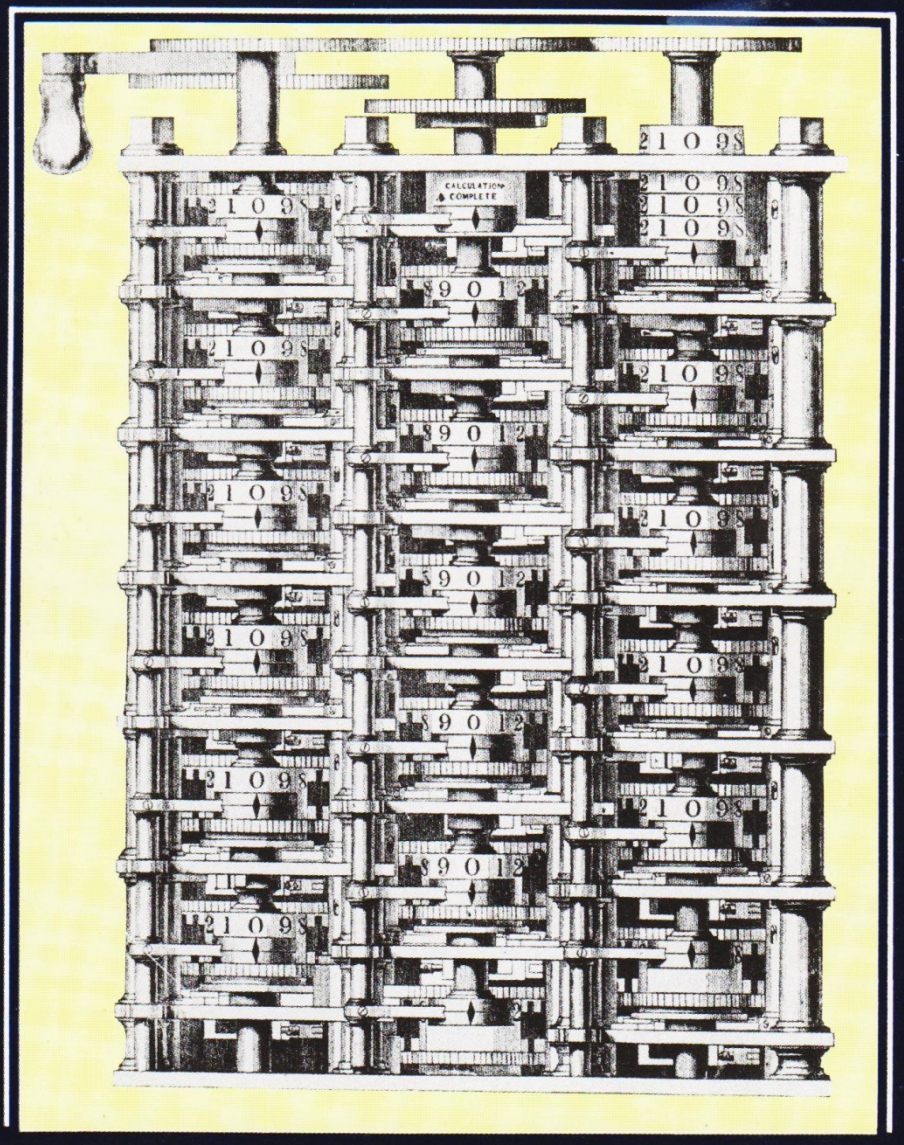
AN ARGUS SPECIALIST PUBLICATION

April 1985 £1.00

COMPUTING Today



THE LESSONS OF HISTORY

Turing and Wilkes, ENIAC and ACE, names from another era.

BUBBLE MEMORIES – the forgotten storage medium

FRAMEWORK: writing a spellcheck

INSIDE CP/M
Z80 random numbers
Hardware and software reviews.
C tutorial

Announcing **MAXAM** for the AMSTRAD CPC464

The start of a complete Expansion System...

SOFTWARE in ROM!

SIDESWAYS ROMS at last!
No more loading...
Leaves 40K free!

The perfect system:
* All-powerful Assembler
* Complete Disassembler
* Full screen editor
* Multi-function Adaptor
* Huge expansion potential in one simple unit!

So easy to use and learn....

```
10 MEMORY HIMEM-10
20 start=HIMEM+1
30 !ASSEMBLE,start
40 'get start
50 'limit &FFFF
60 'ORG start
70 'CP 10:SCF:RET Z
80 'RST 1,&87F2
90 'ORG &BD2B
100'JP start
110'END
```

Meet MAXAM - a new full-feature no-compromise Assembler/Disassembler/Editor - with a difference. It's in a very full 16K EPROM which plugs directly into the AMSTRAD. No waiting while it loads - it's always there! You can still use the Disc unit. You also get, as a bonus, a new expansion socket for Arnor's new range of Sideways ROM cartridges (containing, for example, our forthcoming Word Processor).

MAXAM uses no BASIC RAM space. It lets you mix BASIC and Machine Code - just like the BEEB! Or, you can assemble direct from the Editor, and you can even use the Editor to edit BASIC programs!

MAXAM is ESSENTIAL software for the AMSTRAD enthusiast.

Cassette (reduced specification):£13.50
Disc: £26.90. All prices include p.& p.

MAXAM in ROM £59.90

Arnor LTD

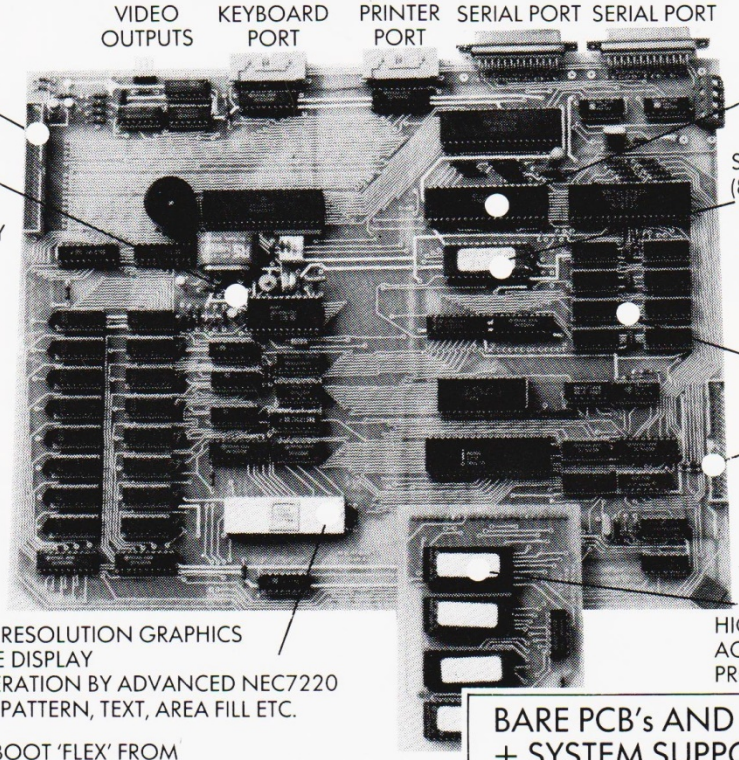
High Quality Software

Technical Data
*Super-fast 3000 lines/min assembly *Conditional Assembly *Plain English error messages *Full Expression evaluation *Unrestricted label names *Directives include:ORG, BYTE, WORD, TEXT, RMEM, LET, IF, GET, PUT, LIMIT, CODE, NOCODE, READ. Commands include: LIST, NOLIST, LISTP, TITLE, PAGE, PLEN, WIDTH, DUMP.
*Menu-driven Screen Editor includes move copy and delete block, tabs, search and replace, print all/part of text, Load/Save all/part of text. Disc/ROM version only: Register display, Memory Edit commands, breakpoint, string search in RAM. Link to AMSDOS.
Technical Enqu. 01 852 2174

Software Houses: We have the perfect low-cost system for software in ROM! Talk to us!

Cheques/P.O.s to: Arnor Ltd, PO Box 619, London SE25 6JL. Order Hotline 01.653.1483 (2pm-6pm)

MICROBOX II THE 6809 SINGLE BOARD COMPUTER THAT YOU BUILD YOURSELF

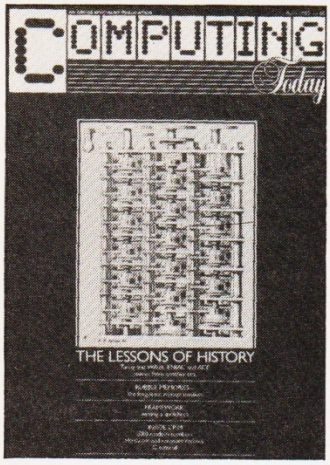


***NOTE:** MICROBOX II WILL BOOT 'FLEX' FROM ANY STANDARD SYSTEM DISC.

BARE PCB's AND DOCUMENTATION
+ SYSTEM SUPPORT MONITOR
+ SYSTEM UTILITIES DISC – **£95.00**
FLEX + EDITOR + ASSEMBLER **£75.00**

Micro Concepts

8 SKILLICORNE MEWS · QUEENS ROAD · CHELTENHAM · GLOUCESTERSHIRE GL50 2NJ
Telephone: Cheltenham (0242) 510525



COVER

This month's cover illustration shows an original engraving of Charles Babbage's Difference Engine. Although the idea for a machine capable of analysing polynomials was conceived by Babbage in 1812, it was to be thirty years before a successful version of the engine was constructed by a Swedish engineer, Per Georg Scheutz, and his son Edvard. Versions of the engine were used up until the early years of this century.

