NEW ZEALAND'S LEADING COMPUTER MAGAZINE

BITS & BYTES

ISN 0111-9826 **August 1986 $2.25**

EXCLUSIVE New Zealand Review

ATARI

The 520ST
– A test of power without the price

Software: **Mac's Excel spreadsheet,**
Hardware: **Atari 520ST, Panasonic JB 3300 Portable, Microbee-Mitac Portable.**
Features: **The GST giddy-up. D-I-Y programme protection. Quest for the Euro-Micro. Review of NZ software industry.**
Columns include Logo organisation, Pascal, Q&A, and a user group directory.

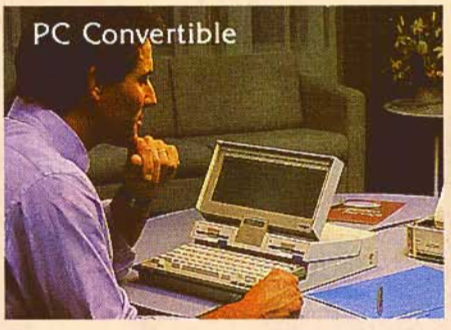
THE CHOICE IS YOURS

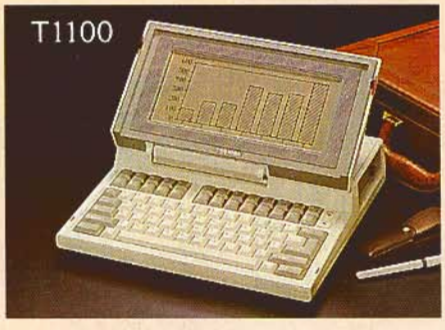
Portable Personal Computing is now a powerful reality and we can offer you the best.

Data General
A Generation Ahead.

IBM

TOSHIBA
TOSHIBA CORPORATION





MODEL	One-2	PC Convertible	T1100
MEMORY			
Minimum/Maximum	256KB/640KB	256KB/512KB	256KB/512KB
DISKETTE DRIVES			
Internal	1 or 2 3.5"	2 3.5"	1 3.5"
External	1 5.25"	1 3.5"	1 3.5" or 5.25"
HARD DISK	10MB internal option	Not Available	Not Available
BATTERY LIFE	4-7 Hours	6-10 Hours	6-8 Hours
PORTS			
Standard	1 Serial/1 Parallel	1 Parallel	1 Parallel
Optional	1 Serial	Serial/Parallel	1 Serial
DISPLAY	LCD or E L 640 x 200	LCD 640 x 200	LCD 640 x 200
WEIGHT	11.2lbs	12.2lbs	9lbs
BUILT IN SOFTWARE			
Basic A	No	Yes	No
System Applications (Notepad, Calculator Calendar etc.)	No	Yes	No
OPTIONS			
Floating point	Yes	No	No
Colour Output	Yes	Yes	Yes
Carrying Case	Yes	Yes	Yes
Automobile Adapter	No	Yes	No
PRICE	$4,847	$4,900 (est.)	$3,500

Each has its own special advantages. One will fill your requirements. We can demonstrate, supply and service them all.

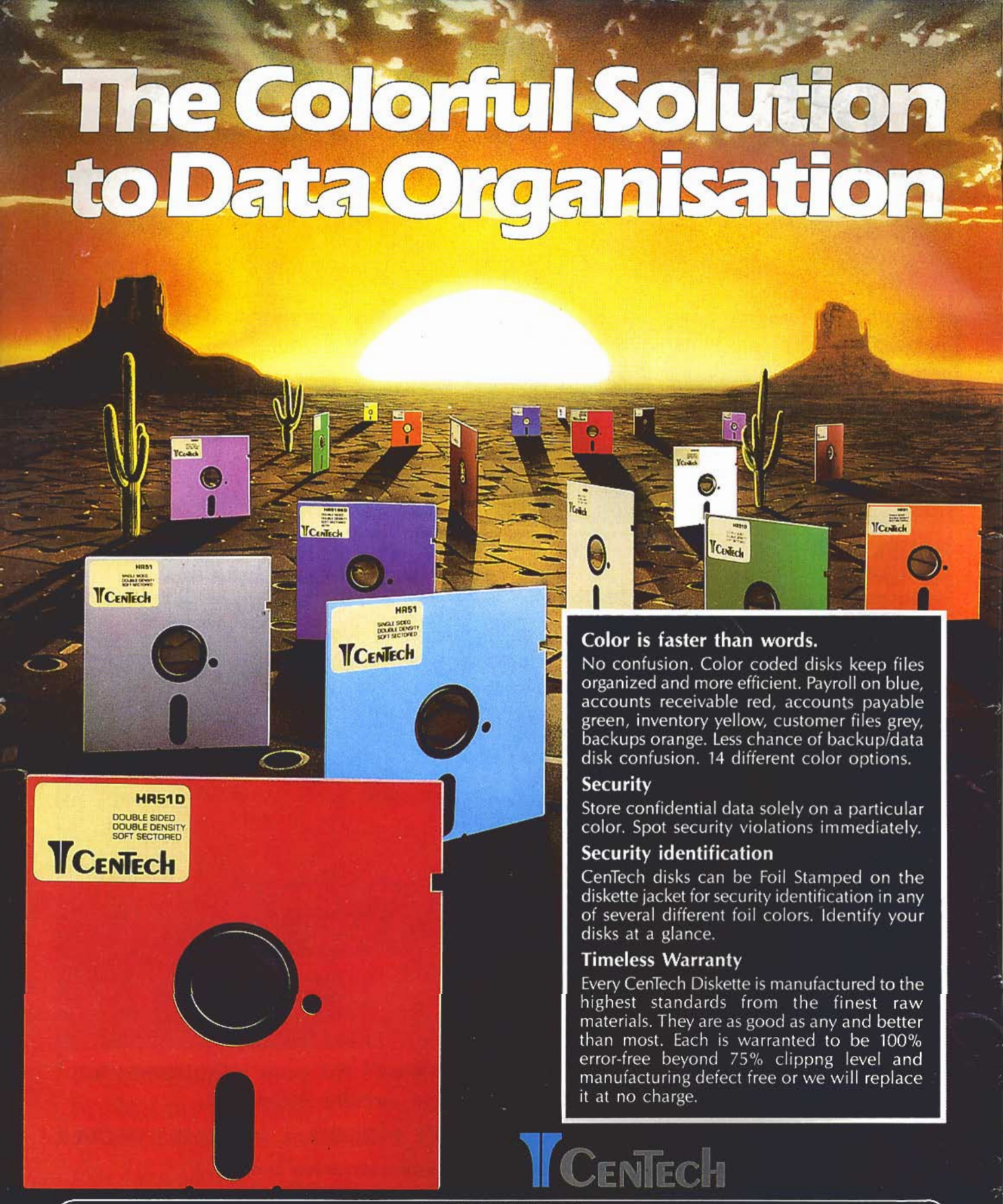
PORTABLE PERSONAL COMPUTING FROM THE PERSONAL PORTABLE PEOPLE.

For more information, a brochure or demonstration contact.

SOUTHMARK
COMPUTERS LIMITED
2 NGAIRE AVENUE, AUCKLAND 3. N.Z. BOX 74-128, TELEX NZ 60101. PHONE 504-609

TCC 0130

The Colorful Solution to Data Organisation



Color is faster than words.

No confusion. Color coded disks keep files organized and more efficient. Payroll on blue, accounts receivable red, accounts payable green, inventory yellow, customer files grey, backups orange. Less chance of backup/data disk confusion. 14 different color options.

Security

Store confidential data solely on a particular color. Spot security violations immediately.

Security identification

CenTech disks can be Foil Stamped on the diskette jacket for security identification in any of several different foil colors. Identify your disks at a glance.

Timeless Warranty

Every CenTech Diskette is manufactured to the highest standards from the finest raw materials. They are as good as any and better than most. Each is warranted to be 100% error-free beyond 75% clippng level and manufacturing defect free or we will replace it at no charge.

CENTECH

SOLE NEW ZEALAND DISTRIBUTORS.

SOUTHMARK COMPUTERS LIMITED

2 NGAIRE AVENUE, AUCKLAND 3. N.Z. BOX 74-128, TELEX NZ 60101. PHONE 504-609

Dealer Enquiries Welcome.

TCC 0126

1986 CHRISTCHURCH COMPUTER SHOW

CHRISTCHURCH TOWN HALL

Monday, 8 September 10am-8pm
Tuesday, 9 September 10am-8pm
Wednesday, 10 September 10am-3pm

See the latest in computer technology and computer communications!

ADMISSION: Adults $3.00
Children (accompanied) $1.00
Business pass holders $2.00

AUCTION
of computers and related products
Tuesday, 9 September
7pm at the Show.

SEMINARS: A range of seminars on major areas of interest will be presented by professionals. For business passes and seminar bookings complete coupon and post.

☐ Please send me .... BUSINESS PASSES
I wish to attend the following seminars (please tick boxes):

- Buying your first Business Computer ..... ☐
- Buying your first Home Computer ..... ☐
- Guide to computer-aided design ..... ☐
- Computers in education-achieving success with software ..... ☐
- Guide to GST and computing ..... ☐
- Productive use of spreadsheet software ..... ☐
- Using wordprocessing software ..... ☐
- Productive use of database packages ..... ☐
- Local area networking ..... ☐
- Integrating a company's communications with computers ..... ☐
- Accessing information using your computer ..... ☐

There is an admission charge of $6,00 for one seminar or $5.00 each for two or more seminars.
I enclose a cheque for $ ....., for tickets to the seminars indicated in the boxes above.

Name: .....
Position: .....
Company: .....
Address: .....
.....

Return To:
Christchurch Computer Show
P.O. Box 827
Christchurch
Phone (03) 66-566

We have hundreds of products available

For the best prices give us a call

SYSTEM UNITS	Retail Price
IPC-XT 256K/2x360K Floppies/Par/Monitor	$2,275
IPC-XT 640K/2x360K Floppies/Par/Ser/Game/Mon/Clock	$2,575
IPC-XT 640K/2x360K Fl/Par/Ser/Game/Mon/Clock/10Mb	$4,150
IPC-XT 640K/2x360K Fl/Par/Ser/Game/Mon/Clock/20Mb	$4,450
IPC-AT 20Mb Hard Disk/1.2Mb Floppy/Mon	$6,850
MAGNUM TURBO 4.77/8.0MHz/640K/1x360K Fl/MS-DOS	$2,290
MAGNUM TURBO 4.77/8.0MHz/640K/2x360K Fl/MS-DOS	$2,590
MAGNUM TURBO 4.77/8.0MHz/640K/1x360K Fl/10Mb/MS-DOS	$4,290
MAGNUM TURBO 4.77/8.0MHz/640K/1x360K Fl/20Mb/MS-DOS	$4,790
MAGNUM TURBO 4.77/8.0MHz/640K/1x360K Fl/40Mb/MS-DOS	$7,190

MONITORS AND GRAPHICS CARDS	
MAGNUM Green Monitor	$295
Thomson Mono Monitor (Composite)	$395
White Knight Hi-Res (TTL) Mono Monitor	$420
MAGNUM Flat Green Screen	$450
MAGNUM Flat Amber Screen	$495
Thomson RGB Colour Monitor	$1,070
White Knight Hi-Res Colour (640x350) plus controller	$2,345
White Knight Colour Graphics Card (2 RCA Ports)	$150
White Knight Mono Graphics Card	$195
MAGNUM Colour Graphics Card	$350
MAGNUM Mono Graphics Plus Printer Card	$395

HARD DISK SUBSYSTEMS - PC/XT	
20 Mb HH Seagate ST225 65ms	$1,995
31 Mb FH Seagate 40ms Self Park	$2,895
42 Mb FH Seagate 40ms Self Park, Drive Splitting BIOS, Plated	$3,935
71 Mb FH Toshiba 22ms Self Park, Drive Splitting BIOS, Plated	$6,400
160 Mb FH Maxtor 28ms Self Park, Drive Splitting BIOS, Plated	$16,600
For all external Half Height Drives add	$760
For all external Full Height Drives add	$950

All hard disk subsystems inlude: Drive, Controller, Cables, Manuals, Hard Disk Booting BIOS ROM, Firmware Hard Disk Preformat program, Park Utility on Oxide media drives and (for external models) Case and Power Supply.
ALL DISK SUBSYSTEMS FITTED FREE OF CHARGE.

STREAMING TAPE SUBSYSTEMS	
10 Mbyte Irwin Tape Backup (Internal) PC/XT/AT	$1,650
24 MByte Irwin Tape Backup (Internal) AT only	$1,825
24 MByte Irwin Tape Backup (External) At only	$2,600
10 MByte Irwin Tape Backup (External) PC/XT/AT	$2,000
60 MByte Archive Tape Backup (Internal) PC/XT/AT	$3,225
60 MByte Archive Tape Backup (External) PC/XT/AT	$3,575

All Streaming Tape Subsystems include: Cables, Manuals, Operating Software and (for external models) Case and Power Supply.

REMARKABLE DISKETTES	
DSDD Premium Quality 5¼" (Suitable for most PC's)	each $4.50
QUAD Premium Quality 5¼" (To 1.2Mb)	each $8.50
SSDD Premium Quality 3½" (To 400K)	each $7.00
DSDD Premium Quality 3½" (To 850K)	each $8.50

Remarkable 5¼" Diskettes are US-made, come boxed with Tyvek Sleeves, Labels, Write Protect Tabs, exceed all ANSI standards. Each track certified 100% errror free, non-woven dust free internal liners, Replacement Guarantee.

GENERIC DISKETTES	
DDSD 5¼"	$3.50

Japanese made, Boxed, Write Protect, Labels, Replacement Guarantee.

SOFTWARE	Retail Price
Word Perfect 4.1	$895
dBase III	$1,095
Clipper	$950
Lotus 123 Ver. 2.0	$995
Open Access	$1,345
Borland Sidekick Ver. 1.5 (Unprotected)	$175
Borland Sidekick Ver. 1.5 (Protected)	$125
Borland Turbo Editor Toolbox	$145
Borland Turbo Database Toolbox	$139
Borland Turbo Gameworks	$145
Borland Turbo Graphix Toolbox	$145
Borland Turbo Tutor	$99
Borland Lightning	$225
Borland Turbo Pascal Ver. 3.0	$150
Borland Turbo Pascal with 8087 Ver 3.0	$225
Borland Reflex	$250
Borland Prolog - NEW -	$250
Microsoft Windows	$250
Microsoft Pascal	$625
Fastback	$295
Keyworks	$195
Norton Utilities Ver. 3.1	$245
Crosstalk XVI	$335
Profax (N.Z. made) - Top software - GL/AR/AP/PO/Inv - NEW -	$1,495
Dac Easy (US Ver.) GL/AR/AP/PO/Billing/Inv./Forecasting	$250
Timeline Ver. 2.0 Project Scheduling Easy to use/Top Performer	$845
Before You Leap - Cocomo-based Software Dev. Cost Model	$995
Generic CAD plus Dot Plot- NEW -	$495
Dan Bricklins Demo Program - NEW -	$495
Prodesign II - NEW -	$895

PC-SIG LIBRARY SOFTWARE	
Catalogue - Hundreds of titles that will save you money	$49.95
PC-Write Ver. 2.6 - Top Word Processor	$29.95
PC-Deskmates - A super alternative to Sidekick	$29.95
PC-Outline - Similar to "Ready" and "Thinktank"	$29.95
PC-Calc - General Purpose Spreadsheet	$29.95
PC-General Ledger - Complete audit trail available	$29.95
PC-DOS Help - Online Help for DOS commands	$29.95
PC-Draw #1 - Create and edit graphics screens	$29.95
123 Worksheets #2 - Loans/Chequebook/Stocks/Printer Control	$29.95
123 Macros - Ratio Analyser, Loan Payoff Calculator & more	$29.95
Touchtype - Typing Tutorial, Turn you PC into a typewriter	$29.95
Ultra Utilities - Useful Tools for every PC Owner	$29.95
Patches - Allows you to make backups of selected software	$29.95
Diskette Utilities - Includes RAMDISK Utilities	$29.95
Best Utilities - A collection of the best in the library	$29.95
Copy Protection/Unprotection Utilities	$29.95
Best Games - A collection of the best in the library	$29.95
Spacewar - Excellent asteroid type arcade game	$29.95
Arcade Games - Colourful games - great on JX	$29.95
Game Mixture - 12 games on Disk	$29.95
Miscellaneous Games - Educational, video and adventure	$29.95

ANY TWO LIBRARY DISKS ONLY $19.95 EACH. BULK DISCOUNTS AVAILABLE.

UPGRADES, CARDS, CHIPS, PRINTERS, AT HARD DISKS..... CALL

All prices subject to change without notice.

All prices include Sales Tax.
Computers are 100% MS-DOS Compatible with 12 Month Guarantee.
Diskettes carry No-Questions-Asked Replacement Guarantee.
Printers carry 90 Day Guarantee.
DEALER ENQUIRIES WELCOME.

FREIGHT AND PACKAGING --
5% for orders under $100. $7.50 for orders over $100.
Systems, Printers, Monitors $50 per package.

To order use this form:

Please supply the following: _____

_____

NAME: _____

COMPANY: _____

ADDRESS: _____

CITY: _____

PHONE No: _____

Cheque Enclosed ☐ Please charge my credit card ☐
Visa ☐ American Express ☐ Diners Club ☐ Bankcard ☐

Card Number: _____

Expiry Date: _____

Signature: _____

All orders must be accompanied by a cheque, money order, postal note or credit card number.
All orders must be signed. Your money will not be banked until after your order is despatched.

For even faster service call (024) 774 464

REMARKABLE ENTERPRISES LTD

(Computer Division)
7 Crawford Street, Dunedin
Freepost 191, P.O. Box 1415, Dunedin
Telephone (024) 774-464, FAX (024) 772-948
Telex NZ 5406 REMPAR

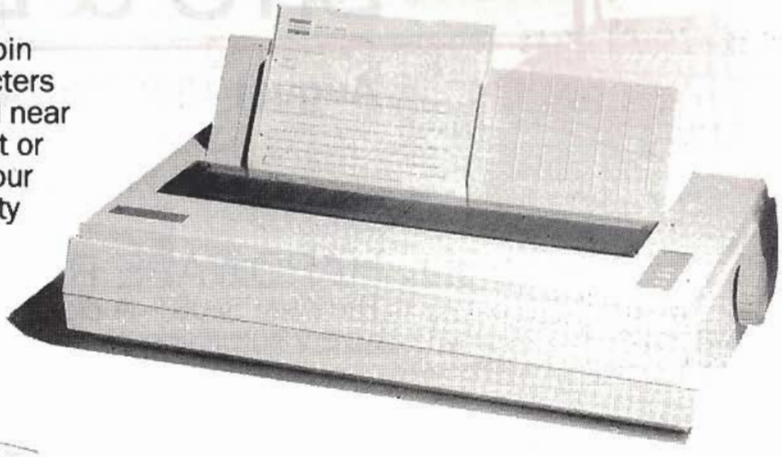
For further information and a complete price list of MS-DOS Computers, Tape Units, Diskettes, Printers and Software, call or write today.

BITS & BYTES

August 1986 Vol. 4, no. 11 ISN 0111-9826

BITS AND BYTES magazine is published monthly (excepting January) by Bits and Bytes Ltd, Denby House, third floor, 156 Parnell Road. PO Box 9870, Auckland 1. Phone 796-776, 796-775. **EDITORIAL:** managing editor, Gaie Ellis; editor, Steven Searle; Christchurch reporter, Dion Crooks, 66-566. **ADVERTISING:** Auckland – David Meyer, PO Box 9870, 796-775; Wellington – Vicki Eckford, 753-207; Christchurch – Jocelyn Howard, PO Box 827, 66-566. **SUBSCRIPTIONS:** First floor, Oxford Court, 222 Oxford Tce, Christchurch, PO Box 827, Phone 66-566. Manager, Mavis Shirtcliffe. **SUBSCRIPTION RATE:** $18 for 11 issues, school pupils rate $16. Overseas subs are $27/year surface mail, and airmail rates of $59 (Australia, South Pacific), $86 (North America and Asia), and $108 (Europe, South America, Middle East). **BOOK CLUB:** manager, Dion Crooks, at above Christchurch address, 66-566. **DISTRIBUTION INQUIRIES:** bookshops to Gordon and Gotch Ltd, computer stores to publisher. **PRODUCTION:** graphic designer, Roger Guise; Typesetter, Monoset; printer, Rodney and Waitemata Times.

This is the Fujitsu DX 2100 9 pin printer which runs at 44 characters per second, produces draft and near letter quality print, can be sheet or continuous fed, will print in colour and is renowned for its reliability and quiet operation.



This is the Fujitsu M3043 band printer which runs at 1200 lines per minute and produces the best quality print (even through 6 part forms). The M3043 provides the speed and unattended reliability to produce large quantities of printed material economically.

The **M3043** is the biggest printer in the Fujitsu range — the **DX2100** the smallest. Between the two are a whole stable of printers to match your requirements. Dot matrix 9 and 24 pin, Daisywheel, band printers, sheet fed, continuous fed, colour options. Fujitsu can provide them all.

Phone Ian Young of Thorn EMI and talk Fujitsu printers. You'll be pleased with the choice, the reliability and the service.

FUJITSU

THORN EMI

THORN EMI TECHNOLOGY N.Z. LTD

190 Great South Road, Remuera.
P.O. Box 74-018, Market Road, Auckland.
Telephone (09) 545-144

TCC 13012

'Privacy Authority' mooted as safeguard

The NZ Computer Society is promoting the concept of legislation to protect individuals from abuse of personal information held by organisations.

The society proposes legislation establishing a Privacy Authority, which would enforce rules limiting the use of such stored information.

In its "position paper" promoting the concept, such an authority is vested with roles of registering commercial or institutional holders of personal information, of monitoring useage of such information, and researching and advising on development of legal protections.

The society says its concern is prompted by the increasing abilities of organisations to compile and use personal information on staff and customers, or potential customers, by means of increasingly sophisticated computer systems.

The society's guiding principles in protecting individuals from misuse of this information are:

- arousing public knowledge of individual rights to be informed by information "custodians" of where and why personal data is being held,
- promoting the right of individuals to have access to information about themselves, and the right to review and modify such data,
- ensuring personal data is kept relevent to holder's business,
- ensuring use of data for only legitimate purposes,
- ensuring data custodians be accountable for safe-keeping of data on individuals.

The society claims such principles are already promoted amongst its members because of "widespread unease" in two areas of personal data files – records held by public order and national security agencies, and by credit agencies.

It wants any individual or business to have the right to access data compiled on themselves, and to be able to contest the accuracy or relevence of such information.

Paperback independent

Paperback Software (NZ) Ltd, which has been operating under the wing of Arcom Pacific (now Ashton Tate New Zealand) for the past 18 months has now taken an independent stance in selling the cheaper range of software.

While Ashton Tate will pursue its policy of promoting the more expensive range of products such as Dbase III Plus, Paperback will be promoting the cheaper, compatible range of databases, spreadsheets and word processing products from the Osborne founded US company.

Poor marketing of local software

A DSIR survey of NZ software companies has revealed significant loss of potential sales through their own lack of marketing skills.

The survey of 250 organisations, compiled by Martin Kaiser, of the DSIR's technology transfer group, highlights weaknesses and strengths of the local software development industry, and records sales and export performances.

Mr Kaiser claims the report has at least stimulated some discussion between trade organisations and software companies to effect more strategic approaches to marketing.

"The NZ Software Industry" survey also looks at the role of software development as an industry and recommends various responsibilities and actions of companies, trade groups and government.

The survey's three resultant papers are available from government print bookshops.

- See survey feature

Datatronics for Alloy

Datatronics, a member of the Quaser group of companies, has been appointed the New Zealand distributor for Alloy computer products.

Alloy has been represented in Australia for the past two and a half years by the parent company in the US and has now been established there as a weholly owned subsidiary with the aim of consolidating in New Zealand before tackling the Asian market next year.

Alloy has a comprehensive range of boards, tapes and drives designed to suit the multi-user environment. Datatronics is also promoting the Persyst range of communications products and the Microvitek monitors.

Scott Green, says it is the company's intention to establish a select few dealers throughout New Zealand.

AMIGA™ is here . . . .

and ASHBY COMPUTER CENTRE LTD is AMIGA™

HARDWARE AND SOFTWARE IN STOCK NOW!

SOFTWARE SELECTION:

Business Application...

AMIGA TEXTCRAFT	
Powerful wordprocessor	$149.95
VIP PROFESSIONAL	
Spreadsheet/Database 256 x 8192 cells	now only $499.00
LATTICE UNICALC	
Fully featured spreadsheet	$249.50
MAXIDESK	
Sidekick type deskset	$249.50
MAXICOM	
Communications package	$249.50
DE LUXE PRINT	$279.00
SCRIBBLES W/PROC	$225.00
GIZMOS PRODUCTIVITY SET	$120

Graphics Applications

GRAPHICRAFT	
Powerful entry level graphics	$149.50
DE LUXE PAINT	
Advanced painting program	now only $299.00
AEGIS ANIMATOR	
Metamorphic animation package	$425.50

Lattice Programming Aids

TEXT UTILITIES	
Language independent utilities	$249.50
SCREN EDITOR	
Enhanced programmer's editor	$239.50
dBC III LIBRARY	
Create dBaseIII compat. files in C	$495.50
MacLIBRARY	
C routines Mac styler!	$299.50
MAKE UTILITY	
Intelligent multiple file utility	$249.50

Entertainment

MINDSHADOW	$ 86.95
HITCHHIKERS GUIDE	$118.95
MUSIC STUDIO	$ 97.95
HACKER	$ 89.95
BRATACCUS	$ 89.95
SKYFOX	$119.00
ARCTIC FOX	$119.00

*List complete at copy deadline. Check for additions
Overviews are available for all software packages on request.

WE ARE THE LARGEST COMMODORE DEALERSHIP IN NEW ZEALAND!
FIRST for Specialised knowledge
FIRST for Service, **FIRST** for **AMIGA**

HARDWARE

Amiga 1000 System complete, comprising Computer with 512K RAM, 3.5" disk drive (880K) colour monitor mouse, keyboard, and software – Kickstart, Workbench. Microsoft Basic and Amiga Tutor, Kaleidoscope. *$3995.*
2nd 3.5" disk drive $795.00 extra.

ASHBY COMPUTER CENTRE LTD

93 ASHBY AVENUE
ST. HELIERS, AUCKLAND 5.

To ASHBY COMPUTER CENTRE
93 ASHBY AVENUE, ST. HELIERS, AUCKLAND 5.
TELEPHONE 588-301.

YES! I want to be among the first with an Amiga! Please record my order now to ensure delivery from the limited first shipment due end June.

Full system order at $3995.00 .....
Extra 3.5" drive at $ 795.00 .....
Software as per attached list .....
Information only than you .....

Enclosed is 10% deposit

$.....

GST OPTIONS IN SOFTWARE

There are hundreds of users out there and many potential users who are becoming increasingly concerned about their software and the proposed inbuilt options for the new GST tax.

If we look at packaged software on micro-computers it would be fair to say that no software house has yet completely tackled the question of GST.

Sure, we are told by some suppliers that provision has already been made for the tax, but if you look at the facilities provided it merely allows for GST to be handled in the same manner as Sales Tax is now.

We are also seeing an influx of UK-based software with its inbuilt VAT control being marketed in this country, and modified to read GST.

Be aware that they don't necessarily address all GST needs.

Palpitations

What is causing us a few palpitations is that we have seen no prototype products being offered where the whole GST question has been tied down to the degree that the bi-monthly return is prepared by the computer.

No doubt each software house will have a different approach to GST and the extent to which they build in controls will vary enormously.

One aspect is certain though, and that is modifications will affect debtors, creditors and invoicing systems and likely sales analysis.

These modifications may be extensive in that they are far from cosmetic changes to reports.

If full GST features are built in, some considerable program code changes are essential.

Impending crunch

Herein lies a potential disaster. As the government has yet to finalise and agree the level of disclosure for GST, software cannot be finalised.

If it cannot be finalised, it cannot be installed on test sites for any length of time prior to 1 October.

If this means we may have dodgy software with GST options available from October, then users beware.

If everyone purchases the GST version of software packages and runs with them, there will also likely be a large scale support problem for software houses who have not had the chance to "bed down" their product.

We would suggest that any business that is computerised and running on non-GST software prior to October 1, continues if feasible, after October 1 without GST versions of software. In this way you can avoid problems that may occur.

"No software house has yet completely tackled the question of GST.'

Wait for proof

Once the software has proved itself in the market place, which may or may not be before GST is introduced, then consider upgrading to the new GST inclusive product.

If existing users have had no word from their suppliers regarding GST, we suggest you ask questions immediately as to:–

- What are the details of proposed modifications?
- When are they going to be released?
- What will they cost and is there an installation fee?
- Is there any change to my software support charge/agreement?
- Are GST changes compatible with my previously modified programs (if you know you have a modified system)?

You only have a little over eight weeks to resolve the questions of GST on your computer. Don't let it drift further!

Charter out

We have all but seen the demise of the Charter Series now that Paxus have

In this regular column we keep the business person in touch with developments in the microcomputer industry.

The research reports are from Phil Ashton and Grant Furley at MicroLab, a "neutral" d.p. consultancy established by the accountancy KMG Kendons, in Auckland.

sold the source code to Tauranga-based Chevron Industries.

It is sad in many ways that a product that for many years was ahead of its time finally ground to a halt through a combination of poor marketing at IAL and a history of ongoing support problems.

In many ways Charter set the standard for microcomputer accounting software, and in some areeas those standards have never been improved upon.

Maybe Chevron have the know how to inject some life into the old girl and bring her to the forefront again.

Magna Charter, which is Paxus's last gasp for air in the software development area, looks set to stall in first gear. It was devised as a networked option of the original Charter series but the reality is that the multi-screen capability in Order Entry is a "Clayton's" option.

Charter was not designed for and will never have multi-user capability.

Multi-user software

With the very popular Pick and Unix operating system now running successfully on micros, we are seeing the transporting of some very reliable mini-based software down to the micros, and allowing users to merely add dumb terminals to extend the number of system users.

MicroLab has details on all the significant products in this range.

These products have traditionally been sold by software houses with small advertising budgets therefore they have not suffered from overexposure.

As a further service to clients Microlab now offers support on a selected range of software products, as well as training courses.

Portable developments

If you're in the market (like us) for an affordable portable, then you'll be interested in the spate of new options launched recently. But before we talk about them – what makes a good portable?

First is size and weight. We've had the Osborne, Corona, and latterly the Panasonic Senior Partner, but these "luggables" and others like them are not exactly convenient.

We were reminded of this when taking one home for an evening's work with Open Access. The Corona, weighing in at near 14 kg in the right hand, and in the left Open Access pack weighing in at about 2! And gripped between your teeth is a briefcase crammed with workpapers?

And what about all the boxes full of diskettes you're wanting to take home too (alas it's a twin floppy).

The solution – two trips to the car, or a balancing act that qualifies for the Moscow Circus!

So there's our first criteria – lightweight (like under 5 kgs) and as small as possible. Anything bigger and you may as well take the desktop model home and save the eye strain!

Second criteria has got to be screen legibility. Soft shades of grey may be great for an interior decorator, but on liquid crystal display screens they'll send you barmy. So it's got to have a gas-plasma, or electro luminescent screen. Or perhaps a backlit LCD.

And if you can plug a separate monitor in as well, then so much the better.

A printer too

Our third criteria for a portable PC is a printer. doesn't have to be letter quality – thermal paper will do, but it's an essential ingredient for our ideal portable. The thing is they're just so handy.

Often it's hard to really get to grips with your work in context, unless you print it out. It just doesn't seem the same when you can only see bits of it on the screen at once – maybe we've been spoiled using the Panasonic Senior Partners.

Our other criteria relate to any PC – decent RAM expansion potential, graphics display capabilities, hard disk option, serial and parallel ports as standard, and you know the rest.

So how do the new players stack up? Here are our comments on those currently available in New Zealand:

"Luggables"

Panasonic Senior Partner, and Executive Partner:

Afraid Panaisonic doesn't pass the scales test – but both are really good machines (if you've been doing some weight training).

Corona *PPC:*

Lovely high res. screen and good reliable box, but showing its age.

Compaq (various models):

Scollay and Businessworld are distributing these in New Zealand now... at those prices, why bother?

Sanyo 770:

The 770 is the only colour screen on the market, but it's a real heavyweight.

The "Totes"

Sharp PC7000:

Nice machine, good price (around $4000), fast processor, backlit LCD screen, twin 5¼ drives, and matching printer available as heavyweight clip on.

Morrow/Zenith Pivot:

They feel a bit plasticy, but have a lot of nice features and twin 5¼ drives. Price and supply in New Zealand a bit uncertain.,

"Lap tops"

Toshiba 1100:

Now this looks like a goody. One 3½ inch drive, nice LCD and a price about $3700. You can plug a monitor in directly too.

IBM PC Convertible:

Looks like IBM and the Toshiba are from the same mold. They deducted a few vital bits (like a parallel port), but they will sell them back to you. Not yet released in New Zealand – but real soon now.

Bondwell 8:

Did you ever see a Bondwell 2 – a laptop CP/M malchine which missed the boat? Well, take out 8-bit and insert 16-bit and you've got the new BONDWELL 8!

Same box, 3½ inch disk, hopefully a legible LCD. But the price is promised to be low – so watch out for this one.

Data General DG One Model 2:

You can now get the DG1 with a backlit or an electro luminescent screen. This will add a lot to what is already a rather impressive portable.

Kaypro 2000

Quite a uniquely styled machine. Looks a little fragile despite the alloy casing, but worth a look.

