.50
PLANETOIDS -defend your self from the Asteroids	AC041-0015	$ 16.45
PLANETOIDS -disk	AD041-0015	$ 20.95
PULSAR TWO -two game disk	AD-SIR07	$ 25.50
SPACE WARS -three game tape	AC096A	$ 10.00
SPACE QUARKS -inject the quarks	ADBR14	$ 25.50

WAR GAMES

PHANTOMS FIVE -fly a fighter bomber	AD-SIR05	$ 25.50
SKY BOMBERS -2 player game	AD271AD	$ 24.50
TANK COMMAND -2 player tank game	AC041-0033	$ 9.00

ATARI

AC= atari cassette
AD= atari diskette

ADVENTURE

ADVENTURE 1 -ADVENTURELAND -the first adventure	AT041-001	$ 16.50
ADVENTURE 2 -PIRATE ADVENTURE -	AT041-002	$ 16.50
ADVENTURE 3 -MISSION IMPOSSIBLE	AT041-003	$ 16.50
ADVENTURE 4 -VOODOO CASTLE	AT041-004	$ 16.50
ADVENTURE 5 -THE COUNT	AT041-005	$ 16.50
ADVENTURE 6 -STRANGE ODYSSEY	AT041-006	$ 16.50
ADVENTURE 7 -MYSTERY FUN HOUSE	AT041-007	$ 16.50
ADVENTURE 8 -PYRAMID OF DOOM	AT041-008	$ 16.50
ADVENTURE 9 -GHOST TOWN	AT041-009	$ 16.50
ADVENTURE 10-SAVAGE ISLAND	AT041-0010	$ 16.50
ADVENTURE 11-SAVAGE ISLAND PART II	AT041-0011	$ 16.50
ADVENTURE 12-GOLDEN VOYAGE	AT041-0012	$ 16.50
ANGLE WORMS	RC051-0092	$ 16.50
CYPHER BOWL	RC103-304	$ 47.00
DELFLECTION	RC051-0078	$ 13.50
GALACTIC EMPIRE	RC051-0026	$ 16.50
GALACTIC TRADER	RC051-0027	$ 16.50
GALACTIC REVOLUTION	RC051-0028	$ 16.50
GHOST HUNTER	RC123-301	$ 29.00
GOLDEN VOYAGE	RC051-0129	$ 16.50
GOMOKU	RC103-302	$ 18.30
LUNAR LANDER	RC051-0094	$ 14.50
MOUNTAIN SHOOT	RC051-0079	$ 14.50
POLKER SOLITAIRE	RC103-301	$ 16.00
PREPPIE	RC050-0147	$ 17.40
REAR GUARD -similar to penetrator	RC050-0143	$ 37.50
REVERSI	RC103-303	$ 18.30
SUNDAY GOLF	RC051-0101	$ 14.50
STAR TREK 3.5	RC051-0025	$ 16.50
3D TIC TAC TOE	RC051-032	$ 16.50

PROGRAMS FOR MULTIPLE MACHINES

E.G. PET/APPLE/TRS80/ATARI

ADVENTURE 1, 2 & 3 -for the TRS80 and APPLE -disk	BD032-0010	$ 35.95
ADVENTURE 4, 5 & 6 -for the TRS80 and APPLE -disk	BD032-0011	$ 35.95
ADVENTURE 7, 8 & 9 -for the TRS80 and APPLE -disk	BD032-0012	$ 35.95

PET

PC= pet cassette

ASSORTED

ARCADE TWO -2 game cassette	PC045P	$ 6.70
CASINO ONE -blackjack/roulette	PC014P	$ 6.70
CASINO TWO -craps	PC015P	$ 6.70
CHIMERA -6 game cassette	PC110P	$ 6.70
CODE NAME CIPHER -4 code programs	PC112P	$ 6.70
DOW JONES -stock market program	PC026P	$ 6.70
ELECTRONICS ENGINEERING ASSISTANT -let your computer do the work for you	PC022P	$ 6.70
HELLFIRE WARRIOR	PC141702	$ 23.50
MIMIC -how good is your memory	PC039P	$ 6.70
PERSONAL WEIGHT CONTROL and bio-rythms	PC005P	$ 6.70
PENNY ARCADE -6 game cassette	PC044P	$ 6.70
SANTA PARAVIA -very popular adventure	PC175P	$ 6.70
QUBIC 4 GO-MOKU -2 game cassette	PC038P	$ 6.70
TANGLE AND SUPERTRAP	PC029P	$ 6.70
TREK X -command a starship	PC029P	$ 6.70
TURF AND TARGET	PC097P	$ 6.70
UTILITY ONE -monitor/programmers calculator	PC105P	$ 6.70

TRS-80 CASSETTES & DISKS

TB0112 -ADVENTURE HINT BOOK	TB0112	$ 7.50

BOARD GAMES

AMAZING BLACK JACK MACHINE	TC010-0091	$ 15.50
MACROPOLY -play monopoly on your computer	TCDF10	$ 7.00
Z CHESS 3	TC010-0080	$ 19.50

BUSINESS

ACCOUNTS PAYALBE	TDDF04	$ 50.00
ACCOUNTS RECEIVABLE	TDDF05	$ 50.00
BILL CONQUERER	TD275RD	$ 25.00
DISK SORT MERGE -MODEL 2	TDDSMMOD2	$120.00
GENERAL LEDGER AND CASH JOURNAL -TARANTO	TDDF06	$ 50.00
INVENTORY CONTROL -TARANTO	TDDF02	$ 50.00
INVOICING -TARANTO	TDDF03	$ 50.00
MTC AIDS -FOR THE MODEL 2	TD8018D	$ 40.00
MTC CALCS -FOR THE MODEL 2	TDCALCS	$ 40.00
MTC DIVERGE -FOR THE MODEL 2	TDDIVERGE	$ 20.00
MTC REBUILD -FOR THE MODEL 2	TDREBUILD	$ 20.00
MTC SUPERSEDE -FOR THE MODEL 2	TDSUPERSEDE	$ 12.00
RACET -GFS -FOR TRS80 MODEL 2	TD2GFS	$ 80.00

EDUCATION

ITALIAN TEACHER	TD1010	$ 15.00
GERMAN TEACHER	TD1011	$ 15.00

FLIGHT SIMULATORS

AIR FLIGHT SIMULATOR	TC017R	$ 10.00
FLIGHT SIMULATOR -3D graphics	TCT80-FS1	$ 28.50

HOME ENTERTAINMENT

ADVENTURE -MICROSOFT DISK	TD1203MIC	$ 32.00
BALL TURRET GUNNER -very fast	TC051R	$ 10.00
BARRYS CUBE -rubix cube on your computer	TCDF07	$ 10.00
DR CHIPS -turn your computer in to psychologist	TC010-0020	$ 15.50
HOUSE OF THIRTY GABLES -adventure	TC219R	$ 15.00
MIND WARP -3 game cassette	TC118R	$ 10.00
MUSICAL YAT-C -musical strategy machine	TC011-0090	$ 15.50
OIL TYCOON	TC023R	$ 10.00
POLYGONS	TC244R	$ 10.00
SILVER FLASH PINBALL -pinball machine	TC010-0086	$ 15.50
TEMPLE OF THE SUN -adventure	TC5012R	$ 15.50
WINNERS DELGHT -4 game cassette	TC124R	$ 10.00

SPACE SIMULATORS

ALIEN ATTACK FORCE/INVADERS	TC240R	$ 10.00
ATTACK FORCE -doge the crusher at convoy and destroy them before they squash you	TCBIG54	$ 15.50
COSMIC FIGHTER -fantastic machine language game	TCBIG53	$ 19.50
COSMIC PATROL	TC223R	$ 15.50
DANGER IN ORBIT -fight the asteroids	TC237R	$ 16.50
DEFENSE COMMAND -very fast machine language	TCBIG57	$ 16.50
GALACTIC TRADER -	TC011-0027	$ 15.00
GALACTIC REVOLUTION	TC011-0028	$ 15.00
METEOR MISSION -can you save your astronauts	TCBIG55	$ 19.50
MISSILE ATTACK -can you defend your cities	TD012-0102	$ 25.00
PENETRATOR -how far can you penetrate the system	TCBIG58	$ 30.00
ROBOT ATTACK -with voice sound affects	TCBIG56	$ 19.50
SIMUTEK	TC012-0024	$ 15.50
SPACE TREK III -2 game cassette	TC031R	$ 10.00
SPACE INTRUDERS	TC010-0016	$ 16.00
SPACE SHUTTLE -pilot a space shuttle	TC322R	$ 15.50
STELLAR ESCORT-escort your mother ship to safety	TCBIG59	$ 17.50
SUPA NOVA -destroy asteroids	TCBIG52	$ 15.00
ZOSSED IN SPACE -complex space simulation	TC010-0049	$ 15.50

UTILITIES

DISK SCOPE	TD139RD	$ 25.00
DISK LABELLING DISASSEMLBER (DLDIS)	TD231RD	$ 25.00
DYNAMIC DEVICE DRIVERS	TC228R	$ 21.50
ENHANCED BASIC	TC077R	$ 19.95
ENERGY CONSUPMTION -record your utility bills	TC132R	$ 10.00
FLOPPY DISK MAINTENANCE	TD012-0122	$ 25.00
PROGRAMMERS PRIMER	TC245R	$ 10.00
SURVEYORS APPRENTICE	TC127R	$ 10.00
THE ELEMENTS -the periodic table of elements	TC216R	$ 12.50
UTILITY 2	TC076R	$ 10.00
VIDEO SPEED READER -improve your reading speed	TC100R	$ 10.00

WAR GAMES

AIRMAIL PILOT -can you get to Chicago on time	TC106R	$ 10.00
AMOURED PATROL -3D arcade	TC010-0140	$ 19.50
BATTLEGROUND	TC141R	$ 10.00
COMBAT -requires 2 computer system	TC010-0123	$ 15.50
JET FIGHTER PILOT	TC159R	$ 15.50
NIGHT FLIGHT -night flight simulator	TC117R	$ 10.00
PROJECT OMEGA	TC010-0076	$ 15.50
SHOWDOWN -can you defeat the fastest draw in town	TC012-0103	$ 25.00
SLAG WAR	TC010-0018	$ 15.00
SWAMP WAR	TC312R	$ 15.50

BOOKS

CUSTOM APPLE & OTHER MYSTERIES	BOKA002	$ 22.50
BASIC WITH STYLE	BOKB009	$ 8.50
MICROCOMPUTERS AND 3 R'S	BOKE056	$ 8.50
PET/CBM COMPUTER GUIDE	BOKG075	$ 14.95
PET & IEEE 488 BUS	BOKG076	$ 14.95
FORTRAN WITH STYLE	BOKL012	$ 9.00
Z80 ASSEMLBY LANGUAGE PROGRAMMING	BOKL015	$ 16.00
Z80 PROGRAMMING FOR LOGIC DESIGN	BOKL023	$ 11.00
6502 ASSEMLBY LANGUAGE PROGRAMMING	BOKL064	$ 15.00
BASIC PROGRAMS FOR THE TRS80	BOKT012	$ 19.95
TRS80 DISK AND OTHER MYSTERIES	BOKT035	$ 15.00
BASIC FASTER AND BETTER	BOKT103	$ 30.00
HOW TO PROFIT FROM YOUR PERSONAL COMPUTER	BOKW021	$ 9.50
COMPUTERS IN SOCIETY	BOKW054	$ 8.50

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

80 Column Cards Compared

CHECKOUT

by Ian Davies

This month, CHECKOUT examines two eighty column cards for the Apple II: namely the Digicard by Maclagan Wright and Associates and the Vision-80 card by Zofarry. The Apple II computer used in the review was kindly furnished by The Logic Shop, Prahran.

In keeping with the high standard of in-depth CHECKOUT investivations, the first action I took after powering up each card was to count the number of columns Eighty on both – drat!! no controversial journalism on this review!

The most noticeable feature about the two cards is their similarity. They both provide much the same features (on the surface) and appear quite similar in construction. Closer inspection of the boards revealed, however, that one was not a rip-off of the other. Each board is based around the 6845 CRT controller chip – a rather gutsy little device that does most of the work of screen control. Both boards also contain two 2716 2k byte EPROMS, one of which is used as a character generator into the 6845 and the other contains 6502 controlling software. The video memory itself is 2k bytes of RAM, implemented through four 2114 static RAM chips (1k x 4 bits) on the Vision-80, and through a single 6116 chip on the Digicard, which appears to be employing slightly more "state of the art" technology. The exception to this is the fact that the Digicard uses an ELEC-TROL Read Relay (presumably to switch between its video output and the standard Apple video output), whereas the Vision-80 employs a 4016 chip, which is a quad bi-lateral switch and is not subject to the problem of mechanical degradation over a period of years.

Installation of the cards is quite straightforward, simply involving plugging the card into port three on the Apple mother board. A few flying clips do have to be connected to strategic points inside the Apple, but this presented no problems. Although the installation procedures are quite similar, the Vision-80 is probably slightly easier to install, but not by a significant margin.

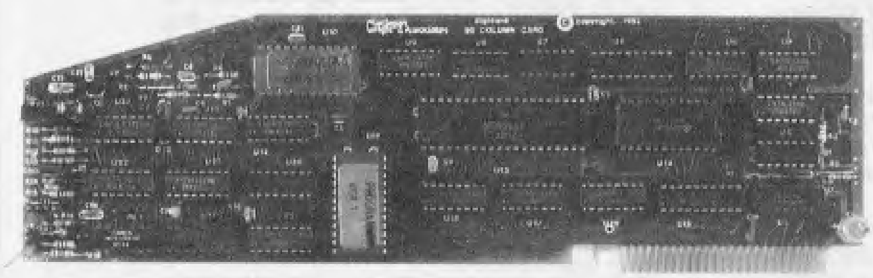
Both cards provide a "shift-lock" facility. The Digicard alters the cursor character to indicate whether it is in upper case or lower case mode, whereas the Vision-80 displays the fact through a red LED. The LED should be mounted on the Apple console, which involves drilling a hole through the Apple casings. Both manuals stated that the shift mode could be changed by pressing the SHIFT key on the Apple keyboard. The Digicard (which was in the machine when we received it) seemed to prefer the CONTROL key for this purpose. This proved to be caused by incorrect installation in the Apple used in the review, and was not the fault of the Digicard itself.

The two cards have extensive screen control features accessed through ESCAPE or CONTROL keys. Included in the features is a rather nice communications package which allows you to communicate with mainframes in various data formats while still using the Apple disk. As an added bonus, the Digicard also provides 17 single-key keyword entries, for example, pressing ESC 6 produces GOTO.

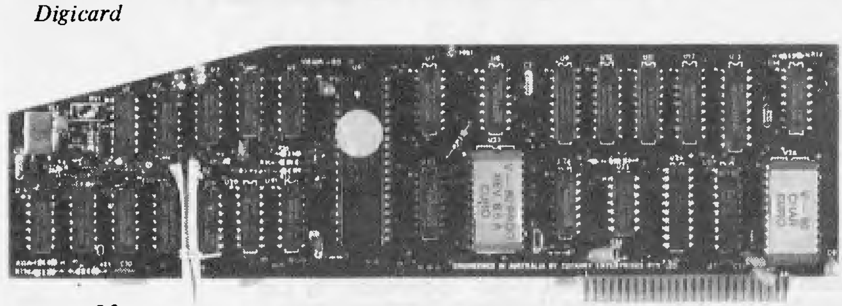
The quality of the output is very good on both boards, although there was some problem with character definition on the Vision-80 (specifically, the horizontal parts of the characters were far too bright) but this appeared to be caused by the fact that the Apple was "tuned" for the Digicard. Both provide excellent high resolution graphics, with the ability to switch between the hi-res screen and the lo-res screen at will. Both boards also allow you to run two monitors – one for hi-res output and one for lo-res. Toggling from one logical screen to the other did produce a slight shudder in the display, but this was common to both.

One area in which the two boards differ is documentation. The manual provided with the Vision-80 seemed significantly superior, including greater detail, and a section at the rear for assembly programmers and trouble-shooting. It also contained a circuit diagram.

In summary, the two cards are very much alike, although the Vision-80 probably does have a slight edge due to a couple of extra features and better documentation. Either one is essential for any sort of serious word processing work. The Vision-80 retails for $337 and the Digicard for $354.



Digicard



Vision 80

APC welcomes correspondence from its readers but we must warn that it tends to be one way! Address letters to: 'Communications', APC, P.O. Box 280, Hawthorn, Vic 3122.

Those factors again

In your May edition, a Mr Hunter asked how he might factor a large number. This drew a 'number' of well-meant but ill-informed replies to the effect that if (as suspected) Mr Hunter was looking for prime factors, it could take him about 10^6 years.

In fact, recent progress in the application of number theory has increased the rate of testing for primality by a 'factor' of 10^7. The products of 100-digit primes are no longer thought secure for encryption purposes.

Mr Hunter should contact Dr H W Lenstra (University of Amsterdam, Tel 020-5259111).
Dr David Fisher

Fix wanted

I would be obliged if you could publish this letter as I am at present unable to obtain any advice locally. I have with me an Osborne 1 and Epson MX80 FT III which I consider to be excellent value for money. However, I find that the 'Install' program is unable to make the best of configuring for my printer as Wordstar expects toggles where the Epson has separate control codes for turning on and off certain modes. Does anyone know how to Install for the Epson? Also, as an instructor in electronics I have to produce inserted graphs of sine waves, etc, in my notes. I have a Basic routine that prints them but it is *very* slow and cannot be patched to Wordstar . . . any ideas? I have a similar problem with calls in Basic as I don't know the best way to pass variables to the subroutine. What it is to be ignorant!!

It may not be commonly known that location 61281 contains the horizontal screen position during scroll and location 61282 the vertical screen position. It is necessary to divide the horizontal by 2, and there are offsets, but they allow a fantastic 3D space simulation with centralised aiming sights for those who like games. It is also much easier to poke direct to the screen RAM as per other machines than to use the Osborne routine. Screen memory starts at 61440 as top left of normal screen. (A Basic listing is enclosed . . . rough and ready but gives a guide.)

Perhaps there is a user who can find someone to act as a 'penpal' or should I say keyboard-pal, as that is what I need here. I will be here for quite a while longer so the address will hold good until Christmas at least.
Richard C Ferryman, 208 Airport Hotel, P.O. Box 2012, Jeddah, Saudi Arabia.

You are certainly not the only one with this Osborne/Epson problem Mr Ferryman. The fix is too involved to explain here, however it is documented in issue 1 and 2 of 'Portable Companion', the magazine put out by Osborne for its users. – Ed'll Fix It.

Dots or brackets...

In response to your challenge (October 1982), I'd like to suggest a way of listing lists of lists of lists for Lisp without brackets. Logicians began using 'Dot Notation' before the inception of set theory. A century on, they are still ranting, enthusing and raving about how convenient and manageable this has turned out to be, and how clever they were to have thought of it.

Let's jump in at the profound end, with Mr Liardet's first example of Lisp data. Using dot notation, it looks something like this:

(SETQ PEOPLE ': BASIL 32M . CYNTHIA 30 F JOHN 28 M)

Here, the colon has the job of delimiting the scope of the quote mark – but see how clearly it divides the list assignment into destination and source. Whereas the action of the brackets is to bind elements together, dots act as element *separators*. They can be compared to ordinary prioritised in fix operators: the more dots in a group, the lower the 'priority' of the separator, and the greater its

scope (the size of chunk it governs in the list).

For a somewhat dramatic illustration, consider next the definition of the function

```
LENGTH:
(DE LENGTH . LIST :
COND : : EQ LIST NIL . 0
     : : T :. PLUS 1 :
LENGTH . CDR LIST)
```

If you don't think that makes the list structure a lot clearer, did you notice the printer's error in the original? There's an erroneous right parenthesis on the end of the second line.

Now to those lists of lists. The simplest I can think of is a pair of pairs of atoms. Take your pick – brackets or dots?

```
(SETQ PAIRS '(((A B)(C D))
((E F)(G H))))
(SETQ PAIRS ':.A B . C D :
E F . G H)
```

John Kerr

...Lisp or logo

Referring to Mr Kerr's letter suggesting that a Lisp interpreter would run faster if reserved words were to be tokenised and brackets removed, it must be made clear that to the interpreter a program, ie, a list, is represented as a set of linked pointers. The brackets are used to delimit lists only in the input/output routines. In Lisp not only are the reserved words tokenised, all words are! If Mr Kerr desires the elegance of Lisp without so many brackets, then I suggest he considers Logo which is Lisp-based and is friendlier in use. This leads me to my second point.

Mr Parr's Logo compiler written in Basic is an interesting program (though lamentably slow) but contains a bug which could confuse a beginner grappling with recursion. If a previously encountered function is again recognised by the compiler, the number of arguments is not fetched and at run time the function is called with wrong values. This occurs in the Branch program in the September 1982 Logo article. It can be fixed by changing line 9730 to read

```
IF SY$=PN$(W) THEN
N=CP(N); RETURN
```

Ian A Stewart

Space defender

It is a shame that so much had to be left out of Names of the Nameless in the October issue. In particular, I wish room could have been found to justify the dangerous use of a space as a mathematical symbol in GSB's arithmetic, since this can, as it does here, involve implicit axioms. It seems that the axioms should include (though one can't be sure):

space	space	=	space
space	⌉	=	⌉
⌉	space	=	⌉
⌉	⌉	=	⌉
	space⌉	=	⌉
	⌉⌉ (nested)	=	space

which are isomorphic to

0 or 0 = 0
0 or 1 = 1
1 or 0 = 1
1 or 1 = 1
NOT 0 = 1
NOT 1 = 0

respectively.

If this is so then GSB's arithmetic is nothing more than Boolean algebra and can hardly be said to be more fundamental.

Your article tries to allay the suspicion that *Laws of Form* is a crank book, and to encourage potential readers, but I must regretfully say 'Not convinced'.

James Crook

This is one of several letters from mathematicians who disapprove quite strongly of Spencer-Brown's work. His proof of the four-colour theorem is by no means accepted 'in the trade', and is as fas as I know not officially published. Not being a professional mathematician, I do not feel threatened by the unorthodoxy of Laws of Form, *nor do I feel that APC readers are likely to come to much harm.*

It is indeed a shame that so much had to be left out; I obviously failed to make clear that 'space' is not a symbol, but precisely the space in which a cross stands. There is only one initial symbol, the cross, which indicates the marked state. The idea is certainly more fundamental than Boolean algebra, to the point where having to write it in ink on paper almost subverts its understanding. It may well be dangerous, but then so are flying and mountaineering. – Dick Pountain.

What's in a name?

I am considering selling software for my ZX81 but have a query about copyright. If I see a program working on a computer other than the ZX81, and I decide to write a program based on this idea, but using my own programming ideas, is it a breach of

copyright if I sell the program?
G Smith

This question is the subject of heated discussion at the moment with the recent threat by Atari to prosecute certain companies who are selling their own representations of Pacman. The question is, can you copyright an idea? If you rewrite a program for the ZX81 then it is very unlikely that it will bear any visual or programming similarity whatsoever, so theoretically you are home and dry. The only point to be careful of is the name, which may be copyrighted. Most companies are very reluctant to allow other companies to benefit from a good name that they have built up. – Ed.

Cramming it in

One of the major uses I have for an Osborne 1 is the production of business letters using Wordstar. Although these letters are usually short (typically four or five lines) I keep running out of disk space, due to the limited (92k formatted) capacity of the Osborne's drives. Is there anything I can do about this, short of altering the hardware, with double density, or external drives?
Name and address supplied.

Yes, there's a lot you can do.

1. Make sure you have all 92k available for text files (incidentally, that 92k is quite generous for single density single sided disks – about 70k would be more typical) by leaving all the Wordstar and other programs (such as PIP and STAT) on the disk on drive A, and having a blank (but formatted) disk on drive B: on entering Wordstar use the L option to log onto drive B.

2. Once you are happy with the final version of a text file, delete the corresponding .BAK file created by Wordstar. If you are not doing this already, this simple action will double your usable space.

3. The minimum file size on a CP/M 2.2 system such as the Osborne 1 is 2k. Obviously this will limit you to 46 files per disk, even if, as in your case the files are actually much smaller. (There is a further limitation in that the disk directory can only accept 64 entries, but this would only be significant with double-density disks.)

4. This limitation can be overcome by merging lots of your older text files into one bigger file, using either PIP, or the 'Control KR' function in Wordstar. If you can use the latter you can arrange for each letter to start on a separate page, so they can be individually printed. If the start of each page includes the letter's reference you can find them in the file using 'Control QF'.

If your letters were of up to 1k size, and you were not already using any of these ideas, you could get 23 letters (and 23 .BAK copies) on a whole disk. Deleting the .BAK copies would increase this to 46. Merging the letters into bigger files would give a capacity up to 92. – Ed.

Better shuffle

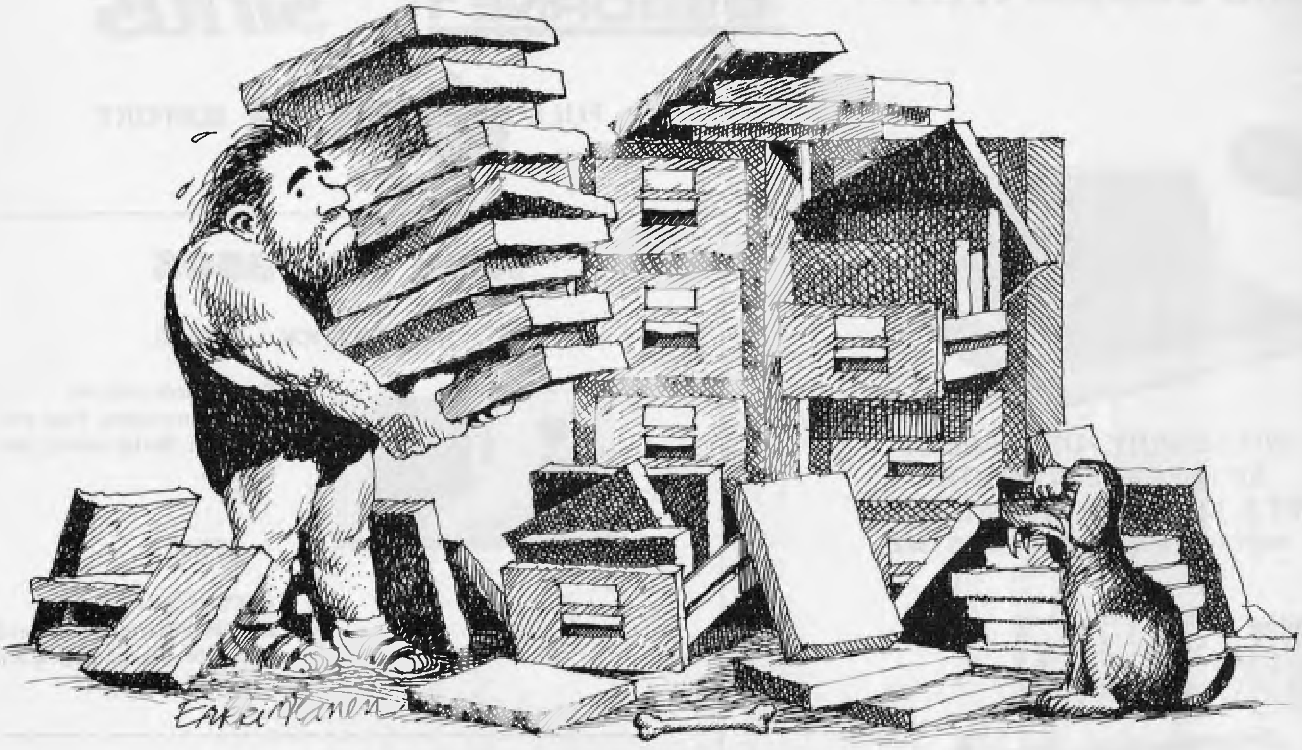
I wish to draw your attention to the card shuffling routine presented by Russell Hutson in the October '82 issue of *APC*.

It may interest some readers to know that the routine is not very efficient. Card shuffling routines were discussed earlier this year in *Byte* and a couple of algorithms presented. One of the more efficient algorithms to result from this discussion is the following:–

If needed, this routine may be invoked several times to keep all the card sharks happy.
Earl Chew

```
100 REM   CARD SHUFFLING ROUTINE
110 REM
120 REM     INPUT– C$(52)  Card description in any order
130 REM    OUTPUT– C$(52)  Shuffled card descriptions
140 REM      USES–
150 REM              X      Card counter
160 REM              Y      Random pointer
170 REM              Z$     Temporary string
200 FOR X=1 TO 52
210     Y=INT(RND(0)*52)+1
220     Z$=C$(X) : C$(X)=C$(Y) : C$(Y)=Z$
230 NEXT X
240 RETURN
```

KEEPING UP WITH FILING SPACE DEMANDS CAN BE TOUGH.



WITH DB MASTER™ IT'S EASY.

It seems to happen before you know it. The business is growing and, naturally, so are your files. But the more room you need for files the less there seems to be. You've added more people, more equipment, more capacity. The trouble is, you're trying to add more file space, when what you really need is more *work* space. Those filing cabinets full of folders are costing you too much—in time, in space, in money.

That's where DB MASTER comes in. It provides plenty of expansion capability to meet the needs of most normal growing businesses that use an Apple II. The floppy diskette version, for example, can be expanded to contain up to 387,000 individual records. Need more? The DB MASTER Special Edition For Hard Disk can be expanded to contain up to 7 million individual records.

Of course, the number of records you may store in your particular business depends on several variables—like the number of characters in each record and the kind of information storage you require. Ask your dealer to spell out for you just how much DB MASTER can help your operation.

We designed DB MASTER to make your life easier. Sure, there are other data base systems for your Apple II. But if you're not using DB MASTER, you're simply not getting all you could from your data base management program.

And there are three powerful accessory programs available to complement DB MASTER's capabilities. UTILITY PAK #1, for example, allows you to access other standard Apple files (including VisiCalc), and even recover damaged files. UTILITY PAK #2 allows you to edit or change up to five "fields" in a single pass through your files. And STAT PAK performs statistical analyses on data in your DB MASTER files. Of course, there's considerably more capability in each of our accessory paks than we have room to talk about here. Ask your dealer for the full details.

STONEWARE INCORPORATED

AVAILABLE FROM YOUR LOCAL MICROCOMPUTER DEALER

APPROVED i SOFTWARE

IMAGINEERING
22-40 Sir John Young Cres., Woolloomooloo
Sydney, NSW 2001. (02) 358 3011

Our dealer of the month
RANDOM ACCESS (MELB.)
MARK THOMPSON 62 1339

PPCB1488APCB

CLOCK IT TO ME

Bruce Marriott continues his description of a clock/calendar card for the Apple II with a description of the software drivers.

Last month I presented the hardware needed to interface a clock/calendar card based on the OKI MSM5832 chip to the Apple II, or — in rather less detail — to other systems. This month we'll look at the driving software. Naturally, this is based on the Apple, too, but where appropriate I have included some information on amending the code for other systems.

The 6521 PIA

Since the clock chip is accessed through a 6821 PIA, it is first necessary to know how to control the PIA. What follows is a resumé on how this is achieved — for a more complete version, see the 6821 data sheet.

The 6821 has two 8-bit ports, A and B; for the purpose of this exercise, they can be considered as identical apart from the unique addresses associated with each. The PIA has four interrupt inputs (two of which may also be used as outputs) which will be discussed later. Each port is controlled by two registers, the data direction register and the peripheral register, which, because of addressing limitations, have the same address (Figure 1), with selection between them being made by the value of bit 2 in a third register (which has no duplicated address) called the control register. The data direction register for each port governs which bits will act as inputs and which will be outputs; writing a 0 to a bit makes it an input while a 1 makes it an output. The peripheral register allows the setting of levels on outputs and the reading of levels on inputs — a 1 in a bit indicates that it is high (+5 volts) and a 0 low (0 volts).

Figure 2 shows how ports A and B are connected to the MSM5832. It repeats information given in last month's circuit diagram but in a more easily assimilated form. The examples

CONTROL OF INTERRUPT INPUTS CA1 AND CB1

CRA-1 (CRB-1)	CRA-0 (CRB-0)	Interrupt Input CA1 (CB1)	Interrupt Flag CRA-7 (CRB-7)	MPU Interrupt Request IRQA (IRQB)
0	0	↓ Active	Set high on ↓ of CA1 (CB1)	Disabled — IRQ remains high
0	1	↓ Active	Set high on ↓ of CA1 (CB1)	Goes low when the interrupt flag bit CRA-7 (CRB-7) goes high
1	0	↑ Active	Set high on ↑ of CA1 (CB1)	Disabled — IRQ remains high
1	1	↑ Active	Set high on ↑ of CA1 (CB1)	Goes low when the interrupt flag bit CRA-7 (CRB-7) goes high

(A_1) RS1	(A_0) RS0	Control Register Bit CRA-2	CRB-2	Location Selected
0	0	1	x	Peripheral Register A
0	[illegible]	0	x	Data Direction Register A
0	1	x	x	Control Register A
1	0	x	1	Peripheral Register B
1	0	x	0	Data Direction Register B
1	1	x	x	Control Register B

CRA

7	6	5	4	3	2	1	0
IRQA1	IRQA2	CA2 Control			DDRA Access	CA1 Control	

CRB

7	6	5	4	3	2	1	0
IRQB1	IRQB2	CB2 Control			DDRB Access	CB1 Control	

CONTROL OF CA2 AND CB2 AS INTERRUPT INPUTS
CRA5 (CRB5) is low

CRA-5 CRA-5 (CRB-5)	CRA-4 CRA-4 (CRB-4)	CRA-3 CRA-3 (CRB-3)	Interrupt Input Interrupt Input CA2 (CB2)	Interrupt Flag Interrupt Flag CRA-6 (CRB-6)	MPU Interrupt Request IRQA (IRQB)
0	0	0	↓ Active	Set high on ↓ of CA2 (CB2)	Disabled — IRQ remains high
0	0	1	↓ Active	Set high on ↓ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high
0	1	0	↑ Active	Set high on ↑ of CA2 (CB2)	Disabled — IRQ remains high
0	1	1	↑ Active	Set high on ↑ of CA2 (CB2)	Goes low when the interrupt flag bit CRA-6 (CRB-6) goes high

Fig 1. 6821 PIA register addressing

	6821	MSM5832	Remarks	
PORT A	PA_0	A_0	Address lines to select	
	PA_1	A_1	specific time/date	
	PA_2	A_2	Register	
	PA_3	A_3	Always 6821 outputs.	
PORT B	PB_0	D_0	Time & date Data	
	PB_1	D_1		
	PB_2	D_2	Bidirectional	
	PB_3	D_3		
	PB_4	HOLD	Control lines.	
	PB_5	READ		
	PB_6	WRITE	Always 6821 Outputs.	
	CA_1	D_0	1024H2	5832 Interrupt
	CB_2	D_1	1 Hz	Outputs at
	CB_1	D_2	1/60 H2	Specified
	CA_2	D_3	1/3600 H2	rates

Fig 2. 6821 PIA to MSM5832 interconnection

which follow should clarify how the PIA/5832 combination is used.

Clock set and read

The program in Listing 1 allows the clock to be set and read from (Microsoft) Basic and should thus be relatively easy to establish on most popular machines. The program was written in Applesoft Basic and has been annotated. If you are using another machine with the 6821, only a few areas will require amendment, the major one being to the address of the PIA. With the Apple, this address changes depending on which slot is used to house the card; lines 110 to 120 handle this. For most other systems the PIA will be mapped into a specific area of memory, allowing lines 110 and 115 to be deleted and line 120 modified so that variable A0 contains the base address (ie, the address of the first location in the PIA).

Other, more minor, amendments will also be necessary. In line 15 the string variable BELL$ is set to beep the Apple's speaker; if your machine has no such facility then set BELL$ to a null value (" "). In line 50 the variable CLR is set to a value which, when CALLed, clears from the current cursor position to the end of the line. The actual CALL is only used once (in line 400) and the two lines should be modified as appropriate. Finally, a few Basic words may require clarification for non-Applers: TEXT declares that an alphanumeric display is required rather than graphics; HOME clears the display and puts the cursor in the top left-hand position; INVERSE declares that all subsequent PRINTing will be black on white; and NORMAL declares that printing will be white on black.

If your design isn't based on a 6821 then, of course, the PEEKs and POKEs to control the interface will also have to be modified; to help you with this, I have REMarked all of them to show what they're doing.