Editor: Don Thomasson
Assistant Editor: Jamie Clary
Technical Illustrator: Jerry Fowler
Additional Illustration: Grant Robertson
Advertisement Manager: Anthony Shelton
Advertisement Copy Control: Sue Couchman, Lynn Collis
Publishing Director: Peter Welham
Chairman: Jim Connell
Origination and Design: Design International
Cover Design: MM Design

Member of the Audit Bureau of Circulation
ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceeding the cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, Lodnon EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

Subscription notes; UK (£16.20) including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, Infonet Ltd., Times House, 179 The Marlowes, Hemel Hempstead, Herts. HP1 1BB England (phone 0442 48432).

CONTENTS

VOL 7 NO 2 APRIL 1985

REGULARS

SERIES

GENERAL FEATURES

SPECIAL FEATURES

EDITORIAL & ADVERTISEMENT OFFICE: 1 Golden Square, London W1R 3AB. Telephone: 01-437 0626. Telex: 8811896.

PRINTERS

DOT MATRIX

All printers have centronic parallel interface unless otherwise stated. All printers have hi-res dot addressable graphic mode. Please send SAE for full details.

EPSON

FX80 160CPS 10" wide friction & pin feed	£347 + VAT	**£399**
FX100 160 CPS 15" wide friction & tractor feed	£499 + VAT	**£574**
RX80 F/T 100 CPS 10" wide friction & tractor feed	£239 + VAT	**£275**
RX80 100 CPS 10" wide tractor feed	£199 + VAT	**£229**
RX100 F/T 100 CPS friction & tractor feed	£385 + VAT	**£443**
8143 RS 23 Interface for FX and RX printers	£39 + VAT	**£45**
8148 RS 232 Interface with 2K buffer x on x off	£60 + VAT	**£69**
Ribbon Cartridge for RX80 FX80 & MX80	£5 + VAT	**£6**
Ribbon Cartridge for FX100 & MX100	£7 + VAT	**£8**

MP165

165CPS 10" carriage friction and tractor feed — £260 + VAT **£299**

SEIKOSHA

BP 420 designed for the business world, 420CPS in draft mode, 110CPS in NLQ mode. — £1095 + VAT **£1259**

SMITH CORONA

Fastext 80: 80 col, 80CPS. Friction feed standard — £149 + VAT **£171**

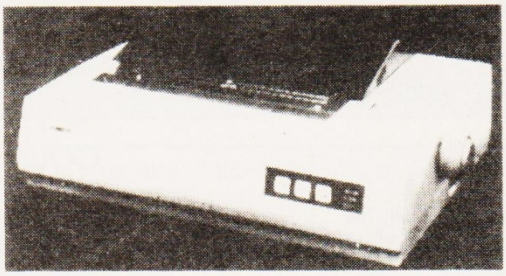
TAXAN KAGA

160CPS 10" wide 27CPS NLQ 24 × 16 matrix	£269 + VAT	**£310**
160CPS 15" wide 27CPS NLQ 24 × 16 matrix	£390 + VAT	**£449**

CANON

PW1080A
160CPS NLQ mode, 27CPS, 10" wide friction & tractor feed
£299 + VAT **£344**

PW1156A
160CPS NLQ mode, 15" wide friction & tractor feed
£399 + VAT £459



COLOUR PRINTERS

Seikosha GP700A 7 colour 50CPS printer	£299 + VAT	**£344**
Canon PJ1080A 7 colour 40CPS ink jet printer	£433 + VAT	**£499**

DAISYWHEEL

JUKI 6100/I PRINT

20 CPS Bi-Directional Logic seeking 10 12 15 CP1 + PS spacing 2K buffer best selling Daisywheel	£324 + VAT	**£373**
Singer sheet feeder unit	£182 + VAT	**£209**
Tractor Unit	£95 + VAT	**£109**
RS 232 Interface	£52 + VAT	**£59**
Spare Daisywheel	£14 + VAT	**£16**

BROTHER HR-15

13 CPS Bi-directional 10, 12, 15 CP1 + PS	**£340** + VAT	**£399**
Keyboard Unit	£139 + VAT	**£159**
Single Sheet Feeder Unit	£217 + VAT	**£249**
Tractor Unit	£95 + VAT	**£109**

QUENDATA

20 CPS Unidirectional 10 12 15 CP1 — **£239** + VAT **£275**

All our printers have 1 year warranty

MONITORS

PHILIPS

7001 High Res Green Screen with sond input £65 + VAT **£75**

GM1211

GM1211 18 MHZ High Res Monochrome Monitor with tilt and swivel stand available in green or amber etched antiglare screen (please specify colour £86 + VAT **£99**

SANYO

DM8112 12" Green screen 18MHZ Hi-Res £86 + VAT **£99**

DM9112 Hi Res Green Screen with tilt stand £109 + VAT **£126**

MICROVITEC CUB

1431 MS 14" RGB Normal Res Colour £173 + VAT **£199**

1451 MS 14" RGB Medium Res Colour £251 + VAT **£289**

1441 MS 14" RGB High Res Colour £417 + VAT **£479**

SANYO COLOUR

Std Res 14" £179 + VAT **£206**
Med Res 14" £299 + VAT **£344**
Hi Res 14" £449 + VAT **£517**

MICROVITEC FOR QL

1451 14" Medium Res Colour. Specially designed for Sinclair QL

£239 + VAT **£275**

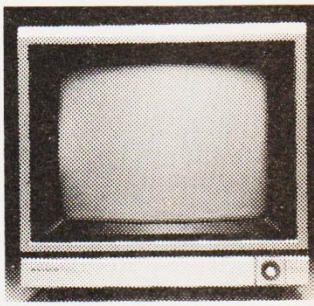
ACORN

SANYO MONITOR

SPECIAL

CM 3125

colour monitor



£179 inc VAT

including free monitor stand

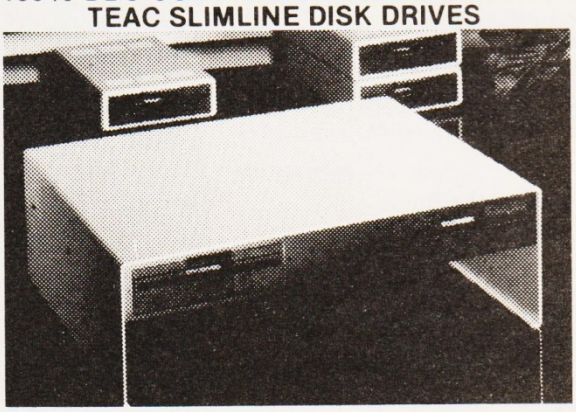
AKHTER

COMPUTER GROUP

28/29 BURNT MILL HARLOW, ESSEX CM20 2HU U.K.
Tel. HARLOW (0279) 443521 Telex: 818894 AKHTER G

DISC

100% BBC COMPATIBLE MITSUBISHI AND TEAC SLIMLINE DISK DRIVES



These drives are supplied ready cased with all the necessary cables formatting program and User Guide
There are some very useful utilities included on formating disc e.g.