HP 110 PLUS

No drives, and pricy. This early entry into the market is going to be upstaged by the newer machines.

So there you have it – one of those could fit your needs. What about us?

Bondwell and Toshiba look like good bets in terms of value for money, and size/performance. But there's no printer clip on or built-in.

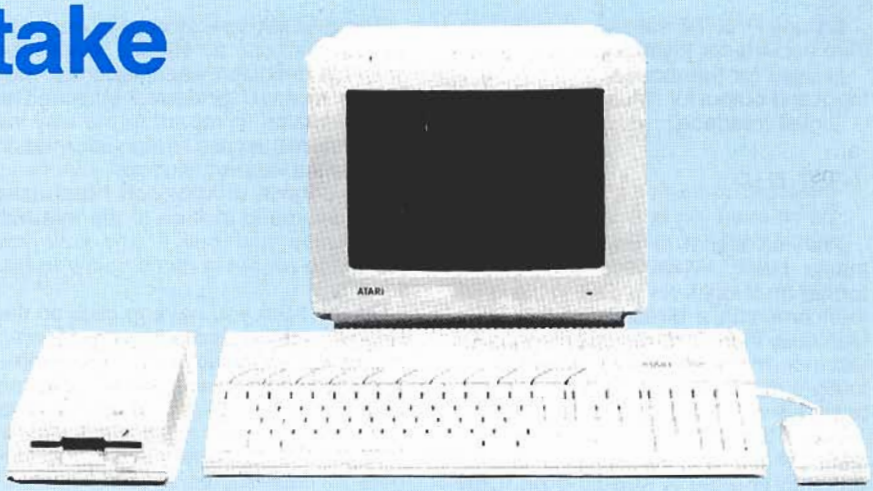
We see Toshiba has just released in the United States the T3100 – an AT compatible laptop with gas plasma display and an internal 10 meg hard disk.

Now perhaps if we wait just a little longer.....

Poised to take the lead

The ATARI 520ST

by John Slane



The 32-bit computers are now generating a great deal of interest and gaining deserved respectability. Although LISA (Apple) had all the right qualities, she didn't make it. Probably too soon, and too expensive to justify its use with a limited software base. However the young upstart, Macintosh, is now established as an "in" machine for specialised users – desktop publishing is a notable example.

With the Atari ST, and the Amiga, we are seeing further machines in what I expect will be a large stable of 32-bit work horses in the not too distant future.

The first in have to get it right to stay in the race. Atari have given it their best shot and offered two versions, the 520ST (512K RAM) and what is clearly intended to be the main business model, the 1040ST (1MB RAM).

Although the operating systems are the same, the physical configurations are quite different. The 520ST is a bundle of bits (keyboard/processor, disk drive/–/3.5" single sided 380K, monitor, two power supplies, and a mouse). The whole is held together with a forest of cables many of which are thick, unyielding and awkward to take around corners.

On the other hand the 1040ST is a "computer-in-a-box". For only $805 dollars extra you get a more convenient unit, twice the RAM and twice the disk capacity.

I'll leave the reader to judge which option I would go for.

First impressions

These days it is more common to have the processor in one box and the keyboard as a separate unit. The 520ST doesn't follow that pattern. Keyboard and processor are all in one and a certain inflexibility in use is introduced as a result.

But the keyboard/computer unit is of pleasing design and remarkably compact when one considers the huge keyboard options available (95 keys) and the substantial memory it includes (192K ROM plus 512K RAM).

Once all the cables and plugs are in place the unit is simplicity itself to operate.

A purchaser will be pleased to find that many desirable, even essential, options are inclusive in the price. The mouse unit, word processing, logo and basic are all there as well as a great graphics program called "Neochrome".

Clear text

If you like windows programmes, you will enjoy the Atari.

The main enhancement is the high quality of graphics and the outstanding clarity of text.

The screen of text comes as close as I have seen to the elegance and sharpness of a page of typeset text.

The reason for the excellent text clarity is, of course, that bit-mapped graphics are used. The advantage is that the high resolution available makes it almost impossible to distinguish the dots that make up the letters.

The disadvantage is that an enormous amount of mapping is necessary to produce text (and graphics) characters in the right place on the screen so the speed of writing and scrolling is as bad as many of the 8 and 16-bit machines (and some of them are terrible!).

On my informal benchmark tests, the Atari ST was breath-takingly fast on arithmetic computation, but very pedestrian in writing information to the screen.

In more detail

The keyboard/computer unit is wide to accommodate all the keys, but the depth is only about a third more than a typical stand-alone keyboard.

The keys have a rather stiff feel and are very sensitive indeed – the slightest touch produces a response. It took me a while to avoid triggering keys I didn't want to use.

CAPS LOCK has no electrical or mechanical indicator to signal the selected position.

I am informed that Atari make two keyboard versions – USA and European. The model available in New Zealand is the UK and European one, and includes the operating system in ROM (192K), which has only recently been developed.

Although not immediately obvious because of the way they blend into the cabinet moulding, there are 10 function keys above the keyboard. The "F1" "F2" etc labels are small and don't stand out well in mediocre lighting conditions. No provision is made for laying down those handy application label strip.

Special consideration has been given to the numeric keypad. All the numbers and symbols needed for mathematical data are grouped here.

Keys such as * / – () and the decimal point are available on the pad without shifting, as well as being in their conventional qwerty positions.

The Atari ST numeric pad could easily become the industry standard.

Built to expand

One big surprise comes when you tip the keyboard over and look around the edges – there's a long row of well-labelled sockets, and even a reset button.

Leaving out the obvious, the ports include:

Cartridge – to take up to 131K of ROM,
Modem – also acts as a "standard" RS232C serial port,
Centronics parallel printer,
Special interface for hard disk,
Floppy disk,
High resolution monitor (monochrome supplied with the review unit),
Modulator output (and cable supplied)

for any PAL TV set,
Two sockets for joysticks, one of which is used for the mouse,
Input and output for "Musical Instrument Digital Interface".

Test run

With no signal to the monitor it remains black. When the computer is turned on it looks for a disk in the drive then presents a directory of that disk. Mercifully there is no tedious diagnostics test nor a mandatory requirement for today's date before it gets on with the business of getting up and running.

Date, time and various parameters such as keyboard response, can be conveniently entered by selecting the "Control Panel" option. And this Atari knew how dates should be written – day, month, year – indicating firmware changes for the European version.

The operating system is uniquely Atari – The Operating System, or TOS. However, the most visible system in many applications is the window management – Graphics Environment Management, or GEM, licensed from Digital Research Inc.

Dynamic

Information and options are in a mixture of text and little pictures ("icons"). A screen arrow directed by the mouse is used to select functions required and then action them either by clicking a button on the mouse or using the ENTER key (on the keypad) or the RETURN key (on the main keyboard).

If you haven't seen window operation in action, you would find a demonstration fascinating and make a lot more sense than an attempt at a verbal description. The essence of windowing is the dynamics of the process – one window partially opening over another, size changing, position changing, and scrolling within the windows side to side or *up and down.*

When I was demonstrating the machine to a group of colleagues it was ten minutes before I touched the keyboard – and we were on to our fifth program by then!

And more windows

Clearly the designers have discovered a new toy and something to impress potential customers. Just about everything is windows – and after a while it all starts to get a bit tedious, and at times frustrating.

But you can go straight to the action by typing in direct commands in the conventional way – an option for which I was grateful.

The first window system I used was Microsoft's on a 16-bit. While the graphics definition was inferior to Atari, I have to say I preferred Microsoft to GEM – mainly in regard to the way the two systems access hidden information outside the current window.

The version of Microsoft I had used did allow you to lock on to the text with the mouse and pull it any way you wanted to (or set in motion slow or fast scrolling).

On the Atari you have to click on the window margins and set a new viewing orientation. Then the screen re-writes itself to the new view. If you've gone too far you won't even see what you have missed.

When the designers and programmers get really clever, and have 10 man-years to spare, they might get around to designing a system that reformats text, with word-wrap, to fit inside whatever sized screen is currently dimensioned.

I don't mind scrolling up and down, but it is a real pain to have to do that AND go from side to side to see all the text.

In the supplied Basic and the word processing programme this text flow on is a severe restriction.

Basic & benchmarks

When I saw the Basic file on disk, I couldn't believe my eyes. File length 138,944 bytes! How could anyone take that much code just to write a reasonably adequate Basic?

To start with, a programmer has a very different approach to his (her?) task when available memory stretches out to the horizon and beyond. The agony of trying to fit a quart into a pint pot is not relevant here and it takes less trouble to be wordy and convoluted – so the programming expands to fill the space available.

The other factor is that sophisticated graphics and windowing techniques *swallow up memory like a hungry whale.* 32K to pixel map one screen. So, programs get BIG.

I was pleased to have Basic supplied as part of the standard package. But I am told by the distributors that Basic was a recent afterthought.

Originally the Atari Corporation thought of the ST owner as a dedicated user of commercial software and that the do-it-yourselfers wouldn't be interested in purchasing an ST. Suddenly the validity of this assumption was being questioned and it was decided that a safer bet would be to provide BASIC after all.

In fact, the distributors are a little embarrassed by the inadequacy of the Basic code provided and assure me that at least one other software house is developing a more sophisticated version to be marketed as third party software.

If this eventuates, it will be to the benefit of potential users such as myself. Frankly, I wouldn't want to persevere with such a clumsy Basic, and I wouldn't buy any computer unless I was able to efficiently write custom programs on it.

The worst feature of Atari BASIC is the idiot way it tries to use windows. It requires separate windows for commands, edits, outputs, and listings. You can only edit as much code as is in the edit window.

Speed

I tried some of my favourite routines to see how the ST coped.

In spite of the large Basic code, there is still 218K of RAM available for BASIC programs, although for some reason arrays are limited to a maximum of 32K. Actually I couldn't create an array bigger than 15K, but that was probably my fault. Even more memory is available if you choose not to have the graphics buffer open.

1500 rather complicated arithmetical computations were processed in only 3.8 seconds. I blinked, and tried it again. No mistake. That's blindingly fast compared with even the best 16-bit 8-Mhz machines I have previously used.

The ST tested for and found all the prime numbers to 50 in only 5.8 seconds.

Accuracy

Long-time readers of Bits & Bytes may recall an interesting arithmetical accuracy test published in the Atari column some years ago. One was to do a test of squares using an integer test, and the other was to step by .01 from 0 to 1. It was claimed in the article that the old Atari was the only 8-bit computer that could do those tasks with 100% accuracy. (I know my computer at the time wouldn't!)

Well, the new Atari found the squares OK, but stumbled on the second test.

It's all a bit academic, really, because there are few instances where this kind of inaccuracy matters (I have had to convert numbers to strings on only one occasion to avoid this lack of accuracy in the fifth or sixth decimal place).

When a basic program requires calculations or strings to be written to the screen, speed performance fails dramatically for the reason I described earlier.

Disk operation

Formatting a new disk (single sided) takes 55 seconds (360K approx).

Writing a random file of 500 records

of 200 bytes took a fairly pedestrian 2 minutes with retrieval (and writing to the screen) about 1 second per record.

Backing up a full disk using a single drive is commendably efficient. I have never really entertained the idea of having only one disk drive on a computer mainly because of the time to do backups. For the first time I have found a computer that does this really efficiently with only one drive.

A couple of quick changes are required at the start, then the ST soaks up the whole of the disk data into RAM, requests the destination disk, and then disgorges the whole lot in one go. Another quick change to write the directory update, and that's it. Remarkable. Time: 2 minutes 50 seconds, and that includes the time I took to swap disks as directed. I can live with that.

Word processing

A program called "1st Word" on a separate disk is part of the purchase bundle.

Since most computers these days are used for at least some of their time on word processing, this offering is a real bonus.

It's only an average program of its type – modelled it seems on the early version of Wordstar.

Pity! With all the power of a 32-bit processor at your command it is weird that you have to manually reformat every paragraph after an editing change, or if you change your mind about width parameters you have to go through the whole document, paragraph by paragraph to get it into the new shape.

Long lines overflow out of the right hand edge of your margin is wider than 75 characters.

It's possible, and surely not too difficult, to do a lot better than this. In fact several alternative word processing programs are listed as being available for the ST – but you have to pay for them, of course.

Print codes

A window of options provides for the setting up of output to your printer, but there appeared to be too little information available to set specific control codes such as would be necessary to get one of the unspecified printers to go into graphics mode, or print italics, etc. The information must be in there somewhere, but I couldn't find it.

With those criticisms out of the way, there are also a lot of good features. I like the mouse/keyboard facility for word processing. Shifting line lengths, setting and cancelling Tabs, marking blocks, and so on. are greatly simplified in the GEM environment.

Above all, is the brilliant sharpness of text and the excellent print style. Also, if you want italics, the screen prints in italics until you change to something else. It's hard not to over-indulge!

Really, this is what computer screens should look like, and it's a pity we have had to wait so long to see the wish realised.

Logo

I'm not sure why Logo was chosen to be included in the standard package. Would parents be buying an ST for their children? Wouldn't a basic database program have been more relevant for what is being targetted as a business machine? Or a spreadsheet?

The supplied manual won't be a great deal of help to a beginner; it's too academic and not well organised.

But an experienced user of LOGO won't have too many complaints about the Atari version. A very good repertoire of commands is available, and the graphics speed is impressive.

The Neochrome

Well, why have a computer that is brilliant with graphics and not have something to exploit its talents? Neochrome is offered as the answer to that.

The way it works is not unlike what is already tried and tested for the 8-bit machines – very similar to a program I reviewed that runs on the Apple.

For this Atari program I used the inbuilt modulator to feed into my domestic TV set. Now we had full colour up and running. There were no complaints about the performance except that the needle sharpness of the monochrome screen was not realised on the colour TV version.

I had to bully my 10 year old daughter to get off the graphics program so I could carry on with the serious work of doing

Microcomputer summary

Name	Atari 520ST
Manufacturer	Atari Corporation, California, USA
Microprocessor	68000
Clock speed	8 Mhz
Ram	512 Kb
Rom	192 Kb
Input/output	Floppy disk, Hard disk, TV, Monitor, Parallel, Serial, digitalised input & output (MIDI), Rom Cartridge, Mouse, Joysticks
Keyboard	95 keys. Separate cursor controls, numeric pad, 10 function keys
Display	9.5" diagonal image on 11.5" screen, monochrome supplied. About 75 characters to line, depending on mode being used. (80 chars. without GEM)
Graphics	640 x 200 (colour), 640 x 400 (monochrome).
Sound	3 voices
Disk	360K single sided 3.5"
Options	Modem, hard disk, cartridge, digitaliser
Operating systems	TOS & GEM
Languages	Just about everything!
Bundled software	BASIC, LOGO, WP, NEOCHROME GRAPHICS Note: Mouse is included
Cost	$2490 (current special price $2290)
Ratings	(1 low, 5 high) Documentation 3, Support 3 (limited dealership at this early stage), Language 5, Expansion 5, Value 5.

Review machine supplied by Montek Equipment Ltd

a review. Then she wanted to know why I was drawing pictures instead of getting on with the other stuff!

Other packages

I also had access to some other programs, not part of the standard package, to experiment with. These included a mapping program and one which could draw and then manipulate three dimensional objects.

Using these confirmed my earlier judgement about the power and speed of the processor. The above two programs included some incredibly complex mathematics and required vast amounts of processing of co-ordinates and data tables. On lesser machines you would have waited all day for the computation results to be displayed.

Documentation

With the Atari 520ST comes relatively limited documentation – sufficient to get you up and going and coping with the software packages provided.

That's not an implied criticism; Atari have made a decision about the needs of the likely purchaser and have provided what they think will be necessary for him/her.

For the users who want to use the ST as a development tool there is a mass of additional material both in documentation and disk utilities, but you have to go out and buy it.

I've no quarrel with that marketing decision. However, a serious programmer would need to total up the dollars to see what the development package was finally going to cost.

Software availability

The NZ agents are taking a cautious approach towards their software inventory for the ST models.

They say that as yet they don't know what NZ customers are likely to want, so have chosen to stock only a limited range of what is available at present.

However, the company tells me that they are encouraging other retailers to stock Atari software and will also take direct orders of specific items for a customer.

It's a chicken and egg situation really. Can you sell enough computers without the software to demonstrate its uses? Will there be enough customers wanting to purchase the stock you have bought in? Will your machine be seen to be a winner by the software developers and worth their investment of time and money to develop product-specific software to expand the program choices available?

As well as providing headaches for the distributors of any new, unique machine, these problems also face early purchasers.

We've waited a long time for a solid software base to be established for the Macintosh. My pick is that we won't have to wait as long for the same developments for the Atari St.

Summary

As is inevitable, I have found some features of the Atari 520ST which are not to my liking. My irritation is fanned by the conviction that designers and programmers could so easily have done much better if they had put their thinking caps on.

However, there's nothing fundamentally inadequate with the 520ST. In fact I would have no hesitation in describing it as a computer with great potential.

It has an efficient processor, it has provided for every possible expansion imaginable and its basic operating system is just waiting to be exploited.

To go back to the metaphor I used at the beginning, this is a very promising colt and with the right grooming and training should be seen to be a thoroughbred in every sense of the term.

As value for money, it leaves everything else standing in the 16 and 32-bit field.

More power to plasma

The Panasonic JB-3300 portable

by Bevan J. Clarke

What's this? Another computer from Panasonic! I'm beginning to feel like Jerry Pournelle. Let us hope that this one **isn't** a success – I'll lose all credibility if I keep on praising Panasonic's computers.

Let's give it a whirl. It takes 19 seconds to un-zip the carry case, set up the computer, pop in a genuine MS-DOS 3.10 disk (with SideKick and SuperKey on it) and flip the ON switch, (which is on the front panel at last).

It takes 27 seconds for ROM and DOS to get the 256 Kb sample machine checked out and booted up. Fair enough!

We can see at a glance that the JB-3300 is a compact portable in the tradition of Panasonic's admirable Senior Partner and its new Executive Partner.

My goodness, it is small! Closed, it measures only 43 x 31 x 15 cm (compared to the Senior Partner's 50 x 34 x 21) – a volume reduction to 56%. It will certainly go under the airline seat (the dream of every self-respecting portable). The slim lines are tidy and functional.

Several hatches prove to cover concealed goodies:

- The front snaps off to become the full-functioned keyboard.
- One lifts the upper hatch to find a plasma-panel screen.
- A flush, concealed carry-handle on the left end is comfortable to grasp. In the interests of consumer testing your dedicated reviewer has twice carried the machine a kilometer to work.
- At the left, beneath the handle, a little hatch covers built-in ports: a serial port (for modem, serial LAN, mouse or whatever) and a parallel port for a printer, which will print the full graphic character-set. These use female and male DB-25 connectors respectively.
- Across the back top a hatch lifts to expose two IBM-compatible full-length expansion slots. Access is as easy as that; no screwdriver needed.

Keyboard

The keyboard in the previous Executive Partner was built in to the body. The removable keyboard in the JB-3300 is much more acceptable. It has a standard layout with two rows of function keys at the left, and is enhanced with useful LEDs in the CapsLock and NumLock keys (Panasonic listens!) and a second Enter key in the numeric keypad.

The function keys are back in the IBM location, that is in two columns at keyboard left.

The Senior Partner's awkward keyboard cable is now replaced with a ready-attached cable, which can, however, be detached. Any industry-compatible keyboard can be connected.

The K and J keys have home-key 'bumps' to guide touch typists.

I found the keyboard noisier than I like (although no more clattery than the Senior Partner and a darn sight quieter than many). My children report that the keys have a slightly over-long travel (a tiresome defect when playing video games).

Silent partner

The JB-3300 is not only small, it is quiet. The original JB-3000, having no fan, was totally silent (as a computer should be). The Snr Partner's fan is irritating. But this machine's fan is very quiet and the two half-height 360Kb drives are the second-quietest I have ever strained to hear.

The 640 x 400 dot plasma panel screen is, of course, the novel feature. Its useful area has a 195 mm diagonal, larger than the bulkier Senior Partner.

These soft neon-orange panels were introduced with the Executive Partner so they can now be considered to be mature technology. They are surprisingly easy on the eyes.

I think that this is both because the colour is orange and because the contrast is rather low. This seems to be caused in part by the etched anti-reflective surface.

But 'white' is only medium bright orange and 'black' is faint orange.

Not very bright

The JB-3300 has acquired a brightness control, which is a kind thought but the control is useless – I itch to make the screen brighter not dimmer! I'd like to study the circuit diagrams; maybe the bright intensity could be tweaked a little brighter.

Several times with the JB-3300 I have had to draw the blinds against the low winter sun.

The panal can be tilted at several angles for viewing comfort and it switches off as you fold it shut. As was the case with the panel of the sample Executive,

several of the 256,000 tiny square pixels are not operational. Perhaps as many as 100 pixels remain obdurately off. But this can be discovered only on close and deliberate inspection.

The vendors advise me that the manufacturing standards consider such a (low) failure rate to be acceptable. The plasma screens have a long life and these few blank pixels do not imply future deterioration of the panel.

As with the Executive Partner, the JB-3300's native screen mode is set to emulate the IBM Colour Graphics Adapter with its line-drawing graphics capability and 16 colours.

Graphics modes of 640x200, 320 x 200 and 640 x 400 are available. The plasma panel is much crisper to view than a CRT in graphics mode.

Orange texture

Because the plasma panel has only two colours, orange and orange, some cunning adaptation is employed. All sixteen colours are mapped into four textures, each chosen to look more intense than the previous. This is surprisingly successful – GATO and the Microsoft Flight Simulator screen are perfectly viewable.

But for coloured text the four textures are mapped on to four type faces: which I would call Roman light, italic light, Roman bold and italic bold. These are also very readable but let's face it – the JB-3300 is definitely not generating sixteen shades of orange (let alone green).

If this lack of colour troubles you, the answer lies in the three expansion slots: two standard 13" full-length slots and one slot reserved for a proprietary hard-disk controller.

Therefore, for an alternative display controller card you could put into either slot any colour graphics card including an EGA adapter. A tiny switch is used to tell the JB-3300 to route output through your added controller to an external monitor (which could sit on the processor) rather than to the plasma panel.

Expansion options

I did not have time to test the slots for compatibility with a wide range of cards. I would expect mice, modems, multi-I/O boards (and anything else beginning with an 'M') to work well.

The sample computer came with 256 Kb but the standard model will have 512 Kb installed. It has sockets on the motherboard for further chips.

The maximum motherboard memory is the usual 640 Mb directly addressable by the standard memory map.

I have not got an Extended Memory 4 Mb card but presume it would function when installed in one of the slots.

The manual raises one point of concern when it states that the slots are not to be used for controllers for external hard disks or external floppy disks. I wondered if this caution was to be believed?

In particular, did it mean that a 'hard-card' (a 3.5" 20 Mb hard disk and controller on one full-length card) would not function? And if not, why not? (A hard-disk option is available. The JB-3300 comes in two versions: FD, with two floppies, and HD, with one floppy and one 20 Mb 3.5" hard disk.)

I mean no disrespect to Panasonic's hard disk but I want only a machine which will accept any and all after-market add-ins.

Well, a telex to Japan produced a long reply referring to power ratings et al. It ended with a suggestion that we try it! I am happy to report that a Hard-Card by Plus (supplied by Imagineering) works like a charm!

Nevertheless, the warning is there.

Before buying I would also want to know whether an 80286 accelerator card would function since there is only the slow 4.77 Mhz processor installed.

True compatible

I found the IBM compatibility to be excellent. With two exceptions every program product which I could fit into 256 Kb would run. For example, I found that programs I have written, which spatter the screen with graphic pull-down menus, ran faultlessly.

One exception was Borland's Superkey keyboard enhancer and macro processor. For any machine this is a severe test of compatibility at the BIOS level and just occasionally the JB-3300 seemed to become confused as to which type-face (colour) it was in or it presented pull-down menus which lacked their inverse video.

This software company says "If <our product> doesn't work then your compatible isn't as compatible as you thought." For a time I thought that another incompatible program was BASICA! Two copies of BASIC which ran perfectly on the Senior Partner produced oddly incorrect characters when " and @ were pressed.

To my relief it transpired that the DOS disk had been installed for the UK keyboard. The problem had nothing to do with the machine.

The JB-3300 corrects one minor short-coming of the Senior partner. A true IBM BIOS chip allows one to enter any of the 256 ASCII characters (decimal codes 000 to 255) by typing that code while holding down the ALT key. The Senior partner could not do this.

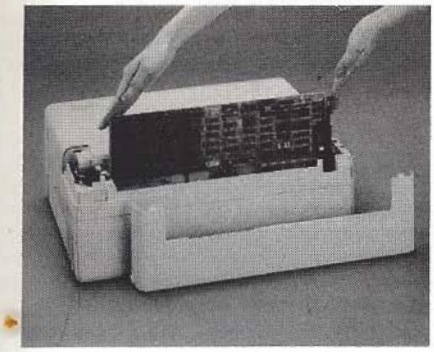
If you have a need for this feature you'll be pleased to find that the JB-3300 has a properly-capable ROM and can do it.

The JB-3300 comes in a stylish NZ-made water-resistant carry-sack with a side pocket for disks, manuals and the power cable. Typical of Panasonic's designing for humans, is a little recess atop the computer where your few most current floppies can rest, protected by the screen.

At 11.6 kg it is heavier than its appearance would suggest. The manuals (User's Guide, User's manual and an MS-DOS Manual) are excellent, written

Microcomputer Summary

Name:	Panasonic JB-3300
Manufacture:	Matsushita of Japan
Vendor:	M.E.C. Ltd of Auckland, Box 9224.
Processor:	16-bit 8088
Clock speed:	4.77Mhz
RAM:	512Kb expandable to 640Kb (plus expansion slots, which could take an 'above-board' 4Mb card.)
Input/Output:	Keyboard; Centronics port; serial port plus slots.
Display:	80 x 25 (640 x 400) flat plasma panel screen; colour RGB port only if expansion card fitted.
Operating Systems:	As for IBM PC
Graphics:	With plasma panel 640 x 400, 640 x 200 and 320 x 200 pixels in monochrome orange; can show four shades (textures) of monochrome.
Sound:	Internal speaker
Cost:	$4995 (includes 512Kb)
Options:	Two 13" slots for expansion cards; One slot for Panasonic hard disk 20Mb.
Reviewer's ratings:	(5 the highest): Documentation 4; ease of use 5; language 5; expansion 4; value for money 4; support 4.



in fluent English and are beautifully clear for the first-time owner.

The DOS manual is not a reprint of the classic set but a thoughtful re-write.

Overall what do we have here? This is a fine little machine, compact, capable and compatible. It is silent and sweet to use and not too heavy to lug for several hundred meters. The restful plasma panel screen is literally a sight for sore eyes.

It accepts fullsize option cards in two slots and can be bought with a hard disk. If you want colour or a CRT monitor then you must buy a graphics card and sacrifice a slot.

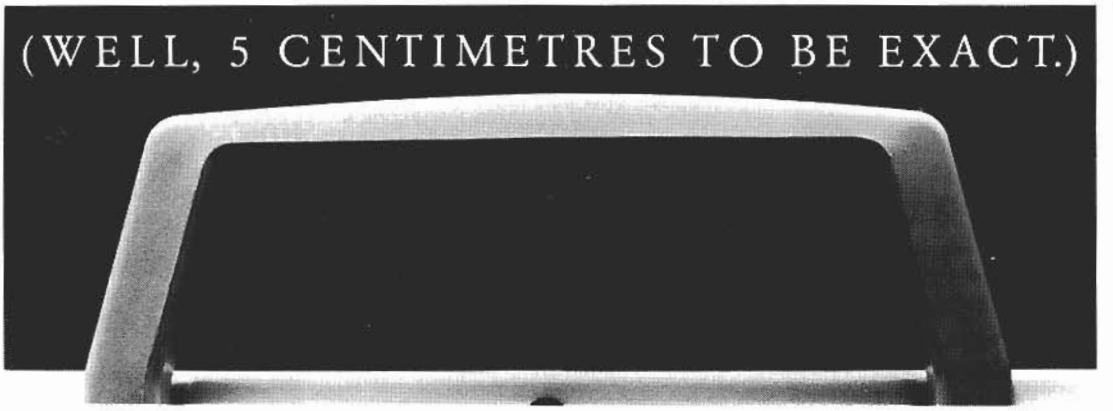
But there **are** a few disappointments:

- The keyboard of the sample machine did not snap quite tightly against the body when closed for carrying and the keys are just as noisy as other machines.
- I still wish Panasonic would give the user an option to store 720 Kb per disk – they make such drives, which are very reliable.
- There is no built-in printer – the past trade- mark of portable Panasonics. Ah well, if you really want a portable printer they make two other computers! Printers don't cost an arm and a leg any more.
- The processor is a plain vanilla 8088 running at 4.77 Mhz. There is a socket for an 8087 mathematics co-processor but I had hoped for the same 8086-2 processor used by the Executive Partner, which can be switched by a key-stroke from 4.77 to 7.1 Mhz. Panasonic, are you trying to protect the Executive's niche in the market?
- The manufacturer warns against plugging in 'hard cards' (hard disk drives on an option card) although experiment shows that one does work.

But it is a machine to be recommended. Well done, Panasonic.

THE ADVANTAGE OF THE NEW MICROBEE MS-DOS* COMPUTER STICKS OUT A MILE.

(WELL, 5 CENTIMETRES TO BE EXACT.)



The more powerful a computer, the more it seems to be desk-bound.

Not so with the new Microbee/Mitac Portable PC.

Unlike most MS-DOS computers, you don't need a physique like Charles Atlas to move it from one place to another.

It's light, weighing in at barely 4.9 kilos. And, by virtue of a handle, it's also easy to carry.

The keyboard and 5¼" disk drive are ingeniously incorporated into one unit, with up to 640K of RAM.

Microbee/Mitac Portable PC with 256K RAM-$1895 (Monitor not included.)

So, if you need to take your Microbee/Mitac Portable PC somewhere, all you need is a free hand.

This new computer opens out to Microbee users the vast range of IBM† PC compatible software.

In fact, the Portable PC uses the internationally accepted Phoenix BIOS.

So, unlike some compatibles, this one is truly compatible.

The inbuilt floppy drive can store 360K bytes per diskette.

If that's not enough, however, a second optional 360K byte floppy disk drive can plug into the rear of the case.

You can also add a 20 Megabyte hard disk when you require large volumes of external storage.

There's a colour graphics adaptor built in as standard.

It also includes two serial communication ports (not just one), plus a parallel printer port as standard.

Other standard features include a real-time clock and calendar, with a battery pack.

You'll also find that it's compatible with the range of Microbee peripherals.

So, no matter what use you have for a personal computer, the new Microbee/ Mitac Portable PC can (dare we say it) handle it.

For more information, contact Microbee Computers now.

Microbee Systems NZ Ltd

438 b Rosebank Road, Avondale, Auckland
Telephone: (09) 88 1138 or 88 1139.
Wellington: P.O. Box 26045 Newlands,
Wellington. Telephone: (04) 78 5548.

*MS-DOS is a registered trademark of the Microsoft Corporation
†IBM is a registered trademark of IBM Corporation. C&MA MBS0024 NZ

Excel puts Mac in business

by Grant Cowie PhD

There has been flattering talk in recent months about Excel, the new spreadsheet system for the Apple Macintosh computer, by Microsoft. I was therefore eager to take a look at this product, even though I was sceptical of the claims made for it.

Excel consists of four main fully integrated parts: a spreadsheet, graphics, database and macros. In developing Excel, Microsoft have obviously taken a good, hard look at the highly successful Lotus 1-2-3. Excel offers similar features to 1-2-3, and can even read 1-2-3 files. Microsoft clearly hope that Excel will be the "standard" next-generation spreadsheet after 1-2-3.

Excel retails in New Zealand for $1,095. For this you get a comprehensive manual, a program disk, a backup disk and a system disk. This latter disk also contains several examples and a copy of "Switcher", a useful program which enables the Macintosh to run more than one program at a time.

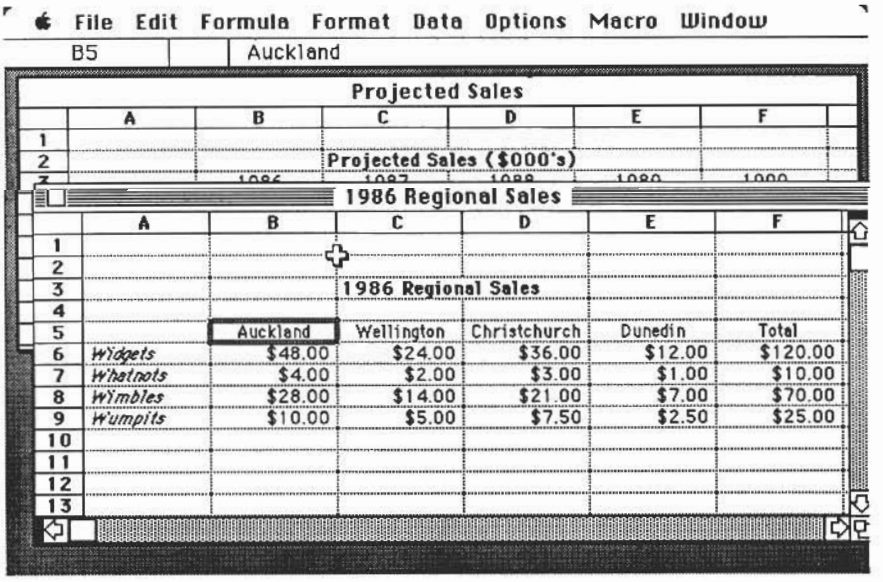
To run Excel you will need a 512K Macintosh with external disk drive or (even better) a Macintosh Plus.