The program has been kept relatively short and simple and will not be dissected. However, it could be substantially improved to make it easier to use. For example, it could allow normal date and time entry (eg, 9/5/82 for date) or automatically calculate the day of the week (see *Some Common Basic Programs*, 3rd edition, by L Poole & M Borchers, pub Osborne/McGraw-Hill, 1979, for a suitable method). Additionally, automatic leap year bit setting, instructions and extensive error-trapping would also be useful, but space does not allow such a lengthy program to be printed here.

```
10 AA$ = " CLOCK SET AND READ "
15 BELL$ =  CHR$ (7): REM  PRINTING BELL$ BEEPS SPEAKER
20  DIM AN(12)
50 CLR =  - 868: REM    CALL CLR TO CLEAR SCREEN FROM CURSOR ONWA
    RDS
100  TEXT : HOME : VTAB 1: HTAB (40 -  LEN (AA$)) / 2: INVERSE : PRINT
    AA$: NORMAL
110  VTAB 3: INPUT "ENTER CLOCK SLOT NUMBER, ";SL
115  IF SL < 1 OR SL > 6 THEN  PRINT BELL$: GOTO 110
120 A0 = 49280 + SL * 16
125 A1 = A0 + 1:A2 = A1 + 1:A3 = A2 + 1
150  VTAB 5: INPUT "<S>ET OR <R>EAD ?,        ";AN$
155  IF AN$ <  > "S" THEN 500
199  REM  SET CLOCK
200  POKE A1,0: POKE A0,255: REM   CONFIGURE ADDRESS LINES ON A S
    IDE AS ALL OUTPUTS
205  POKE A3,0: POKE A2,255: REM   CONFIGURE CONTROL & DATA LINES
     ON B SIDE AS ALL OUTPUTS
210  POKE A1,4: POKE A3,4: REM  GET READY TO SET A & B SIDE LEVEL
    S
220  VTAB 7: PRINT "ENTER;": VTAB 9
225  INPUT "UNITS OF MINUTES                        ";AN(2)
230  INPUT "TENS OF MINUTES                         ";AN(3)
235  INPUT "UNITS OF HOURS                          ";AN(4)
240  INPUT "TENS OF HOURS                           ";AN(5)
245  INPUT "UNITS OF DAYS                           ";AN(7)
250  INPUT "TENS OF DAYS                            ";AN(8)
255  INPUT "UNITS OF MONTHS                         ";AN(9)
260  INPUT "TENS OF MONTHS                          ";AN(10)
265  INPUT "UNITS OF YEARS                          ";AN(11)
270  INPUT "TENS OF YEARS                           ";AN(12)
275  INPUT "DAY OF WEEK (0 SAT, 6 FRI)              ";AN(6)
280  INPUT "NEXT FEB GOT 29 DAYS? (Y OR N)      ";AN$
285  IF AN$ = "Y" THEN AN(8) = AN(8) + 4: REM  ADJUST FOR LEAP YE
    AR
290 AN(5) = AN(5) + 8: REM  WORK IN 24 HR FORMAT
300  PRINT : INVERSE : PRINT "SWITCH WRITE ENABLE ON (SWITCH NO.
    4)";BELL$
310  INPUT "PRESS RETURN TO SET TIME            ";AN$
315  NORMAL
320  POKE A2,16: REM  TAKE HOLD LINE HIGH & STOP CLOCK
330  FOR I = 0 TO 12
340  POKE A2,AN(I) + 16: REM  SETUP DATA LINES
350  POKE A0,I: REM  SETUP ADDRESS LINES
360  POKE A2,AN(I) + 80: REM  TAKE WRITE LINE HIGH
370  POKE A2,AN(I) + 16: REM  TAKE WRITE LINE LOW
380  NEXT I
390  POKE A2,0: REM  TAKE HOLD LINE LOW & START CLOCK
400  VTAB 6: CALL CLR: VTAB 8: INVERSE : PRINT "SWITCH WRITE ENAB
    LE OFF";BELL$: NORMAL
410  REM  AUTOMATICALLY FALL INTO READ CLOCK
499  REM
500  POKE A1,0: POKE A0,255: REM   CONFIGURE ADDRESS LINES ON A S
    IDE AS ALL OUTPUTS
510  POKE A3,0: POKE A2,240: REM  CONFIGURE B SIDE WITH LOWER 4 B
    ITS AS INPUTS (DATA) & UPPER 4 BITS AS OUTPUTS (CONTROL)
520  POKE A1,4: POKE A3,4: REM  GET READY TO SET A & B SIDE LEVEL
    S
530  POKE A2,16: REM  TAKE HOLD LINE HIGH TO STOP CLOCK
540  POKE A2,48: REM  TAKE READ LINE HIGH
550  FOR I = 0 TO 12
560  POKE A0,I: REM  SETUP ADDRESS
570 AN(I) =  PEEK (A2) - 48: REM  READ & STORE DATA
580  NEXT I
590  POKE A2,16: REM  TAKE READ LINE LOW
600  POKE A2,0: REM  TAKE HOLD LINE LOW ( & ALLOW CLOCK TO CONTIN
    UE)
610 AN(5) = AN(5) - 8: REM  TAKE OUT 24 HR BIT
620  IF AN(8) =  > 4 THEN AN(8) = AN(8) - 4: REM  ALLOW FOR LEAP
    YEAR BIT
630  VTAB 10: HTAB 8: PRINT "TIME  ";AN(5);AN(4);".";AN(3);AN(2);
    ".";AN(1);AN(0)
640  VTAB 11: HTAB 8: PRINT "DATE  ";AN(8);AN(7);"/";AN(10);AN(9)
    ;"/";AN(12);AN(11)
650  VTAB 15: INVERSE : PRINT "USE 30 SEC ADJUST SWITCH IF NECESS
    ARY": NORMAL
660  IF AN(6) = 6 THEN TD$ = "FRIDAY": GOTO 690
665  IF AN(6) = 5 THEN TD$ = "THURSDAY": GOTO 690
670  IF AN(6) = 4 THEN TD$ = "WEDNESDAY": GOTO 690
675  IF AN(6) = 3 THEN TD$ = "TUESDAY": GOTO 690
680  IF AN(6) = 2 THEN TD$ = "MONDAY": GOTO 690
683  IF AN(6) = 1 THEN TD$ = "SUNDAY": GOTO 690
686 TD$ = "SATURDAY"
690  VTAB 18: PRINT "( BYE THE WAY: TODAY IS ";TD$;" !!)"
700  GOTO 530: REM   REPEAT READ & PRINT
```

Listing 1

ROM software

As discussed last month, the Apple I/O facility allows each peripheral card a 256-byte driving program. Having an intelligent card makes for much easier application programming when wishing to access time and date. Listing 2 shows a Basic program that gets and prints the time and date (using the yet-to-be-given driving program) and this should be compared with the much longer and more

Will your first computer be able to expand with you?

TODAY

Buying your first personal computer can be a confusing experience. You're entering a world of enormous potential, advanced technology, and a whole new language.

The first thing to remember is that a computer can be an invaluable business tool. The second is to simply relate your purchase to your business needs, now...and for the future.

12 MONTHS

You see, many small computers stay that way. They don't have the capacity for expansion as your business grows, your needs become greater, and you discover how much more a computer is capable of doing. In six or twelve months time, you find you may have to update to a new computer.

Unless you buy a Hewlett-Packard in the first place. The new HP 86 is the ideal first computer.

Small in price. Small in size. But very big in capacity. And it can expand with your needs.

Starting with the basic computer, you can easily add Mass Memory Storage, a Printer, and even Graphics options.

Software programs are available for accounting, data-based management, word processing, and other specific requirements such as statistical and financial analysis. And you can also purchase self-teaching programs.

Now you can take advantage of Hewlett-Packard technology. We have created some of the world's most advanced calculators, instruments and computer systems. Our dealers are fully trained, and more than willing to demonstrate the HP 86 system for you. And our follow-up and service support is first class.

For dealer locations call your local Hewlett-Packard office and ask for the Personal Computation Group. Sydney 887 1611, Melbourne 890 6351, Adelaide 272 5911, Perth 383 2188, Brisbane 304133, Canberra 80 4244.

Think of the future when buying your first computer.

Don't buy a system you'll outgrow. The HP 86 will grow with you.

THE FUTURE

HEWLETT PACKARD

HP 73

CLOCK IT TO ME

fiddly clock read part of Listing 1.

For convenience, I decided to use the well-known 2716 (single rail) EPROM for storing the driving programs. Although this device has room for eight driving programs, it is connected in such a way that only four spaces are available; selection of only one space is determined by the settings on two switches. Figure 3 shows how the switches relate to the EPROM memory map and where the driving program which follows (called 'normal format') should be located.

Listing 3 is the annotated assembler listing of the normal format driving program. This does not follow Apple's standard protocol for slot use (as briefly described in the Apple Reference Manual). Normally the Apple takes input, one character at a time, from the input device, stores it in the input buffer ($20-2FF) and outputs it to the current output device. If the input is the clock driving routine and the output is the Apple screen (as would normally be the case) this would dictate that every time the program wanted the time and date this information would automatically be printed on the screen, which is unduly restrictive and, for most programmers, would be an irritant.

A solution to this is not to use the standard I/O protocol for every character but to fill the input buffer with all characters at the same time and hence fool the Apple into thinking that it has handled each character separately. This works fine but there's still another problem to overcome: whenever an input statement is processed a question mark is sent to the current output device (which could be the printer or a

Address (Hex)		DIL Switch Setting 1 (A8)	2 (A9)
400	Not Available for driving program Storage (Address line A10 permanently tied to ground).		
3FF – 300	NORMAL FORMAT	off	off
2FF – 200	(FREE)	on	off
1FF – 100	(FREE)	off	on
FF – 0	(FREE)	on	on

Fig 3 Relationship between 2716 memory map and Apple clock card Dil switch settings

```
5  TEXT
10 D$ =  CHR$ (4)
20  PRINT D$"IN#4"
23  INPUT DA$,TI$
26  PRINT D$"IN#0"
30  PRINT "DATE ";DA$
40  PRINT "TIME ";TI$
90  END
```

Listing 2

```
C400-   78          SEI              Stop interrupts.
C401-   A9 E0       LDA   #$E0       Load a space character.
C403-   A4 24       LDY   $24        Print space at current output position.
C405-   91 28       STA   ($28),Y
C407-   20 10 FC    JSR   $FC10      Backspace cursor.
C40A-   A9 E0       LDA   #$E0       Repeat above 4 lines.
C40C-   A4 24       LDY   $24
C40E-   91 28       STA   ($28),Y
C410-   20 10 FC    JSR   $FC10
C413-   08          PHP              Save status.
C414-   68          PLA
C415-   8D FC 03    STA   $03FC
C418-   D8          CLD
C419-   20 58 FF    JSR   $FF58      Find which slot we are in.
C41C-   BA          TSX              (See Page 81 in Apple Reference Manual.)
C41D-   BD 00 01    LDA   $0100,X    (Note that there is no need to save)
C420-   8D F8 07    STA   $07F8      (the 6502 Registers, via a call to FF4A).
C423-   0A          ASL
C424-   0A          ASL
C425-   0A          ASL
C426-   0A          ASL
C427-   AA          TAX
C428-   A9 00       LDA   #$00       Configure PIA as follows;
C42A-   9D 81 C0    STA   $C081,X    A side (5832 addresses) — Outputs
C42D-   A9 FF       LDA   #$FF       B side (Bits 4-7; Control) -- Outputs
C42F-   9D 80 C0    STA   $C080,X    8 side (Bits 0-3; Data) — Inputs
C432-   A9 04       LDA   #$04
C434-   9D 81 C0    STA   $C081,X
C437-   A9 00       LDA   #$00
C439-   9D 83 C0    STA   $C083,X
C43C-   A9 F0       LDA   #$F0
C43E-   9D 82 C0    STA   $C082,X
C441-   A9 04       LDA   #$04
C443-   9D 83 C0    STA   $C083,X
C446-   A9 10       LDA   #$10       Take hold and read lines
C448-   9D 82 C0    STA   $C082,X    to 5832 high.
C44B-   A9 30       LDA   #$30
C44D-   9D 82 C0    STA   $C082,X
C450-   A0 0C       LDY   #$0C       Read in all time & date registers,
C452-   84 FF       STY   $FF        setting high bits and storing
C454-   A5 FF       LDA   $FF        temporarily in a part of the
C456-   9D 80 C0    STA   $C080,X    input buffer.
C459-   BD 82 C0    LDA   $C082,X
C45C-   18          CLC
C45D-   69 80       ADC   #$80
C45F-   A4 FF       LDY   $FF
C461-   99 10 02    STA   $0210,Y
C464-   C6 FF       DEC   $FF
C466-   A5 FF       LDA   $FF
C468-   10 EC       BPL   $C456
C46A-   A0 AF       LDY   #$AF       Put '/''s in correct place for format.
C46C-   8C 02 02    STY   $0202
C46F-   8C 05 02    STY   $0205
C472-   88          DEY              Put '.''s in correct place for format.
C473-   8C 0B 02    STY   $020B
C476-   8C 0E 02    STY   $020E
C479-   A9 AC       LDA   #$AC       Put ',' in correct place for format.
C47B-   8D 08 02    STA   $0208
C47E-   AD 17 02    LDA   $0217      Take time and date values
C481-   8D 01 02    STA   $0201      from temporary positions and
C484-   AD 1A 02    LDA   $021A      store in correct part of
C487-   8D 03 02    STA   $0203      input buffer.
C48A-   AD 19 02    LDA   $0219
C48D-   8D 04 02    STA   $0204
C490-   AD 1C 02    LDA   $021C
C493-   8D 06 02    STA   $0206
C496-   AD 1B 02    LDA   $021B
C499-   8D 07 02    STA   $0207
C49C-   AD 14 02    LDA   $0214
C49F-   8D 0A 02    STA   $020A
C4A2-   AD 13 02    LDA   $0213
C4A5-   8D 0C 02    STA   $020C
C4A8-   AD 12 02    LDA   $0212
C4AB-   8D 0D 02    STA   $020D
C4AE-   AD 11 02    LDA   $0211
C4B1-   8D 0F 02    STA   $020F
C4B4-   AD 15 02    LDA   $0215      Tens of hours value — always working
C4B7-   38          SEC              in 24 hour format so remove bit 3
C4B8-   E9 08       SBC   #$08       which is set.
C4BA-   8D 09 02    STA   $0209
C4BD-   A9 08       LDA   #$08       Tens of days value-reread and
C4BF-   9D 80 C0    STA   $C080,X    mask off bits (29 day Feb.) which
C4C2-   BD 82 C0    LDA   $C082,X    might be set.
C4C5-   29 FB       AND   #$FB
C4C7-   18          CLC
C4C8-   69 80       ADC   #$80
C4CA-   8D 00 02    STA   $0200
C4CD-   A9 10       LDA   #$10       Take read and hold lines to 5832 low.
C4CF-   9D 82 C0    STA   $C082,X
C4D2-   A9 00       LDA   #$00
C4D4-   9D 82 C0    STA   $C082,X
C4D7-   A2 11       LDX   #$11       Load x Reg with number of characters in buffer.
C4D9-   AD FC 03    LDA   $03FC      Reload status.
C4DC-   48          PHA
C4DD-   28          PLP
C4DE-   A9 8D       LDA   #$8D       Add carriage return character
C4E0    8D 11 02    STA   $0211
C4E3-   58          CLI              Allow interrupts again.
C4E4-   60          RTS              Back from whence we came.
C4E5-   FF          ???
```

Listing 3

CLOCK IT TO ME

DOS file) and the writing position of that device is advanced twice. If we assume — indeed force — the output device to be the Apple screen, this can be overcome by backspacing the cursor twice with overprinting by blanks.

Listing 3 shows how all this is done in practice. Some of the register contents are based on material in the excellent *What's Where in the Apple?* (W F Luebbert, Micro Ink Inc, 1981). In particular this relates to the overprinting and backspacing requirement and the setting up of registers after the input buffer has been filled.

Although the normal format listing is given in Slot 4 address space, the code itself is slot-independent since it automatically finds which slot it's in, as described in the Apple reference manual. Note, however, that in this context the call to save register values (as suggested by Apple) does not need to be executed and has not been included.

You can easily modify the program to produce different formats; you have only to arrange the time and date differently in the input buffer and remember to add a carriage return character at the end and load the X register with the total number of characters. If you want to extract the day of the week from the 5832, remember that this is available in part of the input buffer as a product of the normal format code; add the following line to the Basic program in Listing 2:
60 PRINT PEEK(534)—176

This will print a number between 0 for Saturday and 6 for Friday, which can easily be decoded to print the day names, as shown at the end of Listing 1.

The ROM software developed here is totally Apple-dependent and can't be used with other machines. However, there's nothing to stop you developing your own intelligent firmware specific to your machine. Probably the easiest way to do this would be to find out how variables are stored and then declare the variables TI$ and DA$ as the first variables in any program, followed by CALLs to the ROM software or USR routine to access the card and fill in the variables whenever necessary.

Timing things

As I discussed last month, the MSM5832 can generate interrupts, allowing accurate time intervals to be produced or, as shown here, accurate timing between events. The interrupt rate should, of course, be as fast as possible for the highest resolution. The 5832 can generate interrupts at 1024 Hz so it's possible to time to the nearest millisecond, which should be accurate enough for most micro-based applications.

Listing 4 shows the machine code to set up and count interrupts, and Listing 5 gives a driving Basic program; both were written for the Apple II. The easiest way to understand how they work is to follow the listings through in the way that the computer would execute them. Starting with the Basic program in Listing 5, a variable, TI%, is zeroed. This is the actual timing variable

```
0000:                 LST  ON
0000:            *************************************************************
0000:            *
0000:            *
0000:            *               DEMO TIMER.FP
0000:            *               -------------
0000:            *
0000:            *
0000:            *   THIS CODE USES THE CLOCK II TO GENERATE INTERRUPTS WITCH
0000:            *   ARE COUNTED INTO AN INTEGER VARIABLE IN THE TIME BETWEEN
0000:            *   PUSH BUTTONS 0 AND 1 BEING PRESSED.  THE INTERRUPT RATE
0000:            *   IS 1024HZ AND THUS THE RESOLUTION IS (APPROXIMATELY) TO
0000:            *   THE NEAREST 1/1000TH OF A SECOND.
0000:            *
0000:            *
0000:            *  LIMITATIONS;
0000:            *
0000:            *       ONLY WORKS WITH APPLESOFT.
0000:            *       CLOCK II IS ASSUMED TO BE IN SLOT 4.
0000:            *       PRESSING RESET STOPS INTERRUPTS.
0000:            *       COUNTS INTO FIRST DECLARED VARIABLE
0000:            *          WHICH MUST BE INTEGER.
0000:            *
0000:            *************************************************************
0000:            *
0000:            *
---- NEXT OBJECT FILE NAME IS DEMO TIMER.FP.OBJ0
0300:                    ORG  $300
03FE:            IRQVEC  EQU  $03FE      ;INTERRUPT VECTOR
C061:            PB0     EQU  $C061      ;PUSH BUTTON 0 - START
C062:            PB1     EQU  $C062      ;PUSH BUTTON 1 - STOP
C0C0:            SLOT4   EQU  $C0C0      ;CLOCK CARD ADDRESS - SLOT NO. 4
0069:            VARTAB  EQU  $69        ;APPLESOFT VARIABLE TABLE POINTER
03FD:            TEMPY   EQU  $03FD      ;TEMP STORE FOR Y REG
03FC:            TSTATUS EQU  $03FC      ;TEMP STORE FOR STATUS REG
0300:            *
0300:            *
0300:78                  SEI             ;NO INTERRUPTS
0301:AD 56 03            LDA  IRQHAND+1  ;SETUP IRQ LINKAGE
0304:8D FE 03            STA  IRQVEC
0307:AD 57 03            LDA  IRQHAND+2
030A:8D FF 03            STA  IRQVEC+1
030D:            *
030D:            * SETUP CLOCK FOR INTERRUPTS
030D:            *
030D:A9 00               LDA  #$00       ;MAKE A SIDE ALL OUTPUTS
030F:8D C1 C0            STA  SLOT4+1
0312:A9 FF               LDA  #$FF
0314:8D C0 C0            STA  SLOT4
0317:A9 04               LDA  #$04
0319:8D C1 C0            STA  SLOT4+1
031C:A9 00               LDA  #$00       ;B SIDE SETUP, LOWER 4 BITS INPUTS
031E:8D C3 C0            STA  SLOT4+3    ;              , UPPER 4 BITS OUTPUTS
0321:A9 F0               LDA  #$F0
0323:8D C2 C0            STA  SLOT4+2
0326:A9 04               LDA  #$04
0328:8D C3 C0            STA  SLOT4+3
032B:A9 0F               LDA  #$0F       ;TAKE ADDRESS LINES HIGH
032D:8D C0 C0            STA  SLOT4
0330:A9 20               LDA  #$20       ;TAKE READ LINE HIGH
0332:8D C2 C0            STA  SLOT4+2
0335:A9 05               LDA  #$05       ;INTERRUPT RATE OF 1024HZ
0337:8D C1 C0            STA  SLOT4+1    ;ROUTED VIA CA1 OF PIA
033A:            *
033A:AD 61 C0    START   LDA  PB0        ;LOOK FOR START SIGNAL
033D:10 FB               BPL  START
033F:58                  CLI             ;START TIMING
0340:            *
0340:AD 62 C0    STOP    LDA  PB1        ;LOOK FOR STOP SIGNAL
0343:10 FB               BPL  STOP
0345:78                  SEI              ;STOP HANDLING INTERRUPTS
0346:A9 00               LDA  #$00       ;STOP PRODUCING INTERRUPTS
0348:8D C1 C0            STA  SLOT4+1
034B:8D C3 C0            STA  SLOT4+3
034E:8D C0 C0            STA  SLOT4
0351:8D C2 C0            STA  SLOT4+2
0354:60                  RTS             ;BACK TO BASIC
0355:            *
0355:4C 58 03    IRQHAND JMP  IRQ        ;DUMMY JUMP - NOT EXECUTED
0358:            *
0358:            *   INTERRUPT ROUTINE
0358:            *
0358:8C FD 03    IRQ     STY  TEMPY
035B:08                  PHP
035C:68                  PLA
035D:8D FC 03            STA  TSTATUS
0360:D8                  CLD
0361:A0 03               LDY  #$03       ;INCREMENT FIRST DECLARED
0363:B1 69               LDA  (VARTAB),Y;BASIC VARIABLE BY 1
0365:18                  CLC             ;IT IS ASSUMED TO BE OF
0366:69 01               ADC  #$01       ;TYPE INTEGER
0368:91 69               STA  (VARTAB),Y
036A:88                  DEY
036B:B1 69               LDA  (VARTAB),Y
036D:69 00               ADC  #$00
036F:91 69               STA  (VARTAB),Y
0371:AD C0 C0            LDA  SLOT4      ;RESET 6821 IRQ REGISTERS
0374:AD C2 C0            LDA  SLOT4+2
0377:AD FC 03            LDA  TSTATUS    ;LOAD 6502 REGISTERS
037A:48                  PHA
037B:28                  PLP
037C:AC FD 03            LDY  TEMPY
037F:A5 45               LDA  $45        ;NOT FORGETTING ACC - SAVED BY MON
0381:40                  RTI             ;BACK TO LOOK FOR STOP SIGNAL
```

Listing 4

CLOCK IT TO ME

and is declared first since its absolute position in Apple memory can easily be established by referencing some page zero locations. Note also that it's an integer variable and values in it are contained in two bytes as opposed to 5-byte floating point numbers. Since we're counting interrupts, we only need an integer variable to store the total, but it does limit the maximum time that can be measured to about 32 seconds — more on this later.

Following the variable declaration, the screen is cleared, a title printed, the machine code routine (Listing 4) is loaded and the instructions are given (lines 10-50, Listing 5). The next line does a CALL to the machine code and attention now shifts to Listing 4.

First, the Apple has to be told the starting address of the routine which handles interrupts ('Setup Irq Linkage'). Then the clock card, which is assumed to be in slot 4, is configured to generate interrupts. The four interrupts which the 5832 generates all emerge together, one on each of the four data lines — see Figure 3 in last month's article. At this point (line 60 in Listing 4), the 5832 is generating interrupts but they are not yet linked to the 6502 interrupt line so the Apple knows nothing about them. Each interrupt rate line (data line) is connected to a 6821 interrupt input (Figure 2) and these are controlled by the control register — see Figure 1. Lines 62 and 63 set up the PIA's CA1 (which is connected to the 5832 data line 0 with the 1024 Hz signal) to pass interrupts through to the 6502 IRQ line. At this point the 6502 receives interrupts but ignores them as an SEI (set interrupt disable status) command was given earlier. The code now looks to see if pushbutton 0 is pressed (active when bit 7 is set) and, when this occurs, interrupts start to be handled (CLI) and a stop signal, on pushbutton 1, is looked for.

When an interrupt occurs, the Apple monitor automatically saves the contents of the 6502 accumulator and jumps through a page 3 address (which we earlier filled) to the routine starting at line 83 in Listing 4. This first stores the Y register and processor status and then increments the timing variable. It does this by using the Applesoft variable table pointer VARTAB, combined with the knowledge that for an integer variable the data is held in bytes 2 and 3 relative to its entry; see page 137 in the Applesoft manual. Before finishing the interrupt routine, and apart from reinstating the temporarily-saved registers, the interrupt flag has to be cleared. When the 6821 actually detects an interrupt, a flag is set and, if the control register is suitably configured, the message that the flag is set is passed on to the micro via the interrupt line. The flag has to be reset by the micro referencing the appropriate peripheral register in the 6821, as it won't reset itself; if it isn't reset, the interrupt line will remain permanently low and the 6502 will continually execute the IRQ routine.

Eventually the stop button is pressed, interrupts are stopped and control returned to Basic. At this point, TI% now contains the number of interrupts which occurred at the 1024 Hz rate. The rest is easy: convert and round the number, print it, reset TI% to zero and start again. The CALL-950 is a reference to an Apple monitor routine that clears from the current cursor position to the end of the line.

Before you go crazy trying to test the timer, note the 32 seconds' maximum time between events. This is because the highest value an integer variable can hold is 32767 — 32767/1024 (the interrupt rate) gives 32 seconds. With a little ingenuity this can be doubled to 64 seconds with one extra line of Basic — try it!

Conversion to Microsoft Basic on a 6502 machine shouldn't be too difficult; you just need to check on where VARTAB is held in your machine's page zero memory and that integer variables are stored in the same way as in the Apple. This is something at the core of most Microsoft implementations and will probably not have changed. If you can't discover this information, then a less elegant solution is to count into two of your own declared locations at the beginning or end of the machine code and then PEEK the contents into Basic.

Concurrent processing

If an interrupt structure is properly set up, a micro can apparently handle two or more jobs at the same time. In reality, of course, the micro is only ever doing one job — but to the user it all happens so quickly that the distinction is invisible. For instance, how many people know that their PET stops working on their problems every 1/60th of a second and goes off to update the time variable?

It would be useful to have the latest time and date continually displayed on the Apple screen, with the variables automatically updated, as on the PET, so that at any point they can be used without the hassle of INPUT TI$, DA$, etc. The code to do this, called CLOCK II.OBJ0.HIGH, is given in Listing 6 and is quite long as I have endeavoured to keep the system as flexible as possible. Because of space limitations a complete and detailed breakdown cannot be given; the rationale for this being that most readers would prefer to have something relatively sophisticated rather than something simple (admittedly with a full explanation) and pointers to the brilliant things that are possible by adding 'a few extra lines'.

Unfortunately, at this level of software and machine interaction, it is hard to give directions for non-Apple users on how to amend the program. The best advice I can give is for you to look at the overall structure of the code and then work out the specifics for your own machine.

The CLOCK II.OBJ0.HIGH code is managed from Applesoft Basic by the use of the '&' command which when encountered causes processing to jump to it via a page 3 vector. Program details and specific control mechanics are given at the beginning of Listing 6.

Before I discuss actual code, most users will want to type it in and play with it. Listing 7 is a short Basic program for testing Clock II Interrupt Handler. The section up to line 40 loads and prepares the system; those after 40 are for experimentation. Note that the machine code is BRUN *not* BLOADed and that ONERR is used if the clock card can't be found. Also note the setting up of the three interface variables (T%, TI$ and DA$) before any other variables are declared.

Lines after 40 demonstrate how to set up for printing and updating time and date to the screen. Line 110 appears somewhat lame; all that happens is that the program sits and waits at the INPUT. By this point in processing, the time and date are periodically (every second) being updated, while the programmer is collecting input from the user. In fact, your options here are many: you can interact with disk files, print results, use graphics etc. The only limitation is that the string variables TI$ and DA$ can't be put on the left hand side of an expression. Thus

TI$ = TI$ + "AM"

is not allowed, but

TT$ = TI$ + "AM" is.

Although not manipulated in the demonstration (but necessary in the program), print the three variables T%, TI$ and DA$ after you have stopped the program. T% will contain the number of seconds since &TI was given, and TI$ and DA$ will hold the last time and date.

The machine code in Listing 6 breaks down into three main parts: initialisation, '&' command handling, and interrupt handling.

Initialisation occurs automatically on BRUNning the code and hence is done only once. This includes discovering which slot the clock card occupies (from the EPROM software on it). If

```
5 TI% = 0
10  TEXT : HOME : INVERSE
15  PRINT " DEMONSTRATION TIMER ": NORMAL
20 D$ =  CHR$ (4): PRINT D$"BLOAD DEMO TIMER.FP.OBJ0"
30  VTAB 4: PRINT "PRESS GAME PUSH BUTTON 0 TO START"
40  PRINT "             AND"
50  PRINT "GAME PUSH BUTTON 1 TO STOP TIMER"
60  CALL 768
70 TI = TI% / 1024
75 TI = ( INT (TI * 1000 + .5)) / 1000
80  PRINT : PRINT : PRINT "WELL DONE!!!!!!"
90  PRINT : PRINT TI;" SECS ELAPSED BETWEEN PUSHES";
92  CALL  - 958: PRINT ""
95 TI% = 0
100  GOTO 30: REM  REPEAT TIMING
```

Listing 5

FMS SOFTWARE AND HARDWARE FOR CP/M BASED SYSTEMS

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

NEW — 16-Bit Software Available for the IBM PC, plus . . .

System Tools:
- Emulator/86 (the CP/M emulator)
- EM80/86
- PMATE 86
- UT86

Mailing List Systems:
- Postmaster

Telecommunications:
- ASCOM

Languages:
- Lattice C Compiler

Numerical Problem-Solving Tools:

DataBase Management Systems:
- T.I.M. III

Word Processing Systems and Aids:
- Wordstar
- Mailmerge
- Microspell
- Spellguerd

Disk Operating Systems:
MS-DOS — soon available configured for CompuPro Sweet 17 and Software Development System. Currently available for OEM license.

Media & Formats:
- IBM Personal Computer G1
- GodBout E1
- Seattle E1
- TecMar. E1
- Victor 9000 . . . . G3
- Sirius. G3

8-Bit Software Available

System Tools:
- BUG and uBUG
- DESPOOL
- DISILOG
- DISTEL
- EDIT
- EDIT-80
- FILETRAN
- IBM/CPM
- MAC
- MACRO-80
- MINCE
- PANEL
- PASM
- PLINK
- PLINK II
- PMATE
- RAID
- Reclaim
- SID
- TRS-80 Model II Cust Disk
- Unlock
- WordMaster
- XASM 05. 09. 18. 48. 51. 65. 67. F8. 400
- ZAP80
- ZDT
- Z80 Development Package
- ZSID

Telecommunications:
- ASCOM
- BSTAM
- BSTMS
- MicroLink 80
- RBTE

Languages:
- ALGOL-60
- APL/V80
- BASIC Compiler
- BASIC-80
- Precision BASIC
- BD Software C Compiler
- CBASIC-2
- CIS COBOL (Standard)
- CIS COBOL (Compact)
- COBOL-80
- KBASIC
- muLISP/mySTAR-80
- Nevada COBOL
- JRT Pascal
- Pascal/M
- Pascal/MT
- Pascal/M+
- Pascal/Z
- PL/I-80
- STIFF UPPER LISP
- S-BASIC
- Timin's Forth
- Tiny-C
- Tiny-C Two
- UCSD Pascal
- Whitesmith's C Compiler
- XYBASIC

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

Language and Application Tools:
- BASIC Utility Disk
- DataStar
- FABS
- FABS II
- Forms 1 for CIS COBOL
- Forms 2 for CIS COBOL
- MAGSAM III
- MAGSAM IV
- MAGSORT
- M/SORT for COBOL 80
- Programmer's Apprentice
- PSORT
- QSORT
- STRING/80
- STRING BIT
- SUPERSORT
- ULTRASORT II
- VISAM

Word Processing Systems and Aids:
- Benchmark
- DocuMate/Plus
- MicroSpell
- Letteright
- Magic Wand
- Spellguard
- TEX
- Textwriter III
- Wordindex
- WordStar
- WordStar Customization Notes

Data Management Systems:
- CONDOR
- Formula
- HDBS
- Hoe
- Microseed
- MDBS
- MDBS DRS QRS RTL
- dBASE II
- PRISM/LMS
- PRISM/IMS
- PRISM/ADS
- T.I.M. III

General Purpose Applications:
- CBS
- CBS Label Option
- Selector III-C2
- Selector IV

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

Numerical Problem-Solving Tools:
- T/MAKER II
- fpl
- PLAN80
- Analyst
- Microstar
- muSIMP/muMATH
- Statpak

Professional and Office Aids:
- Angel
- American Software Property Management Package
- Cornwall Apartment Managment
- Datebook
- GrafTalk
- Guerdian
- Professional Time Accounting
- Property Managemant
- Sales Pro
- Torricelli Author
- Wiremaster
- Univeir 9000 Series Legal Time Accounting

Media & Formats
The list of available formats is subject to change without notice. If you do not see your computer listed or are uncertain, call to confirm the format code for any particuler equipment.