- DISASSEMBLER: This is 6502 machine code disassembler
- DUP: To copy and rename a file on disc
- FORMAT: Formating progam for 40 & 80 tracks
- FREE: This utility provides a disk usage analysis
- MDUMP: Enables you to display and modify any part of BBC memory
- MERGE: Merge a number of text files into one file
- RELOCATE: Downloads a basic program to &E00
- SDUMP: Screen dump for EPSON In all graphic modes
- VERIFY: Verifies every sector on a disk
- MENU: A flexible menu program

PRODUCTS

BBC Microcomputer Model B	£299 + VAT	£344
BBC Mod B - disk interface	£369 + VAT	£425
BBC Mod B - Econet interface	£349 + VAT	£401
BBC Mod B - disk and Econet interfaces	£409 + VAT	£471
BBC Compatible 100K disk drive	£112 + VAT	£129
BBC Compatible dual 800K disk drive	£312 + VAT	£359
Acorn Z80	£347 + VAT	£399
Acorn 6502 Second Processor	£173 + VAT	£199
Acorn Bit stick	£327 + VAT	£375
Acorn IEE Interface	£282 + VAT	£325
Acorn Electron plus 1 interface	£52 + VAT	£60
BBC Prestel Adaptor	£115 + VAT	£132
BBC Telext receiver (Aug)	£196 + VAT	£225
BBC cassette recorder and lead	£30 + VAT	£35
Disk interface kit (free fitting)	£103 + VAT	£118
Mod A to Mod B upgrade kit	£70 + VAT	£80
Fitting charge for A to B upgrade kit	£20 + VAT	£23
16K memory upgrade kit	£30 + VAT	£34
Games paddles	£17 + VAT	£19
User Guide	£15	
Advanced User Guide	£12.95	
Econet Guide	£ 7.50	
Econet interface (free fitting)	£60 + VAT	£69
Speech interface (free fitting)	£47 + VAT	£54
BBC disk manual - formating disk	£30 + VAT	£34
Parallel printer cable	£10 + VAT	£11
BBC word processor (view)	£52 + VAT	£59

YOUR CONTACT AT AKHTER
Tel: 0279 443521 (12 lines)

		EXT
DEALER/BULK ENQUIRIES	HAMAYUN MUGHAL	202
TELEPHONE ORDERS	CARON ANDREWS	210
DEALER ORDERS	JULIA ALLUM	209
EXPORT ENQUIRIES	MOHAMAD EDIB	201
TECHNICAL SUPPORT	ALAN LAFFOLEY	207
ACCOUNTS	JULIE AMBLER	211
LITERATURE REQUEST	JOHN MAULE	201

ORDERING INFORMATION

We accept official orders from UK Government and Education establishments. Carriage is £2.50 + VAT (UK only) for normal delivery. If express delivery is required please add £8.00 + VAT per parcel. We accept telephone orders on Barclay and Access card please ring (0279) 443521 (10 lines). all cheques made payable to "AKHTER INSTRUMENTS".

N.B. All prices are subject to change without notice and are rounded up to the nearest pound

OPENING HOURS: MON-FRI 9am-6.30pm. SAT 10am-5pm.
We welcome callers. no parking problems.

DRIVES

100	Single drive 100K 40 trks single sided	£86 + VAT	£99
202	Dual drive 200K 40 trks single sided	£164 + VAT	£189
200	Single drive 200K 40 trks double sided	£121 + VAT	£139
402	Dual drive 400K 40 trks double sided	£239 + VAT	£275
401	Single drive 400K 80 trks double sided	£152 + VAT	£175
400	Single drive 400K 40 80 trks switchable DS	£155 + VAT	£179
02P	Dual Drive 400K + PSU + built-in monitor stand	£260 + VAT	£299
801	Dual drive 800K 80 trks double sided	£303 + VAT	£349
800	Dual drive 800K 40 80 trks switchable DS	£312 + VAT	£359
800P	Dual Drive 800K 40 80 trks + PSU + built in monitor stand	£373 + VAT	£429

All above drives are low power slimline (0 3 A typ at + 12v and 0 4 at + 5v per drive) Normally extra power supply is not required. The BBC Computer power supply is designed to drive to low power drive (IT IS NOT DESIGNED TO DRIVE INTERNAL ROM BOARD)

SS DD disketts (10 Box) £18 + VAT **£20**

DS DD disketts (10 Box) £23 + VAT **£26**

BUSINESS SYSTEMS

COMPLETE BUSINESS PACKAGE

This system is based on 16 Bit 8088 Processor 128K RAM, 2X730K Floppy Disc Drives. High Res Monitor, fast (160cps) Dot Matrix Printer, Wordstar Wordprocessor, Calcstar Spreadsheet Program, complete integrated Accounts package consisting of Sales Ledger, Purchase Ledger, Nominal Ledger, Invoicing, Stock Control, Payroll and mailing list.
Complete turnkey system at an unbelievable price.
Delivered Only £1495 + VAT **£1719**
Delivered and Installed plus ½ day training £1595 + VAT **£1834**

SANYO PROFESSIONAL COMPUTER

SANYO 550
16 Bit Micro, 256K Ram, Built in full colour graphics, MSDOS Operating System, Single 16K disc and free Green Monitor, free software including Wordstar & Calcstar.
£749 + VAT = **£862**

SANYO 550-IX
As above but with dual drives 2 x 160K
£799 + VAT = **£919**

SANYO 550-360X
As 550 but with dual drives 2 x 360K
£899 + VAT = **£1034**

SANYO 550-800X
As 550 but with dual drives 2 x 800K
£999 + VAT = **£1149**

SANYO 550-10MX
As 550 but with MSDOS 2.11 Operating System and a single 10 Megabyte hard disc and 360K floppy back up.
£1995 + VAT = **£2295**

SANYO 555
16 Bit Micro, 256K Ram, double 160K disc drives, MSDOS Operating System, free Green Monitor, free software Calcstar, Infastar, Datastar etc.
£999 + VAT = **£1149**

SANYO 555-360X
As 555 but with 2 x 360zk Drives
£1099 + VAT = **£1264**

SANYO 555 800X
As 555 but with 2 x 800K Drives
£1199 + VAT = **£1379**

SANYO 555 10MX
As 555 but with MSDOS 2.11 Operating System and a single 10 Megabyte hard disc and 360K floppy back up.
£2195 + VAT = **£2525**

APRICOT PC
"Portable Executive Computer" 16 Bit Micro. 256K RAM up to 1.44 megabytes flopy disk storage. 3½" Sony disks. Portable brief case styling. Modem with auto dialler (optional) hard disk optional. Vast software library (compatible with Sirius
Apricot with Double Drive, Monitor and *Free Printer* £1790 + VAT **£2059**
Apricot with Double Drive, Double & Monitor & Printer £1990 + VAT = **£2289**

APRICOT XI
As above but with 10MB Winchester Drive and Single 315K Drive plus Superwriter, Supercalc and *FREE JUKI 6100 Printer* £2995 + VAT **£3444**

WORD PROCESSING

COMPLETE SYSTEMS FROM £650 + VAT

BBC 1: BBC Micro Model B, View (or Wordwise) Wordprocessor, Quendata 20 CPS Daisywheel Printer, High Res Green Monitor, Cassette Recorder plus 10 cassettes and all the necessary cables £600 + VAT = **£747.50**

BBC 2: BBC Micro Model B + Disk Interface, View (or Wordwise) Wordprocessor, 100K Disk Drive, High Res Green Monitor, Quendata 20 CPS Daisywheel Printer, 1 Box of Disks and all the necessary cables
£749 + VAT = **£862.00**

BBC 3: Same as System BBC2 but with 400K Drive
£825 + VAT = **£949.00**

BBC 4: Same as System BBC 2 but with 400K Drive and JUKI 6100 Daisywheel Printer
£925 + VAT = **£1064.00**

BBC 5: BBC Model B + Disk Interface, View (or Wordwise) Wordprocessor, 800K Dual Disk Drive (Mitsubishi), High Res Green Monitor, JUKI 6100 Daisywheel Printer, 1 Box (10) of 80 Track DS discs and all necessary cables £1095 + VAT = **£1260.00**

SAN 1: Sanyo MBC 550 Series 16 Bit Microcomputer, 256K Ram, Dual 160K drives (2 x 160K), High Res Graphics (600 x 200 pixels in 8 colours), JUKI 6100 Daisywheel Printer, High Res Green Monitor, 1 Box of 10 discs, Wordstar Wordprocessor, Calcstar spreadsheet and all the necessary cables £1095 + VAT **£1260.00**

SAN 2: Same as SAN 1 but with Dual 360K Drives (2 x 360K)
£1245 + VAT **£1432.00**

SAN 3: Same as SAN 1 but with Dual 720K Drives £1295 + VAT **£1490.00**

SAN 4: Sanyo MBC 555 Series 16 Bit Microcomputer, 256K Ram, Dual 160K Drives (2 x 160K), High Res Graphics (600 x 200 pixels in 8 colours) JUKI 6100 Daisywheel Printer, High Res Green Monitor, 1 Box of 10 discs, Wordstar, Wordprocessor, Calcstar spreadsheet, Mailmerge, Spellstar (dictionary), Datastar (database), Reportstar plus all the necessary cables
£1195 + VAT **£1375.00**