Worksheet

In Excel, spreadsheets are called worksheets. A worksheet consists of 256 columns and 16,384 rows, although the amount of information you can actually store in it depends on the amount of memory that your Macintosh has.

The Excel Worksheets have far too many features to enumerate in a review such as this. However features which I found particularly good include:

- In Excel you can use either the A1 terminology of Lotus or the R1C1 terminology of Multiplan.
- Excel uses the *Fill Right* and *Fill Down* commands from Multiplan, which are much simpler than the/Copy option in Lotus. There is also a shortcut keystroke command for entering data or formulas into more than one cell at once.
- You can have as many windows as you want into the same worksheet. This is particularly useful as you can display values in one window and the formulas in another. You can also have as many different worksheets open at the same time, and can reference cells in other worksheets whether they are open or not. This last feature means that in principle you can construct and run models which are larger than the Macintosh's memory.
- You can define your own formats for data display. For example you can specify a format to make a cell display "300 tonnes" instead of just "300".
- Individual cells or ranges of cells can be named and referred to by these names in other formulas. One interesting feature (for which I haven't yet worked out a sensible application) is that ranges of cells can be any shape, and need not be continuous. In 1-2-3 for example you can only select rectangles, something which I have never found to be a restriction!
- Excel has an "intelligent" recalculation facility. Normally on a spreadsheet when you enter or change a value the entire spreadsheet must be recalculated. Excel recalculates only as much of the worksheet as is required. When working on large or complex spreadsheets this facility would save a lot of time.
- You can use any of the fonts on the Macintosh to display your worksheet (although you may not mix fonts within a worksheet). You can also highlight parts of the worksheet by displaying cells in **bold** or *italics*, but you can draw boxes around parts of the worksheet.
- Excel allows you to interrupt it when it is recalculating the worksheet.



1) A sample Excel screen showing two open worksheets, each in its own window.

Mac drawbacks

If Excel has fully exploited the Macintosh user interface it also suffers from some of the Macintosh drawbacks.

Principal amongst these is that cutting, pasting and clearing (the equivalent of Lotus/Move,/Copy and /WErase or /RErase commands) can be slow if you are operating on a large portion of the worksheet. This is caused by the Macintosh facility to "Undo" a command.

However, given the number of Lotus spreadsheets I have seen destroyed by incorrect use of the /Copy command, I think I would opt for slower copying with the ability to "Undo" disasterous mistakes.

The Macintosh mouse, while much better than keyboard commands for most operations, is slightly cumbersome to use in "navigation" around larger models. It would be helpful if Excel provided the "End" function key (or equivalent) of 1-2-3.

To test Excel I built a reasonably sized corporate model based on one which I developed on a mainframe computer. The model itself was about 500 rows and 20 columns, and used several macro functions (these are discussed later). Transferring the model was actually a lot easier than anticipated.

The only problem I encountered was with the model size. The manual states quite explicitly that the size of the model is determined by *the intersection of the last row and column that contain a value, formula or format.*

Unfortunately Excel seemed to determine the size of this particular model by the intersection of the last row used and

the last column on the spreadsheet. So instead of being 500x20 cells, the model was 500x256 cells.

This meant that the saved file had a size of 600K, which does not fit onto a 400K disk (fortunately I was using a Mac Plus which has 800K drives).

Graphics

One of the strongest features of Excel is its simple to use business graphics system. Like the spreadsheet, these offer the same sort of features of Lotus 1-2-3 graphics but are much easier to use. The graphics work in much the same way as in Microsoft Chart, but are not quite so flexible.

To create a graph it is necessary only to highlight the portion of the worksheet that you wish to graph and create a new window of type chart. Excel takes an intelligent guess at how to present the data, so the result is usually a pretty neat graph straight away.

Having got the basic graph prepared in this manner it is then a simple matter to customise it by adding titles, text, arrows, legends and so forth.

Excel offers six basic types of graph: Area, Line, Bar, Column, Scatter and Pie. Each of these has in turn about six different variations which are selectable from a "Gallery". If all of these are not sufficient you can fine tune the graphs, varying the display of the axes, the legends and so forth. In short, the charts from Excel are very easy to use and extremely flexible.

As one would expect from a "Graphics" computer such as the Macintosh, the quality of the charts produced is very high.

Charts printed on the Imagewriter printer are good, but you probably wouldn't want to use them for a presentation to the managing director. The Apple laserwriter produces superb quality black and white graphs, but is unfortunately priced out of the reach of mere mortals.

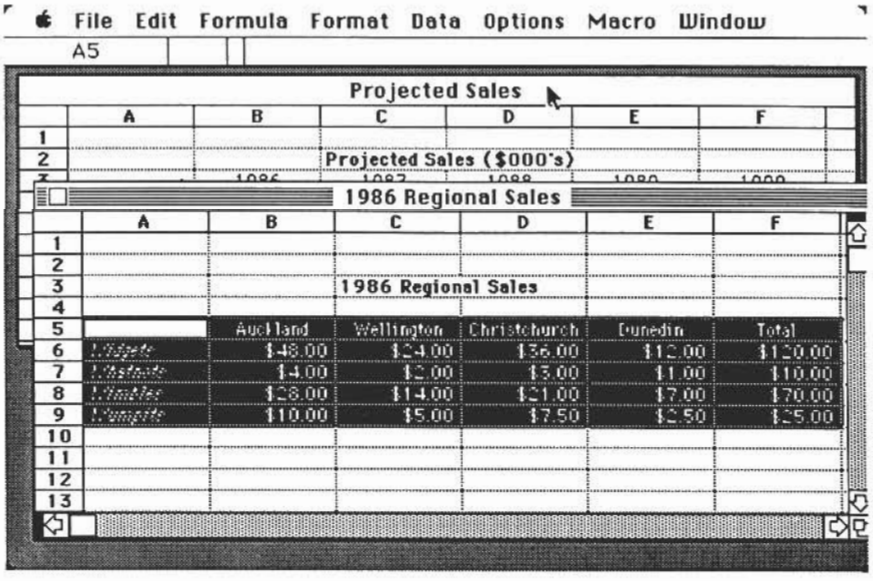
If you want high-quality colour graphics then things become a bit more difficult. Unlike Lotus, Excel does not have the capability to drive a plotter. The Imagewriter II printer can use coloured irbbons, but it is still basically dot matrix quality.

If you do require plotted charts there is apparently a system called MacPlot which can take Excel charts and plot them. Unfortunately I was unable to track down a copy of this, so I cannot comment on how effective it is.

Database

I must apologise at this point to all those people who enthuse over the database functions of Lotus and similar packages.

At the risk of raising the ire of those of you who have developed useful applications in these, I must say that I have always found these less than impressive and only marginally useful. This is probably because if I have any sort of application involving database work I would rather use a "proper" database package. I think the marketing arm of Lotus has done us all a disservice by passing off a simple list management system as a database!



2) The first stage in preparing a chart is to use the mouse to highlight the area which you want graphed. You then create a new window of type "chart".

For any database to be useful it must have at least the following:

- user definable data entry screens with edit checking at time of data entry;
- the ability to have several files open and to relate information across files;
- some sort of language to code operations;
- flexible reporting, with the ability to automatically generate totals and subtotals if required.

Does Excel have these features? Disappointingly no.

The database in Excel offers about the same level of functionality as that in 1-2-3. This restricts its (sensible) use to maintaining lists of (for example) product prices. These can then be accessed automatically from the worksheet.

If you are really enthusiastic you can write Excel macros to prompt you for information and to check it for you. However, to my way of thinking, if you want anything remotely complex you would be better to use a database system such as DBase or Omnis.

Macros

In spreadsheets, a "macro" is a user written "program" which will automate repetitive tasks. It is probably this capability more than any other which has made Lotus 1-2-3 the *de facto* standard spreadsheet system for IBM PCs and compatible computers.

One of the deficiencies of Jazz (the Lotus spreadsheet system for the Macintosh) was that is did not have macros, and this must have limited its sales. Clearly Microsoft didn't want to make the same mistake.

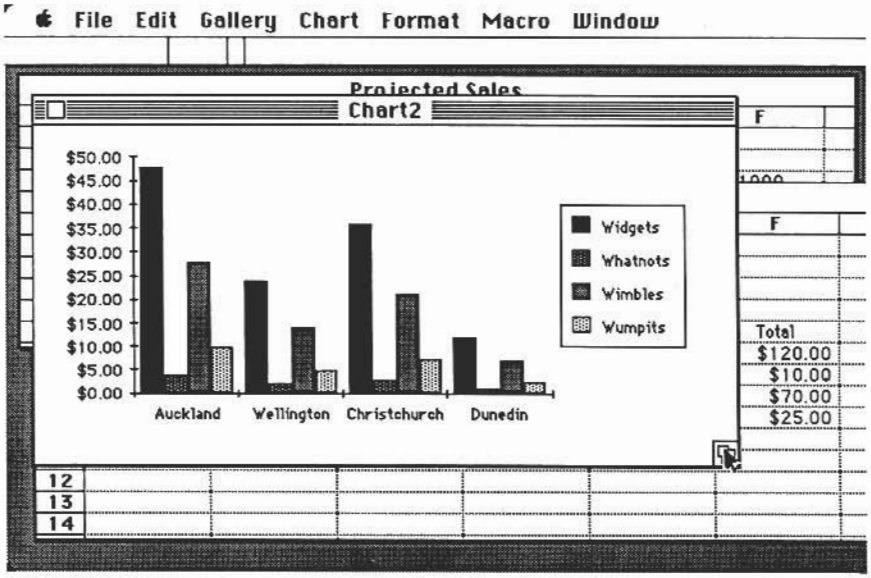
Excel offers a "learning" facility by which you specify your macros. Basically you can turn on a "recorder", and any operations you perform after that are stored in a special "macro sheet". This defines the basic steps in the macro, and these can be edited if required.

In 1-2-3 you have to put the macros somewhere on your spreadsheet. You generally chose some out-of-the-way corner and keep your macros there.

In Excel, macros are placed on separate "macro sheets", so they never clutter up your worksheet. Furthermore you can have as many different macro sheets as required, and a macro sheet can be accessed from more than one worksheet. This enables you to build standard "libraries" of macros.

Macro writing

A particularly powerful feature of Excel macros is the ability to write functions. These are a special type of macro to which you can pass parameters. A typical example would be a macro for calculating straight line depreciation to which you pass the cost of an asset, its expected life and its disposal value.



3) Excel's "guess" at the required chart from figure 2. The legend down the left hand side was added as a customised feature.

It would be helpful if there were macro commands for every operation that can be done in Excel. While this appears to be true for the worksheet and database, it is not true for the graphics. While the basic graphics commands do have macro equivalents, it is not possible to customise charts using a macro.

In 1-2-3 you can create your own means to run a macro, and you can also define special keyboard shortcuts. You can even define a special startup macro which will start running when you open the spreadsheet. In Excel you can only define keyboard shortcuts. This is a pity because it means that you can't "customise" your application as you can do in 1-2-3.

Summary

By and large I found that Excel lives up to its name. It has a very powerful spreadsheet as well as excellent graphics and macro facilities.

The database was the only area I found disappointing.

When the Macintosh was first released there was no worthwhile software available for it. It is only now, more than two years later, that we are starting to see well engineered and useful products such as Excel and Omnis 3 (a database system).

This, at long last, makes the Macintosh a viable business computer.

For the first time I feel I can recommend a Macontish to clients as a solution to specific information requirements. And an integral part of that solution will be Excel.

Grant Cowie, PhD, is a director of Systems, Science & Research Ltd, a software developer and computer consultancy in Auckland.

Attache upgrade

The Attache accounting package has been upgraded with GST reckoning as well as new versions of order entry and payroll modules.

CBA gets Q&A

Cowan Bowman Associates have acquired rights to distribute here the Q&A (Question and Answer) database written by Symantec (US).

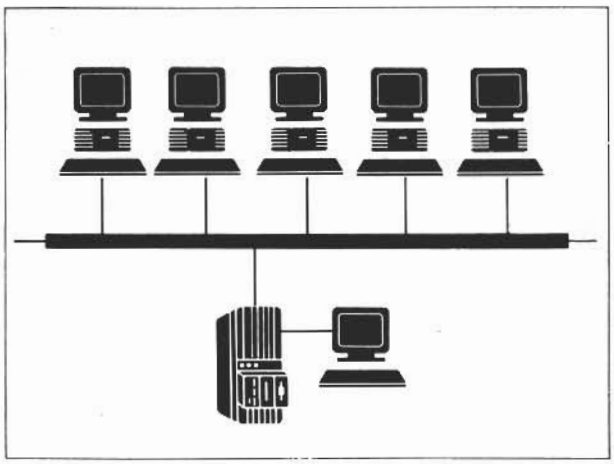
Data is retrieved by "natural language" commands like "show me the computer magazines published locally with a readership of more than 70,000, and give me the cover prices".

Using its "intelligent assistant", Q&A will ask for further details of those terms used which it does not (yet) understand.

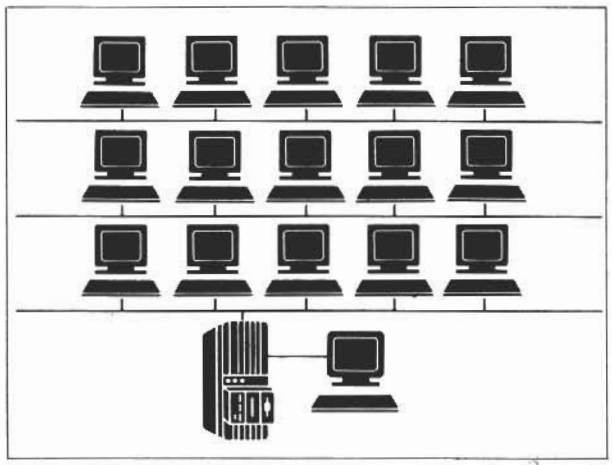
Meanwhile CBA has added a purchase ordering module to its CBA business accounting package. It is integrated with the accounts payable module.

Now there is a new standard in pc performance

The New ITT XTRA XL™ DOS/XENIX Super-Microcomputer

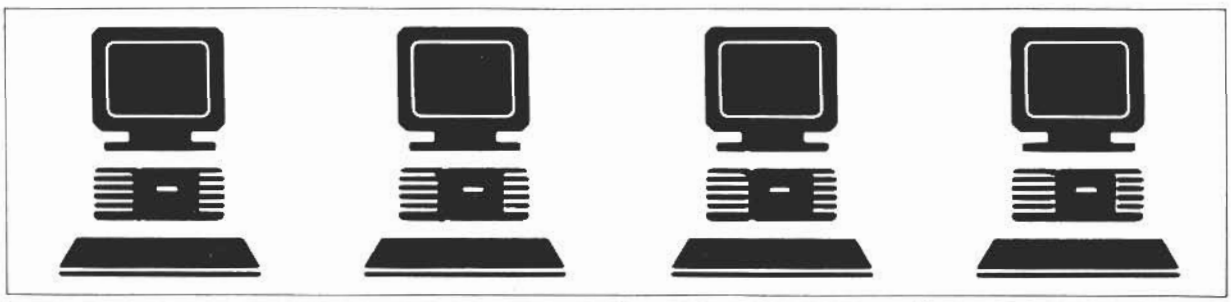


Operating in a LAN Environment



Operating in a Shared-Processor Environment

A powerful processor — ideal as a LAN server with high speed memory and disk storage option — or for multi-user, multi-tasking systems with significant capability for resource sharing and support for up to 16 concurrent users.
It's more than a PC — it's a super micro that rivals mini-computers twice the price!



ITT XTRA XT COMPATIBLE	ITT XTRA XP XT COMPATIBLE 30% FASTER THAN AT	ITT XTRA "WATCH THIS SPACE"	ITT XTRA XL SUPERMICRO & MULTI USER ENVIRONMENTS

STC DATA PRODUCTS (NEW ZEALAND) LIMITED
AUCKLAND PH 500 019, WELLINGTON PH 857 985
P O Box 26 064 Auckland, P O Box 40 140 Wellington

The best ideas are the ideas that help people. ITT

4469

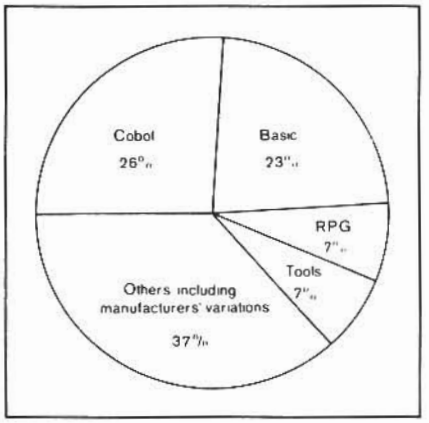
Our software industry exposed

from Martin Kaiser's 'discussion paper'

To establish what the New Zealand software industry companies themselves feel are their strengths and weaknesses, what can be done to improve the industry, and most importantly, who should take action, the DSIR conducted a comprehensive questionnaire survey of the New Zealand software industry during July/August 1984.

The resulting *Discussion Paper* contains the summary of results, conclusions drawn from the survey, and recommendations on the next steps to take.

247 (or 38%) of the organisations in the survey sent back questionnaires containing data on their 'software-for-sale' activities.



Languages used in the New Zealand software industry.

180 organisations in New Zealand are employing 800 staff to write, maintain, and modify software for sale.

Half the software companies have been selling New Zealand developed software for 3 years or less, and three-quarters of them have been selling it for 5 years or less.

Applications areas

Most of the 500 staff in these companies writing new software are engaged in the applications areas of business and accounting. The other main areas are in 'vertical' industries such as pharmacy, motor vehicles, travel agencies, etc.

Other development areas employing considerable numbers of staff are in the fields of process control and real-time systems, as well as fourth generation languages.

About 230 more people are employed in maintaining and improving software in very much the same areas as mentioned above.

When asked about speculative development of software (where there is no clear customer in sight), 113 organisations said they are spending 153 man-years of effort in this area.

About 60 organisations are involved in modifying or customising other people's software, and they used 110 people in this activity. About a quarter of this software has it origin in New Zealand. These people represent an additional pool of software expertise in the New Zealand industry.

Languages

The main languages used are COBOL, followed by BASIC, various software tools, and then assembler languages.

Those to be used more in the future are software tools of all kinds, followed by BASIC, and then other environments and PASCAL.

Nearly 40% of software writing organisations follow a semi-formal system of software writing methodology. A further 36% have some of the features of a formal methodology.

However, only 10% or 18 organisations use a named development methodology.

Motivation

Most of the organisations are selling software because of the deliberate decision to get into the market, or because of a contract froma client to write some software. Probably 20% of the respon-

dents say they are selling software as a by-product of a bureau operation or of a package originally developed for in-house use.

NZ-written sales

Sales figures for 1983 were approximately $19 million for one-off sales in New Zealand plus a further $4 million in annual charges. Export sales were more difficult to establish because of royalty arrangements but were of the order of $3½ million in 1983.

About 150 organisations were selling software in New Zealand, and 40 organisations were exporting New Zealand-written software in 1983.

For 1984, estimates were for about $25 million worth of local sales plus $5 million worth of annual fees, and over $7 million worth of export sales from 52 organisations. Local sales are up 35% and export 120% up.

The estimated figures for 1985 are $30 million for local sales plus $4 million for annual fees, and over $10 million for export sales plus over $1 million for annual fees on exports.

These represent an underestimate of the expectation of future sales.

As well as the above sales, there are some significant sales of modified or customised software; for 1984 about $7 million of local sales and $1 million of exports.

Exports are predominantly to Australia.

Marketing

The marketing technique said to be most effective is predominantly the use of respondents' own sales staff. A further 12% of respondents achieved sales through word of mouth, followed by independent distributors and mail order magazine advertising.

One-fifth of companies say they spend nothing on marketing software in New Zealand, and the average figure was somewhere about 9% of turnover spent on marketing in New Zealand.

Over a third of the respondents rank using their own staff based in New Zealand as the most effective export marketing technique. Independent overseas distributors are ranked next.

Surprisingly, nearly half of the exporters spend nothing on marketing software overseas. The remainder spend about 23% of their income from overseas on overseas marketing. A few developing their markets are spending from 50% to 100% of their overseas income on marketing.

Most software organisations in New Zealand are developing software predominantly for manufacturing and primary processing applications together with finance, insurance, and business sectors. These are followed by the wholesale, retail, and travel industry sector.

The next two sectors in terms of number of organisations active in selling are building and construction, and agriculture, forestry, etc.

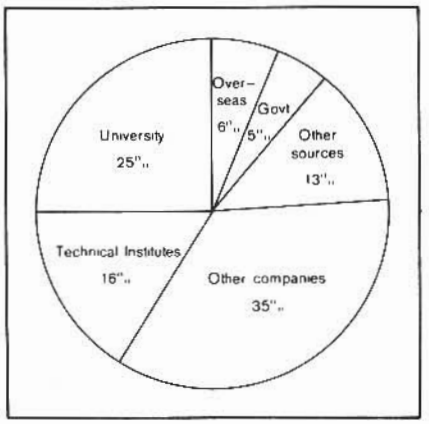
On the export side, the organisations are mainly exporting software to the finance/business sector and the manufacturing/primary processing area.

In terms of sales figures achieved, however, the leading sector is that of finance/insurance/business services.

Local copying

In New Zealand 46 organisations reported that they are copying software under licensing agreements for sale in New Zealand. About 75% of their sales involved this type of software manufacture. It is sourced predominantly from the USA, followed by Australia and the United Kingdom.

14% of software writing staff have a computer science degree, and a further 33% another form of degree; 24% have a New Zealand Certificate in Data Processing or an equivalent polytech diploma.



Sources for recruiting staff.

Staff recruitment

The recruitment sources of software staff are given at 35% from other New Zealand companies, 25% from universities, and 16% from technical institutes. Recruitment from overseas was 6%, and from government departments (in 1984) was 5%. A further 13% are recruited from other sources.

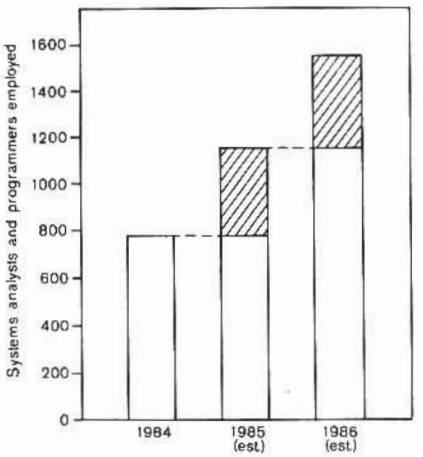
About 40% of software companies claim to take on trainee software staff.

The forecast of analysts and programmers needed for writing software for sale is an increase of 45% on existing staff in 1985 and a further 36% increase in 1986. These figures indicate an extra employment of between 350 and 400 people in each of those years.

Use of government

Under one-quarter of the software writing organisations had contacted government agencies or universities and polytechnics for assistance during 1983. About one-half of the organisations writing software use at least one of the agencies above.

The universities and polytechnics achieved an 84% 'satisfaction rating' for those enquiries, the DSIR 66% satisfaction, the State Services 59%, the DFC Applied Technology Programme 56%, and the Department of Trade and Industry about 58%.



Numbers of systems analysts and programmers employed.

Growth and strengths

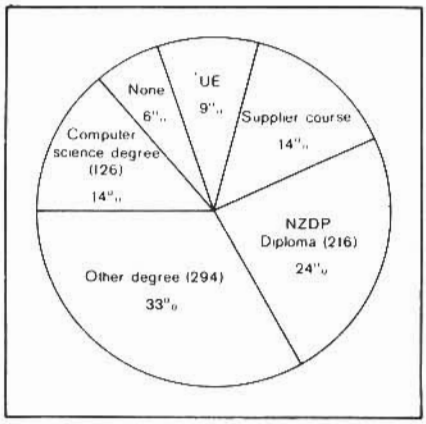
Factors that would help growth of the industry include reducing tax and giving incentives, improving marketing skills, improving the marketing skills, improving the training of staff and the availability of capital.

The strengths of the New Zealand industry are perceived as the flexibility and low wage rates of the staff in the industry, followed (some way back) by the good design and user-friendliness of software and the efficient use of hardware resources by software that is written in New Zealand.

Stagnation and weaknesses

Factors hindering the growth of the New Zealand software industry are perceived as poor skills in the marketing area with a resulting low demand for software, followed by the shortage of experienced staff and the high cost of capital (on about equal footing).

Then follow various comments on poor management skills in software



Educational qualifications of software writing staff.

companies, costs and delays in obtaining hardware, the problems of sales tax, and the low volume of government subcontracting.

Evaluating specific factors that hinder individual companies' growth in the software industry, organisations listed the sales tax on hardware as the most serious problem (since then cut), followed by a lack of finance and the small home market.

Other factors are the difficulty in supporting products overseas and software sales tax.

The least concern is shown by respondents about the quality of their product, about the piracy of staff, or about product liability.

The weaknesses of the New Zealand software industry are perceived very much as the poor marketing skills and the subsequent low demand for software. A long way back came the high costs of capital and the lack of management skills in software companies.

Improving ways

Finally, when asked what can be done to improve the performance of the software industry, people responded with ideas for co-ordination and improvement of marketing skills, reducing tax, and giving incentives, and then improving the training and experience of staff.

There is a clear majority of feeling that the government should take up these ideas, even though there were a few organisations that said no to this action.

The next most popular organisation to take up the ideas are the trade associations, followed by somebody in the respondent's own company.

Of the companies who responded to the survey, most have a low percentage of their turnover representing software sales, but 39 companies had the majority of their turnover representing software sales.

88% of the companies are fully New Zealand owned.

The amount of resource that companies are using for software development and sales were $43 million worth of resources split up as $13 million land and buildings, $23 million computer hardware, $4 million software tools, and $3 million other resources.

This works out at $77,000 of capital for each employee who is writing software for sale. (However, these figures could be inflated by respondents including facilities that are used other than for the development of software for sale).

Brief conclusion

For New Zealand's population, we have a similar proportion of software companies to the numbers found in Australia, USA, etc. However, our companies are smaller on average (at 4 staff per organisation) than the Australian ones (7 per organisation) or the Americans (35-60 per software organisation).

Our software companies on average are young, but our software development staff are skilled, flexible, experienced, and cheap by world standards. 47% had degree-level qualifications.

Our major software exporters are selling medium to large unit cost software products, mainly on mainframe and larger minicomputers.

A few exporters have a higher volume trade in medium to low unit cost accounting packages for microcomputers. Most exports go to Australia, followed by the USA. Little goes to the UK and Europe (similar to the Australian findings).

We are exporting a higher proportion (17% in 1984) of our local software production than is Australia (about 10%).

Marketing is identified as a weakness of the New Zealand industry by respondents, with a certain lack of confidence in exporting.

New Zealand organisations do not use existing (government) agencies much for assistance, and only 39 out of the 180 active organisations were members of the trade group (the New Zealand Computing Services Association).

From a recent Australian survey (discussed in *Resource Paper 11B[8]*), it is clear that they, too, have identified poor marketing of software as a weakness, but their Department of Trade has taken positive steps to develop a market strategy.

It is using its Trade Commissioners' services, has commissioned market research, sent survey missions overseas, and promoted joint industry exhibition stands in order to help the software industry develop its potential for export growth.

Recommendations

Specific recommendations follow for the three main partners in the industry: commercial companies; the trade and professional associations; and the government/public sector.

Commercial companies

1 Software companies should develop or hire good management skills. (It is not good enough to have good computing skills – career paths must be thought through.)

2 Companies need to seek out (niche?) markets and follow them up on a planned basis. (Markets are there for the entrepreneur, but not in an ad hoc way.)

3 Professional marketing strategies have to be developed, particularly for overseas markets. (Using established overseas-based companies who know their markets is essential unless you can afford to run your own overseas branches.) These facilities cost money, and companies must be prepared to spend more money on marketing their products.

4 Small companies will have to look to joint ventures or amalgamation with larger companies who can supply marketing skills, overseas branches, finance, etc. – otherwise they will be taken over or stagnate.

5 Companies should use software productivity tools more, to minimise the effect of skilled staff shortages, as these shortages will exist for many years to come. Only about 10% of software development uses the current software tools.

6 Companies must develop more ongoing professional skills in their existing staff and be prepared to take on trainees and not just poach other companies' staff. Such poaching does not contribute to the future development of the industry as a whole.

Professional associations

7 A formal co-operative should be set up through the trade associations for those selling New Zealand developed software overseas. This organisation could promote a range of software nearly as cheaply as selling just one product. Some individual companies could be persuaded to sell complementary software from other New Zealand suppliers.

8 In presenting a case to government for assistance for their industry (in contrast to other industries) the trade association must clearly represent the whole industry. The NZCSA needs greater membership numbers and increased co-operation from the other interested parties.

9 A formalised system of feedback about market conditions and opportunities should be established (from New Zealand's trade posts to the trade associations). There should be guidelines about quality assurance procedures and documentation for different market places and sectors. Trade associations should investigate and determine standard terms and conditions for overseas contracts.

10 Companies and individuals exhibiting at overseas trade fairs should co-ordinate their exhibits – to jointly display New Zealand software (the Department of Trade and Industry can help).

11 Feedback from the software industry to (government) agencies about services they offer.

12 Action by people in the industry needs to be taken to define the educational standards needed for productive programmers and feedback of this to those in the training field. The current DACUM projects of the Vocational Training Council need wider understanding and support.

13 Guidelines and a code of ethics for the industry need to be developed. (Companies should be seen as professional and supporting their packages on an on-going basis.)

14 Trade associations and professional bodies should continue to increase their involvement in training to ensure a range of professional development seminars.

15 The registration (by the New Zealand Computer Society?) of software expertise in 'uncommon' languages to assist in forming project teams for special projects, etc.

16 The New Zealand Computer Services Association should examine the cost/benefit to its smaller members, and promote its membership services more.

Government

17 Initiatives by the Department of Trade and Industry are needed in the areas of general market feedback from trade posts by reports and by returning officers meeting with trade associations as indicated above.

18 New Zealand Department of Trade and Industry should re-examine the value of software missions to identify software market opportunities overseas. The last such exercise was completed several years ago.

19 Encouragement by government by means of targeted research and development grants and subsidies, market development assistance, and various tax incentives such as faster depreciа-

tion allowances.
20 Equitable and consistent application of taxes to computer and software companies.
21 More subcontracting of work by Government departments to the private sector and to New Zealand companies. There is a tendency to use overseas companies in preference to local companies.
22 Improve the relevance of computer training and education and increase significantly the number of places available for training staff for the fast-growing computer industry. Many employers do not want over-educated staff, and prefer vocational training. Commerce and business skills are as important as computer skills.
23 New Zealand should follow the Australian Department of Trade's actions in supporting its software industry. In particular their suggestions are to gether information on marketing methods and their success; and the development of an export strategy for computer software in close consultation with the industry, with the following elements as a basis:

- The US market to continue as the main focus.
- More attention to market research in UK/Western Europe.
- Medium priority for other markets.
- Assistance to promote specific product areas.
- Specialised trade exhibitions.
- Further investigation into firms interested in software exporting.
- Monitoring export performance of the core group.

24 Identify potential exporters of software, and services they require from both trade associations and government agencies.
25 Improving export of computer-based professional services and software.

Govt's role reviewed

The former Governor General, Sir David Beattie, will head a Ministerial working party set up to review the government's role in science and technology.

"It's timely to stand back and look at how we can best foster science and technology in New Zealand to benefit our country economically and socially for the future," says the Minister of Science and Technology, Mr Bob Tizard.

Working party members will also delve into funding of public sector research and the best areas and types of science and technology activity for the various organisations involved.

The team has also been asked to report on the extent to which Government science agencies should be self funding by charging for work and advice, and ways of encouraging private sector investment in research and development.

Their first meeting was early July and they expect to complete their review by the end of November 1986. The review follows a series of changes by the Government to state funding and the implementation of a user-pays policy.

Word 3 "outliner"

NZ's Microsoft distributor, Brimaur Computer Services, has released here Word Release 3, a word processor having several integrated functions.

New functions include an outliner, mathematical and sorting abilities, and expanded support for laser printers.

Running on MS-DOS, Word 3 also includes the earlier Word's document formatting aids, mail or list merge, spelling checker and windows.

A writer can switch to an outline of an intended document, assigning type styles and placement of headings and body text. Long documents can be telescoped to display only major headings, and when the headings are rearranged the subsequent text is re-ordered accordingly.

Wyse attacks AT

Wyse is attacking the IBM AT with its Wyse PC 286, an AT-compatible micro which runs at 10 Mhz or in a 6 Mhz "compatibility-mode".

The price is the aggressive part. The PC 286 with a 1.2 Mb single floppy drive costs $5,765. The 20 Mb hard-drive model costs $6,935 and 40 Mb model $9,780.

In the US, Wyse has a "co-labelling" agreement recently signed with Businessland, a retail chain, for selling $60m worth of the Wyse PC 286 over a "multi-year" period.

Sybiz dealers

More than 40 dealers of Sybiz business software have been recently appointed for nation-wide support of the Australian-sourced accounting packages.

Sybiz' NZ manager, Paul McLuckie, says the new network and "quality control approach to software marketing" is a radical departure from conventional attitudes of software vendors.

"The attitude has tended to concentrate on achieving the sale, but to then fall down on providing the customer with continuing support and effective user training".

The Microbee-Mitac Portable PC

by Paul Left

As new IBM-compatible personal computers come on the market, it is tempting to ask "why bother"?

Certainly, if all machines were merely clones of the original, the question would be justified.

However, each machine has distinct advantages and is aimed at a particular market.

Some look-alikes seem intended to be general-purpose desk-top work-horses, while others (like the very stylish and very portable Kaypro 2000 reviewed in this magazine recently) are notable because they seem to be aimed at some particular niche in the marketplace.