- AES SUPER PLUS IV M4
- ADDS Multivision. . . . . . . .RT
- ALSPA 8in A1
- Altair 8800 . . . . B1
- Altos A1
- Apple CP/M 13 Sector . . . . .RG
- Apple CP/M 16 Sector . . . . .RR
- Archives 1 . . . . .SG
- AVL Eagle I . . . . .RB
- AVL Eagle II . . . . .ST
- BASF System 7100. . . . . . .RD
- Blackhawk Micropolis Mod II . Q2
- BMC IF-800 . . . . .SR
- California Computer Sys 8in. . A1
- CDS Versatile 3B O1
- COS Versatila 4 . . . . Q2
- Columbia Data Products 8in. . A1
- Columbia Data Products 5 26inS4
- Compal 80. . . . . Q2
- Commodore CBM/PET w/SSE Box + 8050 . . . . C2
- Commodore CBM/PET w/Maddison Z-RAM + 8050C4
- Compal 80. . . . . Q2
- Computer Ops N C HQ. S2
- Control Data 110 . . . . A1
- 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
- Datapoint 1550/2150 DD/SS .AA
- Datapoint 1550/2150 DD/DS .AB
- Datavue DU80-222 5.25in . . .M7
- DECVI 18 X
- Delta Systems A1
- Digi-Log Microterm II . . . . .RD
- Digi-Log System 2000 . . . . .RD
- Digi-Log System 1000 . . . . .RD
- Digi-Log System 1500 . . . . .RD
- Oirect OA1000 . . . . .M2
- DTC Micro 210A . . . . SC
- Durango F-85 . . . . .RL
- Dynabyte DB8/2 . . . . R1
- Dynabyte DB8/4 . . . . A1
- Exidy Sorcerer + Lifeboat CP/M 80 5.25 in .RW
- Exidy Sorcerer + Exidy CP/M 80 8in A1
- EXO A1
- Exxon 510. . . . . .M5
- Exxon 520. . . . . .M5
- Findex P6
- Godbout (MS-DOS). E1
- Heath H8 + H47. . . . . A1
- Heath H89 + Magnolia CP/M 80 . . . . P7
- Heath H89 + Heath CP/M 80 . P7
- Helios II B2
- Heurikon MLZ SS/DD . . . . . SN
- Heurikon MLZ DS/DD . . . . . SO
- Heuristics HCC Spectrum . . . A1
- HP-87 5.25in 5B
- Hewlett-Packard 125 5.25in . . SB
- Hewlett-Packard 125 8in. . . . A1
- IBEX 7100. . . . . .RQ
- IBM Personal Computer G2
- ICOM2411 Micro Floppy . . . R3
- ICOM3712. . . . . A1
- ICOM3812. . . . . A1
- ICOM 4511 5440 Cartridge CP/M 80 V1.4 . . . . D1
- ICOM 4511 5440 Cartridge CP/M 80 V2.X. D2
- ICL PC10 . . . . SO
- IMS 5000 . . . . .RA
- IMSAI VDP-40 . . . . R4
- IMSAI VDP-42 . . . . R4
- IMSAI VDP-44 . . . . R5
- IMSAI VDP-80 . . . . A1
- Industrial Microsystems 5000 .RA
- Industrial Microsystems A1
- Intel IPDS DS/DD M6
- Intel MDS SD/DD. A1
- Intersil Develop System A1
- Inter Systems Ithaca 800. . . . A1
- Intertec Superbrain DOS0.1 . . R7
- Intertec Superbrain DOS0.5-2x RJ
- Intertec Superbrain DOS 3x . .RK
- Intertec Superbrain QD . . . . RS
- ISC Intecolor 8063/8360/8963A1
- Kaypro S6
- Lanier E 1 . . . . M3
- Lanier Super + 4 (AES) M4
- Lanier Super No problem (Lanier) M4
- Lexitron VT1303 DSDD. . . . S8
- Lexor Alphasprint Model S1. . S1
- Lexor Lexoriter. S1
- Meca Delta 1 5.25in P6
- MICOM2001. . . . . B3
- MICOM2001E. B4
- MICOM3003. . . . . .M1
- Micromation. A1
- MicroMega 85 . . . . SC
- Micropolis Mod I Q1
- Micropolis Mod II. Q2
- MITS 3200-3202 . . . . B1
- Monroe OC 8820 DD/SS. . . .SW
- Morrow Discus A1
- Mostek A1
- MSD 5.25in . . . . .RC
- MULTI-TECH-I Q2
- MULTI-TECH-II Q2
- Nascom (Gemini drives) R3
- Nascom II with Lucas Drives . SL
- National MSC 6600. . . . . . . A1
- NCR81 40/9010 . . . . A1
- NEC PC-8001 . . . . .RV
- Nicolet Logic Analyser Model 764 . . . . SX
- NNC 80/80W A1
- NNC 80 . . . . A1
- North Star SD P1
- North Star DD. P2
- North Star QD. P3
- Northern Telecom 503 . . . . .SM
- Nylac Micropolis Mod II Q2
- Ohio Scientific C3 . . . . A3
- OKI IF-800 + MSA CP/M 80 . SP
- OKI IF 800 + OKI/Lifeboat CP/M 80 . . . . .SR
- (Above OKI entries replace catalog entry for OKI IF 800 format code RZ)
- Osborne 1 5.25in . . . . SA
- Otrona Attache . . . . .MC
- Pertec PCC 2000 . . . . A1
- PET/CBM w/Small Systems Engineering Box + 8050 . . C2
- PET/CBM w/Madison Z-RAM + 8050 . . . . C4
- Philips MICOM 2001 8in. . . . B3
- Philips MICOM 2001E B4
- Philips MICOM 3003 . . . . . .M1
- Philips P-2000. . . . . MA
- Processor Technology Helios II B2
- Quasar QDP100 . . . . A1
- Quay 500 . . . . .RO
- Quay 520 . . . . RP
- Quay 900 . . . . A1
- RAIR SD. R9
- RAIR DD . . . . .RE
- Research Machines 5.25in . . .RH
- Reserch Machines 8in A1
- Senco 7000 5,25in . . . . .RQ
- Sanyo MBC 1000 . . . . .SY
- Sanyo MBC 2000 . . . . SS
- Sanyo MBC 3000 . . . . .A1
- Seattle (16-bit) E1
- SD Systems 5.25in R3
- SD Systems 8in A1
- Spacebyte A1
- Tarbell 8in A1
- TEI 5.25in R3
- TEI 8in. A1
- Televideo DD/DS Sys 1(801) . S5
- Televideo DD/DS Sys 2 (806) . S5
- Toshiba T200 6.25in . . . . . . SF
- Toshiba T250 8in A1
- Triumph Adlar Alphatronic Model P2/64K . . . . .SV
- TRS-80 Model I (Standard Model) R2
- TRS-80 I + FEC Freedom . . .RN
- TRS-80 I + Omikron 5.25in . RM
- TRS-80 Model I + Omikron 8in A1
- TRS-80 Model I + Shuffleboard Bin A1
- TRS-80 Model II A1
- Vector MZ Q2
- Vector Systen 2800. . . . . . . A1
- Vector System B/VIP. Q2
- Vista V089 5.25in SD . . . . . RB
- Vista V200 5.25in DD P6
- Wengwriter. . . . . SE
- WORDPLEX. . . . . SZ
- XEROX 820 5.25in. S6
- XEROX 820 860 8in A1
- ZEDA 580. . . . . .SH
- Zenith Z89 + Magnolia CP/M 80 . . . . P7
- Zenith Z89 + Zenith CP/M 80. P7
- Zenith DD/SS . . . . .SK
- Zenith DD/DS. . . . . SJ
- Zilog MC22-20, 25, 50 . . . . . A1

Wordstar training guides $30 each

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

SEND FOR FULL LIFE BOAT CATALOGUE $10

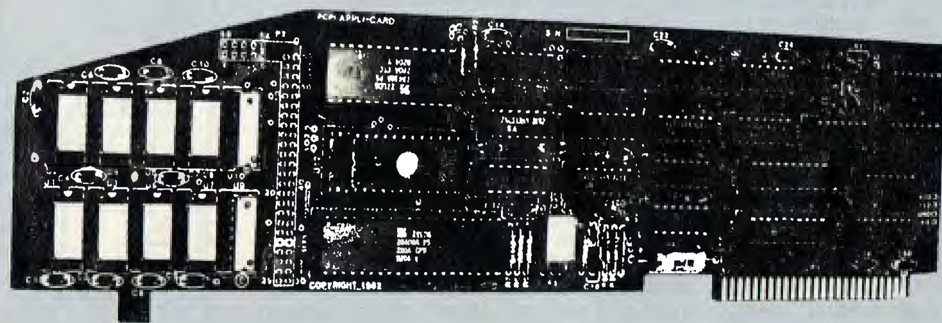
FMS 95 CANTERBURY ROAD, MIDDLE PARK, VICTORIA 3106.
Telephone: (03) 699 9899. Telex: AA 31604.

NEW... For Apple II & III*
from **PERSONAL COMPUTER PRODUCTS, INC.**

$495.00 EX TAX

APPLI-CARD™

4 or 6 mhz Z-80 and 64K ON-CARD MEMORY



THE ONE CARD SOLUTION TO EXECUTE CP/M* APPLICATION PROGRAMS

- Development Languages Available
- Applications Available
- Compare Our Features And Get The Best Value And Performance For Your Money.

FEATURES	Z-CARD*	SoftCard*	APPLI-CARD
6 mhz Z-80 available	No	No	Yes
64K on-card memory	No	No	Yes
CP/M with card	Yes	Yes	Yes
SB-80* available	No	No	Yes
40 col. to 255 col. horizontal scroll	No	No	Yes
Choice of application	No	No	Yes
2K PROM on the card	No	No	Yes
Real time clock available on the card	No	No	Yes
Expansion interface on the card	No	No	Yes
70 col. upper & lower case	No	No	Yes
A self-contained Z-80A or Z-80B with memory	No	No	Yes
One-card WordStar* execution	No	No	Yes
63K available for program development or execution	No	No	Yes
Menu driven set up	No	No	Yes

FMS SOFTWARE AND HARDWARE FOR CP/M BASED SYSTEMS

95 CANTERBURY ROAD, MIDDLE PARK, VICTORIA 3206.
Telephone: (03) 699 9899. Telex: AA31604.

the card is not found a jump to Basic error handling occurs. The interrupt and & page 3 vectors are then set up before a return is made to Basic.

When an '&' command is given processing automatically goes to ENTRYPT at line 183 in Listing 6. Time and date string variable table entries two and three are told where their data is held and the commands following the '&' are decoded. &E (end) is looked for, then &T and finally &I. If &T is found (printing of time and date), then all variables associated with the screen at that point are saved. Note that output does not have to go to the screen — it goes to the output device current when the '&' command was given. This is so 80-column video cards can be catered for, although this will involve extra coding and saving of 80-column screen variables. Whenever &T or &I are decoded there must be an associated number between 1 and 3 to set the rate of date and time revision. The GETRATE routine checks that a valid number exists and stores it for future reference.

Most of the remaining listing consists of subroutines. SETIRQ sets the clock card to interrupt at the specified rate. PRINT updates the time and date on screen by first saving current screen variables, then substituting screen variables it saved when the '&' command was given. The time and date are then printed and original screen variables restored. GETIME sets up the clock card and reads the time and date into variable space — note that this does not leave the clock card as an interrupt producer and JSR SETIRQ should normally be executed after JSR GETIME.

The final section is the interrupt handling code (IRQ), which saves all 6502 registers, updates the first integer variable by one and gets/stores the latest time and date. Depending on the contents of the PRINT.TD flag, the screen may also be updated. Finally the interrupt flags in the 6821 PIA are reset and the 6502 registers reinstated.

The Listing 6 machine code has not been optimised for speed or memory conservation as I suspect the savings would not justify the extra work. As an experiment I have benchmarked the system to determine the overhead involved in using this facility. Ordinarily an empty 1 to 50000 FOR. . . NEXT loop executes in 70.5 seconds. If &I1 is in operation this becomes 71 seconds and if &TI is used this comes to 72.5 seconds — a maximum increase in execution time of about three percent which applies across the board regardless of specific coding.

Conclusion

These articles have described a low-cost, high-specification clock/calendar card for the Apple II and similar micros. The emphasis has been placed on providing suitable driving software both as an example and for direct application. Inevitably there are improvements to be made and I hope that users who develop routines and modifications will document them in a future issue. . .

'We didn't realise he was dumb until I bought that thing.'

```
0000:              1          LST  ON
0000:              3 *
0000:              4 ***************************************************************
0000:              5 *
0000:              6 *
0000:              7 *              CLOCK II INTERRUPT HANDLER
0000:              8 *
0000:              9 *
0000:             10 *                 BY  B.S.T. MARRIOTT.
0000:             11 *
0000:             12 *
0000:             13 *   THIS CODE ALLOWS THE CLOCK II TO CONTINUOUSLY COMUNICATE
0000:             14 *   TIME & DATE INFORMATION TO A RUNNING APPLESOFT PROGRAM &,
0000:             15 *   IF NECESSARY, THE APPLE TEXT SCREEN OR OTHER OUTPUT DEVICE
0000:             16 *   THE APPLE INTERRUPT SYSTEM IS USED, CONTROL BEING
0000:             17 *   EXERCISED THROUGH THE APPLESOFT COMMAND '&'.
0000:             18 *
0000:             19 *
0000:             20 *   USE:
0000:             21 *
0000:             22 *   &I1    CAUSES THE CLOCK TO INTERRUPT EVERY SECOND & UPDATE
0000:             23 *          THE FIRST 3 DECLARED VARIABLES:
0000:             24 *
0000:             25 *      THE FIRST VARIABLE MUST BE INTEGER & IS INCREMENTED BY 1
0000:             26 *      FOR EACH INTERRUPT - AT ANY INSTANT THIS VARIABLE HOLDS
0000:             27 *      THE NUMBER OF INTERRUPTS THAT HAVE OCCURED SINCE
0000:             28 *      THE COMMAND WAS GIVEN.
0000:             29 *
0000:             30 *      THE SECOND & THIRD VARIABLES MUST BE STRINGS & HOLD
0000:             31 *      RESPECTIVELY TIME & DATE.
0000:             32 *      THE 1 CAN BE REPLACED BY EITHER A 2 OR A 3 WHERE,
0000:             33 *               2 CAUSES INTERRUPTS EVERY MINUTE &
0000:             34 *               3 ..................... HOUR.
0000:             35 *
0000:             36 *   &T1    AS FOR &I1 (AGAIN 1 CAN BE A 2 OR 3) BUT IN ADDITION
0000:             37 *          THE CURRENT TIME & DATE ARE OUTPUT TO THE TEXT SCREEN
0000:             38 *          & UPDATED EACH INTERRUPT. THE OUPUT IS LOCATED ON THE
0000:             39 *          SCREEN AT THE CURSOR POSITION CURRENT WHEN THE
0000:             40 *          COMMAND IS GIVEN.
0000:             41 *
0000:             42 *   &E     CAUSES INTERRUPTS TO BE STOPPED.
0000:             43 *
0000:             44 *
0000:             45 *   LIMITATIONS:
0000:             46 *
0000:             47 *          ONLY WORKS WITH APPLESOFT.
0000:             48 *          48K MEMORY NEEDED.
0000:             49 *          DOS MUST RESIDE IN NORMAL POSITION.
0000:             50 *          NOT RELOCATABLE.
0000:             51 *          PRESSING RESET STOPS INTERRUPTS.
0000:             52 *
0000:             53 ***************************************************************
0000:             54 *
0000:             55 *
----- NEXT OBJECT FILE NAME IS CLOCK II.OBJ0
9280:             57          ORG  $9280
9280:4C A8 92     58          JMP  INIT
9283:00           59 SLOT     DFB  $00      ;SLOT NUMBER ( IN FORM 40 FOR 4 ETC.
9284:00           60 RATE     DFB  $00      ;INTERRUPT RATE + 30
9285:00           61 PRINT.TD DFB  $00      ;PRINT FLAG -IF NEGATIVE THEN PRINT
9286:             62 *        TIME EACH INTERRUPT.
9286:00           63 XPNT     DFB  $00      ;SCREEN X COORD FOR PRINTING OF TIME
9287:00           64 YPNT     DFB  $00      ;...... Y ....................
9288:00           65 INVPNT   DFB  $00      ;SCREEN OUTPUT TYPE (NORMAL ETC.
9289:00 00        66 CSWLPNT  DW   $00      ;DEVICE HANDLER FOR PRINTING - NORMALY
928B:             67 *                       MONITOR ROUTINE
928B:00 00        68 TEMP1    DW   $00      ;TEMPORARY STORRAGE
928D:00           69 TEMP2    DFB  $00      ;......
928E:00           70 XTEMP    DFB  $00      ;SCREEN X COORD BEFORE INTERRUPT
928F:00           71 YTEMP    DFB  $00      ;...... Y ...........
9290:00           72 TOP      DFB  $00      ;TEXT WINDOW
9291:00           73 BOTTM    DFB  $00      ;CONFIGURATION BEFORE
9292:00           74 LEFT     DFB  $00      ;INTERRUPT OCCURED
9293:00           75 WIDTH    DFB  $00      ;.......
9294:00           76 YSAVTEMP DFB  $00      ;YSAV VALUE BEFORE INTERRUPT
9295:             77 TIME     DS   $08      ;TIME STRING
929D:00           78 STOP1    DFB  $00      ;STRING DELIMETER
929E:             79 DATE     DS   $08      ;DATE STRING
92A6:00           80 STOP2    DFB  $00      ;STRING DELIMETER
92A7:00           81 STOP3    DFB  $00      ;..........
92A8:             82 CSWLTEMP EQU  TEMP1    ;OUTPUT HANDLING ROUTINE BEFORE INTERRUPT
92A8:             83 INVTEMP  EQU  TEMP2    ;OUTPUT TYPE BEFORE INTERRUPT
0045:             84 ACC      EQU  $45      ;STORRAGE OF REGISTERS
0046:             85 XREG     EQU  $46      ; OVER INTERRUPT
0047:             86 YREG     EQU  $47      ;...........
0048:             87 STATUS   EQU  $48      ;...........
0020:             88 WNDLFT   EQU  $20      ;TEXT WINDOW LOCATIONS
0021:             89 WNDWDTH  EQU  $21      ;.....
0022:             90 WNDTOP   EQU  $22      ;.....
0023:             91 WNDBTM   EQU  $23      ;.....
FDED:             92 COUT     EQU  $FDED    ;MONITOR OUTPUT HANDLING ROUTINE
FC22:             93 VTAB     EQU  $FC22    ;MONITOR ROUTINE TO REPOSITION CURSOR
92A8:             94 *                      FOR SUBSEQUENT OUTPUT
```

```
0032:             95 INVFLG   EQU  $32      ;MON OUTPUT MASK
0024:             96 CH       EQU  $24      ;MON OUTPUT HORIZONTAL POSITION
0025:             97 CV       EQU  $25      ;........... VERTICAL   ......
00FE:             98 PZ1      EQU  $FE      ;TEMP PAGE ZERO LOCATIONS
00FF:             99 PZ2      EQU  $FF      ;ONLY USED DURING INITIALISATION
0035:            100 YSAV1    EQU  $35      ;TEMP STORRAGE LOCATION FOR Y REG BY
92A8:            101 *                       MONITOR COUT1 ROUTINE
00B8:            102 TXTPTR   EQU  $B8      ;APPLESOFT POINTER.TO NEXT TOKEN OR
92A8:            103 *    DATA ELEMENT
D995:            104 DATA     EQU  $D995    ;APPLESOFT ROUTINE THAT MOVES TXTPTR
92A8:            105 *                       TO END OF STATEMENT
0069:            106 VARTAB   EQU  $69      ;APPLESOFT VARIABLE TABLE POINTER
F2E9:            107 HANDLERR EQU  $F2E9    ;APPLESOFT ERROR HANDLING
03F5:            108 ANDVEC   EQU  $03F5    ;& JUMPS TO THIS ADDRESS
03F8:            109 APPSLOT  EQU  $3F8     ;APPLESOFT GETS SLOT NO. HERE (IF REQD)
03FE:            110 IRQVEC   EQU  $03FE    ;INTERRUPT VECTOR
0036:            111 CSWL     EQU  $36      ;ADDRESS OF OUTPUT HANDLING ROUTINE
AA53:            112 DOSCSWL  EQU  $AA53    ;DOS INTERCEPT ADDRESS OF OUTPUT HANDLING
92A8:            113 *    ROUTINE (48K)
C080:            114 BASE     EQU  $C080    ; BASE ADDRESS OF I/O SLOTS
92A8:            115 *      INITIALISATION
92A8:            116 *
92A8:            117 *
92A8:            118 *
92A8:            119 * FIND CLOCK SLOT NUMBER
92A8:            120 *
92A8:A5 FE       121 INIT     LDA  PZ1      ;LOOKING FOR 78 IN THE FIRST
92AA:8D 8B 92    122          STA  TEMP1    ;LOCATION & E0 IN THE THIRD
92AD:A5 FF       123          LDA  PZ2      ;LOCATION OF EACH SLOTS
92AF:8D 8C 92    124          STA  TEMP1+1  ;ROM SPACE
92B2:A9 00       125          LDA  #$00
92B4:85 FE       126          STA  PZ1
92B6:A9 C7       127          LDA  #$C7
92B8:85 FF       128          STA  PZ2
92BA:A2 07       129          LDX  #$07
92BC:A0 00       130 BACK     LDY  #$00
92BE:B1 FE       131          LDA  (PZ1),Y
92C0:C9 78       132          CMP  #$78
92C2:D0 0B       133          BNE  NEXT
92C4:A0 02       134          LDY  #$02
92C6:B1 FE       135          LDA  (PZ1),Y
92C8:C9 E0       136          CMP  #$E0
92CA:D0 03       137          BNE  NEXT
92CC:4C E0 92    138          JMP  FOUND
92CF:CA          139 NEXT     DEX                ;TRY NEXT SLOT
92D0:8A          140          TXA
92D1:F0 0A       141          BEQ  ERROR    ;CANT FIND CARD....
92D3:A5 FF       142          LDA  PZ2
92D5:38          143          SEC
92D6:E9 01       144          SBC  #$01
92D8:85 FF       145          STA  PZ2
92DA:4C BC 92    146          JMP  BACK
92DD:4C E9 F2    147 ERROR    JMP  HANDLERR ;JUMP TO BASIC ERROR HANDLING
92E0:8A          148 FOUND    TXA            ;CLOCK FOUND
92E1:8D F8 03    149          STA  APPSLOT  ;STORE SLOT FOR APPLESOFT
92E4:0A          150          ASL  A
92E5:0A          151          ASL  A
92E6:0A          152          ASL  A
92E7:0A          153          ASL  A
92E8:8D 83 92    154          STA  SLOT     ;SAVE SLOT INDEX RELATIVE TO $C080
92EB:            155 *
92EB:            156 *  SETUP '&' AND IRQ VECTORS
92EB:            157 *
92EB:AD 11 93    158          LDA  IRQHAND+1
92EE:8D FE 03    159          STA  IRQVEC
92F1:AD 12 93    160          LDA  IRQHAND+2
92F4:8D FF 03    161          STA  IRQVEC+1
92F7:A2 02       162          LDX  #$02
92F9:BD 0D 93    163 BACK2    LDA  ANDADDSS,X
92FC:9D F5 03    164          STA  ANDVEC,X
92FF:CA          165          DEX
9300:10 F7       166          BPL  BACK2
9302:AD 8B 92    167          LDA  TEMP1
9305:85 FE       168          STA  PZ1
9307:AD 8C 92    169          LDA  TEMP1+1
930A:85 FF       170          STA  PZ2
930C:60          171          RTS            ;INIT COMPLETE - BACK TO BASIC
930D:            172 *
930D:            173 *
930D:4C 19 93    174 ANDADDSS JMP  ENTRYPT  ;DUMMY JUMPS - NOT EXECUTED
9310:4C 59 95    175 IRQHAND  JMP  IRQ      ; ..
9313:4C 95 92    176 TIMEADDRSS JMP TIME    ; ..
9316:4C 9E 92    177 DATEADDRSS JMP DATE    ; ..
9319:            178 *
9319:            179 *
9319:            180 *     MAIN & ENTRY POINT
9319:            181 *
9319:            182 *
9319:78          183 ENTRYPT  SEI
931A:A0 09       184          LDY  #$09     ;GIVE SECOND & THIRD SIMPLE
931C:A9 08       185          LDA  #$08     ;VARIABLE TABLE ENTRIES (WHICH
931E:91 69       186          STA  (VARTAB),Y ;ARE ASSUMED TO BE STRINGS)
9320:C8          187          INY             ;DETAILS ON TIME & DATE
9321:AD 14 93    188          LDA  TIMEADDRSS+1 ;STRING STORRAGE AREA
```

Molymerx is the largest source of TRS-80® Software in Australasia

250 Programs, 60 pages

- Programming Utilities
- Graphic Adventures
- Word Processing
- Disk Operating Systems
- Business Applications

- Arcade Games
- Education
- Data Base Management

At the lowest possible prices

And if you think that's impressive, wait till you see the catalogue

But low prices are only one feature of our new catalogue. We think you will start us it as an invaluable reference tool.
And that's not all.
As we publish new software we send you an update. Our offer is to keep you abreast of new products, new services and our regular special prices.

Fill in the coupon and return it to:

Molymerx

Molymerx Pty Ltd.,
P.O. Box 900,
GOSFORD,
N.S.W. 2250
Tel: (043) 694-888

Molymerx Ltd,
P.O. Box 60152,
Titirangi, AUCKLAND,
NEW ZEALAND
Tel: (817) 4372

Telephone orders accepted.

Australasian distributors for: Aardvark, Acorn, Big Five, C.A.U., A.J.Harding, Hexagon, ICR Futuresoft, Logical Systems, Micro Systems Software, Mumford Micro Systems, Pioneer Software, Understanding.
® TRS-80 is a registered trademark of Tandy Corporation. Also compatible with System 80.

Please send ☐ copies of the Molymerx Mail Order Software Catalogue

Please use BLOCK CAPITALS

Name/Company..........................

Address..............................

......................Postcode.......

I enclose $....... @ $2.00 per copy (N.Z. $3.00)

☐ Cheque ☐ Money Order

OR Debit my Bankcard Exp.Dte.....

☐☐☐ ☐☐ ☐☐☐ ☐☐☐☐☐

Cardholder's Signature............................

APC12/82

CLOCK IT TO ME

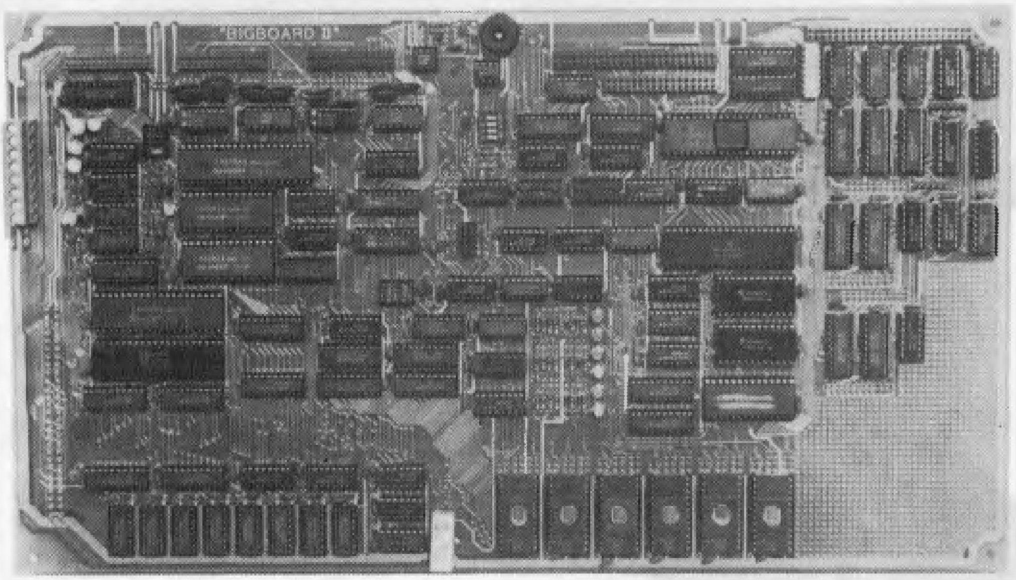
```
9324:91 69       189          STA  (VARTAB),Y
9326:C8          190          INY
9327:AD 15 93    191          LDA  TIMEADDRSS+2
932A:91 69       192          STA  (VARTAB),Y
932C:A0 10       193          LDY  #$10
932E:A9 08       194          LDA  #$08
9330:91 69       195          STA  (VARTAB),Y
9332:C8          196          INY
9333:AD 17 93    197          LDA  DATEADDRSS+1
9336:91 69       198          STA  (VARTAB),Y
9338:C8          199          INY
9339:AD 18 93    200          LDA  DATEADDRSS+2
933C:91 69       201          STA  (VARTAB),Y
933E:A9 00       202          LDA  #$00      ;STOP ANY PRINTING OF
9340:8D 85 92    203          STA  PRINT.TD  ;TIME & DATE
9343:A0 00       204          LDY  #$00
9345:B1 B8       205          LDA  (TXTPTR),Y
9347:C9 45       206          CMP  #$45      ;"E" -- SEE IF &E
9349:D0 14       207          BNE  NEXT1     ;NO - SO LOOK FOR ANOTHER COMMAND
934B:            208 *
934B:            209 *  END INTERRUPTS
934B:            210 *
934B:AE 83 92    211          LDX  SLOT
934E:A9 00       212          LDA  #$00
9350:9D 81 C0    213          STA  DASE+1,X
9353:9D 83 C0    214          STA  BASE+3,X
9356:9D 80 C0    215          STA  BASE,X
9359:9D 82 C0    216          STA  BASE+2,X
935C:4C A7 93    217          JMP  RTS
935F:            218 *
935F:            219 *
935F:C9 54       220 NEXT1    CMP  #$54      ;"T" -- SEE IF &T
9361:D0 29       221          BNE  NEXT2     ;NO
9363:            222 *
9363:            223 * PRINTING OF TIME & DATE REQUIRED.
9363:            224 *
9363:AD 53 AA    225          LDA  DOSCSWL   ;SAVE CURRENT VALUE OF OUTPUT
9366:8D 89 92    226          STA  CSWLPNT   ;ADDRESS.  THIS IS NORMALLY
9369:AD 54 AA    227          LDA  DOSCSWL+1 ;THE APPLE SCREEN (MONITOR)
936C:8D 8A 92    228          STA  CSWLPNT+1 ;HANDLING CODE & HENCE ALL
936F:A5 32       229          LDA  INVFLG    ;SCREEN FUNCTION LOCATIONS
9371:8D 88 92    230          STA  INVPNT    ;ARE SAVED AS WELL.
9374:A5 24       231          LDA  CH
9376:8D 86 92    232          STA  XPNT
9379:A5 25       233          LDA  CV
937B:8D 87 92    234          STA  YPNT
937E:A9 FF       235          LDA  #$FF      ;SET PRINT FLAG ON
9380:8D 85 92    236          STA  PRINT.TD
9383:20 AE 94    237          JSR  GETIME    ;PUT TIME & DATE ON
9386:20 F2 93    238          JSR  PRINT     ;SCREEN IMEDIATLY
9389:4C 90 93    239          JMP  GETRATE
938C:C9 49       240 NEXT2    CMP  #$49      ;"I" -- SEE IF &I
938E:D0 17       241          BNE  RTS       ;NO - NO MORE COMMANDS SO BACK TO BASIC
9390:            242 *
9390:            243 *  INTERRUPT RATE CHECKING
9390:            244 *
9390:A0 01       245 GETRATE  LDY  #$01
9392:B1 B8       246          LDA  (TXTPTR),Y
9394:C9 31       247          CMP  #$31      ;"1" (1 SECOND INTERRUPT RATE)
9396:F0 08       248          BEQ  DOWN
9398:C9 32       249          CMP  #$32      ;"2" (1 MINUTE ..............)
939A:F0 04       250          BEQ  DOWN
939C:C9 33       251          CMP  #$33      ;"3" (1 HOUR ................)
939E:D0 07       252          BNE  RTS       ;NO MORE OPTIONS SO BACK TO BASIC
93A0:8D 84 92    253 DOWN     STA  RATE
93A3:20 AA 93    254          JSR  SETIRQ
93A6:58          255          CLI             ;OK START INTERRUPTING
93A7:4C 95 D9    256 RTS      JMP  DATA      ;MOVE TXTPTR & GO BACK TO BASIC
93AA:            257 *
93AA:            258 * SETUP INTERRUPTS
93AA:            259 *
93AA:AE 83 92    260 SETIRQ   LDX  SLOT
93AD:A9 00       261          LDA  #$00
93AF:9D 81 C0    262          STA  BASE+1,X
93B2:A9 FF       263          LDA  #$FF
93B4:9D 80 C0    264          STA  BASE,X
93B7:A9 04       265          LDA  #$04
93B9:9D 81 C0    266          STA  DASE+1-X
93BC:A9 00       267          LDA  #$00
93BE:9D 83 C0    268          STA  BASE+3,X
93C1:A9 F0       269          LDA  #$F0
93C3:9D 82 C0    270          STA  BASE+2,X
93C6:A9 04       271          LDA  #$04
93C8:9D 83 C0    272          STA  BASE+3,X
93CB:A9 0F       273          LDA  #$0F
93CD:9D 80 C0    274          STA  BASE,X
93D0:A9 20       275          LDA  #$20
93D2:9D 82 C0    276          STA  BASE+2,X
93D5:AD 84 92    277          LDA  RATE
93D8:C9 31       278          CMP  #$31      ;"1"
93DA:D0 06       279          BNE  TRY2
93DC:A9 0C       280          LDA  #$0C
93DE:9D 83 C0    281          STA  BASE{3,X
93E1:60          282          RTS
93E2:C9 32       283 TRY2     CMP  #$32      ;"2"
93E4:D0 06       284          BNE  MUSTBE3
93EA:A9 05       285          LDA  #$05
93EB:9D 83 C0    286          STA  BASE+3,X
93EB:60          287          RTS
93EC:A9 0C       288 MUSTBE3  LDA  #$0C
93EE:9D 81 C0    289          STA  BASE+1,X
93F1:60          290          RTS
93F2:            291 *
93F2:            292 * PRINT TIME & DATE  (NORMALLY TO SCREEN)
93F2:            293 *
93F2:A5 20       294 PRINT    LDA  WNDLFT    ;SAVE EXISTING SCREEN
93F4:8D 92 92    295          STA  LEFT      ;FUNCTION DATA.
93F7:A5 21       296          LDA  WNDWDTH
93F9:8D 93 92    297          STA  WIDTH
93FC:A5 22       298          LDA  WNDTOP
93FE:8D 90 92    299          STA  TOP
9401:A5 23       300          LDA  WNDBTM
9403:8D 91 92    301          STA  BOTTM
9406:A5 35       302          LDA  YSAV1
9408:8D 94 92    303          STA  YSAVTEMP
940B:A5 24       304          LDA  CH
940D:8D 8E 92    305          STA  XTEMP
9410:A5 25       306          LDA  CV
9412:8D 8F 92    307          STA  YTEMP
9415:A5 36       308          LDA  CSWL
9417:8D 8B 92    309          STA  CSWLTEMP
941A:A5 37       310          LDA  CSWL+1
941C:8D 8C 92    311          STA  CSWLTEMP+1
941F:A5 32       312          LDA  INVFLG
9421:8D 8D 92    313          STA  INVTEMP
9424:            314
9424:A9 00       315          LDA  #$00      ;FILL SCREEN FUNCTION LOCATIONS
9426:85 20       316          STA  WNDLFT    ;WITH TIME & DATE
9428:85 22       317          STA  WNDTOP    ;PRINTING DATA.
942A:A9 28       318          LDA  #$28
942C:85 21       319          STA  WNDWDTH
942E:A9 18       320          LDA  #$18
9430:85 23       321          STA  WNDBTM
9432:AD 88 92    322          LDA  INVPNT
9435:85 32       323          STA  INVFLG
9437:AD 86 92    324          LDA  XPNT
943A:85 24       325          STA  CH
943C:AD 87 92    326          LDA  YPNT
943F:85 25       327          STA  CV
9441:AD 89 92    328          LDA  CSWLPNT
9444:85 36       329          STA  CSWL
9446:AD 8A 92    330          LDA  CSWLPNT+1
9449:85 37       331          STA  CSWL+1
944B:20 22 FC    332          JSR  VTAB
944E:            333 *
944E:            334 *
944E:A2 00       335          LDX  #$00      ;PRINT TIME & DATE
9450:BD 95 92    336 AGAIN    LDA  TIME,X
9453:F0 09       337          BEQ  DATEPRT
9455:18          338          CLC
9456:69 80       339          ADC  #$80
9458:20 ED FD    340          JSR  COUT
945B:E8          341          INX
945C:D0 F2       342          BNE  AGAIN
945E:E8          343 DATEPRY  INX
945F:BD 95 92    344          LDA  TIME,X
9462:F0 0D       345          BEQ  FINPRT
9464:E6 25       346          INC  CV
9466:20 22 FC    347          JSR  VTAB
9469:AD 86 92    348          LDA  XPNT
946C:85 24       349          STA  CH
946E:4C 50 94    350          JMP  AGAIN
9471:            351 *
9471:            352 *
9471:AD 92 92    353 FINPRT   LDA  LEFT      ;PUT BACK ORIGINAL
9474:85 20       354          STA  WNDLFT    ;SCREEN FUNCTION DATA.
9476:AD 93 92    355          LDA  WIDTH
9479:85 21       356          STA  WNDWDTH
947B:AD 90 92    357          LDA  TOP
947E:85 22       358          STA  WNDTOP
9480:AD 91 92    359          LDA  BOTTM
9483:85 23       360          STA  WNDBTM
9485:AD 8E 92    361          LDA  XTEMP
9488:85 24       362          STA  CH
948A:AD 8F 92    363          LDA  YTEMP
948D:85 25       364          STA  CV
948F:AD 8B 92    365          LDA  CSWLTEMP
9492:85 36       366          STA  CSWL
9494:AD 8C 92    367          LDA  CSWLTEMP+1
9497:85 37       368          STA  CSWL+1
9499:AD 8D 92    369          LDA  INVTEMP
949C:85 32       370          STA  INVFLG
949E:AD 94 92    371          LDA  YSAVTEMP
94A1:85 35       372          STA  YSAV1
94A3:20 22 FC    373          JSR  VTAB
94A6:60          374          RTS
94A7:            375 *
94A7:            376 *
94A7:            377 *
94A7:9D 80 C0    378 FETCH    STA  BASE,X    ;GET SPACIFIC TIME/BATE ELEMENT
94AA:BD 82 C0    379          LDA  BASE+2,X
94AD:60          380          RTS
94AE:            381 *
94AE:            382 *
94AE:AE 83 92    383 GETIME   LDX  SLOT      ;SETUP REGISTERS TO GET
94B1:A9 00       384          LDA  #$00      ;TIME & DATE
94B3:9D 81 C0    385          STA  BASE+1,X
94B6:A9 FF       386          LDA  #$FF
94B8:9D 80 C0    387          STA  BASE,X
94BB:A9 04       388          LDA  #$04
94BD:9D 81 C0    389          STA  BASE+1,X
94C0:A9 00       390          LDA  #$00
94C2:9D 83 C0    391          STA  BASE+3,X
94C5:A9 F0       392          LDA  #$F0
94C7:9D 82 C0    393          STA  BASE+2,X
94CA:A9 04       394          LDA  #$04
94CC:9D 83 C0    395          STA  BASE+3,X
94CF:A9 10       396          LDA  #$10
94D1:9D 82 C0    397          STA  BASE+2,X
94D4:A9 30       398          LDA  #$30
94D6:9D 82 C0    399          STA  BASE+2,X
94D9:            400 *
94D9:A9 05       401          LDA  #$05      ;LOAD & STORE TIME &
94DB:20 A7 94    402          JSR  FETCH     ;DATE REGISTERS
94DE:18          403          CLC
94DF:69 F8       404          ADC  #$F8
94E1:8D 95 92    405          STA  TIME
94E4:A9 04       406          LDA  #$04
94E6:20 A7 94    407          JSR  FETCH
94E9:8D 96 92    408          STA  TIME+1
94EC:A9 2E       409          LDA  #$2E
94EE:8D 97 92    410          STA  TIME+2
94F1:8D 9A 92    411          STA  TIME+5
94F4:A9 03       412          LDA  #$03
94F6:20 A7 94    413          JSR  FETCH
94F9:8D 98 92    414          STA  TIME+3
94FC:A9 02       415          LDA  #$02
94FE:20 A7 94    416          JSR  FETCH
9501:8D 99 92    417          STA  TIME+4
9504:A9 01       418          LDA  #$01
9506:20 A7 94    419          JSR  FETCH
9509:8D 9B 92    420          STA  TIME+6
950C:A9 00       421          LDA  #$00
950E:20 A7 94    422          JSR  FETCH
9511:8D 9C 92    423          STA  TIME47
9514:A9 08       424          LDA  #$08
9516:20 A7 94    425          JSR  FETCH
9519:29 FB       426          AND  #$FB
951B:8D 9E 92    427          STA  DATE
951E:A9 07       428          LDA  #$07
9520:20 A7 94    429          JSR  FETCH
9523:8D 9F 92    430          STA  DATE+1
9526:A9 2F       431          LDA  #$2F
9528:8D A0 92    432          STA  DATE+2
952B:8D A3 92    433          STA  DATE+5
952E:A9 0A       434          LDA  #$0A
9530:20 A7 94    435          JSR  FETCH
9533:8D A1 92    436          STA  DATE+3
9536:A9 09       437          LDA  #$09
9538:20 A7 94    438          JSR  FETCH
953B:8D A2 92    439          STA  DATE+4
953E:A9 0C       440          LDA  #$0C
9540:20 A7 94    441          JSR  FETCH
9543:8D A4 92    442          STA  DATE+6
9546:A9 0B       443          LDA  #$0B
9548:20 A7 94    444          JSR  FETCH
954B:8D A5 92    445          STA  DATE+7
954E:            446 **
954E:A9 10       447          LDA  #$10      ;DESELECT READING OF TIME
9550:9D 82 C0    448          STA  BASE+2,X  ;& DATE REGISTERS
9553:A9 00       449          LDA  #$00
9555:9D 82 C0    450          STA  BASE+2,X
9558:60          451          RTS
9559:            452 *
9559:            453 * INTERRUPT HANDLING
9559:            454 *
9559:86 46       455 IRQ      STX  XREG      ;SAVE 6502 REGISTERS
955B:84 47       456          STY  YREG      ;(NOTE THAT ACC IS SAVED
955D:08          457          PHP             ;AUTOMATICLY BY APPLE MONITOR)
955E:68          458          PLA
955F:85 48       459          STA  STATUS
9561:D8          460          CLD
9562:A0 03       461          LDY  #$03      ;INCREMENT FIRST DECLARED
9564:B1 69       462          LDA  (VARTAB),Y ;SIMPLE VARIABLE WHICH
9566:18          463          CLC             ;IS ASSUMED TO BE INTEGER
9567:69 01       464          ADC  #$01
9569:91 69       465          STA  (VARTAB),Y
956B:88          466          DEY
956C:B1 69       467          LDA  (VARTAB),Y
956E:69 00       468          ADC  #$00
9570:91 69       469          STA  (VARTAB),Y
9572:20 AE 94    470          JSR  GETIME    ;UPDATE DIME & DATE STRINGS
9575:AD 85 92    471          LDA  PRINT.TD  ;WANT TO PRINT TIME & DATE ?
9578:10 03       472          BPL  OUT       ;NO
957A:20 F2 93    473          JSR  PRINT     ;GO PRINT TIME & DATE
957D:20 AA 93    474 OUT      JSR  SETIRQ    ;SETUP IRQ'S AGAIN (WHICH GETIME
9580:            475 *                         DESTROYED)
9580:AE 83 92    476          LDX  SLOT      ;RESET 6821 IRQ REGISTERS
9583:BD 80 C0    477          LDA  BASE,X
9586:BD 82 C0    478          LDA  BASE+2,X
9589:A6 46       479          LDX  XREG      ;LOAD 6502 REGISTERS
958B:A4 47       480          LDY  YREG      ; ..
958D:A5 48       481          LDA  STATUS    ; ..
958F:48          482          PHA             ; ..
9590:28          483          PLP             ; ..
9591:A5 45       484          LDA  ACC       ;(INCLUDING ACCUMULATOR)
9593:40          485          RTI             ;BACK TO BASIC.
```

Listing 6

RITRONICS WHOLESALE PTY LTD

425 HIGH ST, NORTHCOTE, VICTORIA 3070
(03) 489 7099 (03) 481 1923
48-50 A'Beckett Street, Melbourne*

"BIG BOARD II"



Jim Ferguson, the designer of the "Big Board" distributed by Digital Research: Computers, has produced a stunning new computer that we will begin shipping in November called "Big Board II", it has the following features:

4 MHz Z80 — CPU AND PERIPHERAL CHIPS

The Ferguson computer runs at 4 MHz. Its monitor code is lean, uses Mode 2 interrupts, and makes good use of the Z80-A DMA chip.