SAN 5: Same as SAN 4 but with Dual 360K Drives
£1295 + VAT **£1490.00**

SAN 6: Same as SAN 4 but with Dual 730K Drives
£1395 + VAT = **£1605.00**

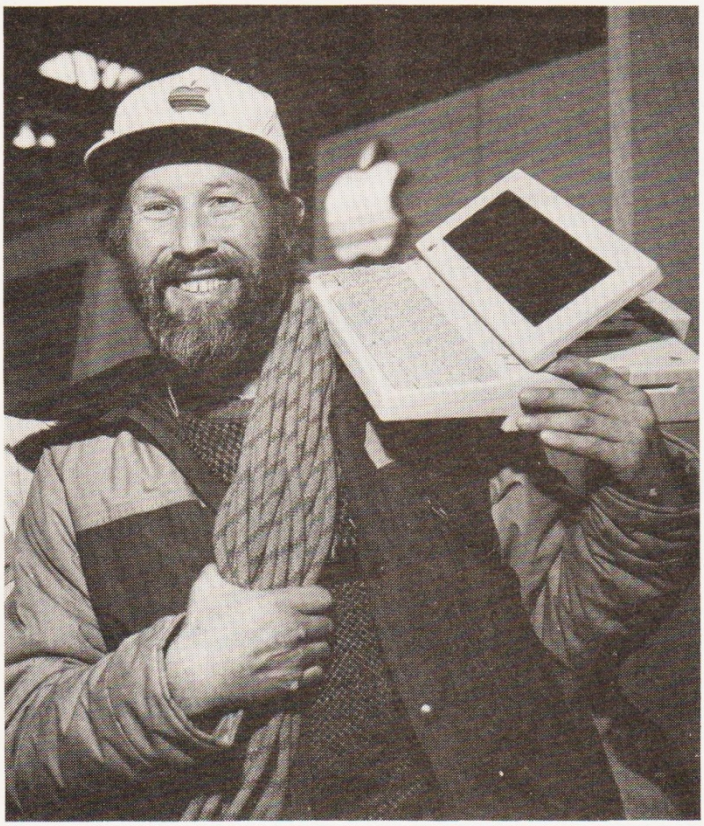
If you require High Res Colour Monitor instead of High Res Green Monitor in Sanyo Systems please add £320 + VAT = £368 to the above prices.

***128K RAM Upgrade for all above Sanyo systems (makes a total of 256K RAM) £150 + VAT = £172.50 including fitting.**

NEWS

BONINGTON PACKS AN APPLE IIc

Chris Bonington, the mountaineer and writer, is using an Apple IIc Portable Personal Computer to help him reach the summit of Everest for the first time. Bonington will be using the Apple IIc with a specially designed software package, Everest Logistics, written by Ian Holt, and the new IIc flat panel display and portable power pack.



Bonington, who has never reached the summit of Everest himself, is participating in the first Norwegian expedition to tackle the mountain. The expedition's leader is Norwegian businessman Arne Neff, who has led a number of previous expeditions to the Himalayas.

The team, which apart from Bonington includes eight Norwegian climbers and twenty eight Sherpas, will leave Oslo on February 23rd and establish Base Camp at the foot of Everest by March 14th. Bonington, using the Everest Logistics program, has calculated that the mimum time it will take to climb the mountain will be 28 days, depending on weather.

Bonington, who has been using two Apple IIe's for the past eight months to help plan the expedition is expecting the Apple IIc to save the team a great deal of time in logistical planning, stock control, accounting and answering the many letters they will receive from well-wishers.

"The IIc will be the key to solving the complicated logistical problem of moving supplies such as food and oxygen up the mountain, and will be an invaluable member of the expedition." says Bonington.

David Hancock, Apple UK's Managing Director, said "We are delighted to be a part of Chris's expedition to Everest and we wish him the very best of luck in his attempt to reach the summit for the first time."

ACORN'S FUTURE UNCERTAIN AS SHARES SLUMP

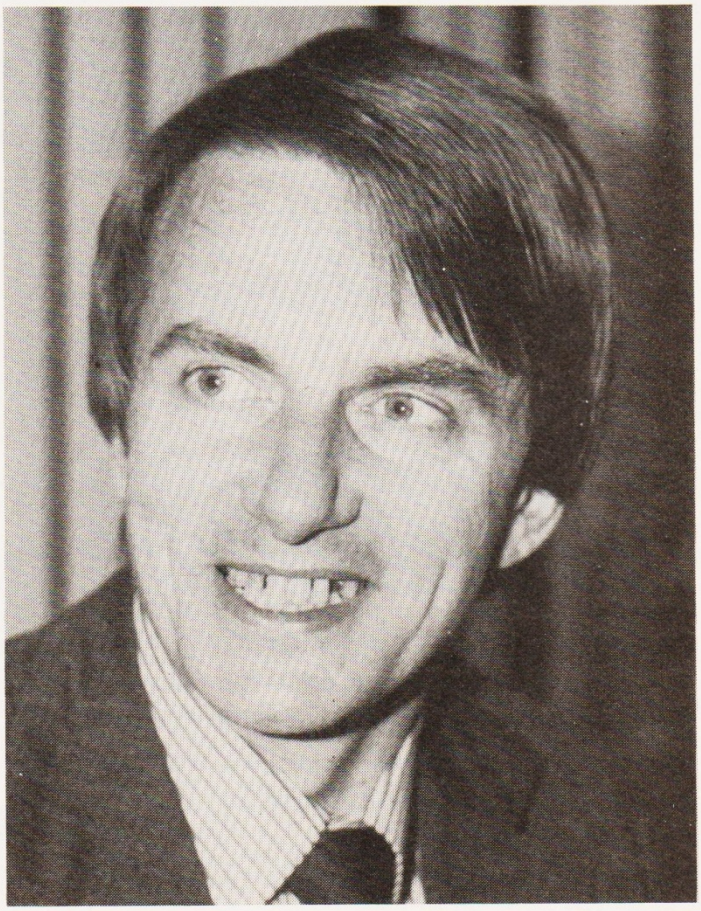
A large question-mark hangs over the future of Acorn, following the suspension of the company's shares at 28p on February 6th.

Falling profits and press speculation were blamed for the calamatous drop in the share price, which couldn't have come at a worse time for Acorn who are currently shipping the first quantities of their recently launched ABC range of business microcomputers, and who urgently need capital to promote and market the new machines.

The announcement of the share suspension follows a disclosure that losses for Acorn's U.S. operation are currently running at £6 million, and despite a £70 drop in the price of the Electron, rumours of a 'disasterous' Christmas sales period have been circulated — closely followed by hot denials from managing director and co-founder, Chris Curry.

Despite pessimistic forecasts by City analysts, Acorn's future may not be as black as industry pundits like to paint it. The top-of-the-range ABC micros are reckoned to be pretty good by those who have seen them, and although it may take a little share redemption to raise the funds necessary to see them through this period of uncertainty, Chris Curry, and Herman Hauser (co-founder and Chairman) who between them have an 86% stake in the company, have every reason to remain quietly optimistic.

A recent showing of Acorn products at a 10-day Computer Exhibition in Moscow prompted the city's Education Institute to place an oder for a £20,000, twenty-station network. More orders are expected to follow, but competition from Sinclair is expected, following the announcement that Jan Tyszka has been appointed Business Development Manager for Sinclair in Eastern Europe. However, competition can only come from sales of the Spectrum+, as exports of the QL to Soviet-block countries are under strict embargo, since the QL's processor — the 68000 — is Manufactured in the U.S. by Motorola, and no permission has been given for the chip to enter the USSR.



HUSKY AT IT AGAIN

A handheld computer package which allows local authorities' trade-waste income to be optimised has been developed by Northamptonshire County Council using Husky Hunter Computers fitted with bar-code readers as data-capture and storage devices.

The system, known as TWIS (Trade Waste Income Survey), is being marketed to other authorities by Husky Computers Ltd. under licence from the Council. It offers considerable savings in manpower and cost over traditional techniques.

The new computer package makes it possible for the Council to monitor the usage patterns of waste sites in terms of types of users, vehicles and waste categories, and this information is then used as the basis for calculations on how much to charge trade-waste users.

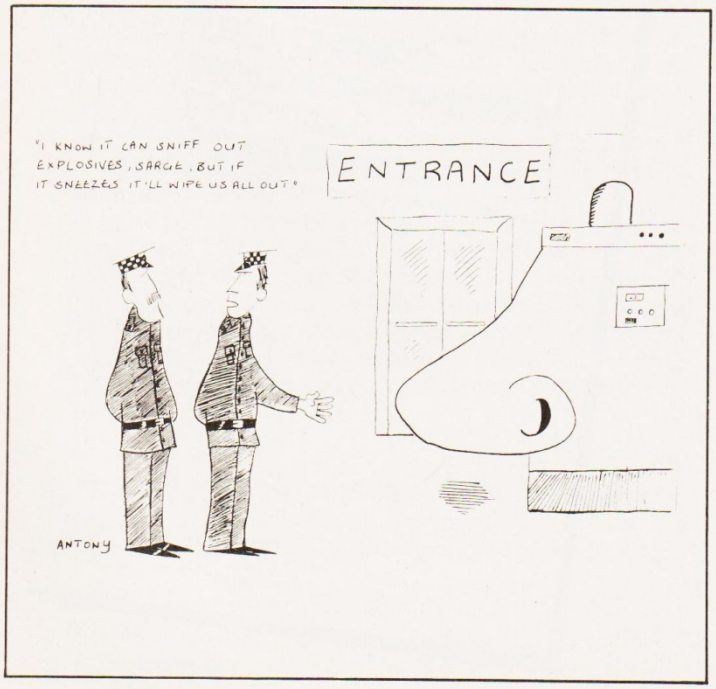
Traditionally, on-site data collection on waste-site usage has been undertaken by operators filling in standard forms which then had to be converted into a form suitable for entry into the Council's mainframe computer. This was then followed by on-site validation and comparison against previous results, with the result that the whole process could take several weeks.