While it is important to determine priorities before buying, any purchase is likely to involve some trade-off.

How many users wish their desk-top machine was not so difficult to lug home for the weekend?

How many portables never leave the office, while their users curse the tiny screens and keyboards?

Australian

This review is based on a brief encounter with one of the latest additions to the IBM-compatible market in this country, the Microbee/Mitac Portable PC.

Microbee has been a dominant force in the Australian market for several years, producing a CP/M machine mainly aimed at the educational market.

Earlier this year, Microbee signed a marketing and distribution deal with Mitac Incorporated, a Taiwanese manufacturer of small computers.

This arrangement has led to the introduction here and in Australia of the 16-bit MS-DOS Portable PC. Microbee's promotions have attempted to contrast their machine with other PC look-alikes, stressing that many of them are 'from unknown and probably short-lived Asian manufacturers'. This implied criticism seems below-the-belt considering their own machine's origins in Taiwan.

The Microbee PC is fairly typical of the MS-DOS machines produced in Asia, which have a firm foothold in the market here.

Be that as it may, this computer may be just innovative enough in overall design to make a significant impact in this market.

Appearances

The main casing is a little reminiscent of the Apple IIc, with its off-white colour and a single 5.25 inch floppy drive around on the right side.

The machine has a small footprint and is light in weight (see the specifications summary), with a carrying handle which folds away under the keyboard end of the housing.

The keyboard has all the familiar PC keys, but with a modified layout.

The 10 function keys are in a row above the number keys, the CAPS-LOCK is up there too, next to the NUM-LOCK key, the four cursor keys are in a row next to the space bar, and the PAGE-UP, PAGE-DOWN, HOME, and END keys are in a vertical row on the right hand edge of the keyboard.

There is a HELP key, and the CAPS-LOCK and NUM-LOCK keys have LED indicators.

There is no separate numeric keypad, as the NUM-LOCK key controls a block of alpha-numeric keys towards the right of the keyboard. The keyboard is not detachable from the main unit.

Despite the compact layout of the keyboard, the keys don't have the cramped feel of some portables.

The keys have a 'clicky' touch but I found them fine to use, although the non-alphabetic keys are a little confusing at first. I was not able to do any serious word-processing to really test the machine, however.

Memory

The machine comes with 256K RAM, expandable to 640K. As the ROM BIOS is that developed by Phoenix Software, IBM compatibility is claimed to be very high.

The DOS included is MS-DOS 3.10, which comes on the only disk packaged with the machine. The built-in disk drive conforms to the now-commonplace 360K double-sided standard.

For serious use you would probably need to add a second drive connected to the port at the rear of the machine.

It is also possible to add a 20 Megabyte hard disk. This requires the connecting of an expansion box, which provides 3 expansion slots and room for 2 floppy drives, the hard disk drive, or 1 floppy and a tape backup drive.

A Colour Graphics Adapter is standard, an RGB and a composite video port are both provided, and an LCD display screen will also be made available.

This will provide 25 lines of 80 characters when plugged into the RGB connector, and will be enabled through software commands.

Other add-ons will include an Enhanced Graphics Adaptor and a Microsoft-compatible mouse.

The EGA will presumably require the use of the Expansion Box.

In/out ports

The Microbee has a Centronics parallel port and two RS232C serial ports built-in, corresponding to LPT1, COM1, and COM2, and a 'game' port for a joystick or mouse.

Also built in are a real-time clock with battery backup, so the machine should always boot up with the correct time and date displayed.

Given the reluctance of most users to bother setting the time when starting their machines, this could be an important convenience if keeping the system time correct is important to you.

Inside the compact case, the Microbee appears well put together and neatly assembled. The whole system seems reasonably robust and without obvious flaws, including the carrying handle, sometimes a weak point on portables.

The amber monitor supplied with the review machine provided a clear, crisp display, although it was very different in appearance from the monitor shown in Microbee's publicity.

The monitor I saw was definitely a no-frills model, while the one in the photographs appeared to be of superior design, with adjustable screen position.

Disk problems

One very serious problem I did have was with the disk drive: I had to reboot some disks several times after an initial Read error.

I also had problems formatting disks, with a mis-spelt message telling me that DOS had 'flaged' (sic) 30 bad sectors.

The disks in fact appeared to be fine when tested later on a different PC look-alike.

I could not use the DOS Diskcopy command as this could not read the source disk, so I was forced to use the original DOS disk for the time I had with the machine.

Also the documentation claimed that I could copy a file with only one drive by specifying drive B: for the target drive.

This did not work, so it appeared that unless some modified DOS is offered in future, you are unable to use the computer in any meaningful way unless you have two drives.

I was unable to determine whether these problems were the result of incompatibility between the machine and the DOS version, a fault with the drive, or just a matter of drive adjustment.

The price

Only extended use will determine the overall reliability of the Microbee PC. My general impression was that of a reasonably well-constructed machine, but with serious doubts about the disk drive, although this was possibly a matter of adjustment.

How is the Microbee going to find its niche in the market for 16-bit MS-DOS machines when there are already so many on the market?

The main selling point will be price.

You will be able to buy just the computer itself for $1895. That means a single-drive, bare-bones system with amber screen will be available for around $2100 and with 2 drives for around $2600.

You could add the 20 Megabyte hard disk for $1795 plus $495 for the expansion box. An LCD screen will knock you back nearly $1000, however.

A cheaper option if you wanted to carry the machine between two locations would be to buy a second monitor (for around $200) so that only the CPU would need transporting.

Compactness

Another selling point will be the small footprint of the Microbee.

While it can't be considered truly portable without the LCD screen, it must be an advantage to avoid that bulky CPU casing that many IBM-compatibles are

lumbered with.

If you don't need the expansion box for adding cards to the system, the Microbee will take up little room on your desk, avoiding that 'no room for my bits of paper' feeling. Small computers can be much friendlier to work with and will become more popular providing they prove they are as reliable as the bigger machines.

A small computer is convenient for the same reasons as a portable television: it's easily moved from room to room so that some of the family can use it without annoying the others.

Sensible outs

I was also impressed with the built-in connection ports of the Microbee. With some of the IBM-compatibles now available, problems arise when adding or changing peripherals.

For example, changing to an RGB monitor often means adding a card to one of the slots. Then, because your parallel printer port is probably on your discarded monochrome card, you also have to add a parallel card so you can use your printer.

With the Microbee, you can plug in colour and monochrome monitors, parallel and serial printers, and a second floppy drive without opening the case or adding cards. This sort of convenient and sensible arrangement should become universal.

Heat

One problem that can occur with very compact machines like this is overheating. The Microbee was running most of one wet Sunday while half the family rediscovered Frogger, however, and stayed quite cool throughout. This was a 256K machine, of course: with its full complement of 640K RAM, however, the Microbee could have difficulties dissipating the extra heat as there is no built-in cooling fan.

However, current chip technology allows machines to run much cooler as power consumption is reduced.

Summary

In summary, the Microbee is a compact, portable MS-DOS machine that should theoretically run many software packages available.

The computer is well-endowed with peripheral ports and has built-in colour capability, all of which should provide a flexible and easily-expanded system.

The computer comes with no bundled software, just a system/DOS disk which is not suited to a single-drive system and possibly is incompatible with the hardware.

The disk drives are the most dubious aspect of the machine; I would need to be convinced that the problems I experienced with copying and formatting were the result of a badly-aligned disk drive head, or some other easily-remedied fault, before I purchased a Microbee Portable PC.

Microcomputer summary

Processor:	80C88, 4.77 Mhz
Operating System:	MS-DOS 3.1
RAM:	256K standard, up to 640K
Disk Capacity:	1 built-in 5.25" floppy, 360K. Connector for second floppy drive. 20 Megabyte hard disk optional.
Dimensions:	340 x 420 x 70
Weight:	4.98 kilograms
Keyboard:	81 keys, keyboard not detachable.
Ports:	1 x Centronics-type parallel. 2 x RS232C serial.
Display:	Colour graphics, RGB and composite connectors. LCD screen optional.
Expansion:	3-card box available.
Prices:	

Microbee/Mitac Portable PC:	$1895
Nylon Carry Bag:	$79
Second floppy drive:	$599
Expansion Box:	$495
20Mbyte hard disk and controller:	$1795
LCD display screen (monochrome):	$999
Serial Mouse:	$189

Microbee Comments

The author mentioned a "serious problem" with the disk drive. It appears that resting a conventional monitor directly on the Portable PC can produce interference with the heads of the drive. We have found that when a monitor stand is used, there is no problem at all. It is possible that Mr Left rested the monitor directly on the computer thus causing an operational problem.

The other difficulty that Mr Left mentioned was regarding the copying of the distribution disk. Portable PCs have internal dip switches to select the amount of memory that is installed and the number and type of drives being used. The Microbee that was reviewed had its switches set for two drives – it should have been set for one drive.

Staff at Microbee Systems in New Zealand have found that disk copying does work as per the manual when the dip switch is selected for one drive. Microbee Systems is emphatic that Portable PCs will be correctly installed for all customers.

The Quest for the Euro-Micro

by Pip Forer

Europe's educational computing market is potentially enormous. Even if you shave off just the EEC it is a market rivalled only by the United States in size.

However, whereas the US is effectively a single market for micros with a single, dominant spoken language, Europe is split by culture, language and national identities into many sub-markets. Even in the EEC, with few trade barriers, national interests and language preferences can have big effects on goods such as microcomputers.

Yet Europe is always trying to keep its own high-tech initiatives alive and free of American domination, and is always dogged by the lack of direction and co-operation of its various countries in so doing.

Examining how Europe has coped with the changes of the last five years tells us a lot about how the educational computing market works. And to understand it we need to know something about two things: the history of educational microcomputing in Europe, and the vested interests that have influenced the European micro market.

Vested interests

The pressure groups acting within educational computing are numerous.

Initially there are the computer companies.

There has long been a conventional wisdom that education is a strategic market for computer manufacturers. In spite of the eccentricities of many of the educational users it remains an attractive area in which to sell.

Computer companies have consistently shown themselves keen to influence the choice of machines that schools and colleges buy, which brings a certain spice to events. The interests of the companies however, are by no means simple to describe.

While all would like to see their machines sold widely the stances adopted in pursuing this aim differ greatly, particularly depending whether they are national firms, European firms or predominantly American firms (the Japanese have little part in this particular tale).

Part of this aspect is reflected in the influence of national pride. A country may not care if its banking is processed by a machine from Boston or Santa Clara, but it is far more sensitive as to who made the machine little Brenda or Bill sits in front of at school.

National firms think that this attitude is a good idea, even if only the badge on the front of the machine was actually made anywhere outside of Taiwan or Singapore. They form a lobby for a national microcomputer.

Standardisation

Another influence has emerged somewhat later. This is a rather complex lobby calling for standardisation of equipment for educational reasons (all the hardware companies applaud this, just so long as it is their machine that gets the nod).

Such standardisation, coupled with policies to introduce machines as regulation school equipment, can create

overnight an enormous and often easily influenced market. Naturally this further excites local hardware manufacturers, who can see a machine that might not be too marketable in free competition instead being imposed by political fiat on a captive sector like education.

There is also a potent lobby for change and growth. This is not comprised just of machine manufacturers (who must innovate to keep sales going) but also members of the spin-off industries. In particular software firms (or educational software developers under contract) want to experiment with new realms. If 8-bit machines are exhausted in terms of feasible innovations then they will argue for 16-bit ones.

Finally the whole shebang gets caught up in national policies for high-tech development. Politicians seek to encourage job growth and new skills by encouraging research and development, plus jobs by manufacturing. Micros, especially those in schools, fit the bill for publicly funded development policies, along with half a dozen other new initiatives.

Governmental support

Where Europe differs from America is that some influences in this are much stronger in the countries of the EEC than in the US. Direct government support is stronger, as generally is national influence on aspects of the national education system.

However, while the bureaucratic forces are stronger the overall market is far weaker, heavily partitioned and the research and development levels far lower.

Standardisation – sensible, convenient and stultifying'

This has produced a particular pattern of response over the last six years as Europe tries to come to terms with competition for U.S. machines in education.

All these pressures at work have the makings for a tale of Byzantine intrigue in which the free market, at best, ends up somewhat compromised.

Certainly it is true that the tensions felt after the first introduction of computers into schools have not subsided. Rather the manoeuverings and demands have grown.

As bigger (if not always better) machines and as more vested interests have entered the fray so the stakes and intrigues have grown.

Not surprisingly, the grail of an acceptable educational microcomputing standard has come no closer to attainment. The New Zealand tale is fairly well known. Rejoice that we are not alone: the problems are world wide.

History lessons

The history of European educational computing gives some clear pointers to how all these forces interact. It really goes in two rounds.

Round one started in Britain in 1981 when the BBC microcomputer led the first European responses to the threatened dominance of American hardware (at that time Apple II+s and Commodore Pets).

Its development can be seen as an alliance between British industry and government in which the British Broadcasting Corporation's stature was enlisted to promote a British microcomputer for home and education.

The surprise winner of the design tender for this, Acorn, produced from a hat a machine for which I still retain enormous respect.

JOIN THE GLO

Yesterday you had a P.C. with a keyboard at your fingertips...

... today you have the world!

COMCOR TECHNOLOGY MODEMS
(formerly Fountain)

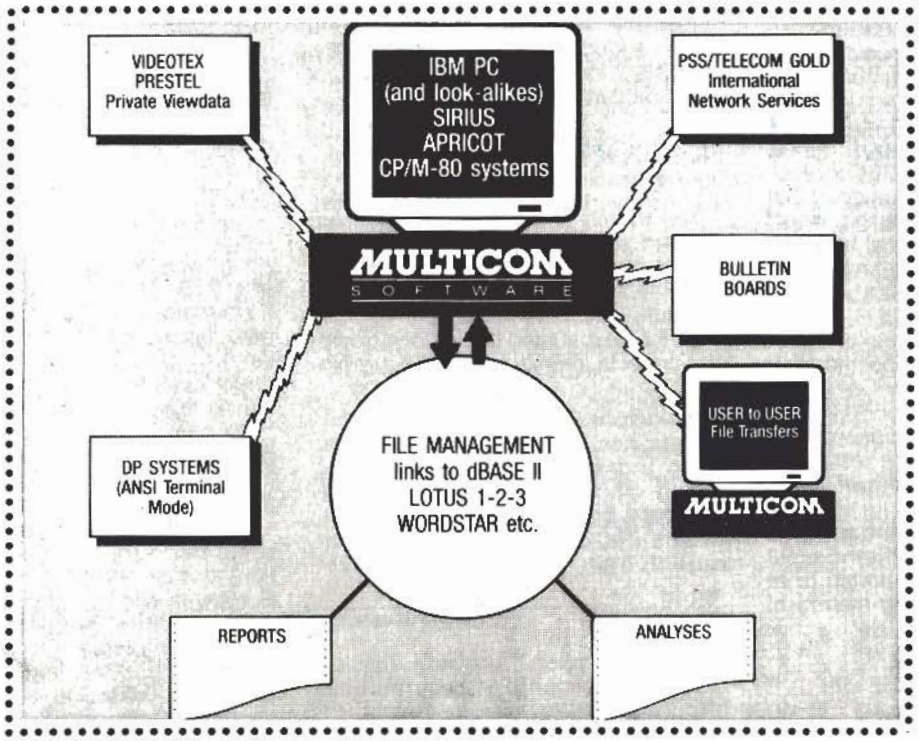
- P.O. Approved
- Reliable
- Low cost
- Simple to Use
- Small
- 1 year warranty

SPECIFICATION • MD 312 – V21, V23 (videotex, 1200/75, 300/300).....$399

• MD 412 – As per 312, but with AUTOANSWER.....$549

- Interrogate data bases like The Source in the USA.
- Connect up to NZ Videotex Systems & have up-to-date information on almost any subject; check your bank account and pay your bills from your keyboard.
- Use STARNET as your electronic mail box to send and receive messages worldwide, and plug into the TELEX system.
- Save a fortune on phone calls, with a modem you can connect your PC to anywhere in the world for $12 per hour.

OBAL VILLAGE



MULTICOM SOFTWARE

Without a doubt, superb COMMUNICATION software for the IBM.PC, XT, AT, & compatibles

- *MULTICOM will turn your PC into a Videotex terminal, with powerful command files to automate data retrieval and processing.*
- *MULTICOM provides SCROLLING MODE for connection to ASCII data base STARNET.*
- *MULTICOM offers VT 100 emulation to connect PC and host. Recommended Retail Price $399.*
- *MULTICOM works with any modem.*
- *The combination of MULTICOM & the CTL MODEM provides your total communication solution for only $798.*

Available from PC dealers throughout New Zealand.

DISTRIBUTED BY

MACHINE WARE LIMITED. PRIVATE BAG
TAKAPUNA

Interestingly the funding supporting the machine's initial sale to schools came not from Education but from the Department of Trade and Industry, which continues to be a major sponsor of new educational technology.

Localised offerings

Just across the Channel, Britain's fellow Europeans watched developments with interest. Would the gambit be an educational and commercial success?

As acceptance mounted in the UK some countries produced local machines in imitation of the BBC project. (Most sat on their hands and hoped that something as expensive as this would go away from the educational budget.)

The French produced elegantly Gallic Micral machines, but stumbled on external sales.

Philips, a very likely source for a new machine, produced a disappointing first offering in Holland.

The Yugoslavs wrestled with implementing a system for two alphabets and five national languages. The resources needed for the development of these machines was minimal.

Jobs and Wozniak had done a fair job in a garage for the Apple II, and most European countries could stretch to at least a few hundred square metres of factory space.

'Civilised chaos'

In general, national funding to purchase micros was also minimal, particularly where only overseas machines were available. Nonetheless innovative institutions and individuals had already found finances from unexpected sources to experiement with Apple IIs and Commodore Pets.

They proceeded to extend their experiments with all manner of machines. From Lisbon to Latvia the cakestalls were busy.

The Germans liked Commodore 64s, the Irish leaned towards Apple... all with a rich variety of alternatives.

With the exception of the BBC micros, most local products came too late and were inadequate in open competition, even in their home markets.

Confronted by two aging American designs the Europeans produced two dozen ineffective competitors. The nett result was typically European; a civilized chaos.

The BBC micro meanwhile headed towards maturity spawning a whole retinue of government-backed, software-design projects to make it educationally ever more useful. By 1984 more and more of this activity had resulted in published software of comparitively high quality across a wide range.

Software, of course, is the essence of any machine' worth. Educational software (as opposed to general-purpose creative software) is the hardest to write and least profitable to produce; one reason why it is usually the slowest to appear.

Three years after the BBC micro's appearance the resource became significantly large. Anglophone countries (especially Australasian ones) found this an added attraction of the machine. Naturally it looked less useful to a Saudi or a Spaniard.

MS-DOS backers

The march of technology caught up with the software developers. There is always a lag between what is an exciting machine to work on and a practical one to teach with.

As they exhausted the options of simpler projects, members of the establishment of educational software developers naturally began to propose more ambitious programs. Many sought the power for this from 16-bit machines, most of which emanated from the business-oriented design of the IBM-PC.

'Will national pride continue to get in the way?'

The natural allies in this were the educational bureacracies.

One of the (some would say spurious) attractions of any MS-DOS machine is compatibility to a standard. This, and the promise of an existing software base, began to attract new admirers amongst educational administrators.

They began to wonder whether this could be the solution to the distressing plurality of machines, as well as offering an experience of 'real world' computing.

The real crunch however was that a third group joined the alliance when it became appreciated that MS-DOS machines were simple to implement. Even a local manufacturer could do it! The stage was set for round two of the game.

Round (ding!) Two

In round two, Europe is moving with more uniformity, at least in timing and intent.

A great deal has been happening in 1985 and 1986 that is moving several countries to a new stage of development.

However, once again, they are moving individually. Several European education systems are set to make a significant commitment to hardware and to recommend 16-bit systems to complement their 8-bit machines.

Ever willing to learn from the 8-bit exercise, the Europeans have predictably decided to do this by designing a whole new suite of strictly national machines.

This time they are not quite so late, and technically many are marginally superior to the obvious overseas rivals.

Sadly however, most of these local designs are incompatible at the hardware level with each other and with the industry standard.

The signs suggest that Britain will buy British machines (and Nimbus and Acorn Master 512). Holland is looking at a three firm consortium to establish a standard on which all three may compete (including Philips and IBM Holland).

Denmark is said to be working along similar lines.

All plan to put significant effort into developing their educational software base to fit this new environment over the next three years.

A true European optimist might welcome all this as a step in the right direction, a true flowering of the spirit of Europe (e pluribus pluribus, for fans of mottos).

A true observer of the computer scene is likely to see it in far less rosy terms. It could be seen as an unfortunate combination of disastrous timing and inappropriate technology: too arcane and/or too late.

Half-baked

There are at least two reasons for this judgement. The first is that any standardisation has very real dangers in times of dramatic technological change. It is sensible, convenient and stultifying.

Half-baked standardisation isn't even sensible and convenient, and is far costlier than buying Taiwanese PC clones.

The more prosaic argument is that if you are designing software for 1989, you do not want to do it within the limits of a basically 1980 sytstem.

If clever people are continuing to offer significant new products with far better pedigrees for use in education then why not use some of their ideas if you are starting a new phase of development? In particular, are 16-bit machines appropriate?

There is an argument that says that for the non-specialist a machine should be simple to use, or clever enough to make itself seem simple. In education the good eight-bit machines represent the former option, the newest generation

of large-volume micros the latter. The MS-DOS machines sit uncomfortably between the two.

New generation

Of course the newest generation machines are all American, and this time they didn't come out of a garage, nor are they reproducible by the sincerest form of flattery. The Macintosh, Atari 1020 and the Amiga all demanded considerable software and hardware research.

In the States the micro market has shaken out dramatically so that only the big producers and big research and development spenders have survived. To match the results that sort of commitment produces, it may not prove enough to hold a competition for the most desirable PC hybrid.

To some degree this is accepted by European educationalists. Talking to teachers, many seem to feel adequately served by their eight-bit machines. Most software developers admit to an admiration for the latest American machine developments. Some actually go so far as to use them.

Asked what they will implement on them however, and most sadly shake their heads and admit they can politically see only one machine they can get support to work on: the local offering, whatever its limitations.

Few believe their governments will handle the concept of national software run on an international machine.

Only a beginning

The misgivings from this approach are heightened when one recognises that the current American machines are only the start.

Carnegie Mellon University in the United States has started releasing details of the 3M micros they have been involved in developing.

These are not produced by the well known tape company: the title represents a piece of numeric fetishism. A million bytes memory, a million instructions per second and a million pixels on the screen (1000 by 1000).

IBM and Digital have worked on the project defining a system standard, a standard in which Apple amongst others have announced intentions to participate.

It has an environment strong on graphics, mice and windows and it is said that a heavily disguised (to the user) Unix forms the foundation of its operating system, which has been designed with the needs of education in mind.

Against this, how will the European 16-bit offerings fare?

In the short run, pride and profit may be satisfied.

It remains to be seen how it will satisfy the real, long-term judges: teachers and students.

High tech is inevitably high cost and high capital investment. Nothing is deader than an obsolete micro, and there have been plenty of new launchings already which have fitted that bill.

To match what the 6800-based machines can offer, resources need to be concentrated – and the Europeans have started to recognise the challenge.

Acorn, Olivetti and Thomson (France) represent a significant proportion of the European market and have spontaneously initiated design studies for a European educational standard.

Clearly only some major collaboration along this line offers any real hopes of success.

The technology is there, and the expertise. The question is whether it can be harnessed.

Will it be third round lucky? Or will two millennia of bad habits assert themselves yet again?

The ST Series

POWER WITHOUT THE PRICE

520ST MONO PACKAGE

With SF 354 360K 3½" Disk Drive
SM 124 Black and White Hi-Res. Monitor, Mouse, GEM and TOS
Plus BASIC Language
LOGO Language
1st Word Wordprocessor
Neochrome Graphics programme $2490
Special Introductory Offer — $2290

520ST COLOUR PACKAGE

With SF 354 360K 3½" Disk Drive
SC 1224 Colour Monitor
Medium Res.
Mouse
Plus BASIC Language
LOGO Language
1st Word Wordprocessor
Neochrome Graphics programme $2995
Special Introductory Offer — $2790

1040ST MONO PACKAGE

With SM 124, black and white Hi-Res. Monitor, Built-in SF314 720K 3½" Disc Drive, Mouse, GEM and TOS
Plus BASIC Language
LOGO Language
1st Word Wordprocessor
Neochrome Graphics programme $3295
Special Introductory Offer — $2990

1040ST COLOUR PACKAGE

With SC 1224 Colour Monitor, Medium Res.
Built-in SF314 720K 3½" Disc Drive, Mouse, GEM and TOS
Plus BASIC Language
LOGO Langauge
1st Word Wordprocessor
Neochrome Graphics programme $3795
Special Introductory Offer — $3490

ATARI®

For further information and the name of your local ATARI ST SERIES dealer, fill out this coupon and send to:
Atari Customer Services Division
P.O. Box 78,118, Auckland.

Name:

Address:

Purpose of your enquiry:
☐ Personal
☐ Business

The case for wider access

By Roger Watson
Managing Director of BOS Software NZ Ltd.

Most companies using micro computers are now realising that access to the information on their computers is needed through more than one point. Two main reasons have caused this:

Firstly, as more applications are used, there is simply not enough time to get all the information processed using a single screen and printer. Good examples are organisations who 12-18 months ago purchased general accounting and job costing packages. Full use has not been made of job costing because there aren't enough "hours in the day."

The second reason people want more access to the computer is that of convenience. Managers today more than ever want fast, simple access to information such as general ledger and work-in-progress.

This requirement has meant that many single user software systems have been changed to allow them to run as multi-user by use of networks. However, it takes more than a network to make software multi-user. The application software and the operating system must be initially designed to be multi-user.

Features which need to be included are:

- Controlled close down at the end of processing since another user may still be working.
- Guarantees that a person taking a back-up has got exclusive access to the data.
- More screens inevitably cause a greater contention for use of printers. The software must give the options to print to a second printer or spool to disk. Poorly designed software may end up with both jobs printing to the same printer.

In spite of the addition of processors to the network, network systems slow dramatically after three or four terminals are added. This is because it is the access to the central base of information that hinders operation in almost all cases.

In other cases the processor with the main disk unit becomes a "file server" only and not available to be used as a data entry point.

A true multi-user operating system handles these disk contentions ("clogging up" of access to the disk) much more effectively than simple networking systems.

True multi-user software allows for the use of inexpensive, unintelligent terminals to be used. This means that a typical configuration of four screens will cost the user as little as $9,000 for hardware. Whereas a network solution will usually cost more than $17,000 for similar resources.

Operating systems

An operating system controls access to the computer by the user. In the case of a multi-user computer, it controls the access of perhaps 20 or more users, each with his own dumb screen and keyboard. For example, a typical micro-computer such as the IBM-AT running BOS software is capable of supporting eight or even sixteen such users.

The operating system also controls the storage of information in files on disk and diskette units and the backing-up of files onto removable disks or tapes for security purposes. It controls the printing of reports on one or more printers and, on a multi-user computer, can arrange for reports to be spooled to the disk and then directed in an orderly fashion to the printers.

When a user runs an application program, it is loaded into memory and executed under the control of the operating system which is held in another part of memory. The access of data on disk and its transfer to and from memory are also controlled by the operating system.

To assist the users of the computer the operating system will have a number of special purpose programs available called "utilities". A typical utility program copies files from one disk to another, or allows menus to be modified to reflect some new system function. An operating system typically has between 50-100 utility programs.

When a micro-computer has just a single user the operating system is obviously able to assume that all attempts to read or write data are being made by that user alone. However, when the computer has two or more users, the operating system cannot make this simplistic assumption and has to ensure that the various users do not interfere with each others' data in any way. This is done either by file locking or record locking.

File Locking

In an operating system using only file locking a complete file of information e.g. Debtors is "Locked". This means that an entire system can be inaccessible by other users.

Record Locking

In this method, only a small part of the file, called a record, is locked. In some software such as BOS, the record is not locked until actual updating occurs, thus effectively sharing records.

Sophisticated access methods are the key to reliable, high-speed performance in multi-user computers and those connected to a Local Area Network. BOS Software uses mostly record locking, with file locking in sensitive areas such as end of month or period end processing.

There are however, occasions when the addition of a second processor is desirable. For example if:

(a) More disk space is required on an existing computer for a new range of applications which will also require access to the original Data Base.

(b) Processing power becomes critical as it is needed to cope with the load of many terminals. BOS Software, realising these needs, have developed BOS/LAN and this is explained below. BOS/LAN allows several processors each with a number of terminals to be linked together thus allowing expensive peripherals to be shared.

BOS/LAN and other LAN operating systems must handle the transfer of information into and out of each computer connected to the Local Area Network. At the electronics level, there is the LAN controller board designed to interface a computer to a particular LAN. The operating system drives this controller board, coding and decoding the messages to be sent and received.

There are several widely available LAN hardware systems such as Omninet, Arcnet and Ethernet. Each has its own characteristics and is incompatible with all the others. You will be able to add new and different computers to it over a period of time, but to change the LAN itself would be costly.

LAN Operating System

You would need to change all the cabling around the building and all the network controller boards in your computers. It is obviously important therefore to choose the right LAN at the outset.

The trade-off in the choice of which LAN to use is broadly speaking, between speed and cost. Omninet is the cheapest in terms of cabling and inter-

The world's best promise of reliability, from the world's biggest seller of floppy disks.

By consistently producing better technology, Verbatim rapidly became the world's biggest seller of floppy disks.

These days, *new* Verbatim disks are *so* reliable we can confidently warrant they'll last a *lifetime* in constant use.

That's a *human* lifetime, not some vague disk lifetime expectation. Only the world's biggest seller of disks could offer such a warranty.

Exceptional accuracy and reliability is built into every single disk we sell, because every disk is individually tested.

Every Verbatim disk is, therefore, as perfect as a disk can be.

Which is probably why we sell more than any other disk maker in the world.

Verbatim.
Word perfect. For life.

VERBATIM NEW ZEALAND LIMITED, WELLINGTON — TEL. (04) 858-615 P.O. Box 11591 — AUCKLAND — TEL. (09) 543-587 P.O. BOX 74035.

face boards and, at 500,000 bits per second it is arguably the ideal medium-speed LAN available.

At the high-speed end of the market comes Ethernet at 10,000,000 bits per second. This also uses co-axial cable, with more complex controller boards to support the higher speed, and consequently is a higher-cost solution. All these LANs have controller boards available for a growing range of computers. The BOS/LAN Operating System supports all three LAN's on a wide range of computers.

Best of both

It is now possible to get the best of both worlds by using a BOS/LAN to connect a number of single-user and multi-user computers, each computer supporting its own dumb terminals and sharing expensive disk storage.

Let's look at a medium-sized production/selling company in New Zealand.

Three 10Mb XT computers and one 20Mb AT on a LAN provide 50Mbs of disk storage accessible as if it were one large disk (unless inhibited by password control). Each 10Mb computer has several inexpensive dumb terminals.

The users in the production department are running programs concerned with manufacture of products. They have one PC/XT with two terminals.

In the accounts department the programs used are debtors ledger, creditors ledger together with a general ledger program to produce management accounts. This requires three terminals on another PC/XT.

In the computer shared by the administration and customer service departments, the main functions are handled by databases. This is held on the local disk with call logs and other departmental reports being printed in their own office.

The sales department enters orders into an order processing system. They have three terminals.

BOS/LAN allows controlled private use of each computer in the LAN. Individual computers within the LAN can be disconnected without affecting the rest of the LAN and used for totally different jobs.

The multi-user system linked through a LAN gives flexibility and power which may otherwise cost much more on a single larger computer.

Various PCs

The other area of flexibility that opens up with BOS/LAN systems concerns the computers themselves.

When you come to add a new function to the company's computer systems you will have the option of doing so with a computer different from those already installed, thereby taking advantage of the latest technology. The new computer may not have been available when the others were acquired, or may have some special feature not needed elsewhere in the company. It might support graphics or colour.

Seagate distributor

Hard-drive manufacturer Seagate (US) is now officially supported here by Christchurch-based Solstat Industries. This means warranties can now be supported locally.

The range is from 10Mb half-height drives to 96 Mb full-heights.

Solstat claims more PC importers will choose the option of importing PCs without drives and installing the Seagates offered here under warranty.

Solstat is buying direct from Seagate's Singapore factory and claims pricing advantages over competitors.

THEOS: Modest in spite of its name

by Mark James

Most of the multi-user operating systems available for microcomputers have been brought down from larger machines, and adapted to run on micros. There is one, however, that was born and bred in the micro world. It made a name for itself in the late 1970s as OASIS; today it is known by the biblical-sounding name of THEOS.

If THEOS sounds more like an archbishop's accounting package than an operating system, you should know that it stands for The Operation System; and if even that sounds too pretentious to be taken seriously, it must be noted that THEOS is an established product, and not just some marketing manager's vulgar contribution to software taxonomy. In fact, THEOS – or at least OASIS – is older than MS-DOS.

When Z-80-based computers first became popular in the late 1970s, OASIS was a competitor to CP/M as an operating system for them. OASIS was by far the more complete product, supporting four different filing systems to CP/M's one. Phase One Systems, the developers of OASIS, actually came out with a multi-user version as early as 1980, several years ahead of anyone else.

For reasons lost in the mists of history, however, the smart money was on CP/M, and the first wave of microcomputer software (Wordstar, Visicalc) was written for CP/M, not OASIS. Later, when IBM and MS-DOS took the micro world by storm, OASIS was sort of lost in the shuffle. Phase One Systems fell on hard times, was reorganised as THEOS Software Inc., and changed the name of its operating system. Today, the growing need for multi-user micros has given THEOS a new lease on life.