64K DYNAMIC RAM + 4K STATIC CRT RAM + 24K E(E)PROM OR STATIC RAM

"Big Board II" has the three memory banks. The first memory bank has eight 4164 RAMs that provide 60K of user space and 4K of monitor space. The second memory bank has two 2Kx8 SRAMs for the memory-mapped CRT display and space for six 2732 As, 2Kx8 staticRAMS, or pin-compatible E(E)PROMs. The third memory bank is for RAM or ROM added to the board via the STD bus. Whether bought as a bare board, a full kit, or assembled and tested, it comes with a 200 nS2732A EPROM containing the monitor.

MULIPLE-DENSITY CONTROLLER FOR SS/DS FLOPPY DISKS

The new Ferguson single-board computer has a multiple-density disk controller. It can use 1793, 1797, or 8877 controller chips since it generated the signal with TTL parts. The board has two connectors for disk signal with 34 pins for 5.25" drivers, the other with 50 pins 8" drives.

VASTLY IMPROVED CRT DISPLAY

The new Ferguson SBC uses a 6845s CRT controller and 8002 Video Attributed controller to produce a display that will rival the display of quality terminals. Characters are formed by a 5x7 dot matrix on 15.75 KHz monitors and 7x9 dot matrix on 18.60 KHz monitors. The display is user programmable with the default display 24 lines of 80 characters.

STD BUS CONNECTOR

The Ferguson computer brings its bus signals to a convenient place on the PC board where users can solder an DSTD, bus cards can be plugged directly into it, and it can as well be connected by bus cable to industry-standard card cages.

DMA

The new Ferguson computer has a Z80-A DMA chip that will allow byte-wise data transfers at 500K bytes per second and bit serial transfers via the Z80-A S10 at 880K bytes per second with serial processor overhead, though the monitor for the new computer uses the DMA chip mainly for transferring data to and from disk, the chip can readily be used for other things since its "wait/ready" pin can be connected under software control to some half a dozen signal lines. When a hard-disk subsystem is connected to the "Big Board II" via its "SASI" interface, the DMA chip makes breathtaking disk performance possible.

"SASI" INTERFACE FOR WINCHESTER DISKS

The "Big Board II" implements the Host portion of the "Shugart Associates Systems Interface". Adding a Winchester disk drive is no harder than attaching a floppy-disk drive. A user simply 1: Runs a 50-conductor ribbon cable from a header on the board to any of several inexpensive controller cards for Winchester drives that implement the controller portion of the SASI interface. 2: Cables the controller to an appropriate drive, and 3: Provides power for the controller-card and drive. Since our CBIOS contains code for communication with hard-disk, that's all a user has to do to add a Winchester to a system!

A Z80-A S10/0 = TWO ASYNCHRONOUS/SYNCHRONOUS SERIAL PORTS

A PARALLEL KEYBOARD PORT = FOUR OTHER PARALLEL PORTS

USER 1/0

The new Ferguson single-board computer has one parallel port for an ASCII keyboard and four others for user-defined 1/0. When the computer is powered-up or reset, the monitor looks for a carriage-return at the keyuboard and serial ports. If the first carriage-return the monitor gets comes from the parallel keyboard, the monitor uses the board's video display circuitry to communicate with the user via a CRT. If the first carriage-return is typed at an ASCII terminal attached to a serial port, the monitor autabauds and makes the terminal the system console.

TWO Z80-A CTCs = EIGHT PROGRAMMABLE COUNTERS/TIMERS

The new Ferguson computer has two Z80-A CTCs. One is used to clock data into and out of the Z80-A S10/0, while the other is for systems and application use.

PROM PRDGRAMMING CIRCUITRY AND SOFTWARE

The new Ferguson SBC has circuitry and drivers for programming 2716s, 2732(A)s, or pin-compatible (E)EPROMs.

CP/M

CP/M with Russell Smith's CBIOS for the new Ferguson computer is available for $190. The CB10S is available separately for $39.50.
Actual board size: 39.6cm x 22.2cm.

Pricing and Availability:

Availability: We should start shipping the second week in November.
In single quantities, full kits cost $750.00 + tax, and A&T'd computers cost $895. There are attractive discounts that range to 35% for OEM's and dealers. For details about them please call Rod Irving on (03) 489 7099. Ie: 3 Ferguson II "Big Board" are less 20% off the one-off price, hard disks disk controllers, boxes and power supply to suit both 8" & 5¼" systems will be available.

* OFFICIAL OPENING JANUARY 4TH 1983

PRINTERS PRINTERS PRINTERS

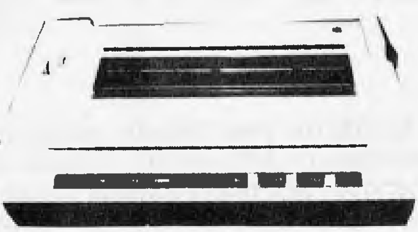
See Rod Irving for the best range and pricing within Australia ...

HIGH QUALITY

Prowriter 8510

- full 136 columns available
- 12CPS print speed
- selectable single & bi-directional print
- multiple characters per inch & variable fonts proportional spacing

$735



Model 1550

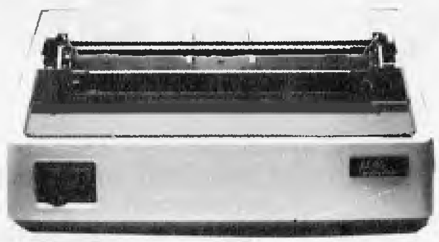
Star DP-8480 Matrix Printer

● low cost ● excellent print quality ● logic seeking, bi-directional ● graphics ● self diagnostics on board ● block graphic characters ● 9 x 7 dot matrix ● 80 chars/second ● low cost spool type ribbon ● tractor feed available **$495**

DOT MATRIX

Microline 80

● Dot matrix 80 CPS ● 40, 80, 132 columns ● rated for continuous operation ● friction & pinfeed paper support What else can we say about this great printer except that for this month only our SUPER LOW PRICE of **$435**



Microline 82A

● serial & parallel interface on board ● upper & lower case characters with true decoders ● multiple character sets ● friction and pinfeed paper tear bar ● 6 and 8 lines per inch ● 40, 80, 132 columns/page ● bi-directional print

MC83 Dot Matrix Printer

● print rate 150 CPS ● 80 ch/line or 132 ch/line ● supports 9½" fan fold 8½" roll paper, 8½" single sheets ● 9 x 7 matrix weight 9kgs ● MC83—2 parallel interface ● MC83—1 serial interface

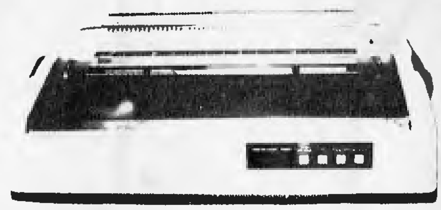
INTRODUCTORY OFFER THIS MONTH ONLY, 10% OFF THE RECOMMENDED RETAIL PRICE.

LETTER QUALITY

F10 Print Master Daisy Wheel

Fugitsu SP830 letter quality Daisy Printwheel Features

● fast max 80 CPS ● exceptional print quality ● selectable fine spacing ● line 3mm increment ● column 2mm increment ● high capacity ribbons ● optional tractor feed ● full graphic representation 11 520 points/sq inch ● supports metal/plastic printwheels



Diablo 620 Serial

New style Daisy Wheel that automatically aligns itself Robust quality printer with excellent letter quality prints 25 CPS bi-directional logic seeking. Proportionally spaced type fonts. 512 byte buffer. Weight 50lbs

Diablo 630

The absolute letter quality printer and industry standard Daisy Wheel Serial or parallel versions. Supports CTK/ACK, DC1/DC3, DTR Protocol. Supports metal & plastic print wheels (68/96 chars). Horizontal & vertical registration guaranteed. 40 CPS print speed 256 bytes buff.

INTRODUCTORY OFFER FOR THIS MONTH ONLY 10% OFF THE RECOMMENDED RETAIL PRICE

ALL PRICES ARE PLUS 20% SALES TAX UNLESS EXEMPT

RITRONICS WHOLESALE

425 HIGH STREET, NORTHCOTE 3070. MELBOURNE. (03) 489-7099.

P.O. BOX 235 NORTHCOTE 3070

Heavy items sent Comet road freight. Allow sufficient freight. Add sufficient money for registration, insurance etc. if required. Prices are correct at the time of advert. As we have no control over suppliers, prices are subject to change without notice. Specifications quoted are those of the manufacturer.

Opening January 4th 1983: 48-50 A'Beckett Street, Melbourne 3000

MAIL ORDER FORM

Please send me the following items:

....

....

Bankcard No. ....

Expiry Date ....

Name & Address ....

....

Signature ....

THE MICROCOMPUTER THAT'S WORTH ITS WEIGHT IN GOLD

The way prices of precious stones and metals shoot up and down these days, how could a wholesale jeweller know the value of his stock each day? And what to charge for it?

Well, one of Australia's biggest bought a Commodore microcomputer from us to solve this very problem.

Their assessment?
Brilliant!

We've just supplied their second Commodore, for general ledger and payroll work, and they're considering two more.

All of which says a lot about Commodore. And even more about the standard of service and general backup that we at Microcomp have made a personal trademark. In fact, most people who buy a microcomputer from us become long-term customers. It's better for us. And much better for them.

If you feel that this standard of service is what you need, please call.
You'll get personal attention, and the benefit of 12 years experience in the computer business.

MicroComp

B. S. MICROCOMP PTY. LTD.,
561 BOURKE STREET, MELBOURNE 3000.
CALL BILL SAUNDERS ON 614 1433.

commodore
COMPUTER

APC SUBSET

by Ian Davies

More machine language subroutines to add to your rapidly expanding library of useful utilities. As always, contributions are very welcome and should be sent to APC SUBSET, care of APC.

From one base to another

Dennis May has sent in routines to convert ASCII encoded numbers in one base to ASCII encoded numbers in any other base from 2 to 36. In bases higher than 10, digits with values 10 to 35 are represented by the letters A to Z. There are three routines. The first, given here in Datasheet XBIN, converts ASCII to unsigned 32-bit binary. The second converts 32-bit binary to ASCII, and the third, using the first two, converts ASCII from one base to another. You might like to have a go yourself at the last two, before Dennis's versions are printed – probably in February. (Read on to see what happens in January.)

Datasheet

```
;=XBIN - convert unsigned base 2-36 number to 32-bit binary
;/CLASS: 2
;/TIME CRITICAL?: no
;/DESCRIPTION: Converts an ACCII base 2-36 number to a
;/              32-bit unsigned binary number
;/ACTION: Clear 32-bit accumulator M0-M3
;/        fetch next digit; if CR then return with C
;/        clear.
;/        Multiply 32-bit accumultaor by base no and add
;/        digit; if overflow, then return with C set
;/        do next digit
;/SUBr DEPENDENCE: none
;/INTERFACES: none
;/INPUT: an ASCII coded number in RAM terminated by ODH
;/       (carriage return) with digits of values 10-35
;/       represented by the letters A-Z
;/       the address of the ASCII coded number in M@(low)
;/       and M5(high),
;/       the base of the number (between 2 and 36) in M6
;/OUTPUT: for a valid number, CY clear. The 32-bit result
;/        in M0-M3 with the most significant byte in M3
;/        M4-M6 unchanged
;/     for overflow: Cy set. M4-M6 unchanged. M0-M3
;/        indeterminate.
;/REGs USED: A,X,Y,P and M0-MC
;/STACK USE: 2
;/LENGTH: 106
;/PROCESSOR: 6502
XBIN:   LDX £4     ;byte count                   A2 04
        LDY £0     ;index counter                A0 00
XBIN1:  STY M0-1,X;clear                         94 ZZ
        DEX        ;accumulator a                CA
        BNE XBIN1  ;                             D0 FB
        STX MC     ;save index counter           86 ZZ
XBIN2:  LDY MC     ;load index counter to Y      A4 ZZ
        LDA M4,Y   ;fetch character              B1 ZZ
        CMP £$0D   ;if CR                        C9 0D
        BEQ END    ;then end                     F0 52
        SEC        ;convert                      38
        SBC £$30   ;                             E9 30
        CMP £$0A   ;to                           C9 0A
        BCC ASCY   ;                             90 02
        SBC £7     ;binary                       E9 07
ASCY:   PHA        ;save digit                   48
        LDX £4     ;byte count                   A2 04
        LDA £0     ;clear                        A9 00
XBIN3:  STA M6,X   ;accumulator                  95 ZZ
        DEX        ;                             CA
        BNE XBIN3  ;                             D0 FB
        LDA M6     ;make copy of base            A5 ZZ
        STA MB     ;in MB                        85 ZZ
        LDY £8     ;bit count                    A0 08
XBIN4:  ASL MB     ;get bit into carry           06 ZZ
        PHP        ;save it on stack             08
        ASL M7     ;shift left accumulator b     06 ZZ
        ROL M8     ;                             26 ZZ
        ROL M9     ;                             26 ZZ
        ROL MA     ;                             26 ZZ
        BCS OVFW1  ;jump if overflow             B0 1E
        PLP        ;pop bit off stack            28
        BCC XBIN6  ;skip if bit=0                90 0E
        LDX £$FC   ;byte pointer                 A2 FC
        CLC        ;                             18
XBIN5:  LDA M4,X   ;add acc A to acc B           B5 ZZ
        ADC MB,X   ;                             75 ZZ
        STA MB,X   ;store result in acc B        95 ZZ
        INX        ;                             E8
        BNE XBIN5  ;                             D0 F7
        BCS OVFW2  ;jump if overflow             B0 09
        INC MC     ;increment pointer index      E6 ZZ
        JMP XBIN2  ;repeat for next digit        4C YY YY
END:    CLC        ;clear carry for OK           18
        RTS        ;return                       60
OVFW1:  PLA        ;remove data                  68
OVFW2:  PLA        ;from stack                   68
OVFW3:  RTS        ;and return                   60
```

Before any 6502 coders leave us, they might like to look at the Z80 routines that follow, because these are to be given next month in 6502 code. That's what happens in January – or part of it, anyway.

Data integrity

Transmission errors are a problem of no mean consequence in computing, where the integrity of stored or transmitted data is of paramount importance and, naturally, there is a multitude of methods designed to deal with it. Some only indicate that an error has occurred while others can identify the error and thus allow for its correction.

John Kerr brings this important subject to the notice of Sub Set for the first time with the routines ECAL8 and EFIX8, which he says are based on the simplest method of data protection.

ECAL8 produces an error correction byte (ECB) for each data block up to 31 bytes in length. The method could accommodate 32 bytes but, as the ECB is to be appended before transmission or storage, John sensibly sets a limit of 31 data bytes so that the total number of bits of data and ECB is not greater than 256.

The ECB is a parity coding formed by using a parity mask unique to each bit in the data block. The mask is simply the binary number of the position of the bit from bit 0 of the byte following the data. Thus the parity shown in each bit of the ECB is of those data bits which have the corresponding bit set in their position number. For example, bit 5 of the ninth byte from the end would have the position number and parity mask 01001101 where the highest five bits of the mask give the byte number from the end of the block and the lowest three bits give the position of the data bit in that byte. Its parity would be included in bits 0, 2, 3 and 6 of the ECB.

Datasheet

```
;=ECAL8 - calculate error correction byte
;/CLASS: 1
;/TIME CRITICAL?: no
;/DESCRIPTION: calculates a one-bit error-correction byte
;/              (ecb) to be appended to a data block of 1
;/              to 31 bytes and subsequently used by the
;/              error detect/correct routine EFIX8
;/ACTION: Abort if no. of bytes=0 or GTHAN 31
;/        initialise mask to 8*(no of bytes)+7
;/        clear ecb
;/        for each byte in data block
;/        for each bit in current byte
;/        if bit is 1 then ecb← ecb XOR mask
;/        decrement mask
;/SUBr DEPENDENCE: none
;/INTERFACES:none
;/INPUT: A=no of bytes  HL points to 1st byte
;/OUTPUT: Cy reset: abort
;/        Cy set: A=ecb HL points to block+1
;/        (ecb can be appended by LD HL,A after the CALL)
;/REGs USED: AF,HL
;/STACK USE: 4
;/LENGTH: 36
;/TIME STATES: 103+332 per byte average
;/PROCESSOR: Z80
ECALB:  AND  A        ;terminate if                    A7
        RET  Z        ;no of bytes+0                   C8
        CP   +32      ;or over 31                      FE 20
        RET  NC       ;                                D0
        PUSH BC       ;save working registers          C5
        PUSH DE       ;                                D5
        RLA           ;multiply A by eight             17
        RLA           ;                                17
        RLA           ;                                17
        LD   D,+7     ;add 7 for initial mask          16 07
        OR   D        ;using D later as a              B2
        INC  D        ;constant for speed              14
        LD   E,A      ;save initial mask               5F
        XOR  A        ;clear ecb                       AF
NBYTE:  LD   B,D      ;B=8 for bit counter             42
        LD   C,(HL)   ;get byte in C                   4E
NXBIT:  SLA  C        ;shift bit into Cy               CB 21
        JR   NC,SKIPX;and if it is not '1'             30 01
        XOR  E        ;ecb=ecb XOR mask                AB
SKIPX:  DEC  E        ;next mask                       1D
        DJNZ NXBIT    ;repeat for all bits in byte     10 FB
        INC  HL       ;point to next byte              23
        LD   B,A      ;temp store ecb while            47
        LD   A,E      ;checking for end of block       7B
        CP   D        ;when mask=7                     BA
        LD   A,B      ;get ecb back in A               78
        JR   NC,NBYTE;repeat for all bytes in block    30 EF
        POP  DE       ;restore                         D1
        POP  BC       ;registers                       C1
        RET           ;                                C9
```

The error detection routine EFIX8 uses ECAL8 to calculate a new ECB for the mask, or bit position number, included in the parity of one received or retrieved data and exclusive-ORs this with the appended ECB to produce a 'correction code' which is the ECB (because the bit was set to '1') but not in the other (because the bit was reset to '0'). If none of the bits is inverted then the correction code will be zero. If error is indicated EFIX8 uses the code to isolate and re-invert the corrupt bit.

Datasheet

```
;=EFIX8 - detect and correct a one-bit error in data block
;/CLASS: 1
;/TIME CRITICAL?: no
;/DESCRIPTION: examines a 1 to 31 byte data block with
;/              appended ecb and corrects a single bit error
;/ACTION: abort if no of bytes=0 or >31
;/        calculate ecb of data block
;/        XOR with stored ecb to show any difference
;/        if no error then terminate
;/        else use highest five bits to point at corrupt byte
;/        terminating if error in stored ecb indicates
;/        a byte outside data block, use three lowest three
;/        bits to produce bit inversion mask and invert
;/        corrupt bit.
;/SUBr DEPEPDENCE: none
;/INTERFACES: none
;/INPUT: A=no of bytes excluding stored ecb
;/       HL points to first byte
;/OUTPUT: C=no of bytes in block
;/        Cy reset: abort
;/        Cy set: N set: no correction made HL points to
;/        block + 1
;/                N reset: correction made A= corrected
;/        byte HL points to corrected byte
;/REGs USED: AF,HL,C
;/STACK USE: 4(including CALL ECALB)
;/LENGTH: 40
TIME STATES: average 360+332 per byte
;/PROCESSOR: Z80
EFIXB:  LD   C,A      ;save no of bytes in block in C  4F
        CALL ECAL8    ;get new ecb of data block       CD XX XX
        RET  NC       ;terminate if ECAL8 aborted      D0
        XOR  (HL)     ;compare new and stored ecb      AE
        CP   +8       ;terminating if                  FE 08
        RET  C        ;no error is indicated           D8
        PUSH BC       ;save B                          C5
        LD   B,A      ;save correction code            47
        RRA           ;divide it by eight to get       1F
        RRA           ;position of                     1F
        RRA           ;corrupt byte                    1F
        AND  1FH      ;from end of block               E6 1F
        INC  C        ;test if position greater than   0C
        CP   C        ;no of bytes in block and        B9
        JR   NC,ESCAP;terminate if error in ecb only   30 11
        LD   C,A      ;move byte position to C         4F
        LD   A,B      ;get three lowest bits of code   78
        AND  07H      ;to give bit pos in byte into A  E6 07
        LD   B,+0     ;move HL to point                06 00
        SBC  HL,BC    ;at corrupt byte                 ED 42
        LD   B,A      ;using B as count                47
        INC  B        ; move a '1' into A              04
        XOR  A        ;in the                          AF
        CCF           ;same position                   3F
SHIFT:  RLA           ;as the                          17
        DJNZ SHIFT    ;inverted bit                    10 FD
        XOR  (HL)     ;re-invert it and                AE
        LD   (HL),A   ;restore it to data              77
ESCAP:  CCF           ;set non-abort flag              3F
        POP  BC       ;restore B                       C1
        RET           ;                                C9
```

John warns that this method can only cope with at most one bit error in the data block and requires that the ECB be received intact. More than one error will not be detected as such and may result in EFIX8 actually inverting a correct bit. Given this limitation, however, the method is fast and is capable of correcting about 95 percent of errors in a system where the probability of bit error is less than 0.4 percent. It also has a very low redundancy figure of 3.125 percent for a 31-byte block.

Flag of convenience

I would like to draw your attention to the use of the N flag in EFIX8 to indicate whether a correction has been made or not. The N result is a consequence of the operation immediately prior to exit, and John obviously noticed this fact and used it for speed and shortness in the routine. But it is difficult to test, unlike Cy and Z, which both have a relative jump conditional on their status. Should a Class 1 routine use any flag for passing information or be limited to the easily tested Cy and Z flags?

Flags are also the subject of a letter from K P Leary, who would like to see the convention Z = OK, NZ = Error used in Sub Set routines, as is used on the TRS-80. Apparently it makes for easier programming than the other way round. He also offers coding to complement the Z flag:

```
PUSH  BC
PUSH  AF
POP   BC
BIT   6,C
```

THIS IS THE COMPUTER THAT WALKS TO WORK...

THAT REVIEWS THE STOCK AND PROJECTS THE PROFITS...

THAT RUNS THE SALARIES...

THAT WRITES TO THE DEALERS..

THAT EVEN A CHILD CAN USE.

WEIGHT: 11.5 kg
Price: UNDER $2995

VIVA LA KAYPRO II. THE NEW REVOLUTION.

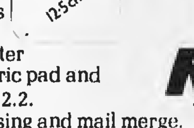
This portable business computer brings you the big screen, double density disks and more standard features.

It is Australia's most cost-efficient business partner and is priced at less than $2995. After a worldwide review of state of the art computers, the President Group chose the KAYPRO II as the optimum portable business computer to meet the demands of Australian business. It is light, easy to operate and is presented in a self-contained, fully-earthed metal case that stores comfortably under an airline seat. The KAYPRO II is today's most aggressive combination of performance and pricing.

KAYPRO II — MORE STANDARD FEATURES

23 cm green screen, restful to read and brightness adjustable. 80 columns x 24 rows. Double density disks, store 200,000 ch. each.

CPU: Z80™ with 64k Bytes. 8 baud rates selectable. RS-232C interface. Full typewriter keyboard, 14 key numeric pad and cursor control. CP/M® 2.2.
SELECT™ word processing and mail merge.
PROFITPLAN™ spreadsheet programme.
TEACH™ — self-teaching disk for word processing. S-BASIC® INSTALL



PRESIDENT GROUP STAR SERVICE

The President Group offers the services of 290 staff and over 160 vehicles, branches in all states, dealers in country areas and a service network of highly experienced professionals that's second to none. We guarantee the product, the service and the customer's satisfaction.

KAYPRO II
The New Revolution

For further information and full specification return the coupon to:
The Marketing Manager, President Office Machine Group,
100 George Street, Hornsby, N.S.W. 2077. Telephone: (02) 476 2700

president COMPUTERS

NAME
ADDRESS
TEL
CD/PG5

Call one of the President's men for the name of your local Kaypro II dealer

N.S.W. DISTRIBUTORS:
PRESIDENT COMPUTERS NSW
100 George Street
Hornsby Tel 476 2700

COMPUTERMAX
539 Pittwater Road
Brookvale Tel 93 1383

DELTA COMPUTERS
12 Fetherstone Street
Bankstown
Tel 705 6636

DESKTOP COMPUTERS
10 Cowper Street
Granville Tel 637 3062

VICTORIA DISTRIBUTOR:
PRESIDENT COMPUTERS VIC
Suite 1 609 St Kilda Road
Melbourne Tel (03) 529 1788

SOUTH AUSTRALIA DISTRIBUTOR:
PRESIDENT COMPUTERS S A
100 Pine Street
Adelaide
Tel (08) 223 6513

QUEENSLAND DISTRIBUTOR:
PRESIDENT COMPUTERS QLD
416 Logans Road
Stones Corner Brisbane
Tel (07) 397 0888

W.A. DISTRIBUTORS:
PRESIDENT COMPUTERS W A
248 Newcastle Street
Perth Tel (09) 328 6522

W J MONCRIEFF PTY LTD
176 Wittenoom Street
East Perth Tel (09) 325 5722

A.C.T. DISTRIBUTOR:
PRESIDENT COMPUTERS A C T
Shop 2 Waramanga Centre
Waramanga Tel (062) 88 2000

DEALER ENQUIRIES WELCOME

```
    POP     BC
```

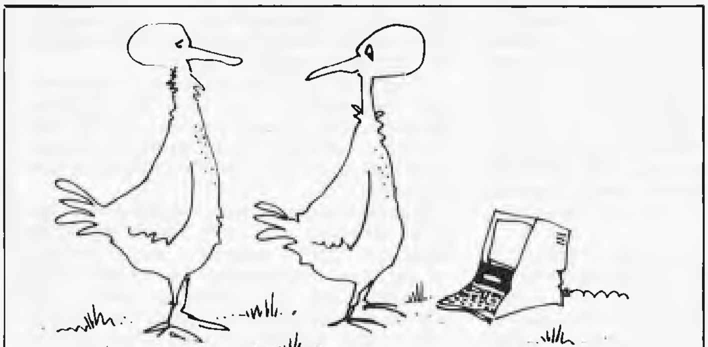
and the Sign and Parity flags:

```
    PUSH    BC
    PUSH    AF
    POP     BC
    LD      A,80H (LD A,+2 for Parity)
    PUSH    BC
    POP     AF
    POP     BC
```

8048 keyboard

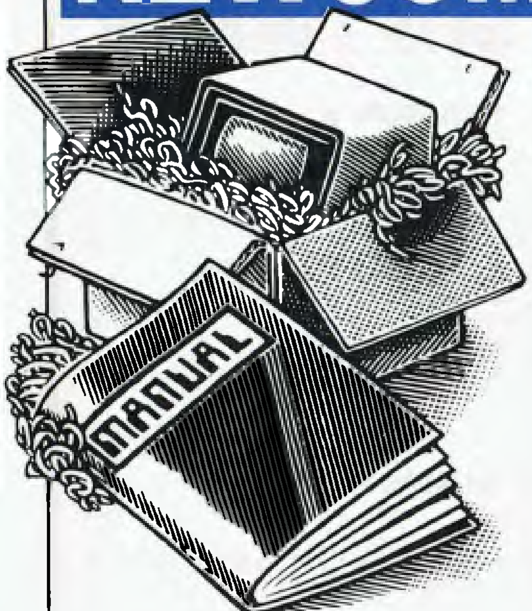
Since we are touching on interfacing, here is a listing of some 8048 assembler code from Jim Chance. It uses port 1 (P1) to read and debounce a 4 x 4 matrix keyboard. Jim says the 8048 processor is much used commercially and can be found in many products; in, for example, the Tandy Quickprinter 1.

```
;P1 0-3 = X drivers
;P1 4-7 = Y receivers routine waits for
;character decode table in ROM
;
KEYBRD:MOV   R7,#0              ;Read 256 times
       MOV   A,#11110000B
       OUTL  P1,A               ;Arouse all rows
KEY1:  IN    A,P1
       CPL   A                  ;-ve logic
       ANL   A,#11110000B
       JNZ   KEYBRD             ;Start again
       DJNZ  R7,KEY1
KEY2:  IN    A,P1
       CPL   A
       ANL   A,#11110000B
       JZ    KEY2               ;Await keypress
       MOV   A,#0
KEY3:  DEC   A
       JNZ   KEY3               ;Delay
       MOV   R7,#KEYCODES       ;Now read
       MOV   R6,#16             ;  16 keys
KEY4:  MOV   A,R7
       INC   R7
       MOVP  A,@A
       OUTL  P1,A               ;Drive row
       MOV   A,R7
       INC   R7
       MDVP  A,@A
       MOV   R5,A               ;Column test
       IN    A,P1
       CPL   A                  ;-ve logic
       ANL   A,#11110000B
       XRL   A,R5
       JNZ   KEY5               ;Not found
       DEC   R6                 ;0-15 range
       MOV   A,R6               ;Answer in R6 and A
       RET
KEY5:  DJNZ  R6,KEY4
       JMP   KEYBRD             ;Illegal
KEYTABLE:                       ;List of 16 pairs
                                ;03p drive code, i/p test
                                ;  1st ->0 etr
KEYCODES EQU KEYTABLE-$/256*256
X1     EQU   11110000B
X2     EQU   11111101B
X3     EQU   11111011B
X4     EQU   11110111B          ;O/p rows -ve logic
Y1     EQU   00010000B
Y2     EQU   00100000B
Y3     EQU   01000000B
Y4     EQU   10000000B          ;I/p columns +ve logic
       DB    X1,Y1,X1,Y2,X1,Y3,X1,Y4
       DB    X2,Y1,X2,Y2,X2,Y3,X2,Y4
       DB    X3,Y1,X3,Y2,X3,Y3,X3,Y4
       DB    X4,Y1,X4,Y2,X4,Y3,X4,Y4
```



'Good God! You don't still migrate by instinct, do you?!!'

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.

ADA–
A BRIEF ENCOUNTER

The US Department of Defence is interested in Space Invaders of a rather more dangerous variety than we micro buffs are used to. It has chosen Ada as the language to implement all future defence software. Mike Parr explains why . . .

Rarely does the design of a new programming language cause emotional discussion, but the introduction of Ada, the language selected by the US Department of Defense (DoD) is the exception. Basic, Forth and Pascal all have their critics, but the Ada debate is at a different level because of the sensitive area of its use — missile control, nuclear power plants, etc. Here I'll present a brief overview of the language, together with the arguments (sorry — discussions) on its value. I will attempt to be neutral!

History

In 1975 the US DoD decided to standardise on its programming language, and put out tenders for a language design to several firms. The winner was eventually called Ada (after Ada Lovelace, who worked with Babbage on an early mechanical computer). The language is based on Pascal, but has important additions to allow for real-time control, and to assist in the production of large software systems by teams of programmers. Because of the area of its use, the handling of errors is also vital — if your Space Invaders game crashes it is merely annoying, but if a fault occurs in a nearby nuclear power station you might be more concerned.

However, before criticising Ada, let's look at some features of the language.

What is Ada?

Fundamentally it is a Pascal-like language with these features:
Strong data typing.
Separately compilable 'modules'.
Facilities for the simultaneous execution of tasks.
(An explanation of this jargon will follow!)

However, Ada is more than just a language — it comes with a set of software tools collectively known as APSE (Ada Programming Support Environment), comprising editor, testing software, etc. In fact, APSE can handle all the paperwork associated with a project. Secondly, as Ada is a higher-level language than Pascal, it can be made use of at the program design stage, where one might have used pseudo-code or flowcharts. Certainly, a detailed study of Ada will affect the way you think about large programs, and the module and procedure facilities will encourage a 'divide and conquer' approach, making programs easy to read: after all, programs are read many times during development. (Forth fans please note!!)

The Ada attitude

Because Ada is intended for large programs, small examples are not always realistic — however it might be useful to approach the Ada way of thinking (on errors, reliability, etc) by criticising some fragments of MicroSoft Basic. Firstly:

```
10 IF F THEN 30
20 PRINT F * 3.142
```

These two lines have the following drawbacks:
The intial value of F is unclear (in fact it is zero).
F is rather meaningless — we would prefer longer names, eg, FLAG.
In Basic 1 (or —1 in some dialects) means 'true'. When F is zero, line 20 will be obeyed, and the missing line 30 will not produce an error.
In line 20 we multiply a logical (boolean) variable by a real number. What is the meaning of 'false'*3.142?

This brings out several points (bearing in mind that programs may be used in life-or-death situations) — ie, that incomplete programs should not be allowed to run, and that the type of each variable should be specified by the programmer to allow the compiler to reject misconceived operations. This feature, also present in Pascal and Algol 68, is called 'strong typing' and though it sounds like a good idea, is not popular among all programmers.

Now a subroutine to find the average value of the elements of an array:

```
1000 REM AVERAGING ROUTINE
1010 S = 0
1020 FOR I = 1 TO 100
1030 S = S + A(I)
1040 NEXT I
1050 AV = S/100
1060 RETURN
```

Again there are drawbacks:
The routine will only work on an array A, of size 100.
It also affects the value of I and S, which may be in use elsewhere in the program.
We may not want the user to have the power of editing the routine — it may be dangerous in certain environments.