With the Husky system, information is entered directly into the handheld units, each of which can store up to four weeks' survey date in its 208k Byte memory. Accuracy of data entry is ensured by the operator using a bar-code wand which he passes over a list of standard codes corresponding to the type of vehicle and the other categories.

Comparison of current and previous surveys can automatically be carried out within the machine, so that there is no need for a separate validation procedure. In addition, reports and costings can be prepared directly by linking the Husky Hunter to an on-site printer.

The Husky Hunter handheld computer is ideal for on-site use of this type because its rugged construction and battery operation allow it to be used out of doors for long periods. The software package used in the Hunters has been specifically designed to combine ease of use with security of stored data, with the operator being led through the procedures with a series of prompts displayed on the Hunter's large display screen.

Commenting on the new system, Tony Bispham, Group Engineer (Waste Management) with Northamptonshire County Council, says 'TWIS is a real breakthrough as far as management analysis is concerned. I am pleased at the speed with which accurate results are produced directly the survey has finished, and the staff time spent previously analysing results can now be spent on other useful work'.



COMPANY POISED FOR ELECTRIC CAR BATTLE

A second Cambridge computer firm is set to enter the electric vehicle market, following the launch of the Sinclair C5 in January of this year.

Details of the car made available to the press show that, although the new car is of unorthodox design — bearing little resemblance to the sleek, aerodynamically countered C5 — it is ". . . ideal for town use and travelling to and from the shops", as the company spokesman who supplied details to us, put it.

"Naturally", commented the spokesman, "the design is intended to reflect our corporate image. However, this is not to say that the vehicle is a mere advertising gimmick — we are very serious about the future of the electric car market, and we are anxious for a piece of the market as it begins to mature."

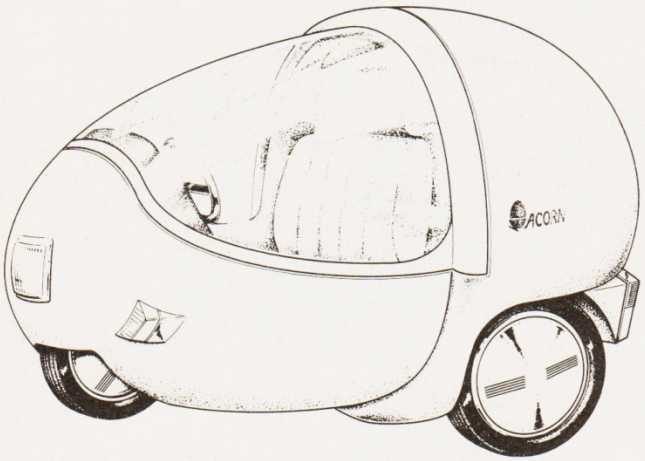
Photographs of the new car are expected to be available shortly, but an artist's impression reveals that it has a narrow but long wheel-base, with a small but 'comfortable' driver cockpit and spring-loaded, swing-back canopy.

Taking advantage of a loophole in the law that states: '. . . vehicles of . . . (a restricted

(continued overleaf . . .)

(continued from previous page)

wheelbase) . . . may be propelled unaided by those of school age providing: 1) the vehicle is supported by a set of stabilising wheels fitted to the left and to the right of the vehicle by a person or persons qualified to do so; 2) the maximum speed of the maximum speed of the vehicle does not exceed 15mph; 3) the driver has the consent of Parent or Gardian . . .' absolutely anyone over the age of 5 is permitted to use the vehicle . . . so long as Mum agrees.



COMPUTING FOR LONDON

Covent Garden hosts the spectacular launch of the 1985 London Festival of Computing.

Running from Tuesday 9th April until Saturday 20th April, the Festival kicks off with a four-day extravaganza of competitions, games and exhibitions — which will be officially opened on Tuesday 9th.

The 1985 London Festival of Computing, a charity supported by the GLC, is a showcase of interests, achievements and ambitions of London's rapidly expanding amateur and professional computer community. Its aim is to promote, in a practical way, the use of information technology products to those people who can most benefit — eg: schools & colleges; voluntary organisations; youth groups; businesses medical establishments; disabled people; hobbyists; central & local government departments; etc.

The Festival is organised by a central consortium consisiting of representatives of a number of bodies involved in the computer industry. These bodies are: GLC — Central Computer Services Division; Department of Industry — IT section; Association of London Computer Clubs; Microelectronics and Information Technology in Islington; Inner London Educational Computing Centre; NCC — Microsystems Centre; Inter-Action — Community Arts and Resources Charity; London New Technology Network; Interactive Storybooks.

COMPUTERS FOR ALL

A study on the ways computers are being used to help ordinary people in the community is published today.

'Computers, Communication and The Community" by Christopher Pilley and Margaret Sutherland, describes how Scotland leading the field in community computing and shows, through a study of projects from all over Scotland, how computers are being used to help young people, disabled people and all kinds of community groups.

Amongst the projects discussed are two very different projects in Edinburgh — Edinburgh Home Computing Club and Microwing. Edinburgh Computing Club started off much like any hobby club but its members are now involved in taking computers out to people who would not normally have a chance to use them. Microwing is the computer arm of the Edinburgh Walk-In Numeracy Centre. This unique centre is a shop where adults can go for help with numeracy. Offering help with computers is a logical next step.

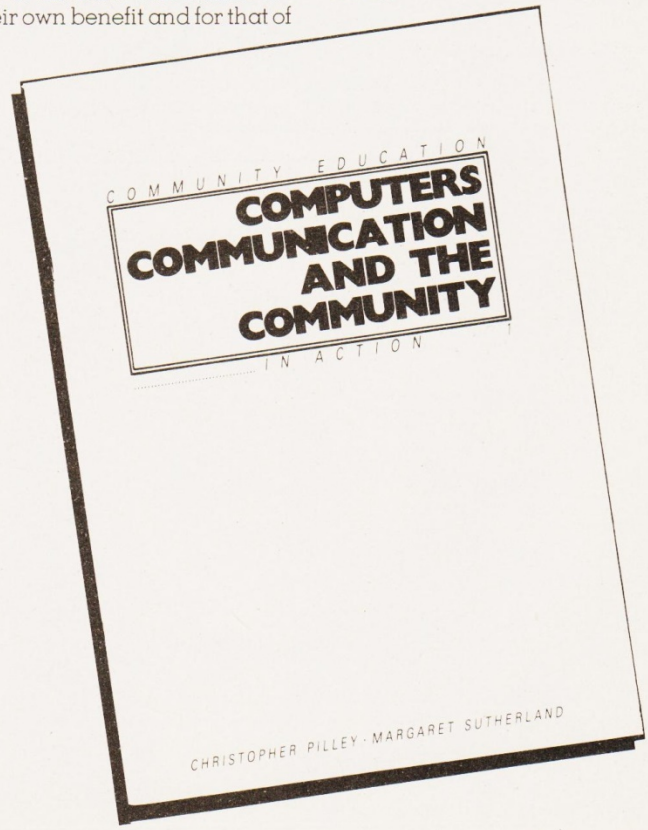
New technology is changing the lives of many people, but none more so than disabled people. A small project running in Inverurie is showing how disabled people can be integrated into society and possibly into work. CID (Computerised Information for the Disabled) is an information service run by disabled people for disabled people and for people who work with disabled people.

The introduction of computers into rural Scotland is happening slowly and unevenly. Rural schools are enthusiastically taking advantage of government schemes which encourage them to buy and use computers. Tiree High School, for example, has seven microcomputers for just over 100 pupils (including primary school pupils). These computers are not just being used by school pupils. Teacher John Christie has arranged demonstrations for local National Farmers' Union, Tiree Slimmers' Club and adult classes in computing.

The study points out that "Scotland was in the forefront of Britain's first industrial revolution, built on coal, steel and shipbuilding. Now, once again, Scotland is playing a significant role in Britain's new revolution based on New Information Technology", and that is reflected in the high number of Scots who own their own computers.

The authors conclude that "The case studies show that we are only just beginning to learn how to use the new technologies. Much of what we have learnt so far has been influenced by the way in which earlier technologies were used. But choices do exist." In showing that the choices exist, the authors aim to encourage people to exercise that choice for their own benefit and for that of the communities they live in.