THEOS differs, in two important ways, from any of the other operating systems that we have considered in this column: it does not pretend to be a portable invironment, and it includes no standard database.

Portability

Unlike BOS, PICK or AMPS, THEOS does not implement any form of virtual machine; that is, when you compile a program under THEOS, the result is a direct machine-code module that can be run on only one kind of processor. This is the approach taken by most single-user operating systems. With each of the other multi-user systems, programs are compiled into an abstract "pseudo-code" that is identical across different machine types.

The advantage of pseudo-code is that any program that runs on one computer will run identically (in theory, at least) on any other computer, regardless of hardware. On the other hand, the advantage of the THEOS approach is that the operating system does not have to handle the pseudo-code, and is therefore leaner. Also, directly-compiled machine code is generally faster in execution than interpreted or compiled pseudo-code.

Although THEOS programs are not 100% portable from one machine type to another, THEOS itself runs on two different machine types (the Z-80 version, called THEOS-8, and the Intel 8086/8088/80286 version, called THEOS-86), with a Motorola 68000-chip version said to be in testing stage. There are C and BASIC compilers available for each version; a program written in one of these languages should be able to be compiled under any version of THEOS, although a few modifications to the code might be necessary.

THEOS does not suffer, as PICK and UNIX do, from a plethora of different implementations or "flavours" of the operating system. There is only one THEOS, and therefore no compatibility problems.

Multi-User series: Part VI

Database

To the user, THEOS looks like a traditional minicomputer operating system, in that it is driven by a command language and operates upon a set of independent disk files. To someone used to MS-DOS or even DEC's VMS, THEOS commands will look reassuringly familiar: COPY, LIST, BACKUP, ERASE, SET. THEOS has no one standard filing system; it has four. They are stream sequential files, random access files, keyed random access files, and indexed sequential files. Each type of file has to be explicitly declared with a file size when created, although the system allows files to grow if their allocated size fills up.

THEOS thus differs from other multi-user systems such as PICK or AMPS, which have integrated databases, or BOS, whose various user tools build and maintain their own files. None of this is automatic in THEOS; the programmer is responsible for the maintenance of all database files.

THEOS itself also lacks such standard database utilities as a query language and a report writer facility. However, specialty THEOS software houses, such as Direction Computers in Auckland, produce their own versions of these features.

System performance can be improved through the use of a RAM disk. A RAM disk under THEOS is like a RAM disk under MS-DOS: The user (or system manager, in a multi-user environment) decides which files are likely to be most frequently used; these are then copied into a special part of memory made to look, to a program, like a disk. Access times to files on a RAM disk are only a small fraction of those to a real disk; however, the RAM disk must be copied back to the real disk periodically, in case a power failure should wipe out the contents of RAM.

A forthcoming version of THEOS, called THEOS-286V, is reported to use a virtual-memory approach to RAM caching. Instead of the system manager having to decide which files to put into RAM, the system itself will keep track of the most frequently-used files, and juggle the RAM cache area accordingly. This is more in line with the disk-caching principles of two of THEOS' competitors, AMPS and PICK.

Programming

THEOS supports two programming languages, C and BASIC. In addition, since the system is not worried about full portability, there are assembly-language programming tools for each of the versions of THEOS. There is also a batch command language called EXEC, through which a user is able to "can" groups of system commands. Neither C

nor BASIC comes with the system; they cost extra.

Unlike PICK BASIC, which is highly particularised to work with the PICK database, THEOS BASIC is fairly standard. There are some enhancements, such as nestable IF statements and file and record locking for multi-user security. The C compiler is claimed to be "100 percent standard Kernighan and Ritchie"; this does not mean much, however, as Kernighan and Ritchie are vague on many points, such as data types and I/O library calls.

Both languages produce re-entrant code – that is, more than one user can run from the same copy of a program. This is important in a multi-user system; much memory could be wasted, for example, if four people were editing, and there had to be four copies of the editor cluttering things up.

Each of the other multi-user systems that we have considered supports only one programming language, which is thoroughly integrated with the system as a whole. These languages tend to have fourth-generation features such as automatic data movement and input validation. Neither of the THEOS languages falls into this category. Like THEOS as a whole, its languages are no-frills affairs, simple and straightforward.

Friendly – not simple

In some areas, THEOS is not at all simple. It has, for example, elaborate facilities for inter-machine communications. Either ASCII or binary files may be sent between two THEOS computers using ordinary serial (RS-232) connections. The protocol used has been published, so that it is theoretically possible to share files with non-THEOS systems as well. There is also a TALK command, whereby two terminals can share a conversation, even if they are on different machines. The system even has an in-built terminal emulation feature, whereby the THEOS system can act as a terminal on another computer.

Another area of functionality unique to THEOS is the flexibility of its system messages. The system manager is able to modify them to suit an individual situation; they could all be translated into Spanish, if that were desired. Different users can be assigned to work with different sets of system messages.

THEOS on larger micros has a virtual device interface for graphics; however, this would be of only limited usefulness, since most multi-user systems are based on inexpensive screens which have no graphics capabilities.

Most of the more elaborate features of THEOS (and some of the mundane ones as well) require the user to learn some rather complex commands. Here, the system's documentation is tried and found somewhat wanting. If, for example, you wish to gain access to a "public" floppy disk drive, you have to issue the ATTACH command first; and if you don't happen to remember that fact, you will spend hours in the massive manuals trying to figure it out. Although all of the information seems to be there, the organisation of the manuals is rather fragmented. There is no index whatsoever, a major shortcoming for so hefty a set of volumes. There is, however, extensive on-line help available.

This is the main drawback of THEOS: It seems to have been built for someone who already knows how to use it. A THEOS system will have to be set up by a THEOS guru, and it is probably wise to have an experienced THEOS person on call if things go wrong. Apart from that, however, the system is simple and efficient. In spite of its overbearing name, THEOS does not try to be omnipotent. It is not as friendly or as functional as the likes of PICK or AMPS, but its overhead is low, and it is a tight, robust system. It packs some surprising features in its small size.

The Ins and Outs of ML

by Joe Colquitt

In/out operations in machine-code are fairly simple when the conventions have been learned.

Most of the time you'll be concerned only with saving and loading, although all other commands, such as those for sequential and relative files follow a similar pattern.

Routine 1 (relocatable)

```
C000 LDA#$36 ;
C002 STA$01  ;BASIC off
C004 LDA#$01 ;
C006 LDX#$08 ;device (1 for tape)
C008 LDY#$FF ;
C00A JSR$FFBA;set file/channel
C00D LDA#$04 ;file-name length
C00F LDX#$2C ;file-name address (lo)
C011 LDY#$C0 ;                  (hi)
C013 JSR$FFBD;set file-name
C016 LDA#$01 ;start address (lo)
C019 STA$FA  ;
C01A LDA#$08 ;start address (hi)
C01C STA$FB  ;
C01E LDX#$08 ;end address (lo+1)
C020 LDY#$21 ;            (hi)
C022 LDA#$FA ;location of start
C024 JSR$FFD8;save
C027 LDA#$37 ;BASIC on
C029 STA$01  ;
C02B RTS     ;
Data
C02C 54 45 53 54  'TEST
```

Routine 1 shows the procedure for a PRG save from machine-code.

PRG in this instance refers to a block of memory, such as a program, character set, bit-map screen etc.

This particular routine will save a program residing between $0801 to $23107, $0801 being the normal start address for saving a BASIC program.

The file 'TEST' can be loaded as a normal BASIC program. Your saves will almost certainly have different start/end addresses, so amend your routine appropriately.

For convenience, the instructions that turn BASIC off/on are included as at some time you may want to save from under the BASIC ROM.

The next four instructions open a channel to the device.

The LDY is used for secondary addresses, which are not required for PRG saves.

The next four instructions set the filename, necessary for disk but not always for tape.

The final bunch set the limits of memory to be sent as the save.

Specific

As usual, the higher level BASIC handles names and save size, but ML routines need to be specific.

As mentioned before, you may want to save an area under one or both of the ROMs.

In the case of the BASIC ROM this is fairly straight-forward. Routine 1 will seitch the BASIC ROM off and procede normally.

The file will then load under the BASIC ROM. Saving from under the KERNAL ROM is a problem, but solvable.

Because the sub-routines used for saving are in the KERNAL, switching it off will obviate any call to it, so you're snookered that way.

The solutions to the problem are 1) move the save sub-routines, 2) move the data to be saved and relocate it on loading.

The first is a bit of a struggle and not really worth the effort.

The second is much more attainable, and opens up a further possibility, ie changing the load address on the disk (sorry tape users).

Routine 2 (relocatable)

```
C000 SEI         ;
C001 LDA#$35     ;KERNAL off
C003 STA$01      ;
C005 LDA#$00     ;set zero-page registers
C007 STA$FA      ;to $E000 and $6000
C009 STA$FC      ;
C00B LDA#$E0     ;
C00D STA$FB      ;
C00F LDA#$60     ;
C011 STA$FD      ;
C013 LDY#$00     ;loop 8K
C015 LDA($FA),Y;
C017 STA($FC),Y;
C019 INY         ;
C01A BNE$C015    ;
C01C INC$FB      ;
C01E INC$FD      ;
C020 LDA$FB      ;
C022 BNE$C013    ;
C024 LDA#$37     ;
C026 STA$01      ;
C028 CLI         ;
C029 RTS         ;
```

Avoid lock-up

Routine 2 will transfer the whole 8K from $E000 to $FFFF to $6000.

The SEI is necessary to inhibit interrupt vectors from attempting to access the KERNAL, and causing a lock-up because it isn't there.

As I don't know which part of the 8K you may want to save, I'll have to leave it to you with regard to the start/end addresses. The main thing is to get the *data out from under the KERNAL*.

If a save from $E100 to $F215 was desired, you would need a loop for $E100 to $F200 and a minor loop for the part block left over **(Routine 3)**, then save from $6100 to $7215. When this is re-loaded, use the same loops as in the original transfer, but transpose the LDA($FA),Y and STA($FA),Y and the LDA$...,Y and STA$...,Y. This will relocate the loaded data to $⅜100-$F215.

The alternative, **Routine 4,** enables 1541 disk users to change the load addresses on the disk. I've tested it thoroughly to make sure it works.

Please, please type it in carefully.

If the wrong bytes are changes on the disk, you could lose data or cause the computer to make an incomplete load or lock-up because it can't find blocks.

Routine 3 (relocatable)

```
C000 SEI          ;block interrupts
C001 LDA#$35      ;KERNAL off
C003 STA$01       ;
C005 LDA#$00      ;set zero-page
C007 STA$FA       ;to $E100 and $6100
C009 STA$FC       ;
C00B LDA#$E1      ;
C00D STA$FB       ;
C00F LDA#$61      ;
C011 STA$FD       ;
C013 LDY#$00      ;major loop
C015 LDA($FA),Y;
C017 STA($FC),Y;
C019 INY          ;
C01A BNE$C010     ;
C01C INC$FB       ;
C01E INC$FD       ;
C020 LDA$FB       ;
C022 LDA$FB       ;
C024 CMP#$F3      ;stop at $F300
C026 BNE$C00E     ;
C028 LDY#$15      ;
C02A LDA$F200,Y;
C02C STA$7200,Y;
C02E DEY          ;
C02F BPL$C000     ;
C031 LDA#$37      ;restore KERNAL
C033 STA$01       ;
C035 CLI          ;and interrupts
C036 RTS          ;
```

Routine 4

```
5 INPUT"□PRG NAME";NA$:LE=LEN(NA$)
10 OPEN1,8,15
20 OPEN2,8,2,"#":T=18:S=1
30 PRINT#1,"U1 2 0";T;S
35 PRINT"□READING TRACK 18 SECTOR"S
40 B$="":FORI=1TO255
45 GET#2,A$:IFA$=""THENA$=CHR$(0)
50 B$=B$+A$:NEXT
60 PRINT"□SEARCHING TRACK 18 SECTOR"S
70 FORZ=1TO255-LE
80 IFMID$(B$,Z,LE)=NA$THEN100
90 NEXT:S=S+1:IFS<21THEN30
95 PRINT"■NOT HERE":END
100 T=ASC(MID$(B$,Z-2,1))
105 S=ASC(MID$(B$,Z-1,1))
110 PRINT"■FIRST BLOCK OF "NA$
115 PRINT"■TRACK"T"SECTOR"S
120 PRINT#1,"U1 2 0";T;S
130 C$="":FORI=1TO255:GET#2,A$
132 IFA$=""THENA$=CHR$(0)
135 C$=C$+A$:NEXT:L=ASC(MID$(C$,3,1))
160 H=ASC(MID$(C$,4,1))
170 PRINT"■START ADDRESS"L+H*256
175 IFC=1THENC=0:GOTO250
180 INPUT"■NEW ADDRESS (DEC)";NA
190 HI=INT(NA/256):LO=NA-HI*256
200 A$=CHR$(LO)+CHR$(HI)
210 PRINT#1,"U1 2 0";T;S
220 PRINT#1,"B-P 2 2"
230 PRINT#2,A$;
240 PRINT#1,"U2 2 0";T;S
245 C=1:GOTO115
250 CLOSE1:CLOSE2:END

READY.
```

This can be remedied if you have a sector editor, but that can be a dead end also.

To be sure your copy works, try it on a new disk with only a couple of PRG files on it.

Loading

Loading on the other hand is comparatively a piece of cake.

Routine 5 sets out the procedure. Again there are small differences for tape/disk.

Routine 5 (relocatable)

```
C000 LDA#$01    ;
C002 LDX#$08    ;disk
C004 LDY#$FF    ;
C006 JSR$FFBA   ;
C009 LDA#$04    ;file-name length
C00B LDX#$18    ;name address (lo)
C00D LDY#$C0    ;             (hi)
C00F JSR$FFBD   ;set name
C012 LDA#$00    ;0=load, 1=verify
C014 JSR$FFD5   ;load/verify
C017 RTS        ;
Data
C018 54 45 53 54 'TEST
or for tape

:C002 LDX#$01
:C009 LDA#$00   ;if no name
:C00B JSR$FFBD  ;
:C00E LDA#$00   ;
:C010 JSR$FFD5  ;
:C013 RTS       ;
```

The routine will perform normal (ie ,8) or relocated loads (,8,1).

That's about it for PRG loads.

In a further issue, I'll cover other file commands.

If you would like a copy of the public domain monitor 'Supermon' with instructions, send me a disk, or a tape with a sample save on it: Joe Colquitt, 6 Martin Ave, Mt Albert, Auckland.

THE TYPE UTILITY: a modular approach for simplicity

by Evan Lewis, Ph.D.

The task of writing a program in BASIC to display or print the contents of a sequential file would not seem too difficult. But if we wish to include a number of convenient optional features, the complexity rapidly increases. However, the technique of modular programming can be employed to keep things reasonably simple. A number of other programming techniques are explained in the description of the program. Although the program was written for a Commodore 64 many of the ideas can be applied to other machines and the program itself would require changes to only a few modules.

The purpose of the program is to get characters from a sequential data file held on disk or tape and display them on the screen or print them on a printer. The characters can be displayed as uninterrupted lines of text or alternatively each character can be displayed with its corresponding ASCII code and a character counter alongside it.

The latter mode allows the detection of special non-printable character codes, which would otherwise be invisible. This is particularly useful for examining files produced by one commercial package so that they can be read by another program package.

When modular programming is used, the central component of the system is the "mainline program" which calls the various subprograms in the appropriate order. Each of the subprograms may make use of other subprograms at the next level down and so on. The maximum depth of nested subprogram calls is 24 on the Commodore 64.

Overview

The mainline program (lines 100-155) provides a summary of overview of the design of the program. First, the subprogram to set up the title screen and its colours is executed and then the requirements subprogram is called to ask seven questions concerning the various options available.

Then there is a choice between two output files which can be opened by appropriate subroutines – the printer and the screen – and the decision is made according to the output option stored in out$. Similarly, either a disk or tape file is opened as an input file according to *inp$*.

PART 1 in a series for the C64 by Dr Lewis of Waikato University.

Instructions regarding the use of functions keys are displayed by a subprogram at line 490 and finally the central subprogram to get and display characters from the input file is executed. In this case the last subroutine in the mainline is the heart of the program. Clearly it must involve a loop which repeatedly gets one character from the input file, i.e. disk or tape (line 545) and prints it out to the output file (line 605). The intervening lines allow interactive control of the program through use of the function keys. We will come back to these statements later. The GOTO statement at line 620 completes the loop to get the next character. Two different methods are used to control the various input and output options. In order to set up the initial conditions a question and answer approach was chosen.

A special subprogram is used instead of an INPUT statement to allow the user to enter a single character in response to each question. Default answers are provided so that the user can press RETURN after every question resulting in the most commonly used mode of operation for the program.

Export/Import

When the program is running and displaying characters, the user can dynamically change the output mode and output device by pressing the function keys. Use of these keys is explained by instructions which are displayed before output begins. This is the second method of mode control.

The TYPE utility was originally written to help solve the problem of "exporting" files from one package and "importing" them to another package, e.g. the transfer of information between various databases and word processors.

TYPE was used in the transfer of data from "The Manager" to "Superbase 64". The report generator of "The Manager" was used to produce a sequential file containing information from each record and including the record number which is stored as register number 101. The resulting file was manipulated to match the "Superbase" screen format specifications by using "Easyscript".

*Running TYPE on the resulting file re*vealed that it started with the special character CHR$(128) which must be removed before it can be read by "Superbase". It was also found by exporting a test file from "Superbase" and examining it with TYPE that each field except the last one in the file in terminated by a blank and than a carriage return character (i.e. CHR$(32) + CHR$(13)). The TYPE program was then modified to detect CHR$(128) and remove it and to add the above terminator to each field. Even then success is not always guaranteed!

The use of multiple input/output files, function keys, the STATUS register, disk error traps and the single character input subprogram will be described in subsequent articles.

TYPE Utility

```
100 rem xxxUtility to 'type' out files xxx
105 rem xxxxxxx Mainline Program xxxxxxx
110 gosub 160 set up screen & header
115 gosub 245 enter requirements
120 if out$="s" then gosub 330 open scre
125 if out$="p" then gosub 345 open print
130 if inp$="t" then gosub 370 open tape
135 if inp$="d" then gosub 390 open disk
140 if el goto 110  error.Start again.
145 gosub 420 display instructions
150 gosub 490 loop to get chars & output
155 goto  110 start again
160 rem xxxxx Set up the screen xxxxxxxx
165 poke 53281,0:rem Set Screen
170 poke 53280,0:rem Set Boarder
175 print CLR :  :rem Clear screen
180 print WHT :  :rem White letters
185 print LOW :  :rem Lower case
190 print
195 print              TYPE UTILITY
200 print             rTYPE UTILITY
205 print
210 print             Written by:
215 print             Dr Evan Lewis,
220 print             Dept Computer Science
225 print             University of Waikato
230 print             Hamilton,New Zealand
235 print
240 return
245 rem xxxxx Enter requirements xxxxxxx
250 fi$=""
255 input qFile name.( <=16 chars) :fi$
260 if len(fi$)>16 then
    print rq Filename too long :goto 250
265 if fi$="" then end
270 print qseq prg or usr  (s/p/u) :
275 qv$="spu" :gosub 820:ty$=qq$
280 print qdisk or tape      (d/t) :
285 qv$="dt" :gosub 820:inp$=qq$
290 print qtext or ascii     (t/a) :
295 qv$="ta" :gosub 820 get character
300 as%=(qq$="a")
305 print qscreen or printer (s/p) :
310 qv$="sp" :gosub 820:out$=qq$
315 print qprinter device No.(6/4) :
320 qv$="64" :gosub 820:pdev=qq
325 return
330 rem xxxx Open screen for output xxxx
335 close 4:open4,3: return
340 return
345 rem xxx Open printer for output xxxx
350 close 4:open 4,pdev
355 if st then gosub 690 status check
360 if qq=6 then
      open 6,6,6:print#6,1:close 6
365 return
370 rem xxx Open Tape file for input xxx
375 el=0 :rem No errors
380 close 8:open 8,1,0,fi$
385 return
390 rem xxx Open disk file for input xxx
395 close 8:close 15
400 open 15,8,15
405 open 8,8,8,fi$+ ,"+ty$+ ,read"
410 gosub 655 error trap
415 return
420 rem xxxxxxxx Instructions  xxxxxxxxx
425 print S       Instructions"
430 print"        rInstructions"
435 print qrf1R pause (stop output)
440 print qrf1R output one character at
    a time
445 print qrf2R or any key for continuou
    s output"
450 print qrf3R output to screen
455 print qrf4R output to printer
460 print qrf5R switch between text/ASCI
    I
465 print qrf7R end output & close files
470 print qrf8R stop without closing fil
    es (CONTinue)
475 print qqq Press any key now to conti
    nue "
480 wait 198,1:get qq$
485 return
490 rem"xxx get characters from file xxx
495 print#4,"qFile '"fi$"' contains:"
500 print#4
505 if as% then
      print#4,"rchr","asc","No."
510 white$=chr$(5)
515 null$ =chr$(0)
520 quote$=chr$(34)
525 del$  =chr$(20)
530 poke 650,128:rem Repeat all keys
535 n=0
540 n=n+1
545 : get#8,a$
550 : if st then goto 690 status check
555 : get qq$
560 : if qq$=""goto 605
565 : fx=(asc(qq$)-132)x2-1
570 : if fx>7 then fx=fx-7
575 : if fx<1 goto 605
580 : rem"             1    2    3    4
585 : rem"          wait,cont,scre,print
590 : on fx gosub 630, 625, 330, 345,
                  635, 635, 640, 645
595 : rem"          asc, asc, end, stop
600 : rem"           5    6    7    8
605 : print#4,a$;white$;
610 : if a$=quote$ then
        print#4,a$;del$;
615 : if as% then
        print#4,"x",asc(a$+null$),"#"n
620 : goto 540
625 return:rem"Key not programmed
630 wait 198,1 :return
635 as%=not(as%):return :rem"Toggle asc
640 close4:close 8:close 15:end:return
645 print:print"qEnter CONT to continue"
650 end:return
655 rem xxxx Error handling routine xxxx
660 input#15,el,e2$,e3$,e4$
665 if el=0 then return
670 print
675 print"r"str$(el)"  e2$;"  "e3$;"  "e4$
680 close 4:close 8:close 15
685 end
690 rem xxxxxx Test status word st xxxxx
695 if st=0 then return
700 print"rqp85 Prog.Reference Guide"
705 print"rqI/O status = "str$(st)"  "
710 if st and -128 then
    print"rqDevice not present":goto 755
715 if st and 1  then
      print"rqTime out for write"
720 if st and 2  then
      print"rqTime out for read "
725 if st and 4 then
      print"rqShort block on tape"
730 if st and 8 then
      print"rqLong  block on tape"
735 if st and 16 then
      print"rqRead error / Mismatch"
740 if st and 32 then
      print"rqChecksum error
745 if st and 64 then
      print"rqEnd of file
750 print"rqAll files closed
755 close 4:close 8:close 15:end
760 return
765 rem"xxxxx Get single key entry xxxxx
770 rem"Valid keys are provided in qv$
775 rem"The first valid key is default
780 rem"which is stored as qd$.
785 rem"The key pressed is in qq$
790 rem"and its numeric value is qq.
795 rem"qq% is a flag for 'yes' entry
800 rem"and is 0 for any other  entry  .
805 qv$="yn":rem"Default is yes/no input
810 rem"Entry at this point allows use
815 rem"of characters other than y & n
820 rem"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
825 back$=chr$(157)
830 cr$=chr$(13)
835 qd$=left$(qv$,1)
840 print"? "qd$;back$;
845 qq$="":q1%=5:q%=0
850 get qq$:q1%=q1%+1
855 if q1%=3 then
      print qd$; back$;
860 if q1%=6 then
      print"r"qd$"R" back$;:q1%=0
865 if qq$="" goto 850
870 if qq$=cr$ then
      qq$=qd$ :goto 900
875 i=1
880 :  if qq$=mid$(qv$,i,1)goto 900
885 :  i=i+1
890 :  if i>len(qv$)goto 850
895 :  goto 880
900 print qq$
905 if qq$="y" then q%=-1
910 qq=val(qq$)
915 return
920 rem"xxxxx Create a Test file xxxxxx
925 dv=8 :rem disk
930 close9:open9,dv,9, "type test,s,w
935 for i=1 to 10
940 print#9, "type test. Line No.="i
945 next
950 print#9
955 close 9
960 end
```

Tips for speeding up Basic

by Geoff McCaughan

Subject: Analogue Ports
System: Commodore 64

[Q] I note that the C64 has A/D comverters POT-X and POT-Y. Is there some reason why these ports are not used more often? I have seen some magazines deny that they are even there. I would like to use them to read temperature from analogue sensors.

[A] The analogue ports certainly exist.

Externally POT-X can be accessed between pins 9 and 7 of either control port, and POT-Y can be accessed between pins 5 and 7.

Internally POT-X can be read at 54297 ($D419) and POT-Y at 54298 ($D41A). Before reading these locations, one must select a control port, this is done by writing 01 to bits 6 and 7 (i.e. 01XXXXXX) of CIA#1 data port A at 56320 ($DC00) to select control port 1, writing 10 to the same location selects control port 2.

As these locations are also written to by the keyboard scan, the keyboard IRQ must be switched off first, so from Basic, one must go through these steps to read POT-X and POT-Y in control port 1:

```
100 POKE 56333, 127: REM turn of keyscan IRQ, for ML use SEI
110 POKE 56320, PEEK (56320) AND 63 OR 64: REM select control port 1
120 REM use POKE 56320, PEEK (56320) AND 63 or 128 for control port 2
130 X = PEEK (54297): REM read X value
140 Y = PEEK (54298): REM read Y value
150 POKE 56333, 129: REM restore keyscan IRQ, for ML use CLI
```

The value returned will be in the range 0-255. Remember to re-enable the keyscan IRQ within your program, as it is rather difficult to do it afterwards if your keyboard is dead! Note that four analogue values can be read, although there are only two registers, thus only two can be read at any one time.

Obviously this is a bit of a rigmarole to go through from Basic, especially if you are looking for a fast response, but there is a clear advantage when doing such operations as moving sprites.

To move a sprite with the joystick, one must read the joystick port, and then increment or decrement the appropriate sprite register (s) according to the direction indicated, thus we must loop through this operation 100 times to move the sprite from location 1 to 100.

Using a paddle controller allows the program to read position information instead of direction, so the program can read the port and place the resulting value directly into the appropriate sprite register, allowing much faster movement, especially from Basic.

There are a number of programs around now that utilise a WIMP (Window-Icon-Mouse Programming) environment driven with a joystick.

Positioning the pointer in such a system can be tricky if it moves too fast, and tedious if it moves too slow.

Some programs get around this problem by moving the pointer slowly at first and then speeding up, which can take some getting used to.

A better solution might be to dispense with the digital joystick and use an analogue one read through the POT-X and POT-Y ports, but there are two problems to overcome here.

Firstly, nobody uses analogue joysticks on 64s, so they would have to be sold with the program (a new form of copy protection perhaps?), whereas virtually everyone has a digital joystick.

Secondly, the sprites move within a 320 by 200 area, and as the POT registers have a maximum value of 255 some juggling will have to be done in software if the whole screen is to be covered.

It is difficult to say why these ports are not used more, certainly I have seen very few programs that use game paddles, which I understand to be the original intention.

Digitising devices such as the Koala pad are perhaps the most prominent applications.

Using the analogue ports to interface to external sensors is not as simple as applying a voltage and reading a value. The A/D converters (which are incidentally in the SID chip) expect a 0-500k resistance between the input and +5 volts. The sampling time is .5 milliseconds. Perhaps the simplest and safest way to interface to this would be to use an opto-coupler.

Subject: Speeding up Basic
System: Not specified.

[Q] Do you have any tips for making Basic programs run faster?

[A] It is difficult to give specific examples when I don't know what sort of computer you have, but I can give some general rules, which in some cases apply to other languages as well as Basic.

1) The Golden Rule of program optimisation is to Experiment.

If your computer has a built-in clock or timer, use that to test routines, otherwise use a stopwatch.

Frequently there is more than one way of doing a particular operation; Basic may even allow a variation in syntax which operates at a different speed. Try this:

```
100 TM$ = "000000": REM zero timer, modify to suit your computer to use a stopwatch
110 FOR X = 1 to 1000
120 Y = X*X: REM operation to test
130 NEXTX
140 PRINT TM$
```

Try these modifications and note the time differences:

Change line 120 to: 120 Y = X ↑ 2
Change NEXTX to NEXT
Use integer variables for X and Y
Put the whole program on one line
Replace the FOR NEXT loop with a GOTO loop that runs 1000 times.

Some of these modifications will not work with your version of Basic: which ones? That can be a useful thing to know too.

If you get very small values when you print the time, try looping 5000 or 10000 times.

Any time you think of a different way of doing something, time it in a loop like this. You can put a single statement, or a whole subroutine in the loop. This technique is known as benchmarking. You may find it helpful to make a note of the best way to do things for future reference.

2) Use variables instead of numbers if yu use a value more than once in a program. Every time the Basic Interpreter comes across a number such as 100 in a program it must convert the ASCII characters 1, 0, 0 to an appropriate binary format which can be used in a calculation.

This conversion is done when assigning a variable (i.e. X = 100), thus it is much quicker to assign a variable once at the start of a program and use the variable from then on.

Strictly speaking this is a constant, but as Basic has no specific constant data type we must use a variable instead.

3) Watch your loops. Because loops are executed many times in a program, it pays to keep a close eye on what goes on inside them.

A common mistake is to place a calculation or other operation inside a loop when it could just as well be placed outside.

Significant time savings can result from careful rearrangement of loops. A simple example:

```
100 FOR X = 1 TO 10
110 Y = SIN (X)+(X*5)
120 NEXT
```

This is acceptable if the loop is only executed once, but if it is used more often, this may well be quicker:

```
10 DIM Z(10)
20 FOR X = 1 to 10
30 READ Y : Z(X) = Y
40 NEXT
50 DATA 5,10,15,20,25,30,
35,40,45,50

100 FOR X = 1 TO 10
110 Y = SIN(X)+Z(X)
120 NEXT
```

Is it faster? Benchmark it and see, put lines 100-120 inside the loop. Lines 10-50 must be included in the timing, but should be outside the loop.

Can you use integer variables for X and Z?

4) If the language you are using allows true integer operation (some Basics don't) use integer wherever possible. Floating point math is a huge killer of computer time.

5) Structure. Unfortunately, carefully laid-out, well commented Basic programs can often be speeded up by squeezing as much as possible on to a line, leaving out spaces and REMarks and throwing structure out the window. This of course makes the program difficult, if not impossible to follow (and debug).

One idea is to write well commented spaced-out programs which once debugged can be run through another program which does the squeezing to get the final ounce of performance. It is a good idea to keep the original if you do this, as it will be easier to modify.

6) Don't GOTO. GOTO not only makes a diabolical mess of a program, it is also rather slow, as the interpreter has to search through the program for the line requested.

Novice programmers often say to me that use of GOTO is unavoidable – Rubbish I say! GOTO is just a "cheats way" out of a tight spot. It is very rare for a GOTO to be unavoidable in a well written program.

One test is to see if it is conditional (i.e. IF <something or other> GOTO). Conditional GOTOs are sometimes acceptable and occasionally necessary. UnconditionalGOTOs are a no-no and a sign of a badly thought-out algorithm.

7) Be careful with subroutines. Frequently called subroutines should be put at the start of your program; infrequently called ones should go at the end.

8) Compile. The best way to speed up an interpreted language is to compile it. This can introduce a whole new set of optimising requirements. For instance using numbers instead of constants will usually make no difference to a compiled program, but a compiler may allow true integer operation where the source language does not.

The solution, as before, is to benchmark anything you are uncertain of.

Another advantage of compilers is that they allow your source program to be as laid-out and well commented as you like without sacrificing performance (although there may be a penalty in compile time).

When optimising programs one often has to make a choice between memory and speed, all too frequently an improvement in one is at the expense of the other.

For instance, it is significantly quicker to sort strings of equal length, but padding the strings to the same length will use up more memory than if you left them alone.

The solution will depend on the particular situation; often, but by no means always, speed is more important than memory.

Within one program different optimising strategies may apply from place to place. One might program for speed inside loops, and for memory elsewhere.

Big grant

Sperry Information Systems Group has chipped in *$240,000* as a grant to fund an expert systems laboratory at Victoria University. The lab is for technology transfer work and development work.

The project will use Sperry's Explorer workstations, made by Texas Instruments, and will run on Lisp, a language for artificial intelligence work.

Software list

A directory of educational software for the Apple II has been published for CED Distributors.

As well, memory and disc drive upgrades for the Mac 512k pc have become available here.

Mainly on the Master

by Pip Forer

The Master has been available now for long enough, and has interested enough purchasers, to encourage more comments on its performance and the programming tricks it offers through its extended operating system calls.

Also in this column comes some comments on compatibility of software and how you may need to adjust your model B programs to run on the Master, plus a mention of two co-processors.

Memory management

When I was an Apple II user one of my favourite pass-times was flipping screens about. A POKE to the right place could flip you from a text page to one of two graphics pages. You could even have a third graphics page stored somewhere as a 'virtual screen', and bring it in for display.

On the model B too, it was possible to maintain more than one screen using some slightly more complex programming, first revealed by Jeremy Rushton.

However, memory restrictions meant you were usually stuck with two or more screens in the less desirable modes. For instance you could not manage two mode 1 screens.