These problems, present in Basic (and to some extent in Pascal) are all overcome in Ada. Bearing in mind the 'safety first' approach, let's look at some areas of the language.

Control structures

Here at last is a fragment of Ada, illustrating the familiar FOR loop of Basic and Pascal —

```
SUM: = 0 -- initial value
FOR INDEX IN 1 . . 100 LOOP
    SUM: = SUM + A(INDEX);
END LOOP;
```

Upper or lower case characters can be used for variables or reserved words, but I'll stick to capitals to differentiate programs from commentary. The layout is similar to Pascal but note that comments are preceded by '--', and that a semi-colon must follow each complete statement.

As well as FOR, we have WHILE, as in: —

```
WHILE X<11 LOOP
    any statements
END WHILE;
```

or even more simply, an infinite loop —

```
LOOP
    any statements
END LOOP;
```

The idea of WHILE is that one collects together the terminating conditions at the head of the loop, though sometimes it is simpler to break out from the loop body; in recognition of this practical requirement. Ada provides an 'exit' statement as in:

```
LOOP
    EXIT WHEN X<0.0001;
END LOOP
    -- next statement
```

We may also use GOTO in such a situation.

As well as repetition, we need statements for selection, and Ada provides IF and CASE, each being more powerful than their Pascal counterpart. The form of an IF is —

```
IF condition THEN statements
ELSIF condition THEN statements
ELSIF condition THEN statements
    etc
ELSE statements
END IF;
```

Note that this is one statement, *not* a series of nested IF's. You *could* choose to nest the statements by writing ELSE IF instead of ELSIF but each IF would need a matching END IF, which becomes clumsy.

The CASE statement at first sight seems similar to the IF, as in

```
CASE EXAM_MARK IS
    WHEN 100 = >
```

Illustration by Ian Wright

```
200 D$="
210 A$(0)="
220 A$(1)="
230 A$(2)="
240 A$(3)="
250 A$(4)="
260 A$(5)="":A$(7)="
270 A$(6)="
280 A$(8)="
290 A$(9)="
300 A$(10)="
310 A$(11)="
320 A$(12)="
330 Z$="":FORI=1T
340 NEXT:Z$=Z$+"
350 IFV<.2THENA$(2
360 IFV<.4THENA$(8
400 FORI=1TO8:T%(I
410 T%(I,3)=X:NEXT
500 PRINT"":FORI
510 GOSUB7000:FORJ
520 GOSUB7100:PRINT
530 GOSUB7000
540 POKE32849,233:POKE329
550 POKE33689,95:POKE337
560 FORI=0TO10:READJ:PO
600 FORI=65TO82:K$=CHR
610 GOSUB7200:IFP%(A
620 IFRND(1)<.5THE
630 PRINT"";K$;"
640 PRINT"
650 B=TI
1000 T=TI
1010 GETK$:IFK$<"A"OR
1020 GOSUB7200:P%(
1030 PRINTA$(P%(
2000 D=T%(
```

Before you buy a p
ask it a few per

1. Does it have 16 BIT technology with the ability to expand to up to 256,000 characters (256K Bytes) of user memory?

2. Does it have external storage which extends to 5¼″ or 8″ diskettes to suit any size business?

If the answers to any of these questions is no, perhaps you should be asking yourself a few questions.

Do I want a computer which can expand at the same pace as my business?

Should I settle for a 3 or 4 colour display monitor when the Panasonic JB3000 offers me 8 colours to easily create charts and graphics to target forecasts and trends?

To have the benefit of the best software available shouldn't I have a computer that's totally software compatible with the IBM and many other personal computers?

If I want to control my computer instead of vice versa, don't I need a computer which will offer a full range of training courses at Australia's leading business colleges included in the purchase price?

What other personal computer offers me up to 256K Bytes of user memory (that's up to four times the capacity of most of our competitors)?

Only available from: Myer, Lonsdale Street, Melbourne 661 3235; Angus & Robertson Business Centres, Sydney 267 5255; Bondi Junction, Parramatta, Newcastle, Canberra,

ersonal computer sonal questions.

3. Does it have an 8 colour high resolution graphic display monitor and software compatibility with IBM's personal computer?

4. Does it have complete computer training courses at Australia's leading business colleges?

Who produces a full range of business software packages like our exclusive Attache™ system specially designed to suit Australian business requirements?

Who provides quick economical after sales service all over Australia?

And of course you want the manufacturer you buy your personal computer from to be around for a long time. Panasonic is the world's largest supplier of consumer electronics. So odds are, they'll be in business as long as your business.

Where do you buy the Panasonic JB3000? In the special business centres of some of Australia's leading retailers, Myer and Angus & Robertson. At a price more than comparable with other brands and with an attractive finance package available.

The Panasonic JB3000.

It's got answers for every question you can ask it.

Panasonic JB-3000

Distributed by The Computer Company Pty. Ltd., a division of Singer Australia Ltd.

Wollongong, Melbourne, Brisbane, Adelaide, Perth, Hobart. Plus selected specialist dealers. For further information ring The Computer Company (02) 436 1733. COM 5021

ADA–A BRIEF ENCOUNTER

```
    PUT ("AMAZING"); --put is
'write'
  WHEN 80 . . 99 => 
    PUT ("GENIUS");
  WHEN 60 . . 80 => 
    PUT ("GOOD");
  WHEN 35 . . 59 => 
    PUT ("AVERAGE");
  WHEN OTHERS => -- ie, default
    PUT ("UNCLASSIFIED");
END CASE;
```

Ada insists that we must cater for every possible value of the selecting item (ie, EXAM_MARK), and also that no two WHEN items include the same values. Moreover, Ada will check this before running the program. In fact, there is an error in my CASE example, because 80 has been specified in two options — the program would produce a compile-time error. You might feel that such a checking facility is impossible, but, to accomplish this, Ada will not allow you to use variables in a WHEN item because variables can be checked only at run-time. The point of all this red tape is a statement which will not inadvertantly fall through to the following statement, and whose effect is not altered by re-ordering the WHEN list.

Finally, one overall point on CASE, IF, WHILE, and FOR is that the Pascal BEGIN/END scheme has been rejected in favour of more meaningful terminators — ie, END CASE, END IF and END LOOP.

Data types

This is a major part of Ada, and occupies about 30 pages of the reference manual. Because of this range, I'll examine the more novel features.

Initially, we have built-in types similar to Pascal, eg:

```
COUNT, I: INTEGER;
SUM: FLOAT: = 1.0; -- and initialise
A:ARRAY (1 . . 100) OF INTEGER;
```

We may also form 'derived' types as in:

```
TYPE VOLTS IS NEW REAL;
TYPE AMPS IS NEW REAL;
X,Y: VOLTS: = 0.0;
A,B: AMPS: = 1.0;
```

and can now write statements such as

```
X: = Y + 6.0;
A: = A + B;
```

but *not*

```
X: = A;
```

because the items are of different types.

The declaration of numeric types is worth looking at, as it is a considerable advance on Pascal. We can declare a limited range for an integer as in:

```
TYPE WEEKS IS RANGE 1 . . 52;
X: WEEKS;
```

but the novel point in Ada is to apply this to reals, and specify the range and precision:

```
TYPE FRED IS DIGITS 10
    RANGE -100000 .. 100000;
X,Y: FRED;
```

Here, we have said that, whatever the size of word on our computer, X and Y must be held to at least 10 digits of precision. Though Ada was not specifically aimed at numerical work, it provides better facilities than Basic, Fortran, or Pascal because programs will run on any machine, irrespective of the number of bits per word.

Arrays are defined in a similar manner to Pascal, but the actual size (within certain limits) can be left until run-time. Because array elements can be of any type (eg, character) and the size of arrays is flexible, we automatically have character strings; and because functions can return arrays as results, we can easily write powerful string functions. A simple example of procedure which prints out every element of an array of any size is:

```
PROCEDURE    ARRAY_PRINT
(A:ARRAY) IS
FOR I IN A'FIRST. . A'LAST LOOP
  PUT (A(I));
END LOOP
END ARRAY—PRINT;
```

Here we have made use of the reserved words FIRST and LAST, which find out the actual limits of the array index.

Program units

An important requirement in a language is that it allows the programmer
a) to split large programs into smaller manageable chunks (eg, subroutines in Basic);
b) to incorporate standard library packages in a program;
c) to write packages which can be safely used by other programmers.

Most languages do not meet these requirements!

Ada provides us with enhanced Pascal-style procedures and functions, plus two types of module called 'packages' and 'tasks'. As usual, I'll concentrate on the differences from Pascal.

Firstly we can 'overload' (enhance the meaning of) the arithmetic operators. As an example, we could overload the '+' operator to perform matrix addition. Assuming that the type MATRIX has been defined as a two-dimensional array of any size, we can add corresponding elements by:

```
FUNCTION "+" (X,Y:MATRIX)
      RETURN MATRIX IS
SUM:MATRIX;
BEGIN
FOR I IN 1 . . X' LAST(1) LOOP
  FOR J IN X'LAST(2) LOOP
    SUM(I,J):=X(I,J) + Y(I,J);
  END LOOP;
END LOOP;
RETURN SUM; -- ie, the result
END "+";
```

and we may now use '+' to work on matrices.

The package

For safe programming, the Pascal rules on the scope of variables are inadequate — ideally we want to provide a set of procedures for other users and also prevent the user from interfering with the internal workings. Consider a TV set — it is sealed up to keep out unskilled hands but can be manipulated to a limited extent by a set of controls on the front panel.

We can see why standard Pascal is unsafe by looking at an attempt to provide a set of 'turtle graphics' procedures. The requirement is to control the heading (angle A) of a

point on the screen by two procedures LEFT(N) and RIGHT(N), where N is any angle in degrees. We could try:

```
PROCEDURE LEFT (N:INTEGER);
VAR A:INTEGER
  Body of LEFT;
PROCEDURE RIGHT(N:INTEGER);
VAR A:INTEGER;
  Body of RIGHT;
```

Here, the programmer attempts to convey that variable A is 'private' to LEFT and RIGHT and must be updated only by these procedures. The problem is that as soon as the program exits from LEFT, all local variables (including A) are lost. In fact, such a variable needs to exist throughout the running of the program, and so must be declared at the outermost global level. Unfortunately, such variables can be accessed by any part of the program.

However, in Ada, we can declare a package, typically consisting of a set of procedures and data items, such that variables within it are inaccessible from outside the package, yet keep their value between procedure calls. For turtle graphics, we could write:

```
PACKAGE TURTLE IS
PROCEDURE LEFT(N:INTEGER);
PROCEDURE RIGHT (N:INTEGER);
END
PACKAGE BODY TURTLE IS
A:INTEGER;
-- followed by the body
-- of LEFT and RIGHT
END TURTLE;
```

Tasks

Many computing applications involve the simultaneous action of several tasks, such as writing to several VDUs at once, controlling a chemical reaction, or animating a Space Invaders game while detecting the player's move. Ada allows several tasks (similar in definition to a package) to be run together by high-level statements, whereas one is accus-

ADA–A BRIEF ENCOUNTER

tomed to dropping into assembly code to deal with interrupts, etc. In Ada, aside from a minimal amount of assembler which may be needed to address novel peripherals, the interrupt is treated as a low-level device that is hidden from the programmer by the language. Indeed, it is interesting to note that Ada, as well as allowing one processor to switch between several tasks in the conventional manner, will also allow the programming of systems in which each task has its own processor — ie, the program is distributed between several machines. In this area of real-time programming, potentially involving dangerous situations, it is realistic to expect errors, either within the programming system itself (eg, division by zero), or errors in results which can be detected by program. Bearing in mind that an error may be detected deep inside a series of nested procedure calls, Ada allows the programmer to specify an action to be taken when a particular 'exception' occurs, and also to control how many partially completed procedures are skipped before normal processing resumes.

...and

Yes, there's more. Particularly in the area of modules, data types, and input/output (provided by a set of procedures written in Ada), but if you want more detail, try one of the books listed below. However, bear in mind that it will be some time before you will be able to run Ada on your micro.

The debate

'Coherent'; 'versatile'; 'unwieldy'; 'baroque'; 'almost impossible to implement'; 'unreliable'.

These phrases come from recent Ada articles in the computing press; the anti-Ada faction maintains that it has features grafted together to allow for any possible requirement, making it a very large language, difficult to understand and use. If the language is incomprehensible to most programmers, will the programs they produce be of poor quality? Well, the point about complexity is true — many programmers will find Ada difficult, so perhaps the choice of a Pascal-like subset might be appropriate. On the other hand, real-time systems are complicated in any language, so perhaps one which acknowledges their existence is an improvement. However, one of Ada's strongest opponents, Professor Hoare says that it must never be used to implement large software systems because it is unreliable in the areas of error-handling and multi-tasking. Bearing in mind that the DoD has had several false alarms in recent years, such criticisms are indeed serious.

Where is Ada?

As Ada compilers are reputed to be six times larger than Pascal compilers, full Ada is not likely to appear on today's typical micro system. Currently though one can purchase micro Ada subsets from the USA.

The future

Will it catch on? Opinions differ, but it will have to overcome Cobol/Fortran inertia from industry — though some firms involved in real-time programming are keen. Ada subsets (eg, without the multi-tasking facilities) will become more widespread, competing with Pascal on CP/M systems.

Perhaps the personal computer language battle will end up being fought between Unix plus 'C', Pascal, Smalltalk, and Ada?

Further reading

The number of Ada books is increasing rapidly, so this is not an exhaustive list.

Programming With Ada by Wegner — published by Prentice Hall.

Ada — An Introduction by Ledgard — published by Springer-Verlag (also includes the Ada reference manual).

Problem Solving With Ada by Mayoh — published by Wiley (uses turtle graphics in many of the examples).

The Ada Programming Language by Pyle — published by Prentice Hall.

END

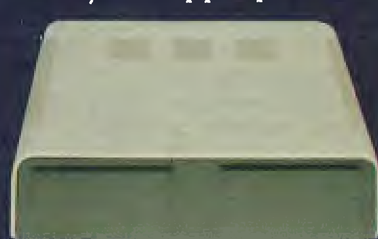
Run rings around other Apple users

Introducing a new and powerful Apple-compatible hard-disk drive . . the RO 100 series from Ampec. This drive/subsystem offers a complete add-on mass storage subsystem for the Apple II microcomputer.

Available in capacities from 4 Mbytes to 16 Mbytes, these Saturn hard-disk drives have on-board microprocessors and feature fast access times to give you considerable improvement in system throughput.

This current release of the Saturn subsystem includes DOS 3.3, Apple, Pascal and CP/M software support. With the subsystem installed, and using the supplied support programmes, you can have up to 16 Mbytes of additional disk storage — that's roughly equivalent to adding 80 Apple mini floppy disk drives to your Apple computer.

Now at your Ampec dealer — the Saturn RO 100 disk drives. They really put the icing on the cake — or, if you like, the cream on your Apple pi.



AMPEC

The absolute in peripheral thinking.

```
0E77   1F 04 0C 0A 12 ; k
0E7C   00 00 1F 00 00 ; l
0E81   1E 02 0C 02 1C ; m
0E86   1E 02 02 02 1C ; n
0E8B   0C 12 12 12 0C ; o
0E90   3E 0A 0A 0A 04 ; p
0E95   04 0A 0A 0A 3E ; q
0E9A   1E 04 02 02 02 ; r
0E9F   14 16 16 16 0A ; s
0EA4   00 02 1F 12 00 ; t
0EA9   0E 10 10 08 1E ; u
0EAE   06 08 10 08 06 ; v
0EB3   0E 10 0C 10 0E ; w
0EB8   12 12 0C 12 12 ; x
0EBD   26 28 28 28 3E ; y
0EC2   12 12 1A 16 12 ; z
0EC7   00 04 0E 11 00 ; {
0ECC   00 00 1F 00 00 ; |
0ED1   00 11 0E 04 00 ; }
0ED6   02 01 02 04 02 ; ~
0EDB   15 0A 15 0A 15 ;
```

Listing 2 Character codes

The main part of the program is the print routine which is called every time a character is output. It makes use of a look-up table to find the dot patterns necessary to form the required character. Five of these are output for each character printed. A single dot-width is then printed to provide spacing between letters.

The program is written to drive the printer through the Z80 PIO fitted to the Nascom. It uses port 5 for data, bit 0 of port 4 in input mode for the BUSY line and bit 1 of port 4 in output mode for the STROBE line. The subroutine AOUT outputs the character in the accumulator to the printer. It outputs the character through port 5 and then repeatedly checks bit 0 of port 4 until it goes low indicating that the printer is ready to accept data. The data is then strobed into the printer buffer by taking bit 1 of port 4 low and then bringing it high again.

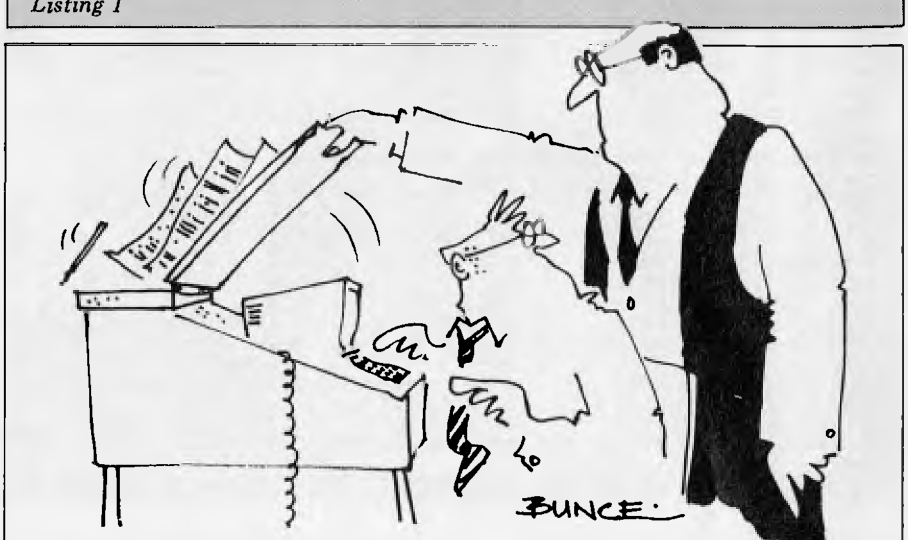
If the printer is connected to the serial port of the computer then lines 14-19 in the program should be replaced by a call to the NAS-SYS external command, DF 59, and the subroutine AOUT should be replaced by calling the NAS-SYS output to serial port routine SRLX; the code for this is DF6F.

The program is presented in two parts: first, the assembler listing of the printer control program (Listing 1), and second, a hex dump (Listing 2) which contains the codes necessary to print the 96 standard ASCII characters supported by the printer in its normal (10 cpi) printing mode. Since all these codes are contained in software, it's not difficult to change the values to redefine single characters or even the entire character set if necessary. To change a character each of the five bytes corresponding to that character should be replaced by codes to form the new character. Bit 0 of the code corresponds to the top dot and the bit 5 of the code corresponds to the bottom dot. As it stands, the program only prints characters in the range 20H to 7FH since these are the only ASCII codes normally supported by the printer. To print codes outside this range will need some modifications to the driving software.

Modification of the program to drive other printers with pin-addressable graphics should not be difficult but the exact details will obviously depend on the way that a particular printer controls the print-head.

```
0CC1              0044          ;** MAIN PRINT ROUTINE **
0CC1              0045          ;
0CC1  F5          0046 PRINT    PUSH AF              ;Save data
0CC2              0047          ;
0CC2  FE 0D       0048          CP   0D              ;If carriage-return
0CC4  28 24       0049          JR   Z,NEWLIN        ;then start new-line
0CC6              0050          ;
0CC6  FE 20       0051          CP   " "             ;If code before space
0CC8  38 1E       0052          JR   C,END           ;then don't print
0CCA              0053          ;
0CCA  E6 7F       0054          AND  07F             ;Reset bit 7 if set.
0CCC              0055          ;
0CCC  5F          0056          LD   E,A             ;
0CCD  16 00       0057          LD   D,0             ;Put 8-bit data
0CCF  62          0058          LD   H,D             ;into 16-bit
0CD0  6B          0059          LD   L,E             ;register and
0CD1  29          0060          ADD  HL,HL           ;multiply by
0CD2  29          0061          ADD  HL,HL           ;character width
0CD3  19          0062          ADD  HL,DE           ;i.e. 5
0CD4              0063          ;
0CD4  11 60 0C    0064          LD   DE,CHRTBL-0A0   ;Calculate address
0CD7  19          0065          ADD  HL,DE           ;of character data
0CD8              0066          ;
0CD8  06 05       0067          LD   B,5             ;Set up loop
0CDA              0068          ;
0CDA  7E          0069 LOOP     LD   A,(HL)          ;Get bit pattern
0CDB  C6 20       0070          ADD  020             ;generate graph code
0CDD  CD ED 0C    0071          CALL AOUT            ;and output it
0CE0  23          0072          INC  HL              ;points to next code
0CE1  10 F7       0073          DJNZ LOOP            ;and get next code
0CE3              0074          ;
0CE3  3E 20       0075          LD   A," "           ;Load accumulator
0CE5  CD ED 0C    0076          CALL AOUT            ;to print space
0CE8              0077          ;
0CE8  F1          0078 END      POP  AF              ;Retrieve data
0CE9  C9          0079          RET                  ;and return
0CEA              0080          ;
0CEA              0081          ;
0CEA  F1          0082 NEWLIN   POP  AF              ;If new-line POP data
0CEB  18 00       0083          JR   AOUT            ;and print it.
0CED              0084          ;
0CED              0085          ;
0CED              0086          ;
0CED              0087          ;** CHARACTER OUTPUT ROUTINE **
0CED              0088          ;
0CED  F5          0089 AOUT     PUSH AF              ;Save data
0CEE              0090          ;
0CEE  D3 05       0091          OUT  (5),A           ;Output data to PIO
0CF0              0092          ;
0CF0  DB 04       0093 LOOP1    IN   A,(4)           ;Input and check
0CF2  CB 47       0094          BIT  0,A             ;BUSY signal
0CF4  20 FA       0095          JR   NZ,LOOP1        ;If high check again
0CF6              0096          ;
0CF6  3E 00       0097          LD   A,0             ;Else take
0CF8  D3 04       0098          OUT  (4),A           ;STROBE low
0CFA  3E 02       0099          LD   A,2             ;and then
0CFC  D3 04       0100          OUT  (4),A           ;high again
0CFE              0101          ;
0CFE  F1          0102          POP  AF              ;retrieve data
0CFF  C9          0103          RET                  ;and return.
0000              0104          ;
0000              0105          ;
0000              0106          ;
0000              0107          ;
0000              0108          ;** PUT CHARACTER DATA HERE **
0000              0109          ;
0000              0110          ;
0000              0111          END
```

Listing 1



'The game's up sport! Your user group dobbed on you.'

WRITING THE SMALL PRINT

Andrew May shows how to produce tiny print on a Centronics 739.

The micro-print program was developed as an experiment with the pin-addressable graphics of the Centronics 739 printer. The program will run on either a Nascom 1 or Nascom 2 computer fitted with a Z80 PIO and either NAS-SYS 1 or NAS-SYS 3 monitor. Since the 739 printer does not have programmable line-feeds there is no way of changing the standard line-spacing of six lines per inch. The program does, however, give the printer the following facilities: printing using a 5x5 character matrix to give 100 (actually 99) characters per line at 12 lines per inch. Printing is at 62 characters per second and lower-case letters have true descenders.

The 5x5 matrix is smaller than most popular printers but the text is surprisingly readable, particularly when printing in upper-case. The program makes use of the fact that when printing in graphics mode the printer uses only the top six pins of the print-head and prints two lines for each line of text. By using the graphics to form a character using only these six pins it is possible to effectively halve the line spacing of printed text. The sixth pin is unused except when printing lower-case descenders; this gives a gap between each line of text since the graphics are designed to run into each other vertically.

The initialisation part of the program puts the address of the main print routine into the user output vector in the monitor workspace — thus, anything printed on the computer screen also appears on the printer. The program then initializes the PIO to drive the printer and sends the ESC,%,0 code to to put it into graphics mode. The system remains initialised until the computer is reset or the printer turned off.

```
0CA5 72 6F 2D 70 0032         DEFM "ro-p"  ;Signal that
0CA9 72 69 6E 74 0033         DEFM "rint"  ;printer is
0CAD 20 72 6F 75 0034         DEFM " rou"  ;ready to
0CB1 74 69 6E 65 0035         DEFM "tine"  ;receive data
0CB5 20 61 63 74 0036         DEFM " act"  ;
0CB9 69 76 65 2E 0037         DEFM "ive."  ;
0CBD 0D 00       0038         DEFM £0D,£0  ;
0CBF             0039         ;
0CBF DF 5B       0040         SCAL MRET     ;Return to NAS-SYS
0CC1             0041         ;
0CC1             0042         ;
0CC1             0043         ;
0CC1             0044         ;** MAIN PRINT ROUTINE **
0CC1             0045         ;
0CC1 F5          0046 PRINT   PUSH AF       ;Save data
0CC2             0047         ;
```

Fig 1 Sample of Micro printing.

```
0C80              0000         ;**************************
0C80              0001         ;*      MICRO-PRINTING    *
0C80              0002         ;*    ON CENTRONICS 739   *
0C80              0003         ;*         PRINTER        *
0C80              0004         ;*                        *
0C80              0005         ;* ANDREW MAY    7/9/1982 *
0C80              0006         ;**************************
0C80              0007         ;
0C80              0008         ;
                  0009         ORG  0C80
0C80              0010         ;
0C80              0011         ;
0C80              0012         ;** INITIALIZE PRINTER FOR MICRO-TEXT **
0C80              0013         ;
0C80 21 C1 0C     0014 INTLZE  LD   HL,PRINT      ;Set up print
0C83 22 78 0C     0015         LD   (0C78),HL     ;routine vector
0C86              0016         ;
0C86 3E 0F        0017         LD   A,0F          ;
0C88 D3 07        0018         OUT  (7),A         ;Initialize PIO
0C8A 3E FF        0019         LD   A,0FF         ;as CENTRONICS
0C8C D3 06        0020         OUT  (6),A         ;printer
0C8E 3E 01        0021         LD   A,1           ;interface.
0C90 D3 06        0022         OUT  (6),A         ;
0C92              0023         ;
0C92 3E 1B        0024         LD   A,27          ;
0C94 CD ED 0C     0025         CALL AOUT          ;Put the printer
0C97 3E 25        0026         LD   A,"%"         ;into pin-addressable
0C99 CD ED 0C     0027         CALL AOUT          ;graphics mode.
0C9C 3E 30        0028         LD   A,"0"         ;
0C9E CD ED 0C     0029         CALL AOUT          ;
0CA1              0030         ;
0CA1 EF 4D 69 63  0031         PRS  "Mic"         ;
0CA5 72 6F 2D 70  0032         DEFM "ro-p"        ;Signal that
0CA9 72 69 6E 74  0033         DEFM "rint"        ;printer is
0CAD 20 72 6F 75  0034         DEFM " rou"        ;ready to
0CB1 74 69 6E 65  0035         DEFM "tine"        ;receive data
0CB5 20 61 63 74  0036         DEFM " act"        ;
0CB9 69 76 65 2E  0037         DEFM "ive."        ;
0CBD 0D 00        0038         DEFM £0D,£0        ;
0CBF              0039         ;
0CBF DF 5B        0040         SCAL MRET          ;Return to NAS-SYS
0CC1              0041         ;
0CC1              0042         ;
0CC1              0043         ;
```

```
0D00   00 00 00 00 00 ;
0D05   00 00 17 00 00 ; !
0D0A   00 03 00 03 00 ; "
0D0F   1E 15 15 11 12 ; £
0D14   12 15 1F 15 09 ; $
0D19   13 0B 04 1A 19 ; %
0D1E   0A 15 15 0A 14 ; &
0D23   00 02 01 00 00 ; '
0D28   00 0E 11 00 00 ; (
0D2D   00 00 11 0E 00 ; )
0D32   11 0A 1F 0A 11 ; *
0D37   04 04 1F 04 04 ; +
0D3C   00 10 08 00 00 ; ,
0D41   04 04 04 04 04 ; -
0D46   00 00 10 00 00 ; .
0D4B   10 08 04 02 01 ; /
0D50   0E 19 15 13 0E ; 0
0D55   00 12 1F 10 00 ; 1
0D5A   19 15 15 15 12 ; 2
0D5F   11 15 15 15 0A ; 3
0D64   08 0C 0A 1F 08 ; 4
0D69   17 15 15 15 09 ; 5
0D6E   0C 16 15 15 08 ; 6
0D73   01 01 19 05 03 ; 7
0D78   0A 15 15 15 0A ; 8
0D7D   12 15 15 0D 06 ; 9
0D82   00 00 0A 00 00 ; :
0D87   00 08 05 00 00 ; ;
0D8C   04 0A 0A 11 11 ; <
0D91   0A 0A 0A 0A 0A ; =
0D96   11 11 0A 0A 04 ; >
0D9B   02 01 15 05 02 ; ?
0DA0   0E 11 15 09 0E ; @
0DA5   1E 05 05 05 1E ; A
0DAA   1F 15 15 15 0A ; B
0DAF   0E 11 11 11 11 ; C
0DB4   1F 11 11 11 0E ; D
0DB9   1F 15 15 11 11 ; E
0DBE   1F 05 05 01 01 ; F
0DC3   0E 11 11 15 0D ; G
0DC8   1F 04 04 04 1F ; H
0DCD   00 11 1F 11 00 ; I
0DD2   08 11 0F 01 00 ; J
0DD7   1F 04 0A 11 00 ; K
0DDC   1F 10 10 10 10 ; L
0DE1   1F 02 04 02 1F ; M
0DE6   1F 02 04 08 1F ; N
0DEB   0E 11 11 11 0E ; O
0DF0   1F 05 05 05 02 ; P
0DF5   0E 11 15 19 1E ; Q
0DFA   1F 05 05 0D 12 ; R
0DFF   12 15 15 15 09 ; S
0E04   01 01 1F 01 01 ; T
0E09   0F 10 10 10 0F ; U
0E0E   03 0C 10 0C 03 ; V
0E13   1F 08 04 08 1F ; W
0E18   11 0A 04 0A 11 ; X
0E1D   03 04 1C 04 03 ; Y
0E22   11 19 15 13 11 ; Z
0E27   00 1F 11 11 00 ; [
0E2C   01 02 04 08 10 ; \
0E31   00 11 11 1F 00 ; ]
0E36   04 02 1F 02 04 ; ^
0E3B   10 10 10 10 10 ; _
0E40   00 00 01 02 00 ; `
0E45   1A 1A 1A 1A 1C ; a
0E4A   1F 14 14 14 08 ; b
0E4F   0C 12 12 12 12 ; c
0E54   08 14 14 14 1F ; d
0E59   0C 16 16 16 16 ; e
0E5E   00 04 1E 05 00 ; f
0E63   24 2A 2A 2A 1E ; g
0E68   1F 04 04 04 18 ; h
0E6D   00 00 1D 10 00 ; i
0E72   00 20 1D 00 00 ; j
```

VECTOR 4

VECTOR 4 SPECIFICATIONS

Central Processing Unit:

Processors:	8-bit Z-80B and 16-bit 8088 (single or multiprocessor operation)
Clock Speed:	5.1 MHz
Memory:	128K Dynamic RAM Standard Expandable to 256K

Video Display:

Screen:	12 inch high contrast green phosphor 20 KHz Horizontal, 60 Hz Vertical
Alphanumeric:	24 Lines x 80 characters High resolution 16 x 13 dot matrix
High Resolution Graphics:	640h x 312v pixels (B/W)
Gray Scale Graphics:	160h x 312v pixels, 16 levels of gray 320h x 312v pixels, 4 levels of gray
Color Graphics:	External RGB Monitor 160h x 312v pixels, 8 colors 320h x 312v pixels, 4 of 8 colors

Keyboard:

Detached, with 8035 auxiliary processor. Capacitance keyswitch with 91 keys, including 15 programmable special function keys, cursor control keys, and 10-key numeric pad for rapid data entry.
Coiled cable with Interface

Input/Output:

Serial Keyboard
Centronics Parallel Interface
Qume/NEC Parallel Interface
RS-232 Serial Printer Interface
RS-232 Communications Interface
RGB color signals
Programmable Tone Generators and Speaker
S-100 expansion slots

Disk Drives:

Type of Disk:	5¼" Floppy	5¼" Winchester
Total Capacity:	630 Kilobytes	5 Megabytes
Transfer Rate:	250 Kilobytes/sec	5 Megabytes/sec
Rotation Speed:	300 RPM	3600 RPM

Configurations:

Model 4/20	Two floppy drives
Model 4/30	One floppy drive and one Winchester hard disk drive

I'd like to know more about Vector computers

☐ Please send me additional information

☐ Please tell me where I can get a demonstration

Name ....

Title ....

Company ....

Address ....

Nature of Business ....

Planned Applications ....

Mail to **Fiona Dicker, Dicker Data Projects Pty. Ltd.**
78 Captain Cook Drive, Caringbah, N.S.W. 2229.

$5295

VECTOR

THE COMPANY COMPUTER.

DEALER ENQUIRIES INVITED

Dicker Data Projects Pty. Limited, 78 Captain Cook Drive, Caringbah, N.S.W. 2229. (02) 525 2122, 525 4707

Hewlett-Packard put it all together.

SPECIAL OFFER

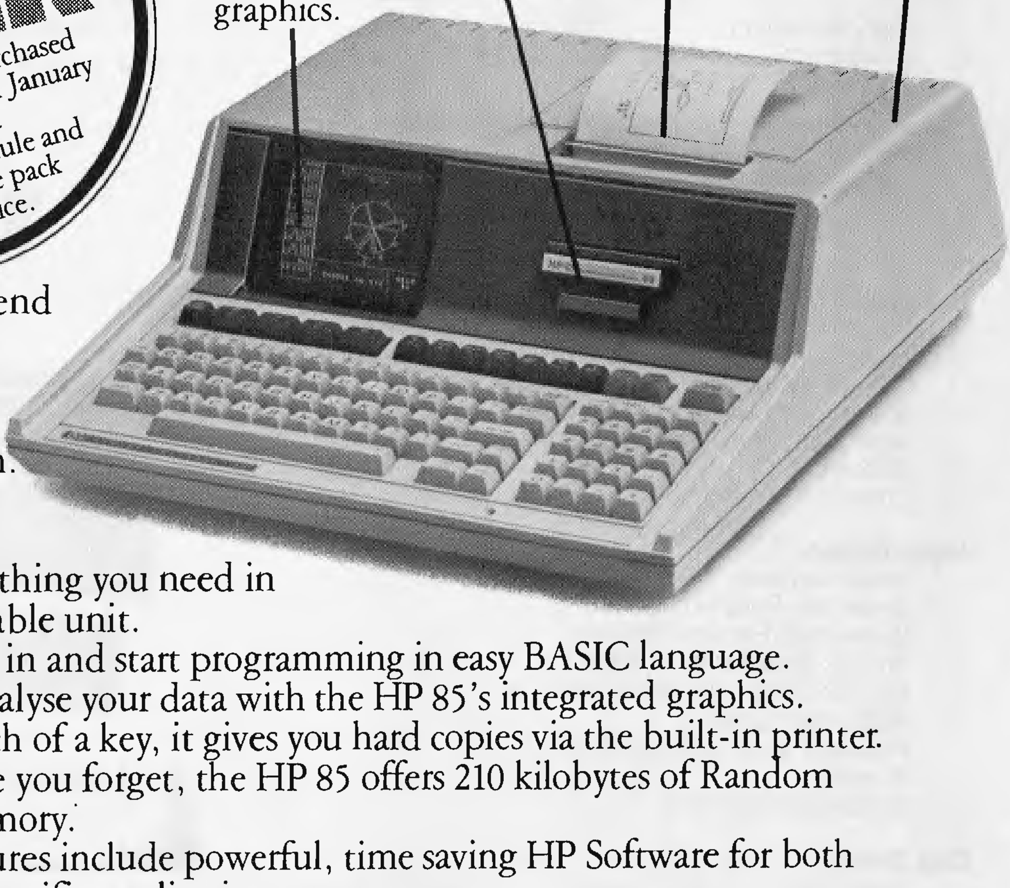
With each HP-85 purchased during December and January receive free a 16K memory module and HP-85 Software pack of your choice.

2. 210 kb data storage with Random Access Tape.

3. Printer.

4. Portable (15 kg).

1. Integrated graphics.



You can spend a lot of time and money building up your own computer system.

Or you can buy the HP 85.

It has everything you need in a compact, portable unit.

Just plug it in and start programming in easy BASIC language.

You can analyse your data with the HP 85's integrated graphics.

At the touch of a key, it gives you hard copies via the built-in printer.

And in case you forget, the HP 85 offers 210 kilobytes of Random Access Tape Memory.

Other features include powerful, time saving HP Software for both business and scientific applications.

There's also a degree of friendliness in the HP 85, unrivalled in a computer of its size. Like helpful error messages and automatic syntax checking.