"Computers, Communication and The Community", by Christopher Pilley and Margaret Sutherland, is published by the Scottish Community Education Council, price £1.95, ISBN 0 947819 04 X, and is available from SCEC, Atholl House, 2 Canning Street, Edinburgh EH3 8EG. Tel: 031-229 2433.



A CREDIT-CARD PROBLEM

We tend to take credit cards for granted, but a slight problem concerning them has come to light that is worth thinking about.

The first thing that happened was that putting a magnetic lock key-card in the same pocket as a credit card was found to erase any magnetic coding on the credit card. Moral; Keep thinks like that apart.

A more alarming phenomenon showed up later, when it appeared that one credit had corrupted magnetic data on another. Now, some people carry a small wad of credit cards in a folder, which is likely to encourage cross-erasure.

Perhaps this was an isolated instance, but it does suggest that those with a number of magnetically coded cards may need to keep them in different pockets . . .

ERRATA

AMSTRAD LETTER WRITER (*Computing Today* March 1985) unfortunately suffered though the over-enthusiastic use of our page-layout person's knife. 18 lines were cut from the top of the program, and this 'unscheduled adjustment' in the program crept through unnoticed. To correct the situation, here are the missing lines:

```
3190 GOTO 3000
3199 REM +- Handle control codes -+
3200 IF I$=CHR$(241) THEN GOSUB 3400:GOTO 3000:REM +- Down line -+
3220 IF I$=CHR$(240) THEN GOSUB 3500:GOTO 3000:REM +- Up line -+
3240 IF I$=CHR$(243) THEN GOTO 3120:REM +- Right one -+
3260 IF I$=CHR$(13) THEN GOSUB 3400:CX=LM+1:GOTO 3000
3280 IF I$<>CHR$(127) THEN 3320
3300 IF CX>LM+1 THEN CX=CX-1:T$(CX,CY+BL)=" ":GOTO 3000
3310 GOSUB 3500:IF FG=1 THEN CX=COLUMNS:T$(CX,CY+BL)=" ":GOTO 3000
3315 CX=LM+1:GOTO 3000
3320 IF I$<>CHR$(242) THEN 3360
3330 IF CX>LM+1 THEN CX=CX-1:GOTO 3000
3340 GOSUB 3500:IF FG=1 THEN CX=COLUMNS:GOTO 3000
3345 CX=LM+1:GOTO 3000
3360 IF I$=CHR$(22) THEN GOSUB 4000
3370 IF I$=CHR$(19) THEN GOSUB 4100:GOTO 3000
3380 IF I$=CHR$(12) THEN GOSUB 4200:GOTO 3000
3381 IF I$=CHR$(16) THEN GOSUB 4400:GOTO 3000
3382 IF I$=CHR$(17) THEN 9800
```

INSIDE BASIC: 2 which appeared in the same issue, enjoyed a similar fate. Several lines were omitted from the paragraph preceeding the side-head MATHEMATIC ABILITY. Redressing the balance, this is how the article should have read:

> The number of terms used in polynomials varies a good deal. In general, the number of terms is directly related to the accuracy of the result, and also to the speed of execution. This one area where a very fast machine may show a balancing disadvantage through loss of extreme precision, but the errors involved are usually small. Measurement of a real angle is rarely very precise, and the accuracy of trigonometric functions is only important where small differences are concerned. It should be noted that the form of the polynomial can vary, some computers dividing and subtracting instead of multiplying and adding.
>
> Some people have been slightly mystified by the advice to use, say X * X * X rather than X 3 when they want to calculate the cube of X. The reason is that the first form carries out an actual multiplication, whereas the second calculates the log of X, multiplies the result by three, and then takes the antilog. This is a much slower process, and is usually a trifle less accurate. It has been known for a computer to report the result as 26.9999 . . .

AVERAGES AND TRENDS of the March issue (page 32) was missing a Sigma character. The first expression after the side-heading HARMONIC MEAN should have read:

$$1/H = 1/N\Sigma 1/X$$

We would like to apologise to those readers who were affected by these omissions.

PRODUCTS NEWS IN BRIEF

• THE MAXAM assembler/disassembler for the AMSTRAD CPC464 is now available from Arnor Ltd. (PO Box 619, London SE25 6JL. Order hotline 01-653 1483). The assembler is in 16K ROM format, and features a full-screen editor and has the advantage that it uses no BASIC RAM space. It also permits assembly language and BASIC to be mised a la the BBC Micro. Cassette and disc versions are also available, but the ROM version is set to retail at £59.90

• A NEW text-processor, described as "The first, second generation word-processor" has been released by Collossus Software Ltd. (310 Finchley Road, London NW3 tel. 01-435 9321). The package is quite unusual, in that it treats documents not simply as a single chunk of text, but as text with a structure. The package is available now for 64K Z80 systems running CP/M, price £99 + VAT.

• THE DIGITAL LA210 dot-matrix letter printer is now available in this country through RAPID Recall (0494-450111). The printer offers draft and letter-quality printing as well as bit-mapped graphics. An interesting feature of the printer — a feature that Computing Today staff have been struggling to comprehend for some while — is the position of the Font Chimney!! Those who insist on knowing exactly what a font chimney is, are invited to ring Rapid Recall on the number given.

• THE MALTRON KEYBOARD is a device which has been developing for some years, but which has refused to catch on, is to be featured on the BBC Horizon programme on the 4th or 11th of March. The keybaord, brainchild of Brian Hobday, is a suggested alternative to the conventional QWERTY keyboard, which is said to cause damage to the hands of regular users. The keyboard is designed to reduce injury, by arranging the keys in logical clusters reducing the work-load on each hand.

• VINE MICROS have announced a new utility ROM for the BBC micro, enabling fast the simple tape-to-disk transfers to takeplace. The T.D. ROM as it is called, allows program transfer of protected programs to take place, leaving the protection mechamism intact. Readers are asked to note that the system solely provides transfer of programs from tape to disc, and does not break program protection. The T.D. ROM is available directly from Vine Micros, Marshborough, Sandwich Kent, tel. 0304 812276.

CT HOTLINE . . . CT HOTLINE . . .

READERS — if you need information on products featured in our news section, why not give us a ring. Phone 01-437 0626 and ask for the Assistant Editor of Computing Today.

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, news and reviews – all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the ..... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

£16.20 for 12 issues UK	☐
£18.70 for 12 issues Overseas Surface	☐
£51.20 for 12 issues Overseas Air Mail	☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £.....
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard* (*delete as necessary)

ACCESS

BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) .....
delete accordingly

ADDRESS .....

.....

..... POSTCODE .....

Signature .....

Date .....

CT April '85

EPROM SERVICES

QL SPECTRUM and ZX81 HARDWARE

QL SERIAL TO PARALLEL PRINTER ADAPTOR – £38.95

★ Supplied with all cables ★ Integral mains power supply — reduced strains on the QL's PSU and high power printer output ★ Power-on LED indicator ★ True RS232 levels — for reliability ★ Fully enclosed black box ★ Crystal controlled BAUD rate generator — default 9600 baud, adjustable down to 300 baud ★ Adjustable parity and bits per character input — for use with other computers ★ Double buffered input — no lost characters ★ SUPERBASIC and QDOS compatible

Also available for ZXSPECTRUM Interface One and other computers with a SERIAL PRINTER PORT.

QL EPROM CARTRIDGE — £14.95
Fits the EPROM Cartridge socket holds two 2K, 4K or 8K EPROMS. Allows SUPERBASIC extensions to be available at power on.

QL EPROM CARTRIDGE with Software — £24.95
A collection of useful utilities supplied on a plug in EPROM CARTRIDGE.
Procedures include:
CLOCK — on screen clock
TIMER — on screen elapsed time
XDIR — extended directory (with pause)
REPEAT — Set key repeate rate and duration

Other EPROM Software also available e.g. Graphics, Basic extensions, multi tasking, machine code monitor.

QL CABLES

QL SER with 1.5 metres of cable	£3.95
QL CTL with 1.5 metres of cable	£3.95
MICRODRIVE approx. 8"	£7.95
QL SERIAL CABLE	£11.95

ESBUS SYSTEM
For ZX81 and SPECTRUM enables interfacing to a wide range and combination of interface cards, e.g.

Motherboard	£16.95
24 line I/O Card	£14.95
Real Time Clock	£24.95
8 Input 8 Bit ADC	£23.95
4 Input 10 Bit ADC	£28.95
8 Input 12 Bit ADC	£57.95
Dual Centronics I/F	£26.95
Single RS232 I/F	£25.95
Dual RS232 I/F	£34.95
3 Output DAC	£26.95
EPROM Programmer	£54.95

ZX81 EPROMS

Eprom Board C/W Toolkit Eprom	£19.95
Graphics Board C/W 4K of Eprom	£24.95
Autostart Card	£9.95

Wide range of software on EPROM e.g. Toolkits rapid load/save, M/C Monitor disassembler, etc. etc.