The Master however, allows more sophisticated screen and display use using a set of *FX calls, especially between numbers 108 and 113, and some OS calls, especially *SRDATA, *SRREAD, *SRROM and *SRWRITE. The last four help bring the sideways RAM space in on the act and will be dealt with in a later article. For now we will look at the *FX calls.

I first came across these because of the problem I mentioned above: a program tried to load a mode 1 map into mode 129 (shadow mode 1) using * LOAD 3000 and got a blank screen. The reason clearly was that my *LOAD 3000 for the screen image was placing the image in a different part of memory from that being displayed.

How then did you load a screen image in to shadow modes?

The simple answer lies in *FX 108.

Used with the parameter 0 (ie. *FX 108,0) the memory addressing in the BBC behaves just like a model B. The area of memory from &3000 to &8000 is just the area of main memory directly above &2FFF.

But with *FX108,1 these addresses apply to shadow memory. This means that ? or ! commands above 3000 will apply to the shadow area.

More importantly here, *LOAD 3000 brings the image straight into shadow area.

One useful application of this is that you can have one image on the screen using one area of memory and load another into the other area of memory for display.

Second image

Then the question becomes, how do I get to see the second image?

Using a new mode statement will clear the contents, and cannot be used within a procedure. The answer is *FX113, which can be used to flip between the main memory 'screen' and the shadow memory 'screen' (using parameters 1 and 2, i.e. *FX113,1).

Programmers have always liked this as a means to achieve two goals. One is that you can have two cimplementary screens (say a graphic and some text about the graphic) which the user can instantaneously switch between.

The other is that you can get much better animation by separating your drawing of the next frame from what the viewer sees (you try to always draw on the nidden page).

For this last trick you need only one extra instruction: you need to be able to control what page you are 'writing' to at any time. *FX 112 does this for you.

You can use *FX113 to display one page for you while *FX112 lets you determine whether new text or graphics goes to the displayed or hidden screen.

Diverse possibilities

Being a * command call these options are available from any language and open the way to some interesting programming possibilities.

When combined with the available memory in sideways RAM even more is possible in terms of handling and managing multiple screens. More on that when we look at the commands useful for this: *SRDATA, *SRREAD, *SRROM and *SRWRITE. They also have some exciting additional uses.

Software compatibility

As I stated in my review of the Master, anything written strictly legally on the model B (always using VDU codes and OS calls to achieve the desired end) should function perfectly first time on the Master.

However, software incompatibility can still occur from several different sources if an author departs from these practices.

The major sources are: hardware access, screen memory access and use of the operating system's reserved memory.

The major problem with hardware access is with protected disk-based (and some ROM) software that makes direct calls to hardware associated with the disk chips.

The change to the 8771 DFS caused some problems here, even before the Master arrived. It is generally not a problem with unprotected software or with software which runs on the network.

The problem of how software accesses the screen is a more common one.

In general, as long as it is not using shadow memory for the screen, or running on a co-processor, the Master will behave like a Beeb.

Some problems can occur however because of the configuration settings held in battery-backed RAM or set by running preceding programs. Ideas on some of these can be gleaned from the comments below on memory management.

Problems

If your software doesn't do things like check how the memory map is configured, especially where the screen RAM is (which may have nothing to do with the mode the program is in), you can get problems.

I had a mapping program that caused

problems when i loaded a screen image of a map: nothing appeared.

The answer was simple but, like many possibilities, hard for an author to anticipate until the Master had arrived.

A few games seem to strike similar problems.

The worst casualty is Elite, which amongst other things presumably has problems because it does such intricate things with the screen.

Has anyone discovered a patch to make it run on a Master yet?

Changed memory

The third problem is that of the reorganisation of memory use on the Master. Even using the disk system, PAGE is always at &E00, and the areas below that which have been used so creatively for extending available memory are now used in various different ways.

The most immediately noticeable is the relocation of the additional character set from &C00 to &&8900 in the shadow RAM workspace.

If you *LOAD symbols into &C00 now, you fail to get the desired affect. Software which makes assumptions about what it will find down here, and in the normal area set aside for disks, may run into difficulties. Problems of class 2 and 3 are as much a problem on the network as anywhere else.

For hackers and their own programs any problems tend to be very easily sorted out. The real challenge is with commercial material.

Even here problems are not too common, although the difficulties can also apply to ROMs.

I am told that ToolKit will not function, and certainly some existing ROMs behave differently on the Master, as shown by View 2.1's problems with shadow modes.

I suppose the good news in those cases is that aspects of ToolKit are covered by built in Master functions (LIST IF in BASIC and the editor do quite a lot between them) and the Master has its customised View.

Second processor

A Master 512 upgrade resides in my machine at present, but arrived just too late to evaluate this month.

I know that bundled with a variety of applications software working under the GEM environment (plus a mouse to drive it) the end retail cost is likely to be about $1500. I hope to give a full review next month.

Meanwhile a report has been sighted benchmarking an Acorn Work Station (effectively a BBC with a 32016 second processor, 4 Megabyte of RAM and a 20 Megabyte disk) as somewhat ahead of a multi-user university mainframe for a series of major, computationally demanding tasks.

The interesting part is that it seems that even with non-commercial charging rates for mainframe use the AWS pays for itself within two months!

For some applications the AWS is not without flaws, but the evidence is that it delivers you a lot of bang for your bucks.

How to protect your programme

by Don Stanley

Firstly, I do not believe it is possible to completely protect a program. If someone wants to break a protection scheme they will probably do it eventually.

So the idea instead is to make it as hard as possible – cut out 95% of the users and leave the other 5% to sweat over it.

Most users don't want to break into software anyway, and I believe a large number of those who do, do so only to see how a program accomplished some task.

So how do we attempt to protect software on the SV and MSX.

Protection requires some work on our part. There can be no simple five minute dream-up method – the method I will show took a week to plan and implement.

One method is to insert delete characters in every line of code of your BASIC programs. This requires one delete per character of code, so doubles your program size.

It is difficult to get the deletes in there, and makes no attempt to prevent programs being copies. They cannot, however, be listed easily.

The method I am going to suggest requires some knowledge of the roms and BASIC's work area. I will discuss what we need to know first.

Variable table

When your program is typed in, BASIC keeps track of where it finishes by using a series of 'pointers'. The start of each line of code is the address in memory where the next line starts. You can look at these by PEEKing the first few bytes.

For instance, suppose you had entered the program in figure 1. In direct mode type

FOR I=&H8001 TO &h800F:?HEX$(I)" "HEX$(PEEK(I)):NEXT

which will list the first 15 bytes of the program in hex.

Note the 8001. This is where all BASIC programs start unless you change an address in high memory (BASIC's work area). We won't get into that here, because it causes all sorts of problems unless you really know the system well.

The bytes you listed should look like this

8001 09
8002 80
8003 0A
8004 ..

The first 2 of these, 09 and 80, are a pointer to the next line of code. In other words, the next line of code starts at 8009 (hex). Look down your list to 8009 and 800A. The bytes here will point to the next line and so on.

Eventually BASIC will find a pointer which points to an address of 0000. This is the signal that the program is finished. For the example program, this should occur at 8157.

Now type in ? HEX$(256*PEEK (&HF7EF) + PEEK(&Hf7EE)).

This should return the value 8159. (If not you probably missed a space or something in the listing).

This value is the start of the variable table, which as I explained previously is where all the variables in the program get their values stored. It always occurs 2 bytes after the end of the program.

Keyboard buffer

When you type a character on the keyboard it goes into the keyboard buffer. This is an area of memory 40 bytes long which the system constantly scans for a RETURN (ENTER). When a return is found, all the characters up to the return are interpreted by BASIC, and the command is carried out.

This buffer is pointed to by addresses FA1A/1B and FA1C/1D.

The first two are the position in the buffer where the next character you type in will go, the last two are the position where BASIC last read an instruction from.

For instance, typing in RUN will leave the last address unchanged, but the first address will advance by 3. Then pressing ENTER tells BASIC to read all those characters (RUN) and work out what to do with them. At this stage the two addresses will be the same.

The buffer can be used by machine code programmers to access ROM routines. Your machine code program simply inserts the instruction into the buffer, ensuring that the address pointed to by FA1C/1D does not get updated.

Insert a return in as well, don't forget to update the first address, and then return control to BASIC. The command is carried out. We will look at an example shortly.

Hooks

Throughout the roms there are hooks. These are calls from the roms into an area of memory known as the hook table.

These calls generally do nothing except go straight back to where the hook was called from. (The disk system is accessed by hooks).

We can utilise the hooks (provided we know where in rom they occur) by altering the area in ram that they call. Usually this consists of a JUMP to somewhere else in ram where a machine code program is run.

Knowing where the hooks occur requires a disassembled listing of the machine code roms. All calls which call a location between FE79 and FFB1 are calls to hooks.

The technical documentation which tells where rom routines are is available from the Wellington Users Group, although it is by no means complete.

Program protection

This protection scheme requires

(1) to know where the variable table occurs

(2) to know the contents of locations *8001 and 8002*

(3) to have the machine code program in figure 2

(4) a little patience.

Figure 1.

```
10 CLS:PRINT
20 PRINT:PRINT "  FILES ON THIS DISKETTE:":PRINT
30 FILES
40 PRINT:PRINT:PRINT
50 PRINT" THIS PROGRAM WILL INITIALIZE THE"
60 PRINT" DISK TO AUTOMATICALLY EXECUTE THE"
65 PRINT" THE PROGRAM OF YOUR CHOICE WHEN "
70 PRINT" YOUR SYSTEM IS FIRST TURNED ON."
80 PRINT:PRINT:PRINT" Enter the name of your program,"
90 INPUT " then press (ENTER).";N$
100 PRINT "finished"
110 REM ----------------
```

We will use the program in **figure 1** to demonstrate the scheme.

If you have a copy of ZEN, load this and type in the program in figure 2. The

first three addresses in here, called BUFFST, BUFFED and VARTAB have already been discussed.

The next one, MAINHK, is the address of a hook which occurs at BASIC's MAIN routine, which is where BASIC always starts its interpreting from.

Then MAINPT is the actual address of this main routine (09C4 in rom).

If you do not have a copy of ZEN, type in the program in **figure 2.**

Figure 2.

```
1                               ORG  8170H
2                               LOAD 8170H
3                       ;
4                       ; EXTERNAL CONSTANTS DEFNS
5                       ;
6                               BUF  FST: EQU 0FA1AH
7                               BUF  FED: EQU 0FA1CH
8                               VAR  TAB: EQU 0F7ECH
9                               BUF  FER: EQU 0FD8BH
10                              MAI  NHK: EQU 0FE94H
11                              MAI  NPT: EQU 09C4H
12                      ;
13                      ; END OF EXTERNAL CONSTANTS
14                      ; NOW PUT BACK PGM POINTERS
15                      ;
16 8170 210980                  LD   HL,8009H
17 8173 220180                  LD   (8001H),HL
18 8176 215981                  LD   HL,8159H
19 8179 22EEF7                  LD   (VARTAB),HL
20                      ;
21                      ; CLEAR KEYBOARD BUFFER
22                      ;
23 817C 218BFD                  LD   HL,BUFFER
24 817F 221AFA                  LD   (BUFFST),HL
25 8182 221CFA                  LD   (BUFFED),HL
26                      ;
27                      ; LOAD "RUN",ENTER IN BUFFER
28                      ;
29 8185 21A381                  LD   HL,RUN
30 8188 ED51AFA                 LD   DE,(BUFFST)
31 818C 010400                  LD   BC,4
32 818F EDB0                    LDIR
33 8191 ED531AFA                LD   (BUFFST),DE
34                      ;
35                      ; ALTER HOOK AT FE94
36                      ;
37 8195 3EC3                    LD   A,0C3H
38 8197 3294FE                  LD   (MAINHK),A
39 819A 21A781                  LD   HL,HOOK
40 819D 2295FE                  LD   (MAINHK+1),HL
41                      ;
42                      ; FIRE UP PROGRAM, STARTING
43                      ; PAST USUAL POINT TO AVOID
44                      ; THE RESET HOOK
45                      ;
46 81A0 C3C709                  JP   MAINPT+3
47                      ;
48                      ; INTERNAL CONSTANTS
49                      ;
50 81A3 52554E    RUN:         DB   'RUN'
51 81A6 0D                      DB   0DH
52 81A7 AF         HOOK:        DB   0AFH
53 81A8 110180                  DB   11H,01,80H
54 81AB 12                      DB   12H
55 81AC 13                      DB   13H
56 81AD 12                      DB   12H
57 81AE 3EC9                    DB   3EH,0C9H
58 81B0 3294FE                  DB   32H,94H,0FEH
59 81B3 C9                      DB   0C9H
60                              END
```

Remove all but the DATA statements and lines 110,120. Change line 110 to I=&H8170 and run this program.

This will load the machine code program in figure 2 into memory. Save it by typing BSAVE "MCODE", &H8170, &H81BF.

This program is set up to protect the program in figure 1. Thus the 8009 on line 16 is the number to be inserted into locations 8001 and 8002 (90 into 8001 and 80 into 8002). This value on line 16 and that on line 18 (the variable table) will need to be altered for each program to be protected.

Lines 23 to 25 reset the keyboard buffer so the addresses discussed above are the same and also so that the buffer starts at its first address (it is possible for the buffer to start anywhere in the 40 bytes allocated for it). The buffer actually begins at FD8B and finishes at FDBA.

We want it set to the start so that we don't need to worry about running out of buffer space.

Lines 30 to 36 insert the word RUN and a carriage return into the buffer. Then the BUFFST is incremented by 4 (the number of bytes inserted). When the system next looks at the buffer, the two buffer pointers will not be the same, and the instruction in the buffer will be interpreted.

Note that since this program will not always run at 8170 as this example does, the 2nd and 3rd bytes of line 30 are also going to need altering. This will be discussed shortly.

From line 37 to 40 the hook which starts the MAIN routine in rom is altered to point elsewhere at a program in ram. Bytes 2 and 3 in line 39 will also need altering.

The program which the hook at MAIN will send BASIC to is going to be loaded at the location HOOK in figure 2. In fact this program is included in the figure 2 machine code program as a series of data statements. In figure 2 this little program is the series of bytes starting at HOOK (line 52). The program actually reads

```
XOR  A
LD   DE,8001
LD   (DE),A
INC  DE
LD   (DE),A
LD   A,C9
LD   (FE94),A
RET
```

Its role in the protection scheme is going to be to fool BASIC into thinking that the program does not exist in memory. To do this it loads two zeroes at 8001 and 8002. Since this is a line pointer and is zero, BASIC considers it as the end of the program!

Also, this little program alters the hook at MAIN back to what it should be, otherwise every program we load from then on would be non-existent to BASIC.

The final stage of the figure 2 machine code program is to jump to the MAIN routine. It has reset the pointers to set the program up, altered the hook, and put a RUN in the buffer. Jumping to MAIN starts the instruction, and thus the program, going. Actually we jump to MAIN + 3, this bypassing the hook at MAIN.

If we jumped straight to the hook, the program would not run as the code at the hook would fill 8001/02 with zeroes immediately, rather than when the program finishes.

Demonstration

To use all this for our demonstration program proceed as follows. First load the demonstration BASIC program. BLOAD the machine code program (figure 2). The BASIC program now exists from 8001 to 8159, and the machine code program from 8170 to 81B3.

In direct mode, poke 0 into locations 8001 and 8002. Now BSAVE the memory image from 8001 to 81B3, and use an entry point of 8170.

The instruction to do this is

BSAVE "TEST", &H8001, &H81B3, &H8170

Now reset the computer. (Switch it off and back on). Then type

BLOAD "TEST", R

Your BASIC program should run. Stop the program (^STOP, whatever) and try to LIST it. The LIST command should do nothing, Try just BLOAD'ing the program and LISTing that. Again nothing should happen. Your program is fairly effectively protected against prying eyes, but can still be run. Of course you would keep a standard BASIC copy SAVEd for your own use, and the protected copy would be what you gave away, sold or whatever.

If the scheme didn't work, in other words LIST did work, then something went wrong somewhere. Start again carefully.

You're probably thinking something like – '... I'm not the only one reading this article, anyong else reading it can break the scheme just by POKEing the relevant non-zeroes back in ...'. True, so use a bit of ingenuity. Alter the small machine code program at HOOK to fill the memory from 8001 to 8159 with zeroes, to do a reset or something else. This program is just a start point, and that small portion can be altered at will.

That protected our demonstration program.

For a general program you need to know VARTAB and the bytes at 8001/02. If you have a copy of ZEN, insert these bytes at lines 16 and 18 of figure 2.

Change the ORG and the LOAD lines to start about 20 bytes after VARTAB. Assemble and then save the memory image from your ORG to the end. (Assemble with option V to see the last address used), then load the BASIC program, poke zeroes into 8001/8002 and BLOAD the saved memory image.

Now BSAVE from 8001 to the end (where the assembled program finished) with an entry point equal to ORG. This should BLOAD and run successfully with no chance of being listed.

Note that if VARTAB exists somewhere in the ZEN program (ie from A000 to BB5D) you must use one of the following methods or ZEN will get overlaid.

Extra work

If you don't have a copy of ZEN, you need to do some extra work. I have supplied a BASIC program (**Figure 3**) to do some of it, some of the time.

The BASIC program can only be used if your program to protect has VARTAB after 858A. If it doesn't, do not use this

Figure 3.

```
10 :
20 ' GET PARAMETERS
30 :
60 PRINT"BYTES AT 8001 & 8002 ? ";:INPUT I0
70 PRINT"VARIABLE TABLE ADDRESS ? ";:INPUT VT
71 IF (VT - 256*PEEK (&HF7EF)+PEEK (&HF7EE)-64*(1024) OR (VT < &HD500) THEN 70
75 LO = VT+255
80 :
90 ' ENTER PROGRAM
100 :
110 I=LO
120 READ A$:IF A$ <> "*" THEN POKE I,VAL("&H"+A$):I=I+1:GOTO120
130 :
140 ' SET UP INTERNAL ADDRESSES
150 :
160 DEFSTR T
170 T1 = HEX$(LO+51):X=22:GOSUB560
190 T1 = HEX$(LO+55):X=43:GOSUB560
200 T1 = HEX$(I0):   X=1: GOSUB560
210 T1 = HEX$(VT):   X=7: GOSUB560
220 :
230 ' TELL USER WHAT TO DO
240 :
250 PRINT
260 PRINT"THIS PROGRAM IS NOW COMPLETE "
261 PRINT
262 PRINT"NOW BSAVE THE MEMORY IMAGE FROM"
263 PRINT
264 PRINT"       &H"HEX$(LO)" TO "
265 PRINT"       &H"HEX$(LO+70)"
266 PRINT
270 PRINT"LOAD YOUR BASIC PROGRAM NOW"
280 PRINT"WHEN THE PROGRAM IS LOADED"
290 PRINT"TYPE"
300 PRINT
310 PRINT"       POKE &H8001,0"
320 PRINT"       POKE &H8002,0"
330 PRINT
340 PRINT"THEN BSAVE THE PROGRAM USING "
350 PRINT"THE FOLLOWING PARAMETERS AND "
360 PRINT"A DIFFERENT NAME TO ORIGINALLY"
380 PRINT
390 PRINT" START -- &H8001"
400 PRINT" END   -- &H"HEX$(LO+70)
410 PRINT" EXEC  -- &H"HEX$(LO)
420 PRINT
430 :
440 ' DATA SET UP TO GO FROM B170
450 :
460 DATA 21,09,80,22,01,80,21,59,81,22,EE,F7
470 DATA 21,0B,FD,22,1A,FA,22,1C,FA
480 DATA 21,A3,81,ED,5B,1A,FA,01,04,00,ED,B0,ED,53,1A,FA
490 DATA 3E,C3,32,94,FE,21,A7,81,22,95,FE
500 DATA C3,C7,09
510 DATA 52,55,4E,0D,AF,11,01,80,12,13,12,3E,C9,32,94,FE,C9,*
520 END
530 :
540 ' SUBROUTINE TO ALTER MEMORY
550 :
560 T=STRING$(4-LEN(T1),"0")+T1
570 POKE LO+X ,VAL("&H"+RIGHT$(T,2))
580 POKE LO+X+1,VAL("&H"+LEFT$(T,2))
590 RETURN
```

BASIC program in figure 3, as it is likely that it will start overwriting itself.

In this BASIC program, it is easiest to enter values in hex, for instance if PEEK(&H8001)=&H18, and PEEK(&H8002)=&H80, then when requested to enter this type &H8018.

To use figure 3, find VARTAB and the bytes at 8001/02. Load and run the BASIC program in figure 3. Follow the instructions carefully.

You are told at the end to BSAVE a memory image. Do as instructed, then LOAD the BASIC program to be protected, poke zeroes into 8001/02, BLOAD the memory image, and BSAVE from 8001 to the point where the BASIC program from figure 3 told you, using the given entry point.

To protect a program with VARTAB above 838A. You could still use the program in figure 3 but give it a value for VARTAB in the correct range. This would give a bigger memory image to save.

Otherwise carry out the following steps carefully.

(1) LOAD the program to be protected and find VARTAB. In the following steps replace VARTAB with this value.

(2) BLOAD the machine code program (figure 2) at VARTAB+20. To do this you use an OFFSET when BLOADing the program. This will be VARTAB+20 – &H8170. The instruction is

BLOAD "filename", VARTAB+20 – &H8170

(3) Calculate
X1 = VARTAB + 20 + 1
X2 = VARTAB + 20 + 7
X3 = VARTAB + 20 + 22
X4 = VARTAB + 20 + 43 (by hand or direct mode)

(4) In direct mode, enter the following
POKE X1, PEEK(&H8001)
POKE X1+1, PEEK(&H8002)
POKE X2, PEEK(&HF7EE)
POKE X2+1, PEEK(&HF7EF)

(5) Now enter? HEX$(VARTAB + 20 + 51). This gives you a 4 digit hex number. The last 2 digits need to be POKEd into X3, and the first 2 are POKEd into X3+1. For example if this gave you AA5E, type
POKE X3, &H5E
POKE X3+1, &HAA

(6) Repeat this using ? HEX$(VARTAB + 20 + 55) and X4.

(7) Now POKE 0 into locations 8001 and 8002 (POKE &H8001,0 and POKE &H8002,0), then type
BSAVE "filename", &H8001, VARTAB +90, VARTAB+20

Your program should now be able to be run in protected mode.

Type BLOAD "filename", r.

If any problems occur repeat the procedure.

For some programs it may be necessary to BSAVE the memory image after step 6, from VARTAB+20 to VARTAB+90. Then load the BASIC program to be protected, and BLOAD the memory image again. Then carry on from step 7.

Naturally there's got to be a catch somewhere in all this.

Some programs (very large ones with lots of variables) will hang the system when you finish them. The reason is that the variable table gets so big that it overwrites the machine code program that the hook at MAIN calls.

When the program finishes, MAIN attempts to run the code, but finds only garbage and runs that instead. It won't affect the program running, but a system reset will be necessary to finish.

All the locations above are for SVI. For MSX here is a table to replace the SVI values.

SVI Value	MSX Value
FA1A	F3F8
FA1C	F3FA
F7EE	F6C2
FE94	FF0C
09C4	4134
FD8B	FBF0

Hooks FE79 – FFB2; FD9A – FFC5

The SVI hooks are 3 bytes each, MSX and 5 bytes.

A Spectrum like the Amiga

by Gary Parker

Sinclair is 'dead' – long live Sinclair.

As you will probably have heard, Sinclair has been sold to Amstrad. Part of the deal is that Sir Clive Sinclair cannot produce another computer bearing the Sinclair name, although it is rumoured that he could produce a computer under the ancestral family name of Saint Claire.

So will we ever see another computer bearing the name Sinclair?

As far as I am aware, Amstrad have not stated whether they will market any computers under that name or not. But most pundits believe that they will produce Sinclair computers, since the name is too well-known to waste.

Amstrad's version?

What sort of Sinclair computer will Amstrad produce?

The Spectrum and QL will not be continued, and obviously Amstrad will not want to produce a computer too similar to any of the Amstrad models.

When Sinclair was sold to Amstrad, Amstrad also bought the rights to a computer that Sinclair was developing – the "SuperSpectrum".

Designed to replace the Spectrum, the SuperSpectrum is a computer "with graphics and sound the likes of which have never been seen even at five times the price", according to an airmailed copy of the June issue of Sinclair User.

Amiga-like

The SuperSpectrum is designed to sell in Britain for £199, and is said to be inspired by the £1500 Commodore Amiga.

So what has the SuperSpectrum got? It has 128K memory, expandable to 1 Megabyte through bank switching. It has several screen modes, with a highest resolution of 512 pixels across the screen, with 16 colours.

It has amazing sound capabilities, with an optional sound sampler and keyboard available. It has built-in CP/M.

Being a bit of a sceptic, I had my doubts when I heard these specifications. How can Sinclair produce an Amiga-style computer for that sort of price?

The SuperSpectrum's capabilities come through clever use of old 8-bit technology, similar to the way Amstrad computers are designed. These capabilities come from two main additions to the basic 8-bit computer.

7 Mhz processor

First of all, the SuperSpectrum contains an upgraded version of the familiar Z-80a processor called the Z-80h. This new processor runs at twice the speed of the Z-80a – at an amazing 7MHz.

It runs fast enough to allow an interrupt handler to control the screen display, while still allowing programs to run faster than on the Spectrum.

Secondly, the Superspectrum contains powerful customised chips, in the same way that the Amiga does. One chip handles the screen display, and another handles sound synthesis. Both have direct memory access. This allows the machine to produce incredibly fast graphics, and hi-fi sound.

Screen memory

Why are the graphics so fast on a SuperSpectrum?

Direct memory access of the custom chip allows large amounts of screen data to be shifted at speeds the CPU couldn't manage. The screen memory is 53K in size (compared with 7K on the Spectrum) and the CPU could not possibly manipulate this much memory fast enough, so the custom memory chip is essential. The chip transfers data and can make logical decisions, so that animation, selective screen scrolling and suchlike, become an automatic process.

The CPU only has to send a few instructions during each interrupt.

What sort of graphics will this produce?

The most obvious differences will be increased resolution and number of colours. The main screen mode allows 256 by 212 pixels and 256 colours.

A high resolution mode has double the number of pixels across the screen, with 16 colours. Another mode allows 256 by 212 pixels, with 64 colours and four sprites.

Built-in software allows the Superspectrum to produce amazing three-dimensional moving wire-frame graphics, similar to the Amiga.

Sound stored

The same principle of using a RAM-sharing custom chip allows the SuperSpectrum to produce amazing sounds. Waveforms are stored in memory, and can be output through a digital-to-analogue converter to a TV speaker, stereo headphones, or hi-fi system.

An optional sound-sampler will allow you to sample real sounds and use them in the music synthesizer, just like very expensive synthesizer systems.

The SuperSpectrum also has a MIDI interface, allowing it to be connected to other electronic instruments, and built-in music composing software.

'Real' keyboard & 'Softcard'

Sound the graphics excepted, what else has the Superspectrum got to offer?

I'm sure I'm not the only one who will be glad to hear that the Spectrum Plus style keyboard has been abandoned, and a real keyboard will be used.

The SuperSpectrum may have a built-in tape recorder, although the main method of storing data will be on Softcards. These credit card sized ROMs can hold a program up to 1 Mb in size and are tipped to be the storage medium of the future.

They sound marvellous, but will Sinclair owners end up out on a limb again with a storage medium no other computer uses? Only time will tell.

The list of connections to the outside world is impressive.

Spectrum

You'll be able to connect an RGB monitor, a composite monitor, or a tv to the SuperSpectrum. With the high resolution possible on the computer, a monitor will probably be necessary.

A floppy disc and a hard disc interface are provided, as well as a serial printer interface, twin joysticks, and a light pen (fitted as standard).

A genlock output will allow the SuperSpectrum to be interfaced with video recorders, laservision optical disk players, and frame grabbers. A compact disc optical ROM drive, mouse, modem, and music keyboard are also supported.

Spectrum mode

The SuperSpectrum is Spectrum compatible. In Spectrum mode, the CPU will slow down to 3.5 Mhz, and most Spectrum software will run. But Spectrum Basic will not be available to the user.

Apparently it is present in the ROM however, so Spectrum Basic programs should run.

The language used on the SuperSpectrum is a development of QL SuperBasic. The keyword entry system has been abandoned, and users will type out keywords (or abbreviations of them). The light pen fitted as standard will be an additional method of information entry.

When to appear?

All this sounds too good to be true – is there anything that this computer hasn't got?

You might doubt that the SuperSpectrum will actually appear, but I would say that while you should take some of the details with a grain of salt, most of it is probably accurate.

The QL conformed closely to the details released at the late development stage, and I have no reason to doubt the information released about the SuperSpectrum.

The basic features of the machine would be too difficult to change at this stage of development.

Features which could still change are those less closely connected with the heart of the machine, such as whether a cassette player is built into the machine, and so on.

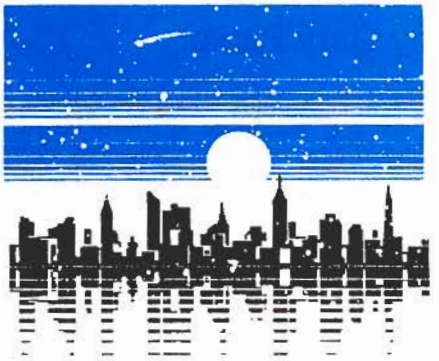
And, of course, it may appear under a different name.

So if you read about a new Amstrad computer which is taking the world by storm, it just may be a Sinclair.

Atari

The 130, comets and circles

by Ian Dopson



Recently we bought our first computer. The price seemed good, and the deal had everything needed to get going.

And just what did we buy? We bought an Atari "pricebuster" package, which included the 130XE along with a 1050 disc drive and a 1029 printer. We also received 2 game discs, Atariwriter and Syncalc.

It soon became apparent that using the TV as a monitor was not good, so we aquired a green-screen.

The "130" impresses me, the disc drive sensible, and the printer adequate but nothing to write home about.

I wanted to get the hang of plotting things on the screen so that I would be able to plot graphs and draw pictures.

A star map was a worthy exercise, I thought, showing prominent stars and then the path of Halley's Comet.

So I copied a star map with the stars and comet already on it.

The next step was to choose a graphics mode. I chose mode 7 because it was the finest at 160 by 80 pixels. Now on a bit of graph paper with 160 by 80 small squares I copied out the star map and the comets path.

Next, counting the graph squares, I plotted out the location of the stars and put them into the programme as a print command. There were lots of these.

I wanted the comet to move across the screen so that had to be treated differently. I had to give the command after the map was drawn.

I put the command in as a "PLOT X,Y", a small time delay, a loop back to the plot x,y, and feed the X,Y values in as a data line.

The whole thing was most impressive, and while we still haven't found the comet I have learnt some things about programming – and the name and locations of some stars.

The second graphics programme was to plot a circle.

Not to draw one but to generate the shape from a mathematical formula. It all starts way back when a knowledgeable gentleman by the name of Pythagoras stated that in a right angled triangle the square of the hypotenuse is equal to the sum of the squares of the other two sides.

Now by manipulating this little bit of wisdom and making the hypotenuse equal to the circle radius we can get the computer to look at lots of triangles and find the lengths of their sides. These side lengths can then be translated into commands to plot points on the screen to draw our circle.

However, on with the programme;

```
10 CLR: GR.7+16
```

This first line just clears the screen and sets the computer into high resolution graphics mode without the window.

```
20 DEG: R = 30
```

This line sets the machine to handle degrees and sets the circle radius at 30.

```
30 D=D+5
```

Here we are telling the computer to look at the circle in 5 degree increments. You can change this value for different results, try 1,2, or 10. Here we get into the good stuff.

```
40 Y=INT(R*SIN(D))
50 X=INT(((R*R)-(Y*Y))^0.5
```

Using an angle and the radius we find the length of the side "Y" in line 40, then using this value we find the length of the other side "X" in line 50. The computer will not recognise these values because they are outside the cursor range so they need to be corrected.

```
60 A=(X-80)*(-1):IF M>91 AND
   M <270 THEN A=X+80
70 B=(Y-40)*(-1)
```

Now all that remains is to plot the points

```
80 PLOT A,B
```

And lets not forget the loop to run the whole thing

```
90 IF M<360 THEN GOTO 30
```

Well that's it then. Even if you do not understand the mathematics of the thing, don't worry, it will still work.

And what can you do with a circle?

I suppose that you could invent the wheel, but that's old hat.

It is the start of a PIE graph or an element in a graphics display.

The purpose I used it for was to check the accuracy of my VDU.

If the display is a true circle then no problem, if on the other hand the shape drawn is like an egg (as mine was) then the VDU is not reproducing truly.

A minor adjustment inside the VDU will correct this. However, if the display has always looked fine to you then leave well enough alone.

Pascal vs Basic

by Bruce Simpson

If my mail is anything to go by there are an awful lot of Pascal programmers out there who read this column (I hope the editor is reading this).

In my last column (June issue) I included a small program that points out one of the ambiguities that can occur within the Pascal language. It looks as if just about everyone who tried the program wrote to me about what I did wrong. Yes, I admit it, there was a deliberate mistake (who am I kidding?) Unfortunately I sent in the wrong listing for publication and as a result, it would not compile properly.

If you change the following line in procedure INIT from...

Number[Count] :=Count;

to

Number[Count].Value :=Count;

and the following line in procedure TEST from...

Writeln('The value of ',Written,' is ',Number)

to

Writeln('The value of ',Written,' is ',Value)

The program will compile and run correctly under Turbo Pascal or MTPLUS Pascal. The resulting output will look like this...

```
Test of the WITH statement
The value of One is 1
The value of Two is 2
The value of Three is 3
The value of Four is 4
The value of Five is 5
End of test
```

So what, I hear you saying, that's just what you should get. In this case the WITH statement has been used according to the rules of the Pascal language. Now change the procedure TEST to look like this...

```
Procedure TEST;
Var Count: Integer;
BEGIN
  With Number[Count] Do
    For Count := 1 to 5 Do
      Writeln('The value of
    ',Written,' is ',Value)
END; (Test)
```

When you run this version of the program the results will range from a runtime error exclaiming that you have tried to access an array entry that does not exist through to a screen full of garbage.