Of course you may eventually wish to expand your HP 85 system. An HP interface lets you plug in up to 14 peripherals without disassembly.

To locate your nearest Hewlett-Packard dealer, call Hewlett-Packard. Sydney: 887 1611; Melbourne 890 6351; Adelaide: 272 5911; Perth: 383 2188; Brisbane: 369 8022; Canberra: 80 4244.

HEWLETT PACKARD

HP91

BENCHTEST
SOFTWARE

CARDBOX

Kathy Lang gets to grips with an electronic card index system.

This month I'm reviewing a CP/M based package which is aimed, not at general purpose data management problems, but at one particular, very familiar, tried and trusted card index stored on a micro, with most of the features you have always wanted on your manual card index but couldn't have, because of the limitations of pieces of card.

The package allows the user to create 'card' formats on the screen, put data into the card index from the keyboard or from other files, have the information indexed on any word specified, and display all or part of the data on the screen or printer. It doesn't provide any facilities for doing calculations on the data, either within the records or by aggregating across records, except that it displays the numbers of records found when a particular search is made. So Cardbox is clearly aimed at textual applications — bibliographies, descriptive records such as medical case histories, and so on, rather than at areas where the manipulation of numbers is an essential part of data management.

Constraints

The orientation of the package is reflected in its limitations. Cardbox allows you quite generous limits on the numbers of records in a file (65,500) and on the number of characters in a record (1404), with the usual CP/M limitation of 8 megabytes on total file size, but the package imposes tighter constraints on the number of fields in a record (maximum 26). Although the layout of a single 'card' or record must be described on one display screen, fields may run over more than one line — and indeed the record could consist of just one very large field if you wanted. Any word or phrase can be indexed, but each indexed item may not exceed 32 characters. Cardbox can deal only with one file of information at a time, so you can't link records across files.

Input and updating

To put information into a Cardbox file, you must first design a screen format for the data. This format is used whenever changes are made to the file. When using the search facilities you can display records using either the initial format or other formats which can be created for particular purposes; providing 'overlay' formats to display the information in different ways. Formats both describe the way the data is displayed and define the length of fields for filing the records away. Cardbox provides a screen editor for use in designing screen formats, with cursor movement using the control keys following the Wordstar pattern — CTRL-E to move up a line, CTRL-S to move left one character, and so on. It would be better still to be able to use the cursor arrow keys — but if you can't then to be able to use a well-known convention is very helpful. This editor has, in addition to the conventional facilities for putting in characters at the cursor position, the ability to 'follow' the cursor round the screen, trailing a line character behind, so you can draw lines round the boxes just by moving the cursor down or across the screen where you want the line to come. The screen format I used for the Benchtest is shown in Figure 1.

On the figure, the plus signs are the 'draw' characters as they came out on my printer. The items AAAAA, BBB ... etc, are the data fields. Since Cardbox doesn't have any calculating facilities, it doesn't need to distinguish different data types but it *does* need to know the maximum length of each field. To describe a field, you give it a caption, which is a field label, of up to 16 characters, an abbreviated field name of two letters which you use to describe the field when editing records or searching them, and a single character identifier which is used when 'placing' the field on the screen during the formatting operation. This is the only time the single character description is seen by the user, which is just as well as I found the use of one character for a field when laying out the format, and two characters at other times, very confusing as both are displayed on the screen while you are adjusting screen formats. So in my example the variable System3 has the identifying letter P and occupies three positions at the point on the screen which follows its caption. None of my fields spread over more than one line, but if one does, you just mark the beginning and end of the field, and Cardbox treats those two points as diagonally opposite corners of a box which is to contain the field. So if System3 had needed two 'words' up to three characters long in the same position, Cardbox would have shown that as

PPP
PPP

and I could still have had the field called Rating exactly in the position shown on the figure.

When you set up the screen format, you also say how you want the field indexed. There are four possibilities: the

```
 CARDBOX(F)   File =  PCWBTBOX.FMT       EDIT SCREEN                   PRINT

+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ REFNUM AAAAA      + NAME BBBBBBBBBBBBBBBBBBBB                           +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Date Cre CCCCCC     Type DDD                                            +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Description EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE       +             +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Supplierl FFFFF   + Pricel GGGGG      + Supplier2 HHHHH  + Price2 IIIII +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Supplier3 JJJJJ   + Price3 KKKKKK     + Supplier4 LLLLL  + Price4 MMMMMM +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Systeml NNN       + System2 OOO       + System3 PPP                     +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Rating QQ                                                               +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+ Refl RRRRRR       + Ref2 SSSSSS       + Refl TTTTTT      + Ref4 UUUUUU  +
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

CURSOR:  ^S=left  ^D=right  ^E=up  ^X=down                     ROW=03 COL=01
EDIT:  ^V=ins col  ^G=del col  ^N=ins row  ^Y=del row
       ^W=graphics  ^Z=draw ON/OFF  ^P=print  ESC=exit
```

Fig 1

```
 CARDBOX(U)    File = A:BOOKS.FIL     PRINT
Level  0 - RECORD 1 OF 434
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Author: Mark Girouard                                                 +
  +                                                                      +
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Illustrator:                                      +Language:          +
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Title: Life in the English Country House                              +
  +                                                                      +
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Publisher: Yale U.P.                              +Date: 1978         +
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Subjects: Society, architecture, history, England, country-houses.    +
  +                                                                      +
  +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


 Enter command: SELECT SU/
   Enter the word to be found.     (hit RETURN at end)
    "?" will match any letter, "+" any sequence of letters.
 LIST:  ^R=lst ^C=last ^A=back ^F=fwd   ENTRY:  ^X=erase ^H=backspace
```

Fig 2

words in a field may always be indexed, or never indexed, or indexed unless the user overturns this decision when the data is input, or indexed only if the user asks for it when the data is input. For either of the optional indexing possibilities, you can choose whether to index or 'unindex' all the words in a field or only some of them. You can also index phrases formed by hyphens — these are indexed both as phrases and under the words which form the component parts of the phrase. All these operations are confirmed when putting the data in with the use of the TAB key, and if you have the right kind of terminal then indexed words and phrases are highlighted on the screen. Cardbox will not allow you to put data in a field beyond the confines which the format allows. If you have records which contain many duplicate fields, you can take a copy of the current record and edit only those fields which have changed, then store it as another record.

In most data management systems, once you have defined the record structure it is hard to change it, and this usually involves copying the whole file out to a new structure. With Cardbox it's very easy; you simply change the main screen format for the data file and Cardbox will show blanks as the value of that field for existing records until you put some data into them. However, if you decide that, while you don't want to change the record structure, you *do* need to index a previously unindexed field, this is much harder, and involves copying the file out in an external format, inserting an index flag before the desired field in each record, and copying the data back into Cardbox. The obvious way round this is to index everything from the start, but this might be expected to slow data entry down; I didn't try it. The same process of flagging index fields and copying the data file into Cardbox can be used to translate any ASCII sequential file into Cardbox format, so it's possible to 'port' data from other programs.

Editing records is achieved by retrieving them using the selection mechanism described later and either modifying or deleting them. During amendment, the cursor can be moved around the record on the screen using control functions much as in the process of formatting the screen, using Wordstar-like control characters.

Displaying data

Records in any file can be displayed using either the screen format employed for data input, or one of any number of different formats set up for particular purposes. These formats do not have to refer to every field in the record; if a field identifier is not used then Cardbox simply ignores the field for display purposes. So, if records consist of a mixture of confidential and 'open' information, then different overlays can be designed so that people with different roles can access only the parts of the record they need for their particular purpose.

Printed reports

Exactly the same mechanism is used for designing printed reports and for screen displays. This has the advantage that you only need one format to get going, and the disadvantage that even if your printer will print more than 80 columns wide, you can't exploit that. (On reflection, as a founder member of the 'A4 is quite big enough for human readers and briefcases' club, maybe that isn't a disadvantage after all.) In addition to the PRINT command, which prints out all the records in the current selection (see below), using a specified format, you can also take a quick copy of the screen at any time with CTRL-P. This could be a very useful feature, not just for data but for taking copies of screen formats and of instruction displays for teaching — I produced all the figures for this article in this way.

```
 CARDBOX(U)    File = A:BOOKS.FIL      PRINT
Level  1 - RECORD 1 OF 25
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Author: Sacheverell Sitwell                                           +
  +                                                                      +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Illustrator:                                     +Language:           +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Title: For Want of the Golden City                                    +
  +                                                                      +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Publisher: Thames & Hudson                       +Date: 1973          +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
  +Subjects: Essays, travel                                              +
  +                                                                      +
  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


Enter command:
  MAsk; SElect, INclude, EXclude; HIstory, BAck, CLear; LIstindex;
 ADd, DUplicate, EDit, DElete;  REad, WRite; FOrmat, PRint; SAve, QUit
LIST:  ^R=1st ^C=last ^A=back ^F=fwd   ENTRY:  ^X=erase ^H=backspace
```

Fig 3

```
   CARDBOX(U)     File = A:BOOKS.FIL      PRINT
  Now at level  3.  File contains 434 records.

  Level  1 - SELECT SU/ESSAYS - 25 RECORDS SELECTED
  Level  2 - EXCLUDE SU/TRAVEL - 20 RECORDS SELECTED
  Level  3 - SELECT AU/S+ - 3 RECORDS SELECTED
```

Fig 4

Selection

Cardbox has an exceptionally powerful and flexible set of selection facilities, always remembering that we're dealing with textual data. There are two groups of selection commands, one for use on indexed fields and the other which can operate on any field. Selection is by identity only, (with case ignored) but you can use wild characters: '?' to match a single character, '+' to match one or more characters. For instance, if you ask for all records in which a particular field matched the word 'PART', Cardbox would select all those records where the specified field matched the word 'PART' or 'part' (or, for that matter, 'PaRt'). A request to match on 'PART?' would match any five-character word starting with 'PART' or 'part', such as 'parts', 'party', and so on, while a request to match 'PART+' would match all those found by 'PART?' plus words like 'partition', 'particular' and so on. Wild characters can of course be put in the middle of words too, so a request to match 'P??TION' would match words like 'portion', while 'P+TION' would match 'portion' but also 'potion', 'partition', 'parturition', 'petition' . . .

When you request matching of a field, you name the field with the two-character abbreviation specified in the screen format, not the full caption — but Cardbox helps you to remember what fields are called by displaying all the abbreviated field names at the bottom of the screen. It also gives you a reminder of what commands are available at any one time and prompts you for further input when a command is given. Figure 2 shows the screen after specifying the data file to use (a file of bibliographic information which is supplied with Cardbox and is used in the tutorial manual) and requesting the field Subject to be used for a selection.

The next step is to enter the key fields for searching. In my example I chose the subject 'essays'. Cardbox found 25 records that matched the specification and displayed the first on the screen as Figure 3 shows. The main point to note about this figure is that the level number has changed from 0 to 1. Cardbox uses the term 'level' to indicate the depth of search to which the selection has gone, meaning the number of selection, inclusion and exclusion commands that have been given. Selection simply chooses all the records in the current set (available at the current level, in Cardbox terms) which match the field value specified. This process can be further modified in two ways. The INCLUDE command allows you to extract a further set of records from the whole file — so it's really a way of having an OR function at the zero level. I might, for instance, have wanted to select titles covering either travel or biography, and I could have asked Cardbox to SELECT travel and INCLUDE biography. The third command for choosing subsets is EXCLUDE, which removes from the current set any records with a field value matching the specified value. So SELECT and EXCLUDE are the inverse of each other and operate on the current set or level, and INCLUDE always operates at the zero level of all the records in the file.

The possibilities for selection are quite wide, because Cardbox allows you up to 99 levels of selection. This can get pretty complex, and thus the HISTORY command, which spells out how you got to the current level, is an extremely valuable option. I've shown in Figure 4 the history of a search which went on from the SELECT shown before through

The Best for Less

ADE 10TH YEAR OF COMPUTER COUNSELLING

COMPUTERS

NORTHSTAR PERSONAL DESKTOP

Advantage 2Q-64K $4758
Advantage 5M-64K $7852
Horizon 2Q-64K $4495
Horizon 5M-64K $7138
Horizon 18M-64K $9518

•

A.D.E. LSI-11 SYSTEMS

11 D.E.C.-compatible minicomputers based on the LSI11-02, LSI11-23, or Dataram M23. Choice of Winchester, floppy or S.M.D. .... $ call

•

D.E.C. PERSONAL COMPUTER OPTION

Add to your VT100, or available complete with VT100.
VT18X-CC CP/M option . $1712

•

We also have a wide range of disk and diskette drives and sub-systems available.

TERMINALS

TELEVIDEO

910 .... $ 895
925 .... $1495
950 .... $1895
Intelligent One $2350
Graphics Option $1595

•

ADE100 .... $1295

•

D.E.C VT100AB $2227
D.E.C. VT101A5 .... $1750
D.E.C. VT131A5 .... $2500
T640 Graphics option .. $2318

•

Teleray 10N $1695
Teleray 10M $1695

•

D.G. D200 .... $2684
D.G. D400 .... $3136
D.G. D450 .... $3824
D.G. G300 .... $4905

•

FULL TECHNICAL BROCHURES AVAILABLE ON REQUEST

* *Ask now for your new A.D.E. catalogue.*
* *Prices shown do not include sales tax.*
* *Full maintenance and extended warranty available.*

[redacted]

* *Quantity discounts available.*

DEALER ENQUIRIES WELCOME

PRINTERS

OKI ELECTRIC

ML80 .... $ 525
ML82A $1099
ML83A $1395
ML84 .... $1795
2350 . .350 c.p.s. $2995

•

QUME

S9/45 ROLP $3120
S9/45ROFPME $3660
S9/35 KSR $1995

•

D.E.C.

LA12C $2595
LA12D $2245
LA34AA $1345
LA34RA $1295
LA34DA $1095
LA100RA $2536
LA100ZA $3080
LA100AB $3275
LA120GA $3149
LA120RA $2771
LA50RB .... $ 996

•

T.I.

743KSR $1500
743RO $1250
745KSR $2100
810 .... $1890
820KSR $2650
820RO $2450

•

Enquire about our range of Plotters and Digitisers.

ADE

P.O. BOX 422, CLAYTON, VICTORIA 3168, AUSTRALIA. PHONE: (03) 544 3444. P.O. BOX 341, PENNANT HILLS, NSW 2120, AUSTRALIA. PHONE: (02) 848 8533. Adelaide: 46 4062. Perth: 387 6055. Hobart: 34 4522. Canberra: 58 1811. Brisbane: 352 5788. Newcastle: 69 4325. Wellington: 69 3008. Auckland: 54 4224. Christchurch: 4 3001. Darwin: 81 2712. Singapore: (65) 449 4433. Bendigo: (054) 42 4322. Wangaratta: (057) 21 6443.

an EXCLUDE and then another SELECT.

Selection is flexible in some ways and restricting in others; the most noticeable restriction (given that identity is the only comparison possible, because the data is expected to be textual) is that there is no real OR mechanism below the first (zero) level. Having requested travel books, it isn't easy to request all those about Greece and Italy, say without writing out a subset of records to another file and then reading them back. Of course, you could get round that particular example by SELECTing Greece, INCLUDing Italy, and then SELECTing travel — but you have to think about the order and there would be some combinations in which only creating a subset file would give the desired results.

The restriction of matching by identity (albeit with wild codes) might be limiting where an application involved searching date fields, when it can be useful to be able to say 'find all the records where "date of last appointment" is more than six months ago.'

Sorting

Cardbox doesn't have any facilities for sorting, and it keeps the data in an arbitrary order, retrieving it in the order in which it is filed. So if you wanted your books (say) listed in author order, you would have to use another program, such as Supersort, to do it.

Tailoring

Cardbox can be preset for one of quite a number of commonly available terminals. If yours is not one of these, it can still be adapted fairly straightforwardly, though I did have some problems with items to which I did not want to give a value. You can't then tell Cardbox that your terminal has arrow keys for moving the cursor about, or give it information about the terminal's highlighting capability. Apart from adapting it to match your terminal, Cardbox doesn't have any tailoring facilities. The feature that I missed in this area was the lack of any ability to store sets of search and select commands in a file. Often one wants to do quite a complex search regularly as data is updated, and it is tedious to have to retype the instructions every time.

Stability and reliability

Cardbox is the first system I've come across with explicit facilities for repairing a set of data which has become corrupted — either through a system glitch such as power surge, or through mishandling. I didn't need to use these, I'm glad to say, but I suspect they could be very valuable in a real-life application over a period.

Housekeeping

Within Cardbox you can copy and erase files, so the only operating system functions required are disk formatting and directory listing information. Cardbox also provides some analysis facilities to give the user aggregate information about the data, to help in tasks such a file sizing.

User image

Cardbox is a menu-driven package, using two ways of specifying what function to carry out next. At the start, the screen displays a set of options as shown in Figure 5.

To get different options, the arrow pointing to the lines of instructions is moved until the correct pair is identified. Once into the data, permitted commands are shown on the bottom of the screen as shown in Figures 1-3, and two-letter abbreviations (which are filled out by Cardbox) used to give the necessary instructions. I found this approach very helpful, especially the display of what commands were permitted at any one time. My reservations were limited to the facilities for setting up screen formats, which have some good features and some unfortunate drawbacks.

The good features include the use of the cursor to move around the screen, defining starts and ends of fields and elements such as lines around 'boxes' of information simply by 'pointing' through pressing a single key when the cursor is in the desired position. Cardbox also tells you what row and column you're on, so you don't have to do anything primitive like counting the number of times you've pressed the cursor key to get a field the right length.

One drawback of form design I've already mentioned — the confusing use of one-letter symbols for fields whose names are abbreviated to two letters, so each field has three 'names' during format definition. I should have preferred the use of a single field definition character for all fields — the use of a caption makes it quite clear which field is intended. Another silly irritation is that the command to insert a character in a position on the screen actually inserts a column — ie, an extra space in every row on the screen — so to juggle around with the spacing on one row after you've set up the rows below involves adjusting them, too.

But these are minor irritations which do little to detract from a generally well-designed package. The accompanying documentation is also written to a high standard. There is an introductory manual for complete beginners and a reference manual which is actually two documents in one — I suspect that, as they say, the last came first since it is in the usual computing mode with numbered sub-sub-sections and so on. The documenters have done an extremely good job in extracting from it the introductory and main reference manuals, which contain all you need to know in a digestible format. I only found one real howler — both manual and screen tell you that functions marked with a star are not available when displayed, then tell you to go ahead and invoke one that happens to be starred! It only occurs once, though, and otherwise the instructions are clear and well-laid-out.

I do wish, though, that people did not feel it essential to put at the front of the tutorial manual the information about configuring the terminal. Nearly every package manual does it, and it must be very confusing for the first-time user. Even in a package which is sold by mail order, it should be possible to say something like 'If this is the first time Package Z has been used on your system, it must be set up for the purpose — see page xxx in the Reference Manual if you need to do this', to avoid confronting the absolute beginner with hex codes unless they really need them. I expect this aspect of the manuals grated on me more even than usual with Cardbox because the documentation is otherwise so well thought out.

Conclusions

Cardbox is a package designed for a particular purpose, namely the automation of card indexes. Within that aim, it succeeds extremely well. Its facilities for indexing and searching are good and very fast (from my initial tests the fastest of any package I've evaluated so far) and the user image of the screen displays and the documentation are in the main excellent. Users I've spoken to particularly liked the ability to index individual words in prose text, the flexibility of the screen design, and the power of the selection facilities. Set against that are the drawbacks: Cardbox has no sorting facilities of its own, cannot store searches for subsequent re-use and does not let you change an unindexed field into an indexed field easily — and of course it has no numerical facilities at all, nor the ability to test for ranges, in date fields for instance. If these limitations aren't a serious drawback, then for bibliographic applications, records (such as medical case histories) containing continuous prose which need part indexing, and the like, at the asking price of $298 Cardbox would be a very good buy. END

```
 CARDBOX                                         +
                                                 +
PRIMARY FUNCTIONS:                               +  SECONDARY FUNCTIONS
                                                 +
  ==>  Database                                  +    ==>  Use
       Format definition                         +         Analyse
       Operating system utilities                +         Create
                                                 +         Repair
                                                 +
                                                 +
                                                 +
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

PRIMARY-FUNCTION = [DATABASE]
SECONDARY-FUNCTION = [USE]

FILE =*
CHANGE-DISKS = [NO]

Hit letter for option (P,S,F,C) or hit ESC:
                    (options marked "*" are invalid)
```

Fig 5

MICRO EXCHANGE

Our new micro 'trading post' will be published monthly in Direct Access. All Micro Exchange ads must be submitted by readers on the appropriate form (or a photocopy). Maximum of 30 words. Print one word per box, very clearly. Contact name and telephone number/s must be included in the 30 words. All ads must be accompanied by a fee of $5.00 for Australian Personal Computer or $7.50 for inclusion in both Australian Personal Computer and Australian Business Computer. Make cheques or Postal Orders payable to Micro Exchange. Ads cannot be repeated (unless sent on another form) and we cannot guarantee to print an ad in any specific issue. Please help the typesetter by printing very clearly. Send your form to: Micro Exchange, P.O. Box 62, Middle Brighton, Vic 3186.

VICTORIA

TRS 80 BUSINESS COMPUTER SYSTEM, DUAL DISC DRIVE, 48K RAM, COMPLETE RANGE OF SOFTWARE, GREEN SCREEN AND SYSTEMS DISC. MAINTENANCE AVAILABLE $200
AH:(03) 578 3445 BH:(03) 328 2988

OSBORNE I, 64K MICRO, NEW IN BOX, COMES WITH GUARANTEE AND ALL ASSOCIATED SOFTWARE, NEVER BEEN USED. MAINTENANCE AVAILABLE $1990
AH:(03) 578 3445 BH:(03) 328 2988

SHARP P.C. 1211 POCKET COMPUTER, CASSETTE INTERFACE, INSTR BASIC AND 300 PAGE APPLICATION MANUALS. EXCELLENT CONDITION $150
(03) 890 6456

COMPUCOLOUR II COLOUR COMPUTER, 32K COMES WITH DISC DRIVE AND COLOUR MONITOR COMES WITH MANY DISCS
$1,850 O.N.O. (03) 890 6456

COMPUTER TELETYPE ASR 38, VERY GOOD CONDITION, COMES WITH KEYBOARD, PRINTER, TAPE READER AND PUNCH, RS232 INTERFACE, ALSO FULL SET OF MANUALS $700
(03) 818 5711

SHARP P.C. 1211 WITH PRINTER CASSETTE INTERFACE AND CARRY CASE, AS NEW IN BOX $295 (03) 878 8895 MR PECK

COMPUTER PRINTER TELETYPE ASR35 WITH FULL TAPE FACILITIES AND ACCESS $140
(03) 877 3409 MR CAMP

SINCLAIR Z X 80 MICRO PROCESSOR, 16K RAM, 8K ROM, MANUALS ETC $290
(03) 417 2774 MR STEWART

DICK SMITH SYSTEM 80 MK.II NEW SCAP BUSINESS COMPUTER SERIAL NO 810311932 VIDEO MONITOR & EXPANSION UNIT, TOTAL OF 32K OF RAM MEMORY, 2 X 40 TRCK FLOPPY DISC DRIVE WITH DAISY CHAIN CABLE AND POWER SUPPLY, 1 X 132 COLUM PRINTER WITH CABLE, IN FULL OPERATING ORDER $2,500
(03) 376 0728 MR CAMERON

COMPUCOLOUR 2, BRILLIANT COLOUR, BASIC/FORTRAN, 36K, 5 INCH DISC DRIVE, MODE PROGRAMMING DISCS AND MANUALS AS NEW $1,600 (03) 469 3130 MR BURNS

OSBORNE COMPUTER, 64K, 2 SINGLE SIDED/SINGLE DENSITY DRIVES, COMES WITH CBASIC/MBASIC/WORDSTAR/MAILMERGE AND SUPERCALC. ALL MANUALS PROVIDED $1,850
AH:(03) 419 4071 BH:(03) 419 7528
MR BURN

COMPUTER, TELETYPE MODEL 33 ASR, COMES WITH PAPER TAPE PUNCH AND READER, RS232 INTERFACE. GOOD CONDITION $285
(03) 792 9301

MICRO COMPUTER SYSTEM 2650 X CENTRAL DATA, IDEAL SET UP FOR ENTHUSIAST, 16K DYNAMIC MEMORY, BOARD EXPANSION TO 65K, S100 BUS/BOARD STANDARD, 53 KEYBAORD POWER SUPPLY, INCLUDES SOFTWARE, TAPES, ETC. COST $1,200 NOW $400
(03) 789 1230 MR SCHUBERT

16K RAM - $80
3K RAM - $50
PROGRAMME GENERATOR - $60
PROGRAMME SOUND GENERATOR - $60
NATIONAL SINCLAIR ZX USERS GROUP
P.O. BOX 148, GLEN WAVERLEY 3150

TRS 80 MICRO COMPUTER SYSTEM, LEVEL 2 BASIC, 16K MEMORY FULL WARRANTY NEVER USED, COST $890 SELL FOR $700. INCLUDES CASSETTE RECORDER FOR ADDED MEMORY AND PROGRAMME MANUALS.
(03) 337 0825 MR ZAFIRIS

COMPUTER, OLIVETTI PAPER TAPE READER FOR READING 7 OR 9 TRACK PAPER TAPE, OPTICAL SYSTEM COMPATIBLE TO MOST OTHER SYSTEMS $110 (03) 311 6670 MR DEVERS

COMPUTER 2 X 81 4K RAM AND 8K ROM COMES WITH ALL LEADS AND MANUALS ETC EXCELLENT CONDITION $210 (03) 311 6670

2 X SPECTRUM (WEBSTER) PDP11 - 64K SINGLE SIDED/DOUBLE DENSITY DISKETTES $2,500 EACH MAINTENANCE AVAILABLE
(03) 531 8788

TI 99/4 KEYBOARD AND MANUAL $100
(03) 758 7983 MR PINE

APPLE II WITH ONE DISK DRIVE AND BLACK AND WHITE MONITOR AND SOFTWARE AND MX80 FT. $2,850 PETER MONGER
BH:(03) 519 9121 AH:(03) 529 2657

QUANTITY OF NEW PERTEC DRIVES, 40 TRACK SINGLE SIDED 163K CAPACITY $220 EACH. MAINTENANCE AVAILABLE (03) 890 1770

8" PERTEC FLOPPY DISK DRIVES (USED) $200 MAINTENANCE AVAILABLE (03) 890 1770

LA36 DECWRITERS (USED)
$500 MAINTENANCE AVAILABLE (03) 890 1770

SENDATA 700 SERIES ACOUSTIC COUPLER $200 MAINTENANCE AVAILABLE (03) 509 5844

QUEENSLAND

2732 DUAL CATRIDGE CHARACTER SET FOR OSI CONTAINS MEDIUM RESOLUTION GRAPHICS BLOCKS AND ALSO EASIER TO READ PRINT. CHESS PIECES AND GAMMON CHARACTERS ALSO MAINTENANCE AVAILABLE (075) 65 4354
MR BERNIE WILLS

BASIC 4 EPROM FOR OSI C1P CONTAINS HIGH SPEED CASSETTE LOAD/SAVE/VERIFY ROUTINES FOR BASIC AND M/C PROGRAMMES MAINTENANCE AVAILABLE $12.80 (07) 266 6229

SOUND GENERATOR IN CASE FOR OSI, SOUND EFFECTS LIKE SIREN, MOTOR, PHASOR, MACHINE GUN, ORGAN NOTES AND MANY OTHER POSSIBLE EFFECTS. $50 O.N.O.
(07) 266 6229

TRS-80 VOICE SYNTHESISER WITH MANUAL AND DEMO PROGRAMMES. SUITE TANDY MODEL 1, WITH INTERFACE TO TRS MODEL III, SYSTEM-80 OR APPLE $200 (07) 398 8361

SOUTH AUSTRALIA

EPSOM EXTENDER, A SIMPLE WAY TO ADD ON AN EXTRA 8K OF ROM TO YOUR OSI COMPUTER. BARE BOARD ONLY MAINTENANCE AVAILABLE $9.50 (08) 258 8647 MR PATTERSON

NEW SOUTH WALES

DISC CONTROLLER BOARD FOR OSI TASKER BUS BARE BOARD ONLY COST $25 SELL $15
(02) 665 7763 MR READ

ANADEX DP-8000 PRINTER NEAR NEW $650 MAINTENANCE AVAILABLE (02) 570 8344

SANDERS MEDIA 12/7 PRINTER WITH SINGLE SHEET FEEDER $2,900 MAINTENANCE AVAIL
(02) 570 8344

MICRO EXCHANGE

CENTRONICS 739 (NEW) $900
MAINTENANCE AVAILABLE (02) 570 8344

TI-810 PRINTER $2,000 NOW $1,200
(02) 922 3977

IMPACT DATA PRINTER $600
(02) 922 3977

CENTRONICS 737 PRINTER $900
(02) 922 3977

SOROC TERMINAL AS NEW IN BOX $1000
(02) 922 3977

NORTHERN TERRITORY

COMMODORE VIC 20 MICRO COMPUTER
COMMODORE VIC 1540 DISK DRIVE
COMMODORE VIC 1515 GRAPHIC PRINTER
INCLUDING MEMORY EXPANDED TO 32K
$2,034.00 (089) 52 3027

COMMODORE VIC 20 MICRO COMPUTER
COMMODORE DATASETTE
GAMES CARTRIDGE (ANY CHOICE)
PACK OF SIX RECREATIONAL GAMES
$613.00 (089) 52 3027

ONE HEWLETT PACKARD COMPUTER DESKTOP
MINI MODEL 9825 + ROMS. ANY OFFER
(089) 52 3027

ONE HEWLETT PACKARD 9885M DISK DRIVE ANY
OFFER (089) 52 3027

ONE HEWLETT PACKARD 9871 DAISY WHEEL
PRINTER ANY OFFER (089) 52 3027

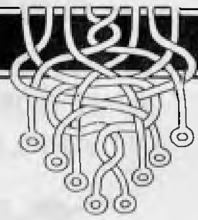
Please find enclosed my cheque/P.O. for $5.00/$7.50 for the following ad.

NAME. . . . .

ADDRESS. . . . .

. . . . .Postcode. . . . .

LAZING AROUND



Quickie

What is special about the number 8,549,176,320?

No answers — no prizes.

Prize puzzle

A difficult puzzle this month, since it's Christmas and everyone will have extra leisure time to work at their micros.

Every odd number, except two, between 1 and 8999 is either a prime or the sum of a prime and twice a square.

Thus:

$321 = 19 + 2.1^2$
$27 = 19 + 2.2^2$

What are the two exceptions?

By the way, remember, *postcards* (or backs of envelopes) only. All other entries go straight into the bin — you have been warned.

September prize puzzle

What a lot of coconuts — and what a lot of entries! Well over 100 replies were received for this puzzle, which is a variation on an old chestnut. Answers received ranged from two coconuts (yes — *two)* to millions and millions.

The correct answer is 279,931 coconuts, which is the smallest positive solution. Although a solution by micro isn't too difficult, an analytical approach can also be used. Since -5 is clearly a solution, then adding or subtracting multiples of 6^7 (six men, seven share-outs) to -5 gives all possible answers. Hence the smallest is -5 x 6^7 which is 279,931.

The winning entry came from P Drapala of Eight Mile Plains. Congratulations, Mr Drapala, your prize will be with you forthwith — if not sooner.

BLUDNERS

A program named 'The Walls' appeared in the November issue with the end of one line missing. The last statement in line 4130 should read 'NEXT A'. And we forgot to thank Calcutronic for the loan of an Atari in the preparation of this listing.

MICROPRO ON SIGMA/OKI if800 PERSONAL COMPUTER

Winston Miller and Margaret Hopkins of Archive Computer Services

Almost everyone knows of Wordstar, one of the most successful word processing systems ever. MicroPro, the company responsible for its development, now has made available special Sigma/OKI versions of this famous package.

All the MicroPro family of systems are ready for demonstration at your nearest dealer. The range also includes: Mailmerge, Datastar, Calcstar, Supersort, Wordmaster and Infostar.

MicroPro software is manufactured and supported in Australia by distributors such as Archives (Brisbane) and Imagineering (Sydney). They are dedicated to providing dealers with documentation advice and after sales service.

The Sigma/OKI library also includes software from suppliers such as Digital Research, Sorcim, Lifeboat and Australian applications from Padmede, Cyres, IMS, Software Source, John F. Rose and others.

The models 10 and 20 are ready for use with any CP/M software, without need for extra options, hardware conversions or extender cards.

Our dealers can give you a special demonstration. Please consult the dealer index earlier in this issue.

SIGMA/OKI

SIGMA/OKI Microcomputers
11th Floor, 157 Walker St.,
North Sydney. 2060
Dealers Enquiries: (02) 436 3777 David Thomas

CHRISTMAS SPECIAL

* APPLEMATE ADD-ON DISK DRIVE AND CONTROLLER FOR ** APPLE II COMPUTER

~~$575.00~~

$495.00 Inc. Tax

☐ HIGHLY RELIABLE

☐ FULLY COMPATIBLE

☐ SOFTWARE TRANSPARENT

BROCHURE AVAILABLE

* TRADEMARK OF MITAC INC
** TRADEMARK OF APPLE INC

apple

DYNETICS Pty. Ltd.

16 LANG ROAD, KENTHURST NSW 2154

Telephone: (02) 654 9055 BUS, (02) 654 9050 AH

THE PERSONAL AND PROFESSIONAL apple SOFTWARE SPECIALISTS

CHRISTMAS SPECIAL

STARTER PACK

- SPELLING AND SPEEDREADING
- GERMAN TUTOR
- FRENCH TUTOR
- AUSTRALIAN GEOGRAPHY TUTOR
- UNIVERSAL TUTOR

PLUS

- ARCHIVES — BIBLIOGRAPHY — FILING

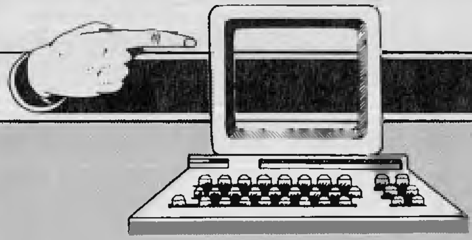
$275

CAVERNS OF MORDIA $75

Lothlorien Farming

Cultivating New Concepts

G.P.O. Box 1033, Sydney 2001. Phone: (02) 398 4023



MICROMARKET

Our bi-monthly guide to microcomputer systems. This first installment is not comprehensive and will be expanded to include most micros available in Australia. Updates and corrections are welcome and should be telephoned to Des Smith on (03) 523 7866.