SPECTRUM

ROM Board	£14.95

Replaces the 16K ROM with two 8K EPROMS plus 4 user definable switches.

Keyboard Interface	£11.95
Add a second keyboard	£19.95

I/O Board plugs into the Spectrum, providing two 8 bit I/O ports plus Strobe line.

EPROM CARTRIDGE SYSTEM
An EPROM system which replaces Sinclairs microdrive storage system with 48K interchangeable EPROM cartridges.

Interface	£34.95
Cartridge 6 x 8K Eproms	£7.95
Programmer	£19.95

Auto loading of programs on power-on. The Z80 vectors may also be redefined by reprogramming the interface's 2K ROM.
★ No mods to the Spectrum needed.
★ Works on 16K and 48K versions.

EPROM SERVICES

3 Wedgewood Drive, Leeds LS8 1EF Tel: 0532 667183

Prices include p&p. Please send SAE for full details. Export and Trade enquiries welcome.

The software that you can program for education and instruction is...

MicroText
by
TRANSDATA

computer based training systems from

TRANSDATA LIMITED

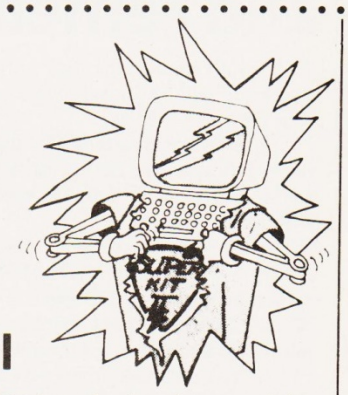
11 South Street, Havant, Hants PO9 1BU Tel: (0705) 486556

developed in collaboration with the National Physical Laboratory

FIND YOUR WAY THROUGH THE DIGITAL ELECTRONICS MAZE

First there was
SUPERKIT

and now ...
NEW
SUPERKIT II



This practical self-instruction kit has been developed to extend the original beginners' **SUPERKIT**. **SUPERKIT II** includes an instruction manual and many more components, enabling you to design and use adders, subtractors, counters (ripple, up/down, synchronous, decade and Gray code), registers, pattern recognisers and 7-segment displays. You need the board and components from **SUPERKIT** to enable you to build the circuits in **SUPERKIT II**. Together the two kits provide an excellent introduction to digital electronics – what really goes on inside a computer.

SUPERKIT (SUP) £22.00 **SUPERKIT II (SUP II) £16.00**
Special price £35.00 for both (SUP + II) (inc. VAT and p & p)

The **SUPERKIT** series is backed by our theory courses, **DIGITAL COMPUTER LOGIC** (beginners' course), which covers the design of logical circuits, and **DIGITAL COMPUTER DESIGN** (more advanced), which covers the design of digital computers. **MICROPROCESSORS AND MICROELECTRONICS** teaches you what a microprocessor is and what it can do.

DIGITAL COMPUTER LOGIC (DCL) £7.00
DIGITAL COMPUTER DESIGN (DCD) £9.50
MICROPROCESSORS AND MICROELECTRONICS (MIC) £6.50

Please send for full information on these and our other courses.

GUARANTEE If you are not completely satisfied, return the item to us within 28 days for a full refund. All prices include worldwide surface postage (ask for prepayment invoice for airmail). Allow 28 days for delivery in UK. Overseas payment by international credit card or by bank draft drawn on a London bank.

CAMBRIDGE LEARNING LTD, Unit 59 Rivermill Site, FREEPOST, St Ives, Cambs PE17 4BR, England. Tel: 0480-67446

VAT No 313026022 Transcash No 2789159 Reg No 1328762

Please send me (initial letters used):

..... SUP	@ £22.00	..... DCL	@ £7.00
..... SUPII	@ £16.00	..... DCD	@ £9.50
..... SUP + II	@ £35.00	..... MIC	@ £6.50

I enclose a cheque/PO payable to Cambridge Learning Ltd for £.....

Please charge my ..... credit card,

No ..... Expiry date .....

Telephone orders from credit card holders accepted on 0480 67446 (24 hrs)

Name .....

Address .....

.....

..... Signature .....

CAMBRIDGE LEARNING
Unit 59, Rivermill Site, FREEPOST, St Ives, Cambs PE17 4BR, England.

NEW!

Now your computer can teach you to read faster – and remember more!

Did you know that the faster you read, the more you absorb?

Now SPEED READ will double your reading speed quickly and easily in your spare time at home.

This is a major breakthrough in home education which will bring you rapid rewards. Businessmen can dramatically cut time spent on paperwork, students and schoolchildren will find study easier and more fun, housewives can double their reading pleasure.

SPEED READ helps your computer to help you to a more rewarding and profitable life. Available on cassette for use with BBC Micro B, ZX Spectrum 48K and Commodore 64 systems, and on disk for use with Sirius (MS dos) and IBM (PC dos) systems.

Available exclusively from HI-YIN Music, Department SR/CT, 43 Church Hill Road, OXFORD OX4 3SG.

Please send me the SPEED READ programme suitable for BBC Micro B/Commodore 64/ZX Spectrum 48K (cassette only)/Sirius (MS dos)/IBM (PC dos) (disk) system quickly!*

Name:

Address:

Postcode:

Signature:

I enclose cheque/P.O.* for £16.95 (inc. VAT & postage & packing) for the cassette. £29.95 (inc. VAT & postage & packing) for the disk.

Post to HI-YIN Music, Dept. SR/CT, 43 Church Hill Rd., OXFORD OX4 3SG.

Please delete as applicable

Vine Micros

T.D.ROM

The Ultimate Tape to Disc System... ...for BBC "B".

For BBC 'B' OS.1.20 with single density interfaces using the standard 8271 DFS chip.
(e.g. Acorn, Amcom S/D, Watford S/D).

ROM based, with the following features:

*Instant access. *Can handle any length of program. *Allows use of all available RAM (&400 to &7FFF). *Can handle any number of programs. *Can save to any disc. *Very user-friendly, (menu driven). *Contains two transfer routines to ensure effective loading of all but a few of currently available software tapes. *Adventures normally taking six minutes to load will now take about five seconds. *The system automatically switches to TAPE filing after loading, so that you can still load/save games positions etc.

PRICE £18.00 (including V.A.T. and post).

N.B. This ROM solely provides convenience in loading, and does not break software protection.

ADDCOMM

For BBC 'B' OS 1.20.

For ELECTRON with ROM board

FORTY NEW BASIC COMMANDS

Unlike other products, ADDCOMM is a ROM of multi-function nature. It gives TOOLKIT, (useful for programmers), GRAPHICS and LOGO GRAPHICS, (not only for the beginner, but also for technical applications), and a good selection of GENERAL PURPOSE statements. All are BASIC commands except *HELP. The ROM comes with an excellent 72-page User Guide and is very good value for money.

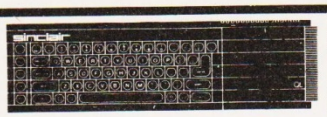
PRICE £28.00 (including V.A.T. and post).

Or send stamp for detailed brochure.

Both items obtainable from:

VINE MICROS, MARSHBOROUGH,
NR. SANDWICH, KENT, CT13 0PG.

(Or your local Computer Dealer.)



ORIC AND SINCLAIR COMPUTERS
Oric 1 computer 48K £85 (£82) £92. Oric Atmos computer 48K £171 (£158) £168. CCP40 Oric colour printer £124 (£115) £132. Sinclair pocket TV £113 (£105) £115. Sinclair QL Computer £406 (£385) £410. Sinclair Spectrum Plus Computer 48K £131 (£131) £153. Kit to upgrade the Spectrum to Spectrum Plus £30 (£30) £40. Microdrive £51 (£50) £60. RS232 interface 1 £51 (£50) £60. Special offer:- Microdrive + Interface 1 + 4 cartridges £102 (£100) £120. Blank microdrive cartridges £5.50 (£6) £7. Standard floppy disc interface for Spectrum £102 (£92) £112. (See Cumana disc section for suitable disc drives). Fuller FDS keyboard for spectrum £52 (£52) £62. Fuller master unit £56 (£56) £62. Interface 2 £20.45 (£20) £24. 32K memory upgrade kit for 16K spectrum (issue 2 and 3 only) £31 (£28) £30. Spectrum Centronics printer interface £51 (£47) £52. ZX printer has been replaced by the Alphacom 32 £71 (£69) £82. 5 printer rolls (state whether Sinclair or Alphacom) £13 (£16) £21. ZX81 computer £45 (£44) £54. 16K ram packs for ZX81 £28 (£25) £30.