The WITH statement is supposed to simplify the task of using structured variables (Records). As I discussed in an earlier article, Pascal has the ability to group many different variables together and give them a collective name. This is done through the use of records.

In the above example we have used a simple record which contains two variables of fields. The RECORD is defined as containing a variable called VALUE which is an integer and a variable called WRITTEN which is a string. When we define a variable of this record type (we used the variable name NUMBER), we must normally tell the compiler the name of the record variable as well as the name of the variable within that record. To simplify...

To use the value of the number contained in the array NUMBER at entry three we must write

NUMBER[3].VALUE

the record variable
the array entry
the variable within the record

Obviously, in cases where we need to

do a lot of referencing of VALUE and WRITTEN, the record variable (NUMER[]) would be repeated over and over if it wasn't for the WITH statement.

By using the WITH statement we can avoid having to constantly mention the record variable. Therefore, instead of having to write:

```
WriteIn(Number[3].Written,' = ',Number [3].Value);
```

we can simply write:

```
With Number [3] Do
  WriteIn(Written,' = ',Value);
```

After seeing the first line, the compiler automatically checks to see if any of the variables used are part of the record NUMBER[3]. If they are then it treats those variables just as if they had been written out in full.

The changes we have made to the procedure TEST in our little program show one important thing about the operation of the WITH statement. The value of COUNT is not re-evaluated each time the loop is run. As a result you could find yourself trying to find the bug in a piece of code that looks deceptively correct.

Many thanks to those of you who took the time to write to me with your observations.

Another piece of correspondence I think worthy of publication (just to show that I have an open mind) comes from Mr William J. Hayes, of Palmerston North.

Dear Bruce,

I have just been reading your PASCAL Part III article in Bits and Bytes (June) and I am intrigued.

I have been programming in BASIC for years and have never come across such a lame excuse for condemning BASIC. We all know how much *some* people hate BASIC and how *some* love PASCAL but if you want to criticize a language and praise another I don't think it is at all fair to do the kind of thing you did.

Yes, I do accept that the way you suggested a DATABASE in BASIC is very unwieldy and dangerous. Have you ever heard of TWO DIMENSIONAL ARRAYS and if necessary the VAL (code for converting STRING to NUMERIC.)

No, you don't have to go and change each of the field subscripts as you suggested – all you have to do is assign a K(J) to each record and sort or anything you like with this variable.

As for advocating PASCAL for a database I do think it is the craziest idea yet. Imagine the situation: I want to find the telephone number of a friend in a hurry. 1 Turn on computer etc. 2 Load

up PASCAL 3 Load up DATABASE program and perhaps switch to the FILE disc. Then you are ready to start searching.

PASCAL as a language is to my knowledge not available resident in (m)any home computers and so must be "loaded up". Also PASCAL is really only used at a few educational (tertiary) institutes as a *good* Structured program. As if you cannot structure in BASIC.

Apart from that yours is a very well written article and perhaps some people will either get PASCAL or improve their standard of BASIC.

William J. Hayes
Palmerston North

I am glad to see that I'm being read by people other than 'dyed in the wool' Pascal programmers and welcome a little open debate on the matter of BASIC vs Pascal, although I feel that neither is in all cases 'better' than the other.

Perhaps my BASIC programming technique is not as refined as Mr Haye's and his method of using multi-dimensioned arrays and converting numbers to and from strings as required could simplify the code. The big problem here however is that there are quite a few BASIC interpreters that don't allow multi-dimensional arrays (I think Atari BASIC is one). An even greater problem with almost all BASIC interpreters is the effect of garbage collection in the string area. If you are constantly assigning and changing string variables eventually the interpreter will run out of fresh string space. At this time your computer will lock up for a small (and sometimes not so small) period of time as the old space is tidied up. The more string variables you use, the more apparent the problem. Converting all variables to strings, therefore, could greatly affect the speed of the program although the use of a 'tag field' sorting technique as suggested would speed up the sort process in both Pascal and BASIC.

I think that Mr Hayes is perhaps not familiar with the type of computer on which most Pascal systems are run. He defines the startup sequence as a 3 step operation. Quite the contrary! Once your Pascal program is compiled into a machine code program all that's required to run the program is to insert the program disk and type the name of the program. That's even simpler than typing RUN "DATABASE" as you would have to on a computer with a resident interpreted BASIC. In fact, there are now third party products that allow you to write memory resident 'pop-up' programs (like Sidekick) in Turbo Pascal. Try that in BASIC.

Finally, modern Pascal compilers are not simply limited to the academic or classroom environment. A considerable amount of 'real' software is being produced using a number of Pascal compilers. Some of this software is pretty impressive and includes:

- DataFlex, the development system used to write the CBA software suite (Pascal MT+)
- SBS Accounts, one of NZ's most successful micro-computer based accounting programs (Pascal MT+)
- Cashlink, an internationally accepted accounting program (UCSD Pascal)
- MicroSoft Fortran Compiler (MS Pascal)
- Trader Series accounting software (Turbo Pascal)

I think most people will agree that although the Jensen & Wirth and ISO standard versions of Pascal are not suitable for 'real world' applications, modern compilers such as MT+, UCSD and Turbo are truely powerful and practical programming tools.

Books On Turbo Pascal

If you are using Turbo Pascal you may find a couple of new books sighted recently in my favourite computer book store to be worthwhile reading. Both books deal with the task of programming in Turbo Pascal and appear to be well written and full of useful information for Turbo users. The best source of learning for novice Turbo programmers however is probably the Turbo Tutor disk and book set from Borland themselves.

A general scan of the shelves revealed that the number of books on Pascal programming has grown considerably in the last 12 months. I have also noticed an increase in the number of job vacancies in the computer industry that list Pascal programming as a pre-requisite.

Workspace management in Logo

by Paul Left

Apple Logo is an excellent tool for teaching programming and as a medium for experimentation for programmers of all ages.

The ability to write programs through a structured and yet intuitive process is its most appealing feature.

Apple Logo is elegant enough to have been an influence on many of the versions around for other micros. Its insistence on logical consistency, however, makes it a little cumbersome to use at times.

When programming, for example, it is not possible to SAVE your present work without first deleting any existing file with the same name.

This is consistent with the Logo philosophy but can be annoying for anyone who has worked in BASIC and DOS and expects a SAVE to overwrite the existing file.

However, because Logo provides a 'seamless' environment with no distinction between Logo commands and DOS commands, it is possible to build procedures to make life easier for the Logo user.

A brief look at how Logo organises procedures and stores them on disk is in order before looking at solutions.

Packages to bury

When you write a Logo procedure, it is stored as text in RAM. Any procedure can be bundled into a larger entity called a PACKAGE.

A package can be hidden from most operations with the BURY command, so that the procedures in it can be called by name but are transparent to the user.

Unless the package name is specifically included, these procedures are not affected by a SAVE or POTS (which displays a list of defined procedures). When you type SAVE filename, all procedures in your workspace which are not in buried packages are saved in a file under that name.

This can cause problems, because the filename may bear no relation to the package names, and trying to UNBURY procedures after LOADing can be a frustrating experience.

While in a classroom situation this arrangement is convenient, it makes the programmer's life less so.

This article attempts to provide a set of Logo tools which add useful commands to the Logo environment (for the programmer or the classroom) and which illustrate the use of interesting Logo techniques beyond the familiar turtle graphics commands.

Procedures

In the accompanying listing, the '@' signs represent a 'ctrl-Q' character, which tells Logo to print the following character.

This is necessary as Logo will otherwise lop off final or initial spaces in a list or word.

When entering the procedures, make sure also that all the brackets are square ones.

You can enter the procedures from within the Logo Editor or from the command mode, one at a time. It is best to

```
TO ONLINE?
CATCH "ERROR [1ON?]
PR [I'M HAVING TROUBLE WITH DRIVE 1]
LINK
END

TO 1ON?
PR [DRIVE 1:]
D1 CATALOG
TYPE [PRESS RETURN...] MAKE "READY RL
LINK
END
```

```
TO 2ON?
PR [DRIVE 2:]
D2 CATALOG
THROW "TOPLEVEL
END

TO LINK
CATCH "ERROR [2ON?]
PR [I'M HAVING TROUBLE WITH DRIVE 2]
THROW "TOPLEVEL
END

MAKE "STARTUP [BURY "DISK.TOOLS MAKE "PACKS []]
```

start with a completely fresh Logo, by pressing ctrl-G as soon as your Logo disk has booted to avoid loading startup procedures.

After typing the listing as is, type PKGALL "DISK.TOOLS and then SAVE "DISK.TOOLS.

If you want this to be your startup package, SAVE it as "STARTUP instead.

The tools will use up about 600 nodes (3K bytes) of your workspace; try typing RECYCLE PR NODES if you want to know how much free space you have left.

With the "DISK.TOOLS package in memory, you have access to these extra commands:

FILES: prints a list of packages and the procedures in each.

FILE: lets you package a procedure and adds its name to a list.

PK: tells you what package a procedure belongs to.

D1: sets the current disk to DRIVE 1.

D2: sets the current disk to DRIVE 2.

STORE: saves any named package to disk under the same name.

UPDATE: deletes the names file and saves new version to disk.

RENAME: renames a disk file.

ONLINE?: catalogs drive 1 then drive 2.

D1 and D2 are trivial; they merely provide 'shortcut' commands.

STORE ensures that a package is SAVEd, whether buried or not, under the same name. If this is used instead of SAVE, controlling packages and procedures becomes easier, especially where there are several users as in a classroom.

After the package has been STOREd once, future versions can be saved with UPDATE, as this will delete the existing file of the same name before saving.

This saves typing ERASEFILE "filename first... if you're like me, you will often forget to do this and have to be reminded by Logo.

Using UPDATE avoids being beeped at by the Apple.

Catch

ONLINE? doesn't really do anything you couldn't do with a couple of CATALOGs, but illustrates the use of the primitive CATCH. The CATCH "ERROR [ION?] line tells Logo to run "ION? and return if there is an error condition.

Its use is different from that of 'ONERR GOTO' familiar to BASIC programmers.

The THROW "TOPLEVEL returns control to Logo and is the 'other end' of the CATCH. The four procedures involved in the operation are worth studying if you are not familiar with THROW and CATCH, which are not restricted to error-handling.

CATCHing an error in this way means that the normal error message is not printed, and control does not pass to toplevel. In this case, if an error occurs in trying to CATALOG drive 1 (if the drive door is open, for example), control passes to LINK and 2ON? to attempt a CATALOG of drive 2.

Without a CATCH "ERROR, a problem with drive 1 would cause a return to toplevel and prevent the CATALOG of drive 2.

FILE puts a named procedure into a named package, and adds the procedure name to a list with the same name as the package.

If the package is a new one, its name is added to the list "PACKS. Note that :PROC is the name of the package, and that THING :PROC is the list with the same name.

NAMEP outputs TRUE if its object is a defined variable, just as DEFINEDP outputs TRUE in "PK if its object is a defined procedure.

```
TO FILES
MAKE "TMP :PACKS
LABEL "LOOP IF EMPTYP :PACKS [MAKE "PACKS :TMP STOP]
TYPE FIRST :PACKS TYPE ":@ PRINT THING FIRST :PACKS
MAKE "PACKS BF :PACKS GO "LOOP
END

TO FILE :PACK :PROC
PACKAGE :PACK :PROC
IF NOT NAMEP :PACK [MAKE :PACK [] MAKE "PACKS LPUT :PACK
:PACKS]
MAKE :PACK LPUT :PROC THING :PACK
END

TO PK :PROC
PR []
IF NOT DEFINEDP :PROC [TYPE :PROC PR [@ IS NOT A PROCEDURE!]
STOP]
MAKE "PK GPROP :PROC "PROCPKG
TYPE :PROC IF EMPTYP :PK [PR [@ IS NOT PACKAGED] STOP]
TYPE ["S@ ] TYPE [PACKAGE:@ ] PR :PK
END
```

```
TO D1
SETDISK 1
END

TO D2
SETDISK 2
END

TO STORE :PKG
SAVE :PKG :PKG
END

TO UPDATE :PKG
ERASEFILE :PKG
STORE :PKG
END

TO RENAME :OLD :NEW
PKGALL "TEMP
BURY "TEMP
LOAD .OLD
SAVE :NEW
UNBURY "TEMP
END
```

Easy to follow

The procedure FILES displays all procedures in the workspace, listed under their appropriate packages. The information is stored in the lists named in "PACKS, which are maintained by FILE.

FILES illustrates the use of the comands GO and LABEL to control the flow of execution within a procedure.

While there are other ways of achieving the same results, possibly more elegant, GO and LABEL are convenient, easy to use, and result in easy to follow procedures.

Note also the use of "TMP to temporarily store the list "PACKS so that it can be restored at the end of the loop, which takes each element of the list in turn, printing the list associated with that name.

Any procedure which has been FILEd and then ERASEd, however, will still be listed by FILES.

You would need to type EDNS to edit the variables and remove the procedure's name from the appropriate package's list.

Modifications

You should find these procedures useful in a classroom or any situation where there are several people working on Logo, or if you are using Logo to further your own programming skills.

They are a starting point, of course, and any set of Logo tools will sooner or later be customised to suit your own needs. That's implicit in the Logo philosophy.

Here are some suggestions for useful modifications, starting with the easiest:

(1) Write procedures CAT1 and CAT2 to catalog drive 1 and drive 2.

(2) Write a procedure PRFILES to send the output of FILES to a printer using the commands .PRINTER 1 and .PRINTER 0.

(3) Write a procedure WIPE to ERASE a procedure and remove its name from the list association with the appropriate package.

The last procedure should prove a challenge.

You can retrieve the name of the package for any procedure by using PK, then search for that name and remove it from the correct list.

There are many other ways to improve the performance of the set of tools, and plenty of scope for adding further commands.

The only restriction is lack of free nodes in Logo workspace.

However, you can use the set of tools to package themselves up into separate packages and files for specific purposes.

Take advantage of the flexibility of Logo to construct your own programming environment, extend your programming skills, and challenge your own imagination.

Prog above

Measuring up Mini Office II

by Craig Beaumont

Mini Office II from Database Software combines word processor, database, spreadsheet, graphics, label printer and communications modules all in the one menu driven package for around $70 (disc version).

Menu driven means that to make things happen you move a cursor between options with the arrow keys, pressing RETURN for the one you want to use.

Considering the price I didn't expect any of the modules in this package to break new ground for the Amstrad. And this is generally true, with some interesting exceptions.

The word processor has similar abilities to manipulate text as Tasman products. With available memory of 17K it falls between Tasword 464 and Tasword 464-D in terms of document size, and in most ways it tends to fit into this category.

One feature it doesn't share with Tasman products is its ability to remember abbreviations for long words. Once you have entered a word, all you do is type the first two letters then press COPY and the rest appears.

A spooling option lets you start typing while printing another document. Print formatting is handled well through menus, but the selection of print mode (italics etc.) uses commands embedded in the text, in which you detail the control codes to be sent to your printer – a rather messy method.

Merges records

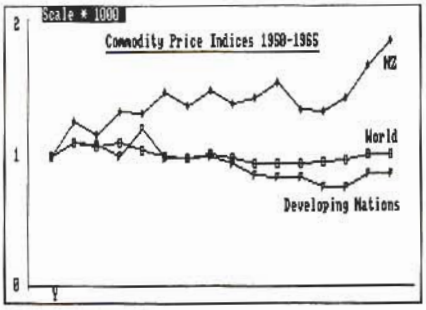
Records from files created by the Mini Office II database can be integrated into documents as you print them – like the mail merge feature of 464-D.

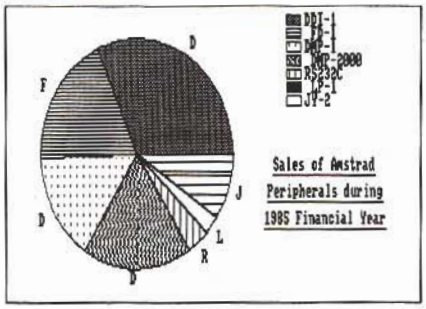
I found the database faster to set up and simpler to use than Campbell Systems' Masterfile. It allows up to 20 fields per record with edit, search, sort and calculate options.

The spreadsheet is not impressive. The main limitation is the number of functions. If you can live with only addition, subtraction, multiplication, division, total, min and max and don't need to insert or delete rows/columns then it might be O.K. Otherwise it's main use will be to save data for the graphics module to display.

If you have Supercalc 2 then this may appeal to you as a cheap alternative to Digital Research's DR Graph.

The graphics module can make bar, line and pie charts from up to 3 sets of 20 data items. Bar charts can show data





side-by-side, stacked or in 3D. Line charts can have up to 3 lines, be cumulative and have a grid background. Pie charts can have emphasised segments and you can select how you want each segment shaded.

It's easy to label axes and title graphs, quickly producing a finished display that can be saved to disc or tape or dumped to printer. This is one of the exceptions mentioned earlier.

Label print

The label printer is for those with a mailing list, like a club newsletter. Its main use is the printing of addresses, held in a database, on to sticky labels ready to be attached to letters.

The communications software is made for use with the Amstrad RS232C Serial Interface. I have no ability to test it, thought it does offer most protocols and allows you to define your own.

The package is compatible with the AMX Mouse – the idea being to make use of the menus easier, also to improve movement around both the spreadsheet and word processor. The Computer Experience who supplied Mini Office II also lent me Mighty Mouse. Unfortunately Mighty Mouse is not fully compatible with the AMX Mouse and also lacks software, unlike the AMX Mouse with its Art and Control software.

Mini Office II's general features include adjustable paper, border and ink colours in all modules. It has the ability to use RSXs like IERA at most menu levels and comes with a program to convert Mini Office I files.

The manual is only 60 small pages long, all features are briefly explained – without any examples however. Overall, Mini Office II is less than the sum of it's modular parts. I think it will be bought for its comms module or its graphics module but not as a total "office".

Mouse control

While working with the mouse I scratched together a little program for 8-directional joystick/mouse control with some random graphics attached.

When running it, just move your joystick or mouse to control the cursor, and press fire or execute when you want to draw.

To get a different effect take GOTO 60 out of line 90. The colours change because a new ink and pen is selected each time you start drawing.

The main part of the program is lines 70 to 90. It finds out what you are doing with the joystick and takes the appropriate action.

```
10 REM Joystick/Mouse Prog
20 MODE 0:RANDOMIZE TIME
30 DEFINT a-z:x=320:y=200
40 a=RND*10:d=RND*13+2:INK d,RND*27
50 b=RND*640-320:c=RND*400-200
60 PLOT x,y,1:PLOT x,y,0
70 IF JOY(0) AND 1 THEN y=y+a
   ELSE IF JOY(0) AND 2 THEN y=y-a
80 IF JOY(0) AND 4 THEN x=x-a
   ELSE IF JOY(0) AND 8 THEN x=x+a
90 IF JOY(0) AND 16 THEN
   DRAWR b,c,d:GOTO 60
100 IF INKEY$="h" THEN 30:REM
    Key H returns cursor to centre.
110 GOTO 40
```

Boulder Dash

Boulder Dash by First Star Software and Mirrorsoft is just as much fun to watch as it is to play. As Rockford the burrowing bug you nimbly dash around the fast scrolling cave collecting gleaming jewels, being careful not to let undermined boulders fall on you. Once you have collected a sufficient number within the alloted time a door opens through which you may go to the next cave.

Once you master level one there are four more to try – with less time, more deviously placed boulders and more baddies on your tail.

The BITS & BYTES Computer Book Club

We have serious books for serious business & professional people

Help yourself from this unmatched selection of the latest releases.

Macintosh Midnight Madness: Utilities, Games & Diversions in Microsoft BASIC for the Apple Macintosh.
Mitchell Waite, Dan Puttman, Don Urguhart & Chuck Blanchard

Explores horizons of Microsoft BASIC for Macintosh. Includes 17 inventive programs, along with programming insights, techniques and listings from The Weaite Group.

Penguin **Our price $44.40. Save $3.60.**

Microsoft Macinations: An Introduction to Microsoft BASIC for the Apple Macintosh
Mitchel Waite, Robert Lafore & Ira Lansing

Hands-on tutorials on the fundamentals for newcomers to programming, while for experienced programmers, there is in-depth information on the Mac's menus, windows and buttons, QuickDraw graphics routines, event-trap programming, graphics animation, and multi-channeled sound generation. Packed with wide range of program examples.

Penguine **Our price $46.25. Save $3.75.**

Excel in Business **Douglas Cobb**

Complete source of detailed, inside information on Excel, the new integrated spreadsheet for the Macintosh. Shows how to maximise the spreadsheets, create illustrative charts, turn the spreadsheet into a database and switch on the power of Excel's macros. Also plenty of advice on authoritative advice and advanced techniques.

Penguin **Our price $55.50. Save $4.50.**

Creative Programme in Microsoft BASIC for Optimal Macintosh Performance
Steve Lambert

Selection of original programs for intermediate programmers, including 17 BASIC utilities and games, communications program, a program which transfers images from BASIC to another program, and an analog to digital converter.

Penguin **Our price $44.40. Save $3.60.**

The BBC B Compedium **Spencer Boston & Ian Thomas**

Combines fun and function as you learn by doing through 40 fully documented programs, plus programming tips and a comprehensive reference section. Programs include games and utilities.

Penguin **Our price $19.40. Save$1.55.**

Command Performance dBASE III
Douglas Hergert

Concise, thorough explanations and practical examples of all dBASE III commands and functions, plus special comments and cautionary notes. Alphabetical format and multiple cross referencing for additional information. Explains how to work effectively from two or more databases, what cautions to keep in mind when using memo fields, how to use date and time values, when to use procedure files for more efficient programming, what other programming tools dBASE III offers.

Penguin **Our price $55.50. Save $4.50.**

Variations in C **Steve Schustack**

Systematic approach detailing basic syntax, and advanced techniques to release full power and flexibility. Schustack, who has taught C seminars to leading computer and software manufacturers, uses in approach based on the proposed ANSI standards, using MS-DOS and the Microsoft C computer, version 3.

Penguin **Our price $46.25. Save $3.75.**

Dictionary of Computers **Anthony Chandor, John Graham & Robin Williamson**

Defines computer language – words, phrases and acronyms in helpful, practical language. Careful cross-referencing allows a non-specialist to track a clear path through the jargon.

Penguin **Our price $12.00. Save 95c.**

The Printed Word: Professional Word Processing with Microsoft Word on the Apple Macintosh
David A. Kater & Richard L. Kater

After taking you through Microsoft Word's features, the authors give advice on choosing and setting up a printer, then include a special projects section covering individualised form letters on print merge command, multi-column newsletters incorporating graphics and tables, customised business forms, high-quality brochures.

Penguin **Our price $41.60. Save $3.40.**

Presentation Graphics on the IBM PC
Steve Lambert

Shows how to select, create and modify Microsoft Chart (Version 2) to column, bar, line, pie, high-low, area, scatter or combination – whichever best communicates your message. Sixty output devices – from plotter to slide makers to colour printers.

Penguin **Our price $48.10. Save $3.90.**

The club: how it works and what you get.

All you have to do to join the club is buy a book. Just pick out the books you want, fill in the coupon in the middle of the magazine, and post it in FREEPOST.

We offer savings on the cash price you pay for each book

Please allow three to four weeks for orders to be processed and the book distributors to get the books to you.

Computer Shows... who needs them?

by Gaie Ellis

The 1986 Wellington Computer Show saw a lively three days trading, wheeling and dealing in the Capital last month. In spite of the cold weather around 8,000 people registered attendance at the show and in light of trends overseas where computer shows are attracting smaller crowds and companies are opting for alternative ways of presenting their goods, the writer took the opportunity to talk with visitors and exhibitors alike about computer shows and their future... or otherwise.

Amongst the business people attending the seminars on Thursday and Friday, there was a keen interest in getting an overview or introduction to the subject presented. Attendees realised they would not be able to learn all there was to know from an hour and a half presentation on a given topic but found the information presented useful.

Asked why they attended the show many people expressed a preference for looking at the latest technology, and comparing it, on the one site and in a relatively neutral invironment. As with the Auckland PC show, visitors were generally looking for specific goods, about which they had read or which they had seen before; now they wanted to compare those goods with the competition and talk with various vendors before committing to a course of action.

One such visitor from a large company was disappointed to find a particular brand he was interested in not represented at the show; he cheered up considerably when an alternative was found that appeared to offer more than his original choice.

Asked about the drop off in numbers at shows, as with overseas shows, several exhibitors said the type of people attending the shows had changed. No longer were people coming to "oggle" at the new technology but they are attending shows for specific information about products and their availability.

Ashton Tate's, Suzanne Walkes, Sales manager from Australia, has been involved in exhibitions worldwide. She said she was impressed with the type of information people were seeking about software. "People know what they are looking for and their questions are very informed. They are obviously here to do business."

Other exhibitors were keen to have the opportunity to talk with representatives from government departments, and there were many at the show. With the tightened budgets in many departments these people were positive about the advantages of being able to compare products and prices in the show environment.

And then there is Saturday... the day many exhibitors express dismay about because this is when they believe they will be hit by the "tyre kickers". It never fails to amaze me that exhibitors at

shows persist in believing they will do little or no trade on a Saturday. This belief seems to come primarily from the fact it is difficult to qualify the people you are talking with, and if they are wearing jeans, a sweatshirt and pushing a pushchair, then the perception is it's unlikely they'll be doing much business.

Have computer industry retailers become so enarmoured with the prospect of selling bulk quantities of machines, large systems and software packages by the gross, that they have forgotten about the small to medium sized business that will grow and will have increasing requirements of them?

Are they so star struck by the hope of securing a contract with the frequently over-served corporate market that they are prepared to forget about the up and coming generations and to focus only on the here and now? Do they really believe that the decision makers from the corporate market do not come out with their families to shop on Saturdays?

And then, heaven help us, there are the kids on Saturdays! Saturdays at the shows appear to me to be the ideal time for exhibitors to get into some good solid public relations and image building. Instead of placing glossy advertisements in publications that might reach the student market and ingrain themselves in the minds of the purchasers of tomorrow, companies would do well to do some one-to-one bridge building. Don't lock up your machines and walk away from the youth and young families who, at your enticement, come keenly to the shows to sample your products and to seek out the options for their future. See these people for the resource they are... employees as well as customers, value their interest and encourage it because it is these people who will strengthen the industry here and who will steer its future course.

Christchurch Show

Organisation for the Christchurch Show is well under way with at least as many companies as last year taking part in what is the South Island's largest annual computer exhibition.

Seminars focussing on current trends will again be a feature of this show, which last year attracted 8,000 visitors. This year, the show will be held on weekdays – September 8, 9 and 10, opening until 8pm on Monday and Tuesday evenings. Any-one wanting further information about the Christchurch Show should contact the Bits & Bytes offices in Auckland, Wellington and Christchurch.

Store opens in Hamilton

Porterfield Computers have opened a new store in Hamilton this month. Like their Auckland and Wellington sites, the new store will cater for the home and low end business markets.

Managing director Bill Porter is no stranger to the trials and tribulations of retailing to this market sector: at a time when many companies are reassessing their stance in this area he has decided to consolidate his company's activities in retail computing.

Porterfield's commenced trading in 1978 with Bill Porter writing software on an early Radio Shack computer. He began selling the lion hearted ZX81 in 1981 and today his stores have developed expertise in the Commodore range, Amstrad and Atari.

For some time now there has been talk that Tandy (nee Radio Shack) has something interesting in the wings but in the meantime, Porterfield's are finding the Tandy 2000's capabilities with graphics and its operating speed ideally suited to the CAD/CAM market.

Mr Ian Potts has been appointed manager of the new Hamilton store which is situated on the First Floor, Government Life Building, Centreplace, Hamilton.

User Group Directory

AUCKLAND

ACES (Auckland Computer Education Society): C/- Director, Computer Centre; Secondary Teachers' College, Private Bag, Symonds Street, Auckland. Meetings, third Thursday of month, at the College.

APPLE USER GROUP: Ross Bryon, ph 761-670 (h). Meetings: 3rde Tuesday.

APPLE JUICE TABLOID: Philip McKenzie, 4/464 Parnell Rd, Auckland 1. Ph 798-179.

ATARI MICROCOMPUTER USER GROUP: Ian Mason, 25 Manutara Ave, Forrest Hill, ph 467-347 (h). Meets 2nd Tuesday, Western Suburbs Radio Club, Gt North Rd, New Lynn.

BBC USER GROUP: Dave Fielder, ph 770-630, Ext 518 (w). Meetings: 2nd Wednesday*.

COLOUR GENIE USER GROUP, (Auckland): Secretary: Mrs Nola Huggins, Ph 655-718, P.O. Box 27-387, Auckland 4. Meets every fourth Monday, All Saints Church Hall, Ponsonby Rd, Auckland.

EPSON HX20 USERS' GROUP. Contact: C.W. Nighy, 231 Khyber Pass Road, Auckland, (Ansaphone, 774-268).

HJP41C USERS' GROUP (Auckland): C/- Calculator Centre, P.O. Box 6044, Auckland : Grant Buchanan, 790-328 (w). Meets third Wednesday, 7pm, at Centre Computers, Great South Rd, Epsom.

LXIV N.Z.: Aligned towards those using Commodore 64's, mainly in education. Contact Brother Bosco Camden, 21E5 Richmond Rd, Auckland 2.

MS-DOS USERS' GROUP: Meets first Monday each month at 112 Mountain Rd, Epsom, Auckland. Contact Peter Biggs, ph 603-274.

MSX/SPECTRAVIDEO COMPUTER CLUB: Contact: P.O. Box 22-620, Otahuhu, Auckland, 6. Meetings, 3rd Wednesday of month at IHC Hall, 56 Ranfurly Road, Epsom. Contact Bill Ferguson (Secretary). Ph 276-1966, ext 803 (w).

NZ AMSTRAD USER GROUP: Roger Manson, 22 Ellice Rd, Glenfield. Ph 444-4044. Meets 2nd Wednesday, Salvation Army hall, Glenfield Road, 3.30pm.

NZ COMMODORE USER GROUP (AK) INC: Kay Coddington. Ph 588-931 (h). Box 5223, Auckland. Meetings: 3rd Wednesday, Remuera Primary School Hall, Dromorne Rd, Remuera.

NZ COMPUTER COMMUNICATION USERS' GROUP: P.O. Box 6662, Wellesley Street, Auckland 1. Stephen Williams.

NZ OSBORNE USERS GROUP (NZOG): C/- P.O. Box 43-182 Auckland. Meetings 2nd Tuesday. Auckland University, Human Sciences Building, 7.30 pm. Neil Williamson, ph 275-4310 Auckland.

NZ TRS-80 MICROCOMPUTER CLUB: Olaf Skarsholt, 203A Godley Rd, Titirangi. Phone 817-8698 (h). Meets first Tuesday OSNZ Hall, 107 Hillsborough Rd, Mt Roskill.

OSI/BBC USERS' GROUP (Ak): Secretary: Ken Harley, 77 Boundary Road, Auckland. Meets third Tuesday, VHF Clubrooms, Hazel Ave, Mt Roskill.

SANYO USER GROUP: Noel Weeks, P.O. Box 28-335, Auckland 5. Ph 540-118 (h).

SYMPOOL (N.Z. SYM user group): John Robertson, P.O. Box 580, Manurewa, ph 2675-188 (h).

TARILAND COMPUTER CLUB INC: A club for Atari enthusiasts living in Auckland. Meeting 2nd Wednesday each month at Remuera Primary School Hall. Contact Allan Clark 8368-922 (h) 505-409 (b).

ZX81 USER GROUP: C/- 28 Haig Ave, Auckland 4. David Pomeroy, 694-339.

NZ MICROCOMPUTER CLUB INC, P.O. Box 6210, Auckland. A meeting is held on the first Wednesday of each month, at the OSNZ Hall, 107 Hillsborough Rd, Mt Roskill from 7.30 pm. Visitors are also welcome at 10am-5pm, at the same hall on the Saturday following the above meeting.

The following user groups are part of the N.Z. Micro Club. Meetings start at 7.30 pm at the OSNZ Hall.

BUSINESS USER GROUP: Sam Chan, phone 678-518 (h). Meetings: 3rde Thursday.

CP/M USER GROUP: Peter Ensor, ph 653-011 (h). Meeting: 4th Monday.

IBM PC USER GROUP: Terry Bowden, ph 452-639 (h). 778-910 (w). Meetings: 3rd Thursday.

GENEALOGICAL USERS GROUP: Maartin de Vries, ph 2674-886 (h). Meetings, 1st Wednesday, 9.15 pm.

KAYPRO USERS GROUP: Russell Clement, ph 817-8525. Meetings, 4th Monday.

POCKET COMPUTER USER GROUP: Peter Raylor, ph 576-618 (h).

SINCLAIR USERS GROUP: Doug Farmer, phone 567-589 (h). Meetings: 4th Wednesday*.

SORCERER USER GROUP (NZ): Selwyn Arrow, ph 491-012 (h). Meets at Micro Workshop.

1802 USER GROUP: Brian Conquer, ph 695-669 (h).

2650 USER GROUP: Trevor Sheffield, ph 676-591 (h).

68XX (X) USER GROUP: John Kucernak, ph 606-935 (h).

The above contacts can usually be found at N.Z. Microcomputer Club meetings and micro workshops, or via P.O. Box 6210, Auckland.