MACHINE Price from	MODEL NO. Model	DISTRIBUTOR Phone No.	HARDWARE	SOFTWARE	MISCELLANEOUS
Altos $6000.00	ACS-8000	Microsolutions (03) 878 6004	Z80A:64K RAM: 8" F/D:S/D IBM: .3 Megs x 2: Seperate Screen	CP/M	H/D:Multi User
Archives $6000.00	Archives 2	Archives (03) 699 8377	Z80A/8086:128K RAM: 2 x 5.25": 750K Bytes	CP/M	H/D:Multi User
ABC N/A	ABC-24	O'Reilly Computers (02) 896 2799	Z80A:64K RAM:	CP/M	
Acorn $1550	BBC	Consolidated Marketing Corp. (03) 419 3033	16K - 32K RAM:	Acorn 1.0	Networking:Printer: 5.25" F/D
Apple $1395.00	Apple II	Apple (02) 212 2833	6502 & Z80A: 16 - 48K RAM: Colour VDU	DOS:CP/M:Basic: PASCAL:GAMES:O/S	5.25" F/D:11 Meg H/D
Apple N/A	Apple III	Apple (02) 212 2833	6502 & Z80A: 96K RAM:	DOS:CP/M:Basic: O/S:GAMES	11 Meg H/D
Atari $575	Atari 400	Futuretronics (03) 579 2011	6502:16K RAM: 5.25" F/D:90K: UHF & VHF	Atari:Graphics	Screen:Disks:Cassette
Atari $1439.00	Atari 800	Futuretronics (03) 579 2100	6502:48K RAM: 5.25" F/D:90K: UHF & VHF	Atari:Graphics	Disks:Cassette:Screen
BBC $1165.00	BBC Micro A	Barson Computers (03) 419 3033	6502:32K RAM: 400K:Colour or Monochrome	BBC	Disks
Canon N/A	CX-1	Canon (02) 20 1331	36 - 96K RAM:	MCX	N/A
Columbia N/A	MPN-16Bit	President Computers (03) 529 1788	8088:128K RAM: 2 x 5.25" F/D: 500K per Drive	MS DOS:CP/M 86	H/D
Commodore $399.00	VIC 20	Commodore (02) 437 6296	5K Bytes RAM:	CBM	3K Additional Mem.: Interface Cartridge: Screen
Commodore $2249	CBM 8032	Commodore (02) 437 6296	6502:36K RAM: 12" Picture Tube	CBM:Basic4	Dual Disk Drive: Dot Matrix Printer
Commodore $1199.00	PET 4016	Commodore (02) 437 6296	6502:16K RAM: 12" Picture Tube	CBM:Basic4	Single/Dual Disk Drive
Compucolour N/A	Compucolour II	Anderson Digital (03) 543 2077	8086:8-32K RAM: 5.25" F/D		
Cromenco $2820.00	N/A	Informative Systems (03) 690 2899	Z80A:64K RAM: 5.25" or 8": 390K Bytes - 1.2 Meg: 12" Green Screen	CDOS: CP/M Compatible Word Processing: S-Basic: Spreadsheet	
Daneva Aust P/L $2000 - $15,000	Western Digital	Daneva Aust. P/L (03) 598 5622	N/A		

List of Abbreviations

D/D .Double Density
D/S Double Sided
F/D Floppy Disk
H/D Hard Disk
N/A Not Available
P/P Parallel Port
O/S Operating System

MACHINE Price from	MODEL NO. Model	DISTRIBUTOR Phone No.	HARDWARE	SOFTWARE	MISCELLANEOUS
Dick Smith $1295.00	Sorcerer II	Dick Smith (02) 888 3200	Z80:8-48K RAM: 30 x 64 VDU	M/DOS:CP/M: EX Basic	RS232 Port:P/P: S100 BUS: High Res Graphics
Digital Microsystems $6500.00	DSC-3	Intelligent Terminals Pty. Ltd. (03) 523 6311	Z80A:64K RAM: 8" IBM Format: 1.2 Meg.	CP/M	H/D: Multi User
Discovery $1424	DPC-180	Archive Computers (03) 699 8377	Z80:64K RAM:	CP/M:DPC/OS	H/D
Dorado N/A	C50 MV	Dorado Micro Systems (02) 29 4884	Z80A:128K RAM: Winchester Disks: 6.7 Meg.	CP/M:OASIS	Multi User
Durango $1100.00	F85 GENERIC	Durango (03) 63 6476	Z80A:64-192K RAM: 5.25":200K Bytes 9" Green Screen	DX-85M:CP/M	H/D: Multi User
ECS N/A	TIME 4500	Electronic Control Systems (03) 699 2633	Z80A:64-256K RAM: 5" D/S & Hard: 1.2 - 10 Meg: Green non reflect screen	CP/M:UNIX	Communications
ERA N/A	ERA 50	Electronic Research Australia (075) 32 5577	Z80A:64K RAM: 2 x 8" D/S:1.2 Meg: Seperate Screen	CP/M	RS232:2 Modem Ports
Fairlight $13.530.00	QASAR	Fairlight Instruments (02) 331 6333	6800/6809 Dual: 64K RAM:2 x 8" S/D D/S:1.2 Meg:High Picture Resolution	MSDOS	62 Station Keyboard Light Pen:Printer: 30 Meg H/D
Hewlett Packard $3550.00	HP 85	Hewlett Packard Australia (03) 89 6351	16-32K RAM:5":200K: black & white VDU	Basic	Dot Matrix:Graphics
Hewlett Packard N/A	HP 125	Hewlett Packard Australia (03) 89 6351	Z80A:64K RAM: N/A	CP/M	
Hitachi $1994.00	Peach 6890	Delta Semi Conductors	6809:32K RAM:24K ROM:12" Green Phos.	DOS	5.25" F/D: 320K per drive
IBM N/A	IBM-PC	IBM (02) 923 5123	Intel 8088 16k-156k RAM 5.25 " F/D 160K Bytes:N/A	CP/M 86	
ICL $6131	ICL-PC	ICL (03) 267 2433	8085A 64K RAM:5.25 " F/D 250K bytes:80 x 24 Green Phosphor	CP/M & MP/M	Dot Matrix Printer H/D
Industrial Microsystems N/A	IMS-5000 IS	S.I. Microcomputer Productions (02) 231 4091	Z80A:64K RAM: N/A	CP/M & MP/M Turbo & DOS	
Intertec $5130	Superbrain	Computer Benefits (03) 699 1633	Z80A:64K RAM: 2 x 5.25 " F/D .6 MEG - 5 MEG Black & white	CP/M	RS232 Port
KDS $6150	KDS 7850	O'Reilly Computers (03) 890 6306	Z80A:64K RAM: 2 x 8" F/D D/S D/D 2.3 MEG	CP/M	10 MEG H/D
Micromation $7600	M&B	Micro Processor Applications (03) 890 0277	Z80:64K RAM: 8" F/D:1.2 MEG	CP/M 2.2	Network:Screen extra
Monroe $6788	8820	Business Control Systems (03) 596 6366	2 X Z80A:128K RAM: 2 X 5.25" F/D 320K Bytes per drive 80 x 24 Green Phos	MOS CP/M avail.	640K per drive Memory Exp to 256K
Monroe $4800	EC 8800	Business Control Systems	2 x Z80A:128K RAM: 1 x 5.25" F/D:320 KB Detach. High Res.	MOS	
National $4207	JB 3000	Megabus Microcomputers (03) 528 6069	8088:128K RAM: 5.25" F/D:80K Bytes Green High Res	MSDOS & CP/M 86	
NEC $4184	PC 8000B	NEC Australia (02) 438 3544	PD780C-1:32-64K RAM: Cassette:N/A High Res. Colour	N/A	Printer 100cps Disk Drives Avail.

MACHINE Price from	MODEL NO. Model	DISTRIBUTOR Phone No.	HARDWARE	SOFTWARE	MISCELLANEOUS
Northstar $5000	Horizon	Anderson Digital (03) 543 2077	Z80A/8086: 64K-256K RAM 5.25" F/D:720K Bytes Green Screen	GDOS & CP/M	Multi User:H/D
OKI N/A	IF800-20	Sigma Data (03) 26 2465	Z80A:64K RAM: 2 X 5.25" F/D 280k Per Drive 12" Green Phos.	CP/M	80 cps Printer:8" F/D ROM Cartridge
ONYX $12825	C 8000	Onyx Australia (02) 498 6611	Z80:64K RAM: H/D:10 MEG N/A	CP/M & Unix	Facility for 8 users
Option $4600	Option-2	Microprocessor Applications	Z80A:64K-128K RAM: 8" F/D D/S D/D	CP/M & MP/M	Screen Avail. Facility for 2 users
Osborne $2595	Osborne-1	Osborne Computers (02) 438 1800	Z80A:64K RAM: 2 x 5.25" F/D 90K per drive 5" screen	CP/M Wordstar:visicalc Mailmerge:C-Basic	Monitor:H/D
Dtrona $4995	Attache	Elmeasco Instruments (03) 233 4044	Z80A:64K RAM: 2 x 5.25" F/D 600k Bytes 5" 80 x 24 Green	CP/M	Monitor:Graphics
Tandy $5300	TRS-80	Tandy Electronics (02) 638 6633	Z80A:32-64K RAM: 8" F/D:500K 80 x 24 VDU	DOS Basic	
Sanyo $3500	MBC 1000	Sanyo (03) 67 5501	Z80A:64K RAM: 5.25" F/D D/S D/D 320K:Green Screen	CP/M	5-10 MEG H/D
Seiko N/A	8300	Tec & Thomas (02) 428 4233	8085A:64K RAM: N/A	N/A	
Seiko $22000	9500	Tec & Thomas (02) 428 4233	8068/8087:512K 5.25" F/D:1.2 MEG Built in Screen	RMX 86	10-20 MEG H/D
Sharp $6000	PC-031	Sharp (03) 763 9444	Z80A:64-128K RAM: 5.25" or 8" F/D 256K-4 MEG	Sharp:F/DOS:CP/M	
Sharp $1895	MZ 808	Sharp (03) 763 9444	Z80A:64K RAM: 5.25" F/D D/S D/D 256K bytes/drive 9" green screen	CP/M 2.2.2.02	
Sinclair $149	ZX81	Barson Computers (03) 419 3033	Z80A:8K RAM: Cassette:N/A	ZX81	Screen Avail.
Sirius $5495	Sirius-1	Barson Computers (03) 419 3033	8088:128K RAM: 5.25" - 8" F/D 600K - 1.2 MEG Colour Screen	MSDOS & CP/M	
Sord $3995	MK3	Mitsui Computer Systems (02) 929 9921	Z80Z @ 4MHZ 128K RAM: 2 x 5.25"F/D D/S D/D 330K per drive 12" Green screen	SORD:FDOS UCSD:Lifebaot	10 MEG Winchester 14" Mo[illegible]
Toshiba $5690	T 2000	Toshiba Australia (03) 561 2752	8085A:64K RAM: 5.25" F/D:N/A 12" Green Screen	N/A	5 MEG H/D
Vector $6000	2600	Dicker Data Products	Z80A:64K RAM: 5.25" F/D:1.2 MEG B/W VDU	DOS CP/M:Basic	H/D
Versatile $5692	Versatile 4	Microprocessor Applications (02) 754 5108	8085:32-56K RAM: 2 X 5.25" F/D:630K B/W VDU	MDOS & CP/M MBasic	
Xerox $5478	820	Novex Data Systems (02) 267 5544	Z80:64K RAM:6K ROM 5.25" or 8" F/D 1 MEG:N/A	CP/M & CO2000	5-35 MEG H/D Avail
Xerox $5940	820-2	Novex Data Systems (02) 267 5544	Z80A:64K-92K RAM: N/A:1 MEG Screen Inclusive	CP/M & CO2000	10 MEG in box

NO, WE WON'T BYTE YOU

BUT WE WILL MAKE YOU PURR WITH OUT PRICES
ON THE INCREDIBLE

HITACHI PEACH

OPENING SPECIAL 1/2 PRICE BOOK SALE

AS WELL AS DISKETTES, PRINTERS, SOFTWARE & BOOKS

BUSINESS, HOME, ANYWHERE IS PURRFECT FOR A PEACH
ESPECIALLY ON A CUSTOM-BUILT DESK DESIGNED TO SUIT YOUR NEEDS

C. ITOH PRINTERS WITH APPLE or TRS 80 interface
8510. . . $1060 incl. TAX M1550 . . . $1430 incl. TAX

KAYPRO II $2950 incl. TAX (Now with $2500 worth of software at no extra charge.)

VERBATIM DISKETTES
5¼" DOUBLE DENSITY SINGLE SIDED $40.00 BOX OF 10
5¼" DOUBLE DENSITY DOUBLE SIDED $60.00 BOX OF 10

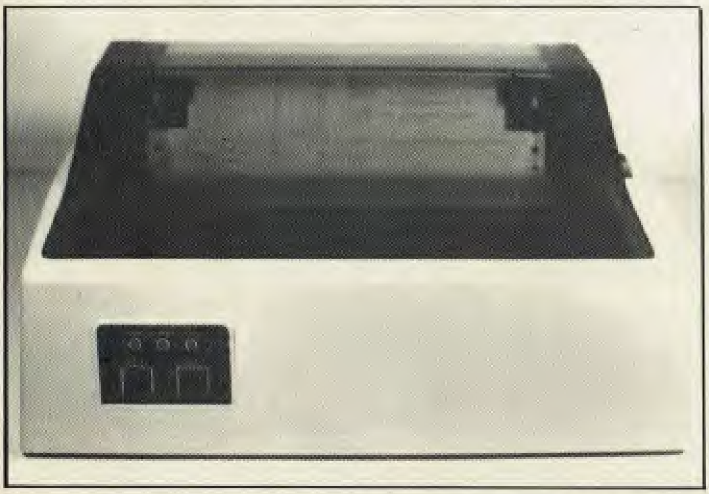
WILDCAT COMPUTERS & SOFTWARE SALES

2/150 CRONULLA ST, CRONULLA
PHONE: (02) 523 3505

IMPORTERS WAREHOUSE CLEARANCE

STAR 80 PRINTER-------ONLY $399**

(Plus Sales Tax if Applicable)



DP 8480 FEATURES
* 80 cps. Bidirectional Logic Seeking.
* Tractor and Friction Feed.
* 80, 96, 132 Chars Per Line.
* Double Width Characters.
* Centronics Compatible Interface Only.

Mail Order Only — No Personal Shopping.
Fill Out The Order Coupon and Mail To:

**ALPHA DIGITAL RESEARCH PTY. LTD.,
1761 FERNTREE GULLY ROAD,
FERNTREE GULLY. VIC. 3156.**

PLEASE SEND VIA: . . . .
. . . . STAR-80 DP 8480 PRINTER
(Qty)
My Cheque for $ . . . . is enclosed.
Name . . . .
Address . . . .
. . . . Postcode . . . .

** Price does not include Sales Tax.
Tax Paid Price is $478.80.

Price does not include Delivery.

Please advise method of shipment and goods will be despatched Freight Forward.

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'.

Dubai	Gulf Computer Exhibition. Contact: Trade Centre Management Co, 01-930-3881	December 13 – December 16, 1983
Atlanta, Georgia	Southcon Electronics Show. Contact: ECL (Exhibition Agencies) Ltd, 01-486-1951	January 18 – January 20, 1983
Sydney	The 1st Australian Personal Computer Show, Centrepoint. Contact: Australian Exhibition Services	March 10–12, 1983

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.

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.

PROGRAMS

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 280, Hawthorn, Victoria 3122.

Teepee Textpro

by Mike O'Regan

This is one of the best programs I've received for the TI99/4A. It's a word processor which works with the TI Thermal Printer, but if anyone wants to reconfigure it for use with a different printer the relevant lines to change are 1720, 1740 and 2630. Lines 1980, 2190, 2240 and 2630 should all have the '28' changed to suit the line length of any other printer. All the instructions you'll need to process your words of wisdom are contained within the program as a menu.

As listed it will print out 28 columns to match the screen. Control characters for upper case (∧) and indent (>) are not included in the characters per line count.

One of its neat features is that it uses the TI99/4A's character redefinition facilities and the Thermal Printer's ability to mimic the chosen character font. The characters used here are customised, offering lower case with true descenders. These can be obtained directly from the keyboard.

TeepeeTextpro does posess some minor limitations — which were deliberate in the interests of saving RAM. It doesn't offer automatic line wrap-around, block deletion or insertion or line length safeguards on edited lines. However, it does SAVE and LOAD your text to and from cassette at the press of a key.

For users of the TI99/4A this program will be useful as a rough word processor, although I couldn't see it being used for business letters and suchlike unless a more conventional printer were used.

```
100 REM TEEPEETEXPRO
110 REM BY MIKE O'REGAN
120 REM UPDATED 2/09/82
130 REM
140 REM CUSTOM CHAR SET
150 DATA 00101010001
160 DATA 002828
170 DATA 00287C287C28
180 DATA 00382070207C
190 DATA 006468102C4C
200 DATA 00205020545824
210 DATA 00101
220 DATA 000810101008
230 DATA 00201010102
240 DATA 002828
250 DATA 0010107C101
260 DATA 000000060204
270 DATA 0000007C
280 DATA 00000000606
290 DATA 000000060204
300 DATA 00384C546438
310 DATA 001030101038
320 DATA 00384408107C
330 DATA 003844184438
340 DATA 000818287C08
350 DATA 007C403C047C
360 DATA 001020784438
370 DATA 007C0810202
380 DATA 003844384438
390 DATA 0038443C081
400 DATA 00303000303
410 DATA 003030003030 2
420 DATA 000204080402
430 DATA 00007C007C
440 DATA 00201008102
450 DATA 00384418001
460 DATA 00384C4C4038
470 DATA 003844447C44
480 DATA 007844784478
490 DATA 003C4440403C
500 DATA 007844444478
510 DATA 007C4078407C
520 DATA 007C4078404
530 DATA 003C44404C3C
540 DATA 0044447C4444
550 DATA 003810101038
560 DATA 000404044438
570 DATA 004448704844
580 DATA 00404040407C
590 DATA 00446C544444
600 DATA 004464544C44
610 DATA 007C4444447C
620 DATA 0078444478 4
630 DATA 00384444 4C3C
640 DATA 00784478504C
650 DATA 003C40380478
660 DATA 007C1010101
670 DATA 004444444438
680 DATA 0044444281
690 DATA 004454545428
700 DATA 004428102844
710 DATA 00442810101
720 DATA 007C0810207C
730 DATA 001810101018
740 DATA 004020100804
750 DATA 00301010103
760 DATA 00102844
770 DATA 00000000007C
780 DATA 001008
790 DATA 00003844443C
800 DATA 004078444478
810 DATA 00003C40403C
820 DATA 00043C44443C
830 DATA 00003844784038
840 DATA 00384070404
850 DATA 00003844443C0478
860 DATA 004078444444
870 DATA 00100010101
880 DATA 00040004044438
890 DATA 004048704844
900 DATA 00301010101
910 DATA 000068545454
920 DATA 000078444444
930 DATA 000038444438
940 DATA 00007844447840 4
950 DATA 00003C44443C0404
960 DATA 00005864404
970 DATA 00003840380438
980 DATA 002070202038
990 DATA 00004444443C
1000 DATA 00004444281
1010 DATA 000044545428
1020 DATA 000044283844
1030 DATA 00004444443C047C
```

PROGRAMS

```
1040 DATA 00007C08107C
1050 FOR K=33 TO 122
1060 READ C$
1070 CALL CHAR(K,C$)
1080 NEXT K
1090 CALL CLEAR
1100 GOSUB 5040
1110 PRINT "          TEEPEETEXPRO
": : : : :
1120 PRINT "         BY MIKE O'REG
AN"
1130 PRINT "FOR THE T.I. THERMAL
 PRINTER"
1140 PRINT : : : : : : : : :
1150 GOSUB 5090
1160 CALL HCHAR(1,1,30,32)
1170 CALL VCHAR(2,1,30,22)
1180 CALL VCHAR(2,32,30,22)
1190 CALL HCHAR(24,1,30,32)
1200 GOSUB 4990
1210 CALL CLEAR
1220 PRINT "99.4 OWNERS PRESS KE
Y 4"
1230 CALL KEY(0,K,S)
1240 IF S=0 THEN 1230
1250 IF K<>52 THEN 1270
1260 TT=1
1270 DIM A$(100)
1280 REM - MASTER MENU
1290 CALL CLEAR
1300 GOSUB 5040
1310 PRINT "        ### MENU ###"
: :
1320 PRINT TAB(5);"0.   WRITE TEX
T"
1330 PRINT TAB(5);"1.   DRAFT ON
SCREEN"
1340 PRINT TAB(5);"2.   PRINT FIN
AL DRAFT"
1350 PRINT TAB(5);"3.   DELETE LI
NE"
1360 PRINT TAB(5);"4.   INSERT LI
NE"
1370 PRINT TAB(5);"5.   REPLACE L
INE"
1380 PRINT TAB(5);"6.   REPLACE W
ORD"
1390 PRINT TAB(5);"7.   SAVE ON T
APE"
1400 PRINT TAB(5);"8.   READ FROM
 TAPE"
1410 PRINT TAB(5);"9.   START AGA
IN"
1420 PRINT TAB(5);"?.   INSTRUCTI
ONS": :
1430 PRINT TAB(8);"KEY?"
1440 GOSUB 5000
1450 GOSUB 5090
1460 CALL KEY(0,K,S)
1470 IF S=0 THEN 1460
1480 CALL CLEAR
1490 A=K-48
1500 IF K=48 THEN 2030
1510 IF K=49 THEN 4060
1520 IF K=50 THEN 2510
1530 IF (K<54)*(K>50)THEN 1830
1540 IF K=54 THEN 3430
1550 IF K=55 THEN 3890
1560 IF K=56 THEN 3970
1570 IF K=57 THEN 1760
1580 IF K=63 THEN 4740
1590 GOTO 1280
1600 GOSUB 5040
1610 PRINT "PICK CHARACTER SET":
 : :
1620 GOSUB 5000
1630 PRINT TAB(6);"1.PRINTER SET
(LARGE)"
1640 PRINT TAB(6);"2.CUSTOM SET
(SMALL)": : :
1650 PRINT "IS YOU PRINTER SWITC
HED ON?": : :
1660 GOSUB 5090
1670 CALL KEY(0,KEY,S)
1680 IF S<1 THEN 1670
1690 IF KEY=77 THEN 1290
1700 IF (KEY<49)+(KEY>50)THEN 16
00
1710 IF KEY<>49 THEN 1740
1720 SET$="TP.E"
1730 IF K=49 THEN 4080 ELSE 2530
1740 SET$="TP.U.E"
1750 IF K=49 THEN 4080 ELSE 2530
1760 FOR I=1 TO L
1770 A$(I)=""
1780 NEXT I
1790 L=0
1800 PRINT "FILE NOW BLANK"
1810 GOSUB 4990
1820 GOTO 1290
1830 INPUT "LINE NUMBER?":B
1840 IF (B<1)+(B>L)=-1 THEN 1850
 ELSE 1880
1850 PRINT "NON-EXISTENT LINE"
1860 GOSUB 4990
1870 GOTO 1280
1880 PRINT "OLD LINE #";B
1890 PRINT A$(B)
1900 FOR PSE=1 TO 250
1910 NEXT PSE
1920 ON A GOTO 1290,1290,3220,33
00,1930
1930 PRINT "ENTER REPLACEMENT LI
NE:"
1940 INPUT A$(B)
1950 IF SEG$(A$(B),1,2)<>"BB" TH
EN 1980
1960 A$(B)=" "
1970 GOTO 1280
1980 IF LEN(A$(B))>28 THEN 2180
ELSE 1280
1990 REM -- ENTER TEXT
2000 TEMP=L
2010 L=B
2020 GOTO 2350
2030 GOSUB 5040
2040 PRINT "READY FOR TYPING --
STARTING WITH LINE";L+1
2050 IF TT=1 THEN 2060 ELSE 2070
2060 PRINT "^=UPPER CASE"
2070 PRINT "SLASH(DIVIDE SIGN)=C
OMMA    STAR(MULTIPLY SIGN)=QUOT
ES   DOLLAR SIGN=$#
>=INDENT"
2080 PRINT "N.B. # CUSTOM CHARS
ONLY"
2090 PRINT "BB=BLANK LINE": : :
2100 PRINT "KEY-IN ZZ TO FINISH
ENTERINGTEXT": : : : :
2110 GOSUB 5090
2120 GOSUB 5000
2130 PRINT L+1
2140 IF L+1<>52 THEN 2160
2150 CALL SOUND(1000,440,2,444,2
,448,2)
2160 INPUT A$(L+1)
2170 LC=LEN(A$(L+1))
2180 LL=LC
2190 IF LC>28 THEN 2200 ELSE 225
0
2200 FOR I=LL TO 1 STEP -1
2210 IF SEG$(A$(L+1),I,1)<>CHR$(
94)THEN 2230
2220 LC=LC-1
2230 NEXT I
2240 IF LC>28 THEN 2310
2250 IF LEN(A$(L+1))<1 THEN 2130
2260 IF SEG$(A$(L+1),1,2)="ZZ" T
HEN 1290
2270 IF SEG$(A$(L+1),1,2)<>"BB"
THEN 2290
2280 A$(L+1)=" "
2290 L=L+1
2300 GOTO 2130
2310 REM - SHORTEN LONG LINE
2320 REM BELL
2330 CALL SOUND(100,-5,1)
2340 L=L+1
2350 PRINT "LINE TOO LONG -- SHO
RTENED AS SHOWN:"
2360 FOR I=28 TO 1 STEP -1
2370 IF SEG$(A$(L),I,1)=" " THEN
 2420
2380 NEXT I
2390 A$(L)=""
2400 L=L-1
2410 GOTO 2430
2420 A$(L)=SEG$(A$(L),1,I-1)
2430 PRINT A$(L)
2440 IF (TEMP=B)*(TEMP<>0)THEN 2
450 ELSE 2460
2450 L=TEMP
2460 IF A$(L+1)="" THEN 2130
2470 IF (L>TEMP)*(TEMP=0)THEN 12
90
2480 L=TEMP
2490 GOTO 2160
2500 REM -- PRINT TEXT
2510 C=0
2520 GOTO 1600
2530 GOSUB 5000
2540 PRINT TAB(5);"READY TO PRIN
T               CHOOSE FORMAT:"
2550 PRINT "0 = NUMBERED
          1 = SINGLE SPACED
        2 = DOUBLE SPACED"
2560 CALL KEY(0,K,S)
2570 IF S=0 THEN 2560
```

PROGRAMS

```
2580 CALL CLEAR
2590 IF K=77 THEN 1280
2600 B=K-48
2610 IF (K<48)+(K>52)THEN 2540
2620 IF (B<0)+(B>2)THEN 2510
2630 OPEN #1:SET$,OUTPUT
2640 FOR I=1 TO L
2650 P$=A$(I)
2660 IF B<>0 THEN 2680
2670 PRINT #1:I;
2680 IF LEN(P$)=0 THEN 3070
2690 PRINT #1:TAB(3);"";
2700 FOR K=1 TO LEN(P$)
2710 P=ASC(SEG$(P$,K,1))
2720 REM -- CONVERT UNSHIFTED LE
TTERS TO LOWER CASE
2730 IF (P>64)*(P<91)THEN 2740 E
LSE 2760
2740 P=P+32
2750 GOTO 2950
2760 REM -- CONVERT SHIFTED LETT
ERS TO UPPER CASE
2770 IF (P>96)*(P<123)THEN 2780
ELSE 2800
2780 P=P-32
2790 GOTO 2950
2800 REM -- CONVERT SLASH (/) TO
 COMMA (,) - * TO " AND > TO IND
ENT
2810 IF P<>47 THEN 2840
2820 P=44
2830 GOTO 2950
2840 IF P<>42 THEN 2870
2850 P=34
2860 GOTO 2950
2870 IF P<>62 THEN 2900
2880 GOSUB 3110
2890 GOTO 2950
2900 IF P=94 THEN 2910 ELSE 2930
2910 GOSUB 3150
2920 GOTO 2950
2930 IF P<>95 THEN 2950 ELSE 295
0
2940 GOSUB 3180
2950 PRINT #1:CHR$(P);
2960 REM - TEST FOR BLANK LINE (
BB)
2970 IF (P=66)*(LEN(P$)=2)THEN 3
020
2980 NEXT K
2990 IF B<>2 THEN 3020
3000 PRINT #1
3010 C=C+1
3020 PRINT #1
3030 C=C+1
3040 CALL KEY(0,K,S)
3050 IF S=0 THEN 3070
3060 GOTO 3080
3070 NEXT I
3080 CLOSE #1
3090 CALL SOUND(50,262,0,524,0)
3100 GOTO 1280
3110 PRINT #1:TAB(4);
3120 PRINT TAB(1);
3130 P=32
3140 RETURN
3150 P=ASC(SEG$(P$,K+1,1))
3160 K=K+1
3170 RETURN
3180 PRINT #1:TAB(15);
3190 P=32
3200 RETURN
3210 REM  -- DELETE LINE
3220 L=L-1
3230 FOR I=B TO L
3240 A$(I)=A$(I+1)
3250 NEXT I
3260 A$(L+1)=""
3270 GOTO 1290
3280 REM
3290 REM -- INSERT LINE
3300 L=L+1
3310 FOR I=L TO B+1 STEP -1
3320 A$(I)=A$(I-1)
3330 NEXT I
3340 PRINT "ENTER NEW LINE":
3350 GOSUB 5000
3360 A$(B)=""
3370 INPUT A$(B)
3380 IF SEG$(A$(B),1,2)<>"BB" TH
EN 3400
3390 A$(B)=" "
3400 GOTO 1290
3410 REM
3420 REM -- REPLACE WORD
3430 INPUT "WORD TO BE CHANGED:"
:R$
3440 R$=" "&R$&" "
3450 D=LEN(R$)
3460 IF D>2 THEN 3510
3470 PRINT "ERROR"
3480 FOR PSE=1 TO 250
3490 NEXT PSE
3500 GOTO 3430
3510 N=0
3520 INPUT "NEW WORD?":N$
3530 INPUT "FROM LINE (0=ALL)?":
B
3540 IF B<=L THEN 3570
3550 PRINT "NO SUCH LINE"
3560 GOTO 3530
3570 IF B>1 THEN 3610
3580 C=L
3590 B=1
3600 GOTO 3640
3610 INPUT "THROUGH LINE:":C
3620 IF (C<B)+(C>L)<>-1 THEN 364
0
3630 C=L
3640 PRINT "SEARCHING"
3650 FOR K=B TO C
3660 P$=" "&A$(K)&" "
3670 IF LEN(P$)<D THEN 3740
3680 A=LEN(P$)+1-D
3690 FOR I=1 TO A
3700 IF R$<>SEG$(P$,I,D)THEN 373
0
3710 GOSUB 3780
3720 P$=SEG$(P$,2,LEN(P$)-1)
3730 NEXT I
3740 NEXT K
3750 PRINT N;" CHANGE(S) MADE"
3760 GOSUB 4990
3770 GOTO 1290
3780 CPOS=POS(P$,R$,1)
3790 IF CPOS<>0 THEN 3810
3800 RETURN
3810 T$(1)=SEG$(P$,2,CPOS-1)
3820 T$(2)=SEG$(P$,CPOS+D,LEN(P$
)-1)
3830 P$=T$(1)&N$&" "&T$(2)
3840 A$(K)=P$
3850 N=N+1
3860 PRINT "LINE #";K
3870 PRINT A$(K)
3880 RETURN
3890 PRINT "READY CASSETTE FOR R
ECORDING:"
3900 OPEN #1:"CS1",INTERNAL,OUTP
UT,FIXED 192
3910 PRINT #1:L
3920 FOR I=1 TO L+5 STEP 6
3930 PRINT #1:A$(I),A$(I+1),A$(I
+2),A$(I+3),A$(I+4),A$(I+5)
3940 NEXT I
3950 CLOSE #1
3960 GOTO 1290
3970 PRINT "READY CASSETTE TO RE
AD"
3980 OPEN #1:"CS1",INTERNAL,INPU
T ,FIXED 192
3990 INPUT #1:X
4000 FOR I=L+1 TO X+L+5 STEP 6
4010 INPUT #1:A$(I),A$(I+1),A$(I
+2),A$(I+3),A$(I+4),A$(I+5)
4020 NEXT I
4030 L=L+X
4040 CLOSE #1
4050 GO TO 1290
4060 REM PRINT FINAL TEXT ON SCR
EEN ONLY
4070 C=0
4080 PRINT "CHOOSE FORMAT:": : :
4090 PRINT TAB(8);"0=NUMBERED"
4100 PRINT TAB(8);"1=SINGLE SPAC
ED"
4110 PRINT TAB(8);"2=DOUBLE SPAC
ED": : :
4120 GOSUB 5000
4130 CALL KEY(0,K,S)
4140 IF S=0 THEN 4130
4150 CALL CLEAR
4160 B=K-48
4170 IF (K>48)+(K<52)THEN 4190 E
LSE 4080
4180 IF (B<0)+(B>2)THEN 4070
4190 FOR I=1 TO L
4200 P$=A$(I)
4210 IF B<>0 THEN 4230
4220 PRINT I
4230 IF LEN(P$)=0 THEN 4590
4240 PRINT "";
4250 FOR K=1 TO LEN(P$)
4260 P=ASC(SEG$(P$,K,1))
4270 IF (P>64)*(P<91)THEN 4280 E
LSE 4300
4280 P=P+32
4290 GOTO 4440
```

```
4300 IF (P>96)*(P<123)THEN 4310
ELSE 4330
4310 P=P-32
4320 GOTO 4440
4330 IF P<>47 THEN 4360
4340 P=44
4350 GOTO 4440
4360 IF P<>62 THEN 4390
4370 GOSUB 4610
4380 GOTO 4440
4390 IF P<>94 THEN 4420
4400 GOSUB 4640
4410 GOTO 4440
4420 IF P<>95 THEN 4440
4430 GOSUB 4670
4440 PRINT CHR$(P);
4450 CALL KEY(0,KE,ST)
4460 IF ST<1 THEN 4480
4470 GOTO 1280
4480 REM TEST FOR BLANK LINE
4490 IF (P=66)*(LEN(P$)=2)THEN 4
540
4500 NEXT K
4510 IF B<>2 THEN 4540
4520 PRINT
4530 C=C+1
4540 PRINT
4550 P=32
4560 C=C+1
4570 IF C<51 THEN 4590
4580 C=0
4590 NEXT I
4600 GOTO 4700
4610 PRINT TAB(1);
4620 P=32
4630 RETURN
4640 P=ASC(SEG$(P$,K+1,1))
4650 K=K+1
4660 RETURN
4670 PRINT TAB(15);
4680 P=32
4690 RETURN
4700 PRINT "
     PRESS ANY KEY TO CONTINUE"
4710 CALL KEY(0,K,S)
4720 IF S=0 THEN 4710
4730 GOTO 1210
4740 CALL CLEAR
4750 REM INSTRUCTIONS
4760 PRINT "THIS PROGRAM HANDLES
 UP TO  100 LINES OF TEXT": :
4770 GOSUB 4990
4780 IF TT<>1 THEN 4820
4790 PRINT "IT PRINTS IN LOWER C
ASE       (SMALL LETTERS) UNLESS T
OLD OTHERWISE."
4800 GOSUB 4990
4810 PRINT "TO PRODUCE UPPER CAS
E        LETTERS, PRESS '^' BEFOR
E   EACH LETTER.": :
4820 PRINT "99.4A USERS USE SHIF
T KEY TOPRODUCE UPPER & LOWER CA
SE": :
4830 GOSUB 4980
4840 PRINT "TO PRODUCE A BLANK L
INE        PRESS 'BB', OR AN INDENT
 '>'"
4850 PRINT "FOLLOWED BY THE APPR
OPRIATE NUMBER OF SPACES": : :
4860 GOSUB 4990
4870 PRINT "IF YOUR TRY TO TYPE
A LINE  WHICH IS TOO LONG THEN Y
OU  WILL BE TOLD THAT THIS IS TO
"
4880 PRINT "BE SHORTENED TO THE
NEAREST WHOLE WORD AND THE NEW L
INE IS DISPLAYED.  LINES WITH"
4890 PRINT "MORE THAT 28 CONTINU
OUS      CHARACTERS(I.E. NO SPACE
S)  WILL BE TOTALLY DISCARDED":
: :
4900 GOSUB 4980
4910 PRINT "AFTER 52 LINES YOU W
ILL GET AN AUDIO INDICATION THAT
 YOUHAVE ENOUGH TEXT TO FILL ONE
"
4920 PRINT "SIDE OF AN A4 SHEET
(SINGLE SPACED)": :
4930 PRINT "TEXT PRINTOUT ON SCR
EEN OR  PRINTER MAY BE STOPPED A
T   ANY TIME BY PRESSING ANY KEY
": :
4940 PRINT "PRESS ANY KEY FOR ME
NU"
4950 CALL KEY(0,K,S)
4960 IF S=0 THEN 4950
4970 GOTO 1280
4980 CALL SOUND(4000,10000,30)
4990 CALL SOUND(4000,10000,30)
5000 FOR BEEP=1 TO 3
5010 CALL SOUND(50,1000,1)
5020 NEXT BEEP
5030 RETURN
5040 FOR DIS=1 TO 12
5050 CALL SCREEN(8)
5060 CALL COLOR(DIS,8,8)
5070 NEXT DIS
5080 RETURN
5090 FOR APP=1 TO 12
5100 CALL COLOR(APP,2,8)
5110 NEXT APP
5120 RETURN
```

PET Firebird

by Michael Fok

There must be thousands of sticky-fingered, bug-eyed space game addicts who'll find this game keeps them happy over Christmas. It combines many of the dubious qualities of Asteroids, Space Invaders and Phoenix in a series of four little sub-games and it's even got a story behind it. Unfortunately the text in this listing could be improved by some discreet editing, but it's readable and easily absorbed.

The rather comprehensive instructions are entirely optional, as they're a separate program occupying almost 5k. If you don't feel like typing them in you can read them from the listing. The

COMPARE OUR PRICES FOR HITACHI PEACH SYSTEMS

AUSTRALIA'S LOWEST PRICES

HURRY WHILE STOCKS LAST

SYSTEMS FROM $1995 Plus tax

(CPU, MONITOR, DOCUMENTATION AND DRIVE)

RADARO COMPUTER CENTRE

316 QUEEN STREET, MELBOURNE TELEPHONE ENQUIRIES (03) 67-6638

OPEN SATURDAY MORNINGS

N.B. FREE DELIVERY FOR COUNTRY AND INTERSTATE CLIENTS

Software compatible

HARD DISK SYSTEM

for TRS-80™-Model 3

The MICRO-80 Winchester Hard Disk System plugs onto the expansion port of a Model 3 disk system in seconds to give you over 27 times the storage capacity and up to 20 times the data transfer rate of a standard double density disk drive. Now your Model 3 can become a serious business computer capable of handling the most demanding applications. The DOSPLUS 4.0 operating system supplied is compatible with most Model 3 programs and adds the following enhancements to your operating system and disk basic.

- Single volume addressing/Double sided floppies seen as one drive—one file can expand to limit of the hard drive
- Hard Disk—disk editing utilities
- Incredible I/0 speed
- Runs any combination of densities or tracks
- Also operates 8" drives with special hardware — comes with expanded users guide and complete DOS technical section on I/0 calls and DCB organization
- Ability to use hard drive as the "system" drive
- BASIC array sort—multi key, multi array
- Tape/Disk-Disk/Tape utility (with relocator)
- Input (controlled screen input)
- Random access and ASCII modification on Diskdump
- BASIC checks for active "DO"
- Backup and Format from a "DO" file
- Much improved Backup (more reliable)
- I/0 package much faster (disk access time reduced)
- Repeat last DOS command with "/" (ENTER)
- Short directory (filename and extension) available
- Short directory of Model III TRSDOS disks
- Single file convert from Model III TRSDOS
- COMPLETE device routing supported (DOS and BASIC)
- Ability to save BASIC programs directly to another machine's memory (if equipped with DOSPLUS 3.4)

MICRO-80 Winchester disk drive subsystem complete with DOSPLUS 4.0

5 Mbyte $2995 10 Mbyte $3775

TM TRS-80 is a trademark of Tandy Corporation

BRUERPYM 2364

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

MICRO-80

game itself takes up just over 7k. Both programs use machine code which occupies the second cassette buffer and when the instructions have run the program exits, so to avoid keying-in twice (as you have to type NEW between instructions and game), save the instructions before you run them. Once you've typed NEW, run the game. Pressing key 'N' after running will prevent the 'STOP' key from being disabled — on a 'new ROM' PET.

In the first three stages of the game you can cheat by pressing 'C' (if you're the cheating type), but this won't actually make things much easier as you can't cheat on the fourth stage. Firebird will run on any PET. It was tested using Basic4 and Basic2.

Game

```
READY.

 1 REM      ***************************
 2 REM      ***FIREBIRD VERSION 1.1***
 3 REM      *********M.FOK -82**********
 4 REM      ***************************
 5 REM
 6 REM
 7 REM
 8 FORI=1TO100:GETA$:IFA$="N" THEN30
 9 NEXTI
 10 IF PEEK(50003)=1 THEN POKE144,49
 15 IF PEEK(50003)=160 THEN POKE 144,88
 20 IF PEEK(50003)=0 THEN POKE 59458,255
 30 ER=515-(364*SGN(PEEK(50003)))
 100 REM***FIREBIRD ATTACK***
 110 DEF FNA(AX)=32768+AX+AY*40
 120 AX=20:AY=21:EX=2:EY=1:F1=0:F2=0:F3=0:DI=1
 130 Y$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 140 RESTORE:GOSUB8350:GOSUB7000:POKE FNA(AX),1
 200 PRINT"▒"RIGHT$(Y$,EY);TAB(EX);"    "
 210 IF F1=1 THEN 300
 220 EX=EX+DI:IF EX=2 THEN DI=1
 230 IF EX=35 THEN DI=-1
 240 IF RND(1)>.95 THEN F1=1
 250 PRINT"▒"RIGHT$(Y$,EY);TAB(EX);
 260 IF H=1 THEN H=0:PRINT" ╲A╱";:GOTO280
 270 H=1:PRINT" ╱A╲";
 280 IF RND(1)>.6 THEN PRINT"▒●"
 290 GOTO400
 300 G=G+1:IF G=2 THEN G=0:EX=EX+SGN(AX-EX)
 310 EY=EY+1:IF EY=22 THEN EY=1:F2=1
 320 PRINT"▒"RIGHT$(Y$,EY);TAB(EX);
 330 IF H=1 THEN H=0:PRINT" ╲V╱";:GOTO350
 340 H=1:PRINT" ╱V╲";
 350 IF F2=1 AND RND(1)>.95 THEN F1=0:F2=0
 360 IF RND(1)>.6 THEN PRINT"▒●"
 400 SYS826:IF PEEK(FNA(AX))<>1 THEN AP=FNA(AX):GOTO9050
 405 POKE FNA(AX),32
 410 A=PEEK(ER):IF A=42 THEN AX=AX+(AX>2)
 420 IF A=41 THEN AX=AX-(AX<36)
 430 IF A=48 AND F3=0 THEN F3=1:LP=FNA(AX)-40
 435 IF A=31 THEN 900
 440 POKE FNA(AX),1
 450 IF F3=0 THEN 200
 460 POKE LP,32:FORI=1TO3:LP=LP-40:IF LP<32768 THEN F3=0:GOTO200
 470 A=PEEK(LP):IF A=1 OR A=22 THEN S=S+20:GOTO900
 480 NEXTI:POKELP,34:GOTO200
 900 AP=FNA(AX):FORI=1TO30:PRINT"▒";RIGHT$(Y$,EY);TAB(EX);" ╲V╱"
 910 PRINT"▒";RIGHT$(Y$,EY);TAB(EX);"    ▒▒▒▒╲V╱▒"
 920 NEXTI:PRINT"▒";
 930 POKE AP,32:AP=AP-40:IF AP<33208 THEN 950
 940 POKE AP,1:FORI=1TO50:NEXTI:GOTO930
 950 POKE AP,79:FORI=1TO50:NEXTI
 960 POKEAP,32:AP=AP+SGN(33198-AP):IF AP=33198 THEN1000
 970 POKE AP,60:FORI=1TO50:NEXTI:GOTO960
 1000 REM *** ASTEROIDS ATTACK ***
 1010 POKE AP,60:FORI=826 TO 980:READA:POKEI,A:NEXTI:POKE768,2:F1=0
 1100 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";:FORI=1TO10:GETA$:NEXTI
 1110 POKE768,2:AP=33198:POKE AP,60:MI=32768:F1=0:F2=0:F3=0
 1150 PRINT"     ╱╲_         ╱╲          ╱‾‾╲
 1160 PRINT"   ╱╲╱   ╲___   ╱‾  ╲_    ╱‾   ╲_  ╱‾╲
 1170 PRINT"╱‾          ╲_╱      ╲____╱       ╲╱   ╲╱
 1200 FORI=1TO1000:IF RND(1)>.95 THEN POKE32769,87
 1210 GOSUB1300:NEXTI
 1220 FORI=1TO500:GOSUB1300:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";500-I;"▒ ":NEXTI:S=S+
50
 1230 FORI=1TO800:POKE 33519,32:GOSUB1300:NEXTI:GOTO1900
 1300 IF PEEK(AP)<>60 THEN 7500
 1305 POKE AP,32
 1310 GETA$:IF A$="Q" THEN AP=AP-40:IF AP<MI THEN AP=AP+40
 1320 IF A$="A" THEN AP=AP+40:IF AP>33519 THEN AP=AP-40
 1325 IFF1=0ANDA$="C"THENI=0:POKEAP,60:PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ ▒":F1=1:GO
TO1230
 1330 POKE AP,60:SYS826:RETURN
 1900 FORI=1 TO 8:POKE AP,32:AP=AP+1:POKE AP,60
 1910 FORN=1TO50:NEXTN,I
 2000 REM *** LASER ATTACK ***
 2010 POKE 891,96:E1=32769:MI=32768:MA=33519:F1=0:F2=0:POKE33648,32
 2200 SYS840:EX=INT((E1-32768)/40):AX=INT((AP-32768)/40)
 2210 POKE E1,32:F1=F1-(F1<>10)
 2215 IF F1<>10 THEN E1=E1-SGN(AX-EX)*40:GOTO2225
 2220 E1=E1+SGN(AX-EX)*40
 2225 IF E1<32768 THEN E1=E1+40
 2226 IF E1>33518 THEN E1=E1-40
 2230 IF F1=10 AND ABS(AX-EX)<3 AND RND(1)>.5 THEN F1=0:GOTO2500
 2240 POKE E1,62
 2300 POKE AP,32:F2=F2-(F2<>10)
 2310 A=PEEK(ER):IF A=64 THEN AP=AP-40:IF AP<MI THEN AP=AP+40
 2320 IF A=48 THEN AP=AP+40:IF AP>MA THEN AP=AP-40
 2325 IF A=31 THEN 2900
 2330 IF F2=10 AND A=42 THEN F2=0:GOTO2600
 2340 POKE AP,60
 2345 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";F1"▒ ",F2"▒ "
 2350 GOTO2200
 2500 PRINT"▒"RIGHT$(Y$,EX);">────────────────────────────────"
```

PROGRAMS

```
2510 IF EX=AX THEN 7500
2520 PRINT"ﾇ"RIGHT$(Y$,EX);"                                        "
2530 GOTO2300
2600 PRINT"ﾇ"RIGHT$(Y$,AX);"─────────────────────────────────────────C"
2610 IF EX=AX THEN S=S+30:GOTO2900
2620 PRINT"ﾇ"RIGHT$(Y$,AX);"
2630 GOTO2200
2900 FORI=1TO50:POKE E1,60:POKE E1,170:POKE E1,62:POKE E1,32:NEXTI
2910 GOSUB7000:GOTO4000
4000 REM***PLANET ATTACK***
4010 GOSUB8300:GOSUB8000
4100 AP=33707:MA=33727:MI=33688:FO=40:MP=32768:S1=160:E1=33757:DI=-1
4110 FORN=33728 TO 33767:POKE N,99:NEXTN
4120 F1=0:F2=0:F3=0
4200 POKE AP,32
4210 A=PEEK(ER):IF A=42 THEN AP=AP+(AP>MI)
4220 IF A=41 THEN AP=AP-(AP<MA)
4225 IF F1=1 THEN  4250
4230 IF A=48 THEN F1=1:LP=AP-FO
4250 POKE AP,1
4260 SYS826
4300 IF F2=0AND ABS(33709-AP)<3 THEN F2=1:EL=33229+INT(RND(1)*5)-2:S1=PEEK(EL)
4310 IF F2=1 THEN GOSUB 4700
4320 IF INT(RND(1)*3)=0 THEN 4440
4400 POKE E1,99:E1=E1+DI:IF E1<33729 THEN E1=33729:DI=1
4410 IF E1>33766 THEN E1=33766:DI=-1
4420 IF INT(RND(1)*10)=0 THEN DI=INT(RND(1)*2):IF DI=0 THEN DI=-1
4430 POKE E1,22
4440 IF INT(RND(1)*6)=0 THEN 4900
4450 IF F1=0 THEN 4200
4500 POKE LP,32:FORI=1TO3
4510 LP=LP-40:IF PEEK(LP)<>32 THEN F1=0:GOTO4600
4515 IF LP<MP THEN F1=0:GOTO4200
4520 NEXTI:POKELP,34:GOTO4200
4600 A=PEEK(LP):IF A>128 THEN POKE LP,120:GOTO4200
4610 IF A<>99 THEN POKELP,32:GOTO4200
4620 S=S+30:A=LP:S1=80:GOSUB9100:A=LP:S1=32:GOSUB9100:RESTORE:GOTO100
4700 POKE EL,S1
4710 EL=EL+40:IF EL>33687 THEN 4800
4720 S1=PEEK(EL):POKE EL,81:RETURN
4800 FORI=-2 TO 2:POKE EL+I,86:NEXTI
4810 IF ABS(EL-AP)<3 THEN 9050
4820 FORI=-2 TO 2:POKE EL+I,32:NEXTI
4830 F2=0:RETURN
4900 POKE E1-40,30
4910 IF AP=E1-40 THEN 9050
4920 POKE E1-40,32
4940 GOTO4450
7000 REM***DRAW RING***
7010 PRINT"ﾐ";:FORI=1TO25:PRINT"ﾇ****************************************■"
7020 NEXTI:PRINT"ﾇ";:FORI=1TO25:PRINT"
7030 NEXTI:RETURN
7500 FORI=1 TO 50:POKE AP,60:POKE AP,170:POKE AP,62:POKE AP,32:NEXTI
7520 P=P+1:IF P<>3 THEN RESTORE:GOTO100
7560 PRINT"ﾐﾘﾘﾘ"
7570 PRINT"●●●● ●●●  ●●●●  ●●●● ●●●  ●●● ●●●  ●●● "
7580 PRINT"●     ●   ●   ● ●     ●  ●  ● ● ●  ●   ●"
7590 PRINT"●     ●   ●   ● ●     ● ●   ●  ●  ● ●  ●"
7600 PRINT"●●●   ●   ●●●●  ●●●   ●●●   ●  ●●● ●   ●"
7610 PRINT"●     ●   ●●    ●     ● ●   ●  ●● ●    ●"
7620 PRINT"●     ●   ● ●   ●     ●  ●  ●  ● ● ●   ●"
7630 PRINT"●     ●   ●  ●  ●     ●  ●  ●  ●  ● ●  ●"
7640 PRINT"●    ●●●  ●   ● ●●●● ●●●  ●●● ●   ● ●●● "
7660 A$="       VERSION 1.1 BY M.FOK-82"
7670 FORI=1TOLEN(A$):PRINT"ﾇﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘ";:PRINTRIGHT$(A$,I):NEXTI
7680 RESTORE:GOSUB8350:RESTORE
7690 FORI=1 TO 25:SYS826:FORN=1TO20:NEXTN,I
7700 PRINT"ﾐﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘﾘ          PRESS 'G' FOR A NEW GAME"
7710 PRINT"              YOUR SCORE IS";S
7715 GETA$:IF A$="-" THEN END
7716 IF A$<>"G" THEN 7715
7720 RUN
8000 REM *** DRAW PLANET ***
8100 PRINT"ﾐﾘﾘﾘﾘ               ▄ ▙▔▌ ▌▙ "
8110 PRINT"             ||||  ▙▔▔|||"
8120 PRINT"            ▔▔▄\___/▄||"
8130 PRINT"           | █|| (OXO) |||▙"
8140 PRINT"          ||| ▛▔ \W/ █▔| |||"
8150 PRINT"         | █|▔▄▔ ▔██▄ ▄▄"
8160 PRINT"       ──▒.▒.▒.▒.▒.▒.▒.▒.▒.▒.▒.■──"
8180 PRINT"        ◥▒                      ◤"
8190 PRINT"         ◥▒                    ◤"
8200 PRINT"          ◥▒                  ◤"
8210 PRINT"           ◥▒                ◤"
8220 PRINT"            ▔▔▀▄▄      ▄▄▄▀▔"
8230 RETURN
8300 REM *** ROTATE RING ***
8310 FORN=826 TO 843:READ A:POKE N,A:NEXTN
8320 RETURN
8350 REM ***BOMB M.C.***
8360 FORI=826 TO 929
8370 READ A:POKE I,A:NEXTI:RETURN
9000 REM ***U.EXPLOSION***
9010 IF S1=32 THEN S1=32:S2=32:S3=32:S4=32:GOTO9030
9020 S1=77:S2=66:S3=78:S4=64
9030 FORI=1TO10:POKE A-41*I,S1:POKE A-40*I,S2:POKE A-39*I,S3
9040 POKE A-I,S4:POKE A+I,S4:FORN=1TO50:NEXTN,I:RETURN
9050 A=AP:S1=80:GOSUB9000:A=AP:S1=32:GOSUB9000:GOTO7520
9100 REM ***F.EXPLOSION***
9110 IF S1=32 THEN S1=32:S2=32:S3=32:S4=32:GOTO9130
9120 S1=77:S2=66:S3=78:S4=64
9130 FORI=1TO10:POKE A-41*I,S1:POKE A-40*I,S2:POKE A-39*I,S3:POKE A+I,S4
9140 POKE A+41*I,S1:POKE A+40*I,S2:POKE A+39*I,S3:POKE A-I,S4
9150 FORN=1TO30:NEXTN,I:RETURN
9800 REM*** BOMD DATA ***
9805 DATA 162,240,189,255,127,201,81,240,42,202,208,246,162,240,189,239
9810 DATA 128,201,81,240,43,202,208,246,162,240,189,223,129,201,81,240,44
9820 DATA 202,208,246,162,240,189,207,130,201,81,240,45,202,208,246,96,0,0,169
9830 DATA 32,157,255,127,169,81,157,39,128,76,67,3,169,32,157,239,128,169,81
9840 DATA 157,23,129,76,79,3,169,32,157,223,129,169,81,157,7,130,76,91,3,169
9850 DATA 32,157,207,130,169,81,157,247,130,76,103,3,255
9860 REM*** SCRAMBLE DATA ***
9870 DATA 172,0,3,136,140,0,3,208,56,160,2,140,0,3,162,39,172,111,131,140,72
9880 DATA 131,189,71,131,157,72,131,202,208,247,162,39,172,71,131,140,32,131
9890 DATA 189,31,131,157,32,131,202,208,247,162,39,172,31,131,140,248,130,189
9900 DATA 247,130,157,248,130,202,208,247,173,247,130,141,0,128,169,32,141,247
```

```
9910 DATA 130,162,255,189,255,127,201,87,240,31,202,208,246,162,255,189,254
9920 DATA 128,201,87,240,32,202,208,246,162,249,189,253,129,201,87,240,33,202
9930 DATA 208,246,96,0,0,0,169,32,157,255,127,169,87,157,0,128,76,143,3,169,32
9940 DATA 157,254,128,169,87,157,255,128,76,155,3,169,32,157,253,129,169,87,157
9950 DATA 254,129,76,167,3
11000 DATA162,20,172,174,129,189,154,129,157,155,129,202,208,247,140,155,129,96
```

Instructions

```
READY.

 1 REM      ***************************
 2 REM      *FIREBIRD I INSTRUCTIONS**
 3 REM      ***WORDS BY S.J.MORRIS***
 4 REM      ***PROGRAM BY M.FOK 1982**
 5 REM      ***************************
 6 REM
 7 REM
 8 REM
 9 REM
 10 Y$="▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 100 FORI=826 TO 931:READA:POKEI,A:NEXTI
 110 GOTO9500
 200 READA$:IF A$="*"THENPRINT:PRINT:PRINT"▒PRESS ANY KEY■":Y=1:GOSUB8000:PRINT"
*" READA$
 210 IF A$="@" THEN 500
 220 FORI=1TO39:PRINT"▒";RIGHT$(Y$,Y);RIGHT$(A$,I):NEXTI:Y=Y+1
 230 GOTO200
 500 PRINT"▒":END
 510 GETA$:IFA$<"1" OR A$>"8" THEN 510
 520 A=VAL(A$):LOAD"FIREBIRD I",A
 4000 DATA 162, 39, 189, 199, 129, 157, 200, 128, 202, 208, 247, 162, 39
 4010 DATA 189, 23, 129, 157, 24, 129, 202, 208, 247, 162, 39, 189, 103
 4020 DATA 129, 157, 104, 129, 202, 208, 247, 162, 39, 189, 183, 129, 157, 184
 4030 DATA 129, 202, 208, 247, 162, 0, 189, 241, 128, 157, 240, 128, 232, 224
 4040 DATA 38, 208, 245, 162, 0, 189, 65, 129, 157, 64, 129, 232, 224, 38
 4050 DATA 208, 245, 162, 0, 189, 145, 129, 157, 144, 129, 232, 224, 38, 208
 4060 DATA 245, 162, 0, 189, 225, 129, 157, 224, 129, 232, 224, 38, 208, 245
 4070 DATA 96, 0, 0, 0, 255, 0, 255, 0, 255, 0
 5000 DATA"▒","               INTRODUCTION"," "," "
 5005 DATA "  BACK IN THE 20TH MAN REALISED THE"
 5010 DATA"EARTH IS ON ITS LAST LEG-NO RESOURCES"
 5020 DATA"OR SPACES.THEN BEGAN A PHENOMENAR"
 5030 DATA"SPACE RESEARCH PROGAM BEGAN IN A LAST"
 5040 DATA"DITCH ATTEMPT TO SURVIVED.IT PROVED"
 5050 DATA"FRUITFUL,YIELDING THE EARTH-LIKE "
 5060 DATA"PLANET ZOG IN THE ORWELLIAN MAJOR "
 5070 DATA"GALAXY.HOWEVER,IT HAS ONE DRAWBACK-"
 5080 DATA"THE NEARBY PLANET PLANET OF ARGON."
 5090 DATA"THE ARGONITES RESENT THE HUMAN RACE"
 5100 DATA"FOR INTRUDING AND PATROL YOUR PLANET."
 5110 DATA"AGRONITES HAVE COMMITTED HORRIBLE "
 5120 DATA"ATROCITIES ON YOUR PEOPLE.NOW THEIR"
 5130 DATA"LEADER ZARGON HAS BUILT A STARBASE"
 5140 DATA"FROM WHICH TO ATTACK YOUR PEOPLE.",*
 5150 DATA"MANY OF OUR BEST MEN HAVE ALREADY"
 5160 DATA"BEEN EXTERMINATED IN AN EFFORT TO"
 5170 DATA"ATTACK HIS BASE.ONLY 50 CENTRONS"
 5180 DATA"AGO A WHOLE FLEET WAS DESTROYED.YOU"
 5190 DATA"ARE OUR LAST RESORT,YOU MUST FIRST "
 5200 DATA"SUCEED TO LIFT OFF IN AN UNPATROLLED"
 5210 DATA"AREA AFTER YOU DESTROYED ALL FIREBIRDS"
 5220 DATA"IN SIGHT.AFTER TAKING OFF YOU MIGHT"
 5230 DATA"GET CAUGHT IN A METEOR STORM WHICH"
 5240 DATA"OFTEN OCCURS IN ZOG REGION.ANYTIME"
 5250 DATA"DURING THE JOURNEY YOU COULD WELL"
 5260 DATA"RUN INTO AN ARGONITE FIGHTER WHICH"
 5270 DATA"YOU HAVE TO DESTROY IT BEFORE IT "
 5280 DATA"DESTROY YOU.FINALLY THEIR STARBASE"
 5290 DATA"IS WELL DEFENDED.GOOD LUCK"
 5295 DATA" ","                    MAJOR GENERAL"
 5300 DATA"                       S.J MORRIS" *
 5310 DATA"MISSION-1"
 5320 DATA"YOUR AIM IS TO DESTROY ANY ATTACKING"
 5330 DATA"FIREBIRD(S).BUT WATCH OUT !FIRBIRDS"
 5340 DATA"ARE DEADLY SINCE YOU CAN ONLY DESTROY"
 5350 DATA"IT BY HITTING ITS SOFT STOMACH"
 5360 DATA"ON THE CONTROL BOARD YOU:"," "
 5370 DATA"PRESS '4' TO MOVE LEFT"
 5380 DATA"PRESS '6' TO MOVE RIGHT"
 5390 DATA"PRESS 'A' TO FIRE",*
 5400 DATA"MISSION-2"
 5410 DATA"YOU MUST AVOID THE ASTEROIDS WHICH "
 5420 DATA"APPEAR FROM THE TOP OF THE SCREEN "
 5430 DATA"AND MOVE LOWER AND LOWER FOR A SET"
 5440 DATA"TIME(TIME IS DISPLAY UNDER THE LAND-"
 5450 DATA"SCAPE AFTER THE FIRST ASTEROID REACH"
 5460 DATA"THE BOTTOM OF THE SCREEN)."
 5470 DATA"ON THE CONYROL BOARD YOU:"," "
 5480 DATA"PRESS '0' TO MOVE UP"
 5490 DATA"PRESS 'A' TO MOVE DOWN",*
 5500 DATA"MISSION-3"
 5510 DATA"YOU MET AN ARGON FIGHTER WHICH WILL"
 5520 DATA"ATTEMPT TO ZAP YOU SO YOU MUST ZAP"
 5530 DATA"IT BEFORE IT SUCEED.HOWEVER YOU MAY"
 5540 DATA"ONLY FIRE WHEN YOUR ENERGY(LEFT ONE"
```

```
 5350 DATA"IT BY HITTING ITS SOFT STOMACH"
 5360 DATA"ON THE CONTROL BOARD YOU:"," "
 5370 DATA"PRESS '4' TO MOVE LEFT"
 5380 DATA"PRESS '6' TO MOVE RIGHT"
 5390 DATA"PRESS 'A' TO FIRE",*
 5400 DATA"MISSION-2"
 5410 DATA"YOU MUST AVOID THE ASTEROIDS WHICH "
 5420 DATA"APPEAR FROM THE TOP OF THE SCREEN "
 5430 DATA"AND MOVE LOWER AND LOWER FOR A SET"
 5440 DATA"TIME(TIME IS DISPLAY UNDER THE LAND-"
 5450 DATA"SCAPE AFTER THE FIRST ASTEROID REACH"
 5460 DATA"THE BOTTOM OF THE SCREEN)."
 5470 DATA"ON THE CONYROL BOARD YOU:"," "
 5480 DATA"PRESS '0' TO MOVE UP"
 5490.DATA"PRESS 'A' TO MOVE DOWN",*
 5500 DATA"MISSION-3"
 5510 DATA"YOU MET AN ARGON FIGHTER WHICH WILL"
 5520 DATA"ATTEMPT TO ZAP YOU SO YOU MUST ZAP"
 5530 DATA"IT BEFORE IT SUCEED.HOWEVER YOU MAY"
```

```
5540 DATA"ONLY FIRE WHEN YOUR ENERGY(LEFT ONE"
5550 DATA"IS FOR AGRONITE FIGHTER WHILE THE"
5560 DATA"RIGHT ONE IS YOURS)IS TEN."
5570 DATA"ON THE CONTROL BOARD YOU:"," "
5580 DATA"PRESS 'Q' TO MOVE UP"
5590 DATA"PRESS 'A' TO MOVE DOWN"
5600 DATA"PRESS '4' TO FIRE",*
5610 DATA"MISSION-4"
5620 DATA"YOU MUST SHOOT THROUGH THE WELL "
5630 DATA"DEFENDED STARBASE AND KILL THEIR"
5640 DATA"LEADER ZARGON.(SINCE NO-ONE HAS GOT"
5650 DATA"THAT FAR YET,THEREFORE THE INFORMATION"
5660 DATA"MIGHT BE WRONG"
5670 DATA"ON THE CONTROL BOARD YOU:"," "
5680 DATA"PRESS '4' TO MOVE LEFT"
5690 DATA"PRESS '6' TO MOVE RIGHT"
5700 DATA"PRESS 'A' TO FIRE"
5710 DATA" "," ","SPECIAL COMMAND:"
5720 DATA"PRESS 'C' AT ANY MISSION(EXCEPT LAST"
5730 DATA") FOR CHEAT.",*
5740 DATA"                    SYMBOLS"
5750 DATA"'A'...........................FIREBIRD"
5760 DATA" O...........................ASTEROID"
5770 DATA" >...................ARGONITE FIGHTER"
5780 DATA" \___/    "
5790 DATA"(OXO)    "
5800 DATA" \W/ ...........................ZARGON"
5810 DATA"  ‾      ",*,@
8000 GETB$:IF B$="" THEN 8000
8010 RETURN
9500 PRINT"▮▮▮▮▮"
9510 A$(1)=" ●●●● ●●●  ●●●  ●●●● ●●●  ●●● ●●●  ●●●  "
9520 A$(2)=" ●      ●   ●  ● ●     ●  ●  ●  ●  ●  ●  ● "
9530 A$(3)=" ●      ●   ●  ● ●     ● ●   ●  ●  ● ●   ● "
9540 A$(4)=" ●●●    ●   ●●●  ●●●   ●●●   ●  ●●●  ●   ● "
9550 A$(5)=" ●      ●   ●●   ●     ● ●   ●  ●●   ●   ● "
9560 A$(6)=" ●      ●   ● ●  ●     ●  ●  ●  ● ●  ●   ● "
9570 A$(7)=" ●      ●   ●  ● ●     ●  ●  ●  ●  ● ●   ● "
9580 A$(8)=" ●     ●●●  ●  ● ●●●● ●●●  ●●● ●  ● ●●●  "
9590 A$=" INSTRUCTION FOR FIREBIRD I BY M.FOK-82"
9600 FORA=1TO37
9605 PRINT"▮▮▮▮▮",
9610 FORB=1TO7STEP2
9620 PRINT"▮";MID$(A$(B),(38-A),1)
9630 PRINT"▮▮▮";MID$(A$(B+1),A,1);"▮"
9640 NEXTB
9650 SYS826
9660 NEXTA
9670 PRINT"▮▮▮▮▮▮";:FORI=1TO8:PRINT"▮";LEFT$(A$(I),39);:NEXTI
9680 FORI=1TO39:PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮";RIGHT$(A$,I):NEXTI
9690 FORI=1TO500:NEXTI:GOTO200
```

Atari Colour Selector

by Derek Lees

The colour graphics on the Atari 400/800 allow for a vast number of screen combinations. In Atari Basic text mode 0, 16 foreground colours, with eight possible luminosities can be combined with a similar selection of border colours. Added to those, eight text brightnesses can be chosen.

Some of these combinations are gaudy and unpleasant. Others are very pleasing. As there are so many, it is often laborious to select the combination which looks best. This program will help in doing just that. It allows you to step through every possible combination using the space bar and escape key. As each combination changes, the SETCOLOR parameters and relevant addresses are displayed and can be 'frozen' for use in later programs. It runs straightforwardly in 4k. Underlined characters in this listing should be in inverse video. The vertical line is a shifted down-arrow and the curly bracket is escape-control-delete (clear screen in program mode).

```
1 REM .  GRAPHICS 0 COLOUR UTILITY
2 REM .  D.J.LEES, AUGUST 1982
3 REM .  ---------------------------
4 REM .  USE BREAK KEY TO TERMINATE
5 REM .  ---------------------------
10 INTRO=800:FGND=200:BDR=300:TEXT=400:I
NFO=600:BLINK=700:N16=16
15 GOSUB INTRO
20 GRAPHICS 0:POKE 752,1:POKE 712,148:PO
KE 82,0:POKE 83,40:OPEN #1,4,0,"K:"
23 ? :? " Cntl-Q, Cntl-R 37 times, Cntl-E
"
25 ? " |Number Colour    Number Colour
   |"
26 ? " |
   |"
30 ? " |  0    GREY        08   LIGHT-BL
UE   |"
35 ? " |  1    GOLD        09   BLUE-GRE
EN   |"
40 ? " |  2    ORANGE      10   AQUA
   |"
45 ? " |  3    RED         11   GREEN-BL
UE   |"
50 ? " |  4    PINK        12   GREEN
   |"
```

PROGRAMS

```
55 ? " | 5    VIOLET        13   YELLOW-G
REEN|"
60 ? " | 6    BLUE-PURPLE 14   ORANGE-G
REEN|"
65 ? " | 7    BLUE          15   ORANGE
     |"
70 ? "  Cntl-Z, Cntl-R 37 times, Cntl-C
     "
75 GOSUB BLINK
80 POSITION 1,16:? "SE.1,0,":POSITION 14
,16:? "SE.2,":POSITION 21,16:? ",":POSIT
ION 27,16:? "SE.4,":POSITION 34,16:? ","
85 POSITION 1,18:? "ADDR.709=":POSITION
14,18:? "ADDR.710=":POSITION 27,18:? "AD
DR.712="
90 C1=INT(PEEK(709)/N16):L1=PEEK(709)-C1
*N16
95 C2=INT(PEEK(710)/N16):L2=PEEK(710)-C2
*N16
100 C4=INT(PEEK(712)/N16):L4=PEEK(712)-C
4*N16
110 GOSUB INFO
125 GET #1,CHOICE
130 IF CHOICE=70 THEN 150
135 IF CHOICE=66 THEN 155
140 IF CHOICE=84 THEN 160
145 GOTO 125
150 GOSUB FGND:GOSUB BLINK:GOTO 125
155 GOSUB BDR:GOSUB BLINK:GOTO 125
160 GOSUB TEXT:GOSUB BLINK:GOTO 125
190 REM ***FOREGROUND COLOUR***
200 POSITION 2,20:? "Hit SPACE to alter
FOREGROUND Colour"
205 POSITION 2,21:? "Hit ESCAPE Key to f
reeze parameters "
210 GET #1,K:IF K=32 THEN C2=C2+1:GOTO 2
25
215 IF K=27 THEN POSITION 21,20:? "the B
RIGHTNESS    ":GOTO 240
220 GOTO 210
225 IF C2>15 THEN C2=0
230 SETCOLOR 2,C2,L2:GOSUB INFO:GOTO 210
240 GET #1,K:IF K=32 THEN L2=L2+2:GOTO 2
55
245 IF K=27 THEN 270
250 GOTO 240
255 IF L2>14 THEN L2=0
260 SETCOLOR 2,C2,L2:GOSUB 600:GOTO 240
270 POSITION 2,20:? CHR$(156):POSITION 2
,20:? CHR$(156)
275 RETURN
290 REM ***BORDER COLOUR***
300 POSITION 4,20:? "Hit SPACE to alter
BORDER Colour"
305 POSITION 4,21:? "Hit ESCAPE to freez
e parameters "
310 GET #1,K:IF K=32 THEN C4=C4+1:GOTO 3
25
315 IF K=27 THEN POSITION 22,20:? " BRIG
HTNESS    ":GOTO 340
320 GOTO 310
325 IF C4>15 THEN C4=0
330 SETCOLOR 4,C4,L4:GOSUB INFO:GOTO 310
340 GET #1,K:IF K=32 THEN L4=L4+2:GOTO 3
55
345 IF K=27 THEN 370
350 GOTO 340
355 IF L4>14 THEN L4=0
360 SETCOLOR 4,C4,L4:GOSUB INFO:GOTO 340
370 POSITION 4,20:? CHR$(156):POSITION 4
,20:? CHR$(156)
375 RETURN
390 REM ***TEXT BRIGHTNESS***
400 POSITION 3,20:? "Hit SPACE to alter
TEXT brightness"
405 POSITION 3,21:? "Hit ESCAPE to freez
e parameters    "
```

PROGRAMS

```
410 GET #1,K:IF K=32 THEN L1=L1+2:GOTO 4
25
415 IF K=27 THEN 440
420 GOTO 410
425 IF L1>14 THEN L1=0
430 SETCOLOR 1,0,L1:GOSUB INFO:GOTO 410
440 POSITION 4,20:? CHR$(156):POSITION 4
,20:? CHR$(156)
445 RETURN
590 REM ***DATA UPDATE***
600 POSITION 9,N16:? "  ":POSITION 9,N16
:? L1:POSITION 19,N16:? "  ":POSITION 19
,N16:? C2:POSITION 22,N16:? "  "
610 POSITION 22,N16:? L2:POSITION 32,N16
:? "  ":POSITION 32,N16:? C4:POSITION 35
,N16:? "  ":POSITION 35,N16:? L4
615 POSITION 10,18:? "   ":POSITION 10,1
8:? PEEK(709):POSITION 23,18:? "   ":POS
ITION 23,18:? PEEK(710)
620 POSITION 36,18:? "   ":POSITION 36,1
8:? PEEK(712)
630 RETURN
690 REM ***BLINK INVERSE VIDEO***
700 POSITION 1,14:? "SELECT (T)ext,(F)or
eground, or(B)order"
710 NORM=PEEK(755)
715 FOR W=0 TO 5
720 FOR DELAY=0 TO 10:NEXT DELAY
725 POKE 755,0:SOUND 1,50,10,10
730 FOR DELAY=0 TO 10:NEXT DELAY
735 POKE 755,2:SOUND 1,0,0,0
740 NEXT W:POKE 755,NORM
745 RETURN
790 REM ***INTRODUCTION***
800 ? "}":DIM A$(18):POKE 752,1
805 A$="DJL COLOUR UTILITY"
810 FOR J=0 TO 22:POSITION J,J
815 ? A$:POKE 710,J*8:POKE 712,J*4:SOUND
 0,J,10,10:FOR W=0 TO 5:NEXT W:NEXT J
820 SOUND 0,0,0,0:RETURN
```

CHIP CHAT

The 'state of the art' in micro technology in the USA seems to be getting itself into more and more of a state. For instance, an ad spotted recently in an American micro mag boasts 'Now you can use computer technology to search the Bible on any subject.' You see, some redneck fundamentalist computer freaks have put the King James Bible onto disk and are supplying it with a piece of look-up software called (believe it or not) The Word processor. Going to the other extreme, an outfit calling itself 'The Softcore Software Co.' is advertising a game (at least, we *think* it's a game) called 'Sewers of Moscow' – with sound! . . . When 80 column cards were advertised recently for $130 there would, one expects, be a spate of calls to the distributor (no address was supplied in the advertisement). Several nights after distribution of subscribers copies of that issue, *APC's* publisher was working late and received a call: "What?" (our customary late-night greeting) "Is this Howard Productions?" "Yep " "Do you publish a magazine called Australian Personal Computer?" "Yes." "For the last couple of evenings I've been getting dozens of calls from people wanting to know about a computer board that was advertised in your October issue and somehow my number was put in the ad." "Oh dear . . . " "But that'll probably stop because your November issue should be out pretty soon." "Well, actually, this issue hasn't . . er . . hit the streets yet." Words couldn't properly describe the sound emitted by this unfortunate gentleman at the thought of jumping up to night calls ad nauseam for the next few weeks. Our sincere apologies and a free twelve month subscription to *APC* sir, if you are reading this . . . A report was dropped onto our desk not so long ago, the first item of which was a paragraph about computerised lavatory flushes of all things. Apparently one of Reckitt & Colman's R&D labs has (quote) 'an international selection of lavatories flushing 24 hours a day at the command of a computer'. These flushes are performed by pneumatic controls which 'compress months of normal household use . . . into weeks or even days'. No suggestion is given as to why exactly there is a need to wear out all these cosmopolitan khazis, but it could be that Reckitt & Colman will produce a 'superlav' down which micro industry rubbish might be forever flushed at the command of a Cray 1 . . . For worried users of the VIC-20, we've heard from the top that, despite rumours to the contrary, the VIC will not be dropped in the wake of the Commodore 64. So if anyone tells you that the VIC-20 won't be on sale for much longer, you know what to do – tell them to VIC off . . .