COMMODORE COMPUTERS
Commodore C16 Starter Pack £145 (£142) £162. Commodore Plus/4 £305 (£281) £301. Commodore 64 £222 (£215) £235. Convertor to allow most ordinary mono cassette recorders to be used with the Vic 20 and the Commodore 64 £9.78 (£9) £11. Bargain package:- cassette convertor + compatible cassette recorder £37 (£38) £44. Commodore cassette recorder £43 (£44) £50. Printer interfaces for Vic20 and the Commodore 64:- Centronics £45 (£41) £46, RS232 £45 (£41) £46. Disc drive £233 (£209) £234. 1520 printer/plotter £99 (£96) £111. MPS801 Printer £235 (£200) £245.

ENTERPRISE, MSX AND ACORN COMPUTERS
New Enterprise 64 computer £261 (£236) £256. MSX Goldstar computer £203 (£193) £213. Acorn Electron £173 (£179) £199. BBC Model B £404 (£357) £387. BBC Model B with disc interface £480 (£423) £453. Kenda double density disk interface system £149 (£131) £141. See below for suitable disc drives.

CUMANA DISC DRIVES
To suit disc interfaces of Sinclair spectrum, BBC B and Videogenie. Single:- 40 track single sided £176 (£158) £178, 40 tr double sided £218 (£195) £215, 80tr ss £207 (£186) £206. 80tr ds £234 (£209) £229. Dual:- 40tr ss £299 (£280) £320, 40tr ds £395 (£353) £393, 80tr ss £372 (£334) £374, 80tr ds £437 (£390) £430.

PRINTERS
Brother HR5 £162 (£146) £170. Shinwa CTI CPA80 £237 (£228) £258. Cannon PW1080A £382 (£344) £374. Epson RX80 £277 (£251) £282. Epson RX80F/T £314 (£286) £316. Epson FX80 £339 (£298) £328. Combined matrix printers and electric typewriters:- Brother EP22 £173 (£166) £186, Brother EP44 £258 (£235) £260. CCP40 Oric Colour printer/plotter £124 (£115) £132. Interfaces to run the above printers from Vic and the Commodore £45 (£41) £46. We can supply interfaces to run the above printers from Sharp computers £58 (£52) £55.

UK101, SUPERBOARD AND VIDEOGENIE
We still support these Computers. Write for our list.

COMPUTER REPAIRS
We offer a world-wide repair service. Write for a quotation.

SWANLEY ELECTRONICS

Dept CT, 32 Goldsel Road, Swanley, Kent BR8 8EZ, England.
TEL: Swanley (0322) 64851

Official orders welcome. All prices are inclusive. UK prices are shown first and include post and VAT. The second price in brackets is for export customers in Europe and includes insured air mail postage. The third price is for export customers outside Europe (include Australia etc) and includes insured airmail postage.

LIGHT TO THE POINT

SPECTRUM 48K/COMMODORE 64
£17.25
DRAGON/TANDY
£17.25
inclusive
NO INTERFACE REQUIRED

VIC 20
£17.25
REQUIRES SUPER EXPANDER

Discover the exciting world of creating your own graphics on screen.

The Trojan Light Pen will draw boxes, circles, lines, freehand pictures, save and load pictures with full erase facility.

All in Hi-Res screen in any of 4 colours for the Dragon/Tandy, 8 colours for the Spectrum, and 16 colours for the Commodore 64.

For educational or leisure use.

DEALER ENQUIRIES WELCOME

TROJAN

Micro Computer Software & Accessories

Send cheque/P.O. to.
TROJAN PRODUCTS
166, Derlwyn, Dunvant, Swansea SA2 7PF
Tel: (0792) 205491.

TALKING SHOP

Don Thomasson

During a discussion on the future of Computing Today, a question as asked which gave rise to a lot of thought: "Can we get some well-known 'names' to contribute?"

Now, there are plenty of well-known people in the computer world, but they tend to be entrepreneurs, rather than representatives of the technical side of the industry. Their actions and pronouncements may make the headlines, but are rarely concerned with computing, as such.

There was a time when the situation was rather different. There was Turing, who started it all, and Von Neumann, who expanded Turing's ideas, Professor Williams, who initiated the Manchester development work, Professor Wilkes of the EDSAC at Cambridge, Professor Booth at Birkbeck.

These were the pioneers, to be followed by people like Dr Pinkerton, the father of LEO, Colin Haley and Tom Elliot, who turned the ACE into DEUCE, S L H Clarke of Elliot Brothers, Tony Headley and many others.

Such men had — and some still have — considerable stature. We may not have agreed with all their ideas, but they provided the combination of drive and technical know-how which was needed to make the industry work.

Who are their peers today? Who could we call upon to make authoritiative statements about computer matters? Those named above have remained within the mainframe section of the industry to a large extent, and view microcomputing through a telescope, perhaps looking into the wrong end. This is not a criticism, it is only natural that they should concentrate on what they know and understand best, but it leaves the micro world rather lacking in real authorities.

Should any individual reader feel that this is uncomplimentary to his own standing, he should remember that true authority is given by others, not taken. Writers who simplify and popularise usually provide second-hand material. We are looking for original thought.

There are, of course, exceptions, and we hope to see them represented in our pages in due course, but some of them present a paradox, in that they work full-time in the big computer world, and see microcomputing as a hobby — sometimes as a hobby-horse.

So where should we seek interesting and authorititive information on the microcomputer world? It should not be contaminated by mere sales-talk, which can sometimes make black look white, or even sky-blue-pink, but should be factual and logical.

One might expect to find suitable sources in the larger microcomputer companies, but such companies often debar their employees from offering contributions in case confidential information is accidentally revealed. There are instances where this is circumvented by a change of name, but content and style can give the game away.

The real authorities are often retiring individuals who need a strong prod to persuade them to publish their knowledge. Some are voluble enough in conversation, which can be denied, yet reluctant to commit themselves in print.

This is not just a problem facing technical journals. It affects the industry as a whole. Lack of communication leads to re-invention of the wheel — in octagonal form. Analyse the circuitry of a number of computers, and you will find many instances where a technique used in one could be adopted with advantage in another. There is a strong body of opinion which contends that a super-micro could be created by using the best parts of all existing machines, but that remains to be proved.

A major underlying reason for all this is the large-scale integrated circuit, which has eroded the freedom and responsibility of the system designer. His contribution is limited to selecting a given chip component and working out the interconnections. This calls for a certain amount of ingenuity and expertise, but does not encourage inspirational thinking. It degrades the hardware design function, and in some cases it also degrades the hardware design.

In consequence, we have had a spate of small computers that can be readily classified into family groups, with some sub-grouping to cater for oddities. Some have 'sprites', others manage well enough without. Some use the Z80, some use the 6502, and the 'IBM family' use the 8086/8. Other processors are less widely represented.

Producing such devices does not call for super-human design abilities. The production of an inexpensive but effective keyboard may emerge as a vital factor, while the general case design standard may be significant, printed circuit board design can be farmed out, as can many other aspects.

It has been shown that very effective microcomputers can be created by a team consisting of a man and a boy, but such a combination is commercially vulnerable. Success can be as fatal as a lack of public interest, for failure to deliver is sometimes seen as more important than failure to function. A sales force is necessary, if on occasion somewhat bad, and an effective crystal ball is essential.

These factors do not encourage the creation of a sound home-based industry. There is a tendency towards a faceless mass of products bobbing around at random, with one or another showing clear of the ruck from time to time, only to be swamped out of sight by action in favour of a competitor.

Importers are in a different position. A given number of machines arrive in this country, and must be sold — by hook or by crook. Technical considerations are of little importance, unless they can be covered by reference to a 'standard' created on an arbitrary and obsolescent basis. This is all a matter of sales, not technicalities.

Meanwhile, there are opportunities going begging. There is a large untapped market for microcomputers in industry, but that cannot be reached by a sales organisation geared-up to sell games machines. The larger companies which might buy in quantity want to talk technicalities, and few micro vendors are able to satisfy that desire.

So we need personalities who are genuinely technical, not mere entrepreneurs. Entrepreneurs are needed, yes, but they cannot provide a substitute for hard technical fact.

Pipe-dreaming? Maybe, but it would do microcomputers no harm to have a few 'names' to bandy about...

DON'T VAT THE PRESS

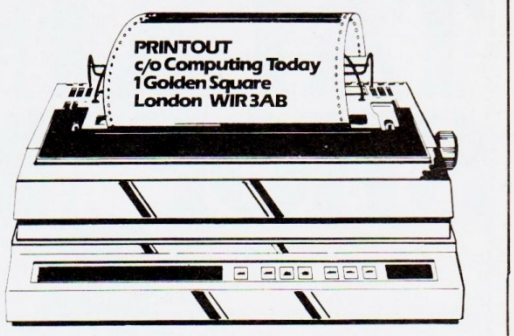
There are strong reasons to believe the Chancellor of the Exchequer is planning to impose VAT on your magazine. Such a move would turn the clock back 130 years — the last tax on newspapers and