NORTHLAND

BAY OF ISLANDS COMMODORE 64 USER GROUP: Contacts – Mrs B. McLeish, P.O. Box 119, Okaihau (secretary), or Mr H. Perry, 143 Church Street, Onerahi.

KERIKERI COMMODORE 64 USER GROUP: Contact: Brett Snow, Skudders Beach, KeriKeri. Ph 78-868.

WHANGAREI COMPUTER GROUP: 15 James Street, Whangarei. Phone 84-416. Meets every second Wednesday of the month at Northland Community College.

WHANGAREI SINCLAIR USERS CLUB: Meets 1st Sunday, 1pm-5pm, Whangarei Community College. Contacts: B.M. van Gelder 83-886 or president, 81-733.

BAY OF PLENTY

A.Z.T.E.C.: Brian Mayo, Church Street, Katikati. Phone 490-326. Members all use micros.

BAY MICROCOMPUTER CLUB (Tauranga): G.L. McKenzie, Secretary, Snodgrass Road, Tauranga. Phone: 25-569.

BAY OF PLENTY TAURANGA COMMODORE USERS GROUP: Contacts – phone 62-083, 65-311, and 83-610. Meets first and third Monday of month, 7 pm.

BAY SHARP USERS GROUP: Contact – phone 86132. Meeting 2nd and 4th Monday, 417 Cameron Road, Tauranga.

BEACH COMPUTING CLUB (Waihi): Jamie Clarke, Box 132, Waihi (Ph: 45-364 Waihi Beach).

TAURANGA MAC GROUP: Contact, Clive Bolton, 81-779 (w), 62-811 (h).

WAIHI COMPUTER ENTHUSIASTS: Contact G.C. Jenkins, 10 Smith St, Waihi (h) WAH 8478. Workshops 2nd Tuesday. Meetings last Tuesay of month.

WAIKATO

THE ATARI CONNECTION. Contact Paul Cormack, 29 McDiarmid Cres, Huntly. Ph (h) 88-695.
HAMILTON SUPER 80 USERS: P.O. Box 16113, Glenview, Hamilton.
HP HANDHELD USERS GROUP: Don Garry. 11 Leonard Place, Hamilton. Ph 78-001, 391-199 (w).
WAIKATO ATARI USERS' CLUB: P.O. Box 6087, Heaphy Terrace, Hamilton. Ph Bob (071) 78-434, Albert 73-380.
WAIKATO COMMODORE USERS' GROUP: Secretary, Mrs Eileen Woodhouse, 32 Kenny Crescent, Hamilton.
WAIKATO COMPUTERS IN EDUCATION SOCIETY. Secretary, Geoff Franks, Fraser High School, 72 Elliott Road, Hamilton. Phone (h) 81-050.
WAIKATO SPECTRUM USERS' GROUP: Secretary: Roger Loveless, 18 Heath St, Hamilton. Phone 492-080. Meetings; First Tuesday of the month.
WAIKATO SPECTRAVIDEO USERS' GROUP: P.O. Box 16113, Glenview, Hamilton.
WAIKATO PC USER GROUP. Bruce Ferguson, P.o. Box 9535, Hamilton. Ph 391-670.
WAIKATO MACINTOSH USER GROUP: Geoff Pooch, Waikato Computer Services. Ph 391-670 (h).

CENTAL N.I.

ROTORUA AREA SANYO USERS' GROUP: Pam Dowling, (073) 24-17. Meets 1 pm 2nd Saturday.
ROTORUA COMPUTER CLUB: Contact: Ken Blackman, 6 Urquhart Place, Rotorua. Third Tuesday of each month at 7 pm, Waiariki Community College, Rotorua.
GLOWWORM COMPUTER ENTHUSIASTS: Meets every second Sunday of the month in the Otorohanga District Council's board room. Contacts: president, Colin Wilkins, Oto 8331; vice-president, Hugh Button, Oto 7228; secretary, Laurence Bevan, Oto 7066.
ELECTRIC APPLE USERS' GROUP: Noel Bridgeman, P.O. Box 3105, Fitzroy, New Plymouth, Phone 80-216.
TARANAKI MICRO COMPUTER SOCIETY: P.O. Box 7003, Bell Block, New Plymouth: Mr K. Smith. Phone 8556, Waitara.
SOUTH TARANAKI MICROCOMPUTER SOCIETY: Contacts: Apple, Jim Callaghan, 86-667 Hawera; S80, TRS80, John Roberts-Thompson, 84-495 Hawera; Sega, Dave Beale, 85-108 Hawera; Spectrum, Guy Oakly, 8060 Manaia. Sub groups meet on the third Wednesday of the month. The whole society meets periodically in the Hawera High School computer room. Written inquiries to Allen Goodhue, 21 Princes Street, Hawera.
TOKOROA COMMODORE USERS' GROUP: Peter Rowe, 220 Balmoral Drive, Tokoroa. Meetings 2nd Wednesday at Balmoral School Library.
WANGANUI COMMODORE 64 USER GROUP: Contact – P. Northway, Phone (h) 42-916. 7 Broadhead Avenue, Wanganui. Meets first and third Thursdays of month at Wanganui Community College.
MOTOROLA USER GROUP: Harry Wiggins, (ZL2BFR), P.O. Box 1718, Palmerston North. Phone (063) 82-527 (h).
MANAWATU MICROCOMPUTER CLUB: Contact: Richard Anger, 64-108 (w) or 63-808 (h). Meets twice a month at PDC Social Club rooms.
MANAWATU ATARI USERS' GROUP: Savern Reweti, 29 Rewi Street, ANZAF Base, Ohakea. Ph 49-746, 69-109 (ext 830).

HAWKES BAY

SHARP PC1500 USER GROUP – Contact: Allan Thomas, P.O. Box 155, Napier, Newsletter.
NAPIER VZ-200 USERS' GROUP – Contact: Peter Cox, Ph 435-126 after 4 pm or write to Peter Cox, 9 Cranby St, Orekawa, Napier.
HAWKE'S BAY APPLE COMPUTER CLUB: Meets 1st and 3rd Mondays, 7 pm Napier Boys' High School. Contacts: Bert Tripp 700-412; Charlie Lum 438-005.
HAWKE'S BAY MICROCOMPUTER USERS' GROUP: Bob Brady, Pirimai Pharmacy, Pirimai Plaza, Napier. Phone 439-016.
HAWKE'S BAY COMMODORE COMPUTER CLUB: Contacts: Raymond Chaplow, 83-589, Ann Ferguson, 83057, 1022 Sussex St, Hastings. Meetings: first Tuesday of month at H.B. Community College.
HAWKE'S BAY SPECTRAVIDEO USER GROUP: Meets first Tuesday of month at hawke's Bay Community College. Contact P. Lawrence, P.O. Box 799, Napier.
INTERACT USERS GROUP: For more information write to Denis Clark, 43 Charles Street, Westshore, Napier.
HBCES (Hawke's Bay Computers in Education Society): Contact – Grant Barnett, 89 King Street, Taradale, Napier. Ph: 446-992.
GISBORNE MICROPROCESSOR USERS' GROUP: Stuart Mullett-Merrick, P.O. Box 486, Gisborne. Phone 88-828.

NATIONAL

NZ AMIGA USER GROUP, Tony Wills, 156C Queens Drive, Wellington 3. Ph 877-258, 850-559.
BBC/ACORN COMPUTER USER GROUP OF NZ, P.O. Box 9592, Wellington. Local meetings – Auckland: 2nd Wednesday of month at Conference Room, Auckland Teachers' Training College, 60 Epsom Ave. Ph Dave Fielder, 770-630, ext 518 (b). Wellington – meets last Thursday of each month in staffroom, first floor, Correspondence School, Portland Cres, Thorndon. Local contact, Anton, 286-289. Hamilton – Waikato Tech B-block staffroom; last Wednesday of the month 5 pm. Local contacts Peter (Ham) 393-990 or Alison (Morrinsville) 6695. Hawke's Bay – Hastings and Napier alternate months. Local contacts: Kendall (Napier) 435-625, Bob (Taradale) 446-955; Mitch (Hastings) 778-235. Christchurch – fortnightly, Tuesdays, 7 pm, Hagley High School. Local contact Michael, 582-267.
SANYO USER GROUPS have been set up in Auckland, Wellington and Christchurch. Contact P.O. Box 6810, Auckland for further information.
NZ UNIX USERS GROUP, P.O. Box 7087, Auckland 1. Membership P.O. Box 13-056, Hamilton.
NOTE: Clubs would appreciate a stamped self-addressed envelope with any written inquiry to them.

WELLINGTON

AMSTRAD USERS GROUP (Wgtn), P.O. Box 2575 Wellington. Murray Thessman 722-627.
HOROWHENUA MICROCOMPUTER CLUB: Meets on second and fourth Thursday of month. President, Wally Withell, P.O. Box 405, Levin; secretary, Dennis Cole, 28 Edinburgh Street, Levin. Ph (069) 83-904.
WAIRARAPA MICROCOMPUTER USERS' GROUP: Geoffrey petersen, 27 Cornwall St. Masterton. Ph (h) 87-439.
CENTRAL DISTRICTS COMPUTERS IN EDUCATION SOCIETY: Rory Butler, 4 John Street, Levin (069) 84-466 or Margaret Morgan, 18 Standen Street, Karori, Wellington (04) 767-187.
UPPER HUTT COMPUTER CLUB: Shane Doyle, 18 Holdsworth Avenue, Upper Hutt. Phone 278-545. An all-machine club.
ATARI USERS' GROUP, Wellington: Eddie Nickless. Phone 731-024 (w), P.O. Box 16011. Meetings: first Wednesday of month.
CROG (Central Region Osborne Group): For Osborne, Amust, Kaypro & other CP/M computer users. Contact: Bruce Stevenson, 5 Dundee Pl, Chartwell, Wellington 4; ph 791-172. Meetings: 1st Wednesday, 7.30 pm.
KAPITI COMMODORE USERS GROUP INCORPORATED: President – Derek Millett, 115 Matai Road, Raumati South; secretary – Mrs Faye Deakin, 9 Buckley Grove, Paraparaumu, phone 87-869 (or 859-799 Wellington bus.). Meets 1st Friday in month Paraparaumu Primary School Library, Ruapehu Street, 7.30 pm.
MICROBEE USERS' CLUB: P.O. Box 871, Wellington, 2nd Sunday of month.
NEC COMPUTER USERS' GROUP: C/- P.O. Box 3820, Wellington.
NZ SUPER 80 USERS' GROUP: C/- Peanut Computers, 5 dundee Pl., Chartwell, Wellington 4. Phone 791-172.
OHIO USERS' GROUP: Wellington. Secretary/Treasurer: R.N. Hislop, 65B Awatea Street, Porirua.
POLY USERS GROUP, Wellington: Contact – Christine Greenbank, Computer Studies, Wellington Teachers' College, Private Bag, Karori, Wellington.
WELLINGTON APPLE USERS GROUP: Inquiries to secretary, Grant Collison, P.O. Box 6642, Wellington. Ph 872-537, evenings. Meets last Saturday of month.
WELLINGTON COMMODORE USERS' GROUP: P.O. Box 2828, Wellington. Contacts: Peter March (h) 86-701, Robert Keegan (h) 789-157, or phone 886-701.
WELLINGTON MICROCOMPOUTER SOCIETY INC.: P.O. Box 1581, Wellington, o. Bill Parkin (h) 725-086. Meetings are held in the Fellowship Room, St Johns Church, 176 Willis Street, on the 2nd Tuesday each month at 7.30 pm.
WELLINGTON SEGA USER GROUP: Meets first Thursday of month at Paprangi School Hall. Contact Shaun Parsons, P.o. Box 1871, Wellington. Phone: 897-095 after 6 pm.
SEGA OWNERS CLUB: Lower Hutt. Meets 1st Monday each month. Contact: Murray Trickett. (w) 724-356, (h) 662-747.
WELLINGTON SPECTRAVIDEO CLUB: Contact – Don Stanley, C/- Box 7057 Wellington South. Ph 746-906 (w). Meets on one Monday a month at Staff Common Room (Level D), Wellington Clinical School, Mein Street, Newtown.
WELLINGTON SYSTEM 80 USERS' GROUP: Contact: W.G. (Bill) Lapsley, day 286-175; evenings, 268-939; or Andrew Vincent 780-371 (evenings).
HUTT VALLEY COMMODORE USER GROUP: Contact – P.O. Box 46047. Phone 671-992 or 646-254 evenings. Meetings, first and third Mondays of month at St Bernard's College, from 7.30 pm.

CHRISTCHURCH

CANTERBURY COMPUTER EDUCATION SOCIETY: Contact – Graeme Sauer (secretary), P.O. Box 31-065, Liam, Christchurch 4.
CHRISTCHURCH APPLE USERS GROUP – Contact: Peter Fitchett, ph 328-189. Meets first Wednesday of month, third floor, Tower Building, Christchurch Teachers' College.
CHRISTCHURCH ATARI USERS GROUP: Contact Donna Hosking, 184 Milton Street, Christchurch 2. Ph 370-354.
CHRISTCHURCH SPECTRAVIDEO USERS GROUP – Contact: Lester Reilly, ph (h) 428-686. Meets third Tuesday of month.,
CHRISTCHURCH TRS-80 COLOUR USER GROUP: Meetings: last Wednesday of month. Contact: Dennis Rogers, 21 Frankleigh Street, Christchurch 2. Phone 34-731.
CHRISTCHURCH '80 USERS' GROUP: Brendan Thompson. Phone (h) 370-381. P.O. Box 4118, Christchurch.
IBM & COMPATIBLE PC USERS GROUP, John Kirby 881-032. 10 Rockinghorse Rd, Christchurch. Meets 1st Thursday.
OSI USERS' GROUP (CH): Tony Martin, 9 Innes Rd. Phone 555-048.
SINCLAIR USERS' GROUP CANTERBURY, INC: Contact: Gary Parker (president). Phone 894-820. P.O. Box 4063. Meets 7.30 pm last Monday of month. Phone for latest meeting place.
CHRISTCHURCH COMMODORE USERS GROUP: John Kramer, 885-533 and John Sparrow. Phone 896-099.
CHRISTCHURCH BBC and ELECTRON USERS GROUP. Meets alternate Monday nights at 6.30 except Saturday or Secondary-School holidays, at Hagley High School. Secretary, Mrs R.D. Nolan, 87 Palmers Road, Christchurch, 9. Jack 883-787, Mike 497-939.
PANASONIC (JB-3000) USERS' GROUP: Contact: Prof. B.J. Clarke, Dept of Accountancy, University of Canterbury, Private Bag, Christchurch, 1.
CHRISTCHURCH COLOUR GENIE USERS' GROUP: Meets 2nd Wednesday, 7.00 pm, Abacus Shop, Shades Arcade, Secretary, D. McEchen, P.O. Box 25-125. Ph 327-063.
CHRISTCHURCH SORD MS USERS GROUP: Meets first Thrusday of month, 7 pm.
CHRISTCHURCH SEGA USER CLUB: George Cox, ph 33-007. 17 Hillsborough Tce Christchurch 2.
DICK SMITH WIZZARD COMPUTER CLUB, Christchurch: Contact – Tony Dodd, 34 Mayfield Ave. Ph: 557-327.
CHRISTCHURCH VZ-200 USERS GROUP: Meets second Tuesday of month. Contact Brian Sheppard 228-778, Graham Dillon, ph 324-117, or P.O. Box 22-094, Christchurch 1.
CHRISTCHURCH CP/M USERS GROUP. Murray Tasker, P.O. Box 4574, Christchurch. Ph 227-252. Meets 3rd Sunday, 7 pm.

SOUTH ISLAND

ASHBURTON COMPUTER SOCIETY: Meets first Monday of month, 7.30 pm. Enquiries to Pete Boyce, 4 Willow St, Ashburton. Ph 83-664.
SOUTH CANTERBURY COMPUTER GROUP: Caters for all machines from ZX81 to IBM34, Geoff McCaughan. Phone Timaru 60-756 or P.O. Box 73.
NORTH OTAGO COMPUTER CLUB: Contact: Peter George, P.O. Box 281, Oamaru. Phone 29-106 (b) 70-646 (h).
LEADING EDGE HOME COMPUTER CLUB: Elaine Orr, Leading Edge Computers, P.O. Box 2260, Dunedin. Phone 55-268 (w).
NELSON COMMODORE USERS' GROUP: Peter Archer, P.O. Box 860, Nelson. Phone (054) 79-362 (h).
NELSON HOME COMPUTER CLUB: Contact – Mike Jenkins, Box 571. Ph 87-930. Meets, 7 pm, first and third Tuesdays of the month at Nelson Intermediate.
BLENHEIM COMPUTER CLUB: Club night second Wednesday of month. Ivan Meynell, Secretary, P.O. Box 668. Phone (h) 85-207 or (w) 87-834.
MARLBOROUGH COMMODORE USERS GROUP: Secretary, Murray Herd, 32 Rousehill Street, Renwick. Meetings: second Thursday of month, 7.30 pm. IHC rooms, Weld Street.
BULLER COMPUTER USERS GROUP: P.O. Box 310, Westport. Phone: 7956 Wpt. R.J. Moroney (secretary).
HOKITIKA COMMODORE COMPUTER USERS GROUP: Contact – ADrian Mehrtens, 185 Sewell Street. Ph: 943.
OTAGO COMMODORE 64 CLUB: Meets first Tuesday of month, 7.30 pm. Contact: Geoff Gray, 41 Eglinton Road. Ph 53-986.
DUNEDIN SORD USERS' GROUP: Terry Shand. Phone (024) 771-295 (w), 881-432 (h).
CENTRAL CITY COMPUTER INTEREST GROUP: Contact: Terry Stevens, Box 5260, Dunedin. Phone 882-603. Meetings every second Tuesday.
OTAGO COMPUTER EDUCATION SOCIETY: Jim Ferguson, Arthur Street School, 26 Arthur Street, Dunedin. Ph 776-524.
ATARI USER GROUP, Dunedin: Meets fortnightly on Thursday. Phone Graeme Wheeler 737-907 for the date, time and place of next meeting, or write to 38 Calder Avenue, North East Valley, Dunedin.
SPECTRUM AND QL COMPUTER CLUB – Contact: James Palmer, 37 Sunbury St, Dunedin. Phone 44-787, Monday to Friday after 4 pm.
SOUTHLAND MICRO USERS GROUP – Contact: Secretary B.J. Brown, 40 Elm Cres, Invercargill. Ph 88-920. Meets every second Monday at St Paul's Church Hall, 7.15 pm.
SOUTHLAND COMMODORE USER GROUP: (VIC 20 and 64s). Address: C/- Office Equipment Southland, Box 1079, Invercargill.
SOUTHLAND COMPOUTER EDUCATION SOCIETY: Secretary, Bob Evans, Southland Boys' High School, Herbert Stret, Invercargill. Ph (h), 73-050 or ZL4LX.
GORE COMPUTER CLUB: Meets first and third Tuesdays of month, 7 pm. Contacts: Allan Rodgers, ph 7488, Dave Clarke, ph 5836.
NZ SOFTWARE EXCHANGE ASSOCIATION: Non-profit group for exchange of software written by programmer members. Contact: Ian Thain, Box 333, Tokoroa.

Classified Advertising

C/16/PLUS/4 Software. Send SASE for list to Wirecom Software, Moeraki, 2RD Palmerston, Otago. Apologies are extended to anyone who has written for this list and had their letter returned. the fault was with the Post Office and has now been rectified.

SOLUTION 1 (Apple imitation) for sale – 64k RAM; disk drive; green screen monitor; detached keyboard with numeric keypad $900. Phone (067) 68148 Inglewood or write to Box 215 Inglewood.

WANTED TO BUY. 3K Super Expander for Vic 20. Pam Miles, 149 River Road, Kawerau. Phone 8867.

COMMODORE 64: LATEST COPY PROgramme from USA. Ultrabyte Disk Nibbler. Copies 99%+. As advertised in Compute Gazette. Regular update. Send S.A.E. to:
Authotech
2/34 Buffon St
Christchurch 2

COMSEC warranted computers. IBM AT style. Superb quality units. Warehouse prices. Also TRS-80 Model 4P 128K. Kaypro 2. Both with bundled software & guaranteed. Offers. S.A.E. to PO Box 30, Waihi Beach South.

GET RICH with your microcomputer. It's amazing! I will show you how. For more information send S.A.E. to Paul M. Baylis, 7 Norana Ave, Remuera, Auckland 5.

WANTED TO BUY. SV-806 80 column card for SV-601/605 expander. Please contact W. Gailer, Ph (Ak) 466-236.

APPLE II AND IBM PC SOFTWARE FOR sale. Write for large list: Software, 125 Sparks Rd, Christchurch 2.

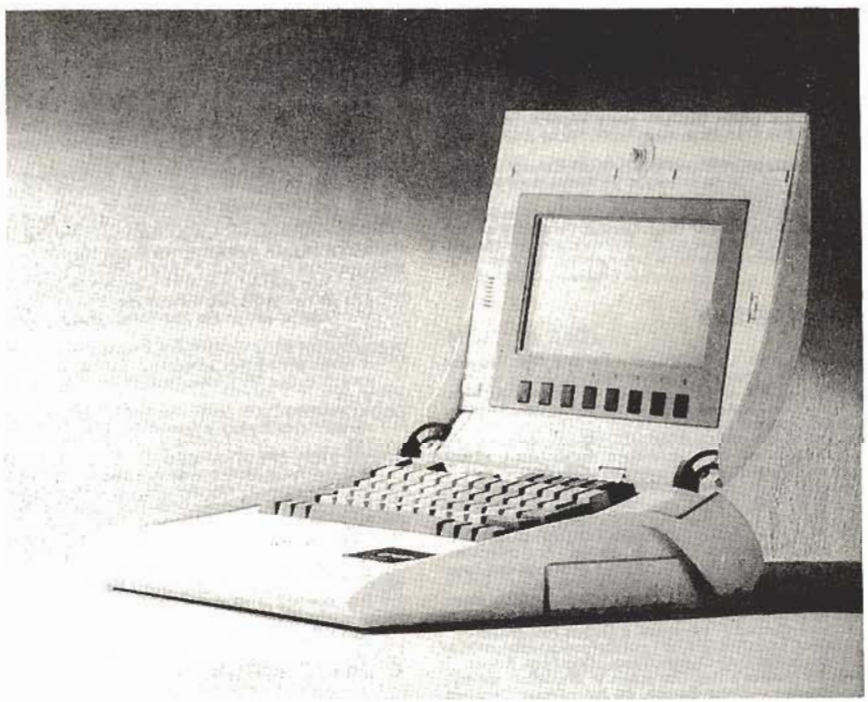
NEW SPECTRAVIDEO SOFTWARE for the SV.318 SV.328. Good quality! Write away with S.A.E. for a catalog to: "NBS Software", 313 Brougham St, Christchurch, 2. Ph 56-664.

BBC-B (32K) for sale. Includes speech unit, heaps of software and documentation, $850. Phone Neville, Christchurch 228-622.

Advertisements for our Classified section will be accepted subject to the following conditions:

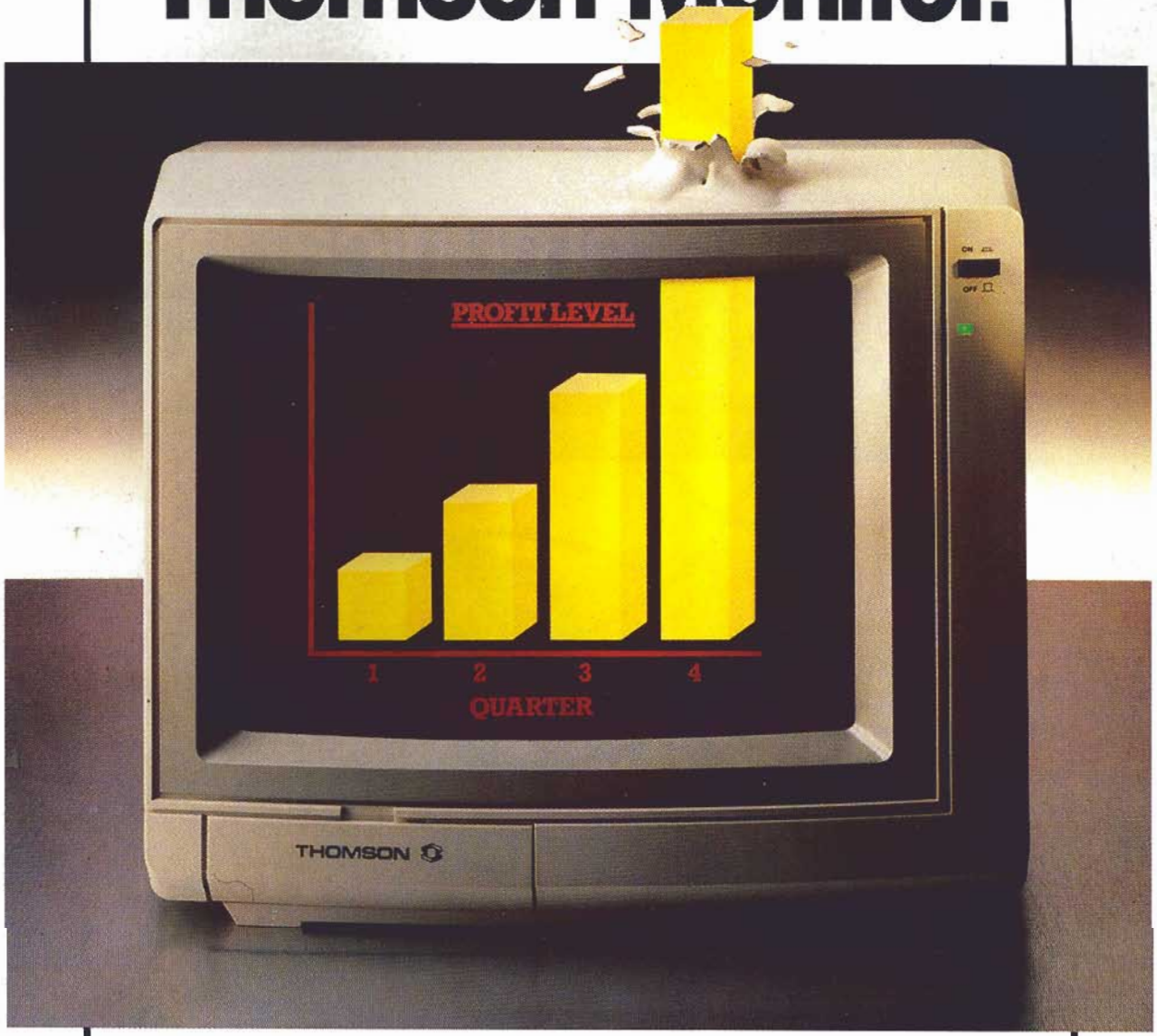
1. The full name and address of the advertiser must be provided.
2. Where a response (i.e. stamped, addressed envelope) is asked for, a street address must be provided.
3. A minimum charge of $10 applies, and advertisements are charged at 60 cents a word.
4. Our deadline for classified advertising is the 10th of the month previous.

Index to Advertisers



This computer is the only one of its type in New Zealand. It went missing in transit with the NZ Railways between Christchurch and Auckland sometime between July 10th and July 21st. Any one who has information relating to the machine's whereabouts should contact Bits & Bytes Ltd.

You get more out of a Thomson Monitor.



Thomson's superior standards show through – clearly. Sharper characters. More brilliant graphics. Eye fatigue reduced. Headaches eliminated. You get much more from your computer hours.

Ten economical green, amber and colour models for all popular PCs.

For a dealer near you contact –

CONTROL

P.O. Box 68-474, Auckland, N.Z. Telex NZ 61102 'DATASYS'
45 Normanby Road, Mt Eden, Auckland 3. Ph (09) 600-687.

CLAUDE 3817

The only communications network ECL doesn't service.

It's not surprising. Two tin cans and a piece of string was never the most efficient communications system. It only worked in a straight line and wasn't particularly private.

Today effective communication is the lifeblood of industry. Communication systems take many forms and perform many tasks from simple telephone, intercom and paging duties to the more complex energy and security monitoring functions.

Electronic Components Limited specialise in the designing and installing of sophisticated communication systems in commercial and industrial buildings.

Whatever business you're in ECL can tailor a system to suit you now and in the future. A well designed system would be able to cope with a network of computers, videos, televisions, telephones, smoke detectors, burglar alarms, temperature controls and be able to receive and transmit signals from satellite receivers, all using the same cable system.

Ideally ECL should be involved in the design stage of your building to enable them to work closely with the architects. This allows them to cater precisely for your needs at minimum cost. And when business expands the network will have the capability of growing with it.

ECL's association with Tisco means they can offer you a complete 'Turn-key' service.

(1) ECL will, in conjunction with you determine exactly what you need.

(2) ECL will design a system to suit.

(3) ECL will install the system.

(4) ECL will instruct your personnel in its usage.

(5) Tisco will then service and maintain the system.

If you're considering updating your present system, or thinking of installing one, your best bet is to start communicating now, with ECL.

Call us on (AKL 669 591) when you can get an outside line.

ECL

ELECTRONIC COMPONENTS LTD.

BGH1905E

"The future . . . . a little sooner than you expected"

Panasonic

the JB-3300

The Panasonic JB3300 is equipped with a newly developed flat Plasma Display, incorporated into the machines compact design. This new display has a full-size 12 inch 80 Character x 25 Line Screen, equivalent to that of a desk top P.C.

The display speed is faster than conventional LCD's with a high resolution of 640 x 400 dots. Characters are displayed in a high quality 8 x 16 dot matrix. The Plasma display results in no flickering or residual images. The new display is easy to read and minimises eye fatigue. Reliable, portable and totally versatile the JB3300 is an important new development for today's progressive business environment.

Panasonic JB3300 — bringing the future's technology to today's top executives.

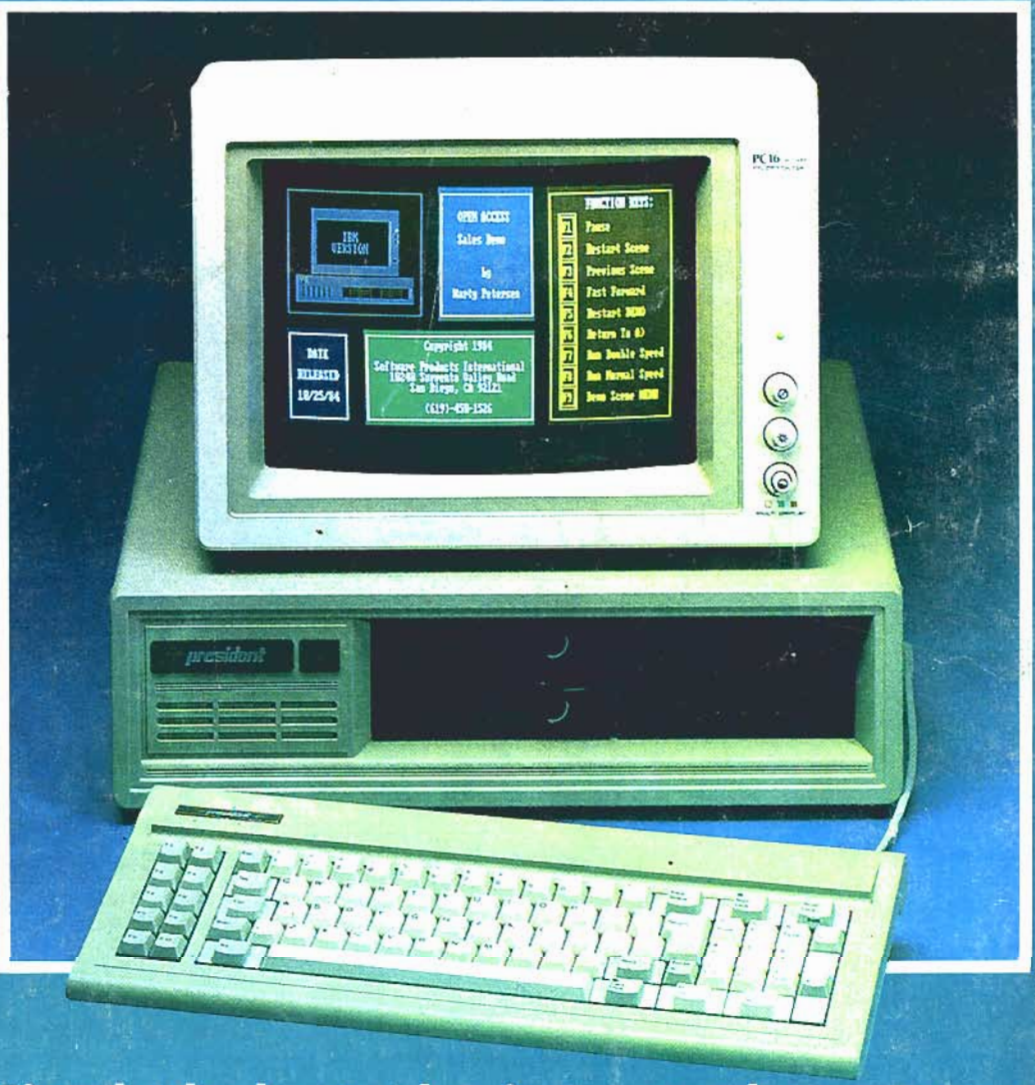
MEC DEALER PRODUCTS making sense of business!

THE MICROCOMPUTER ELECTRONIC COMPANY LTD

P.O. Box 9224, Newmarket, Auckland. Phone 504-774.

PC16-220

Personal Computers



Simply the best value in personal computers.

Only available from:

HITEC MICRO LTD.,

91 Custom Street East, Auckland 1. Ph. (09) 399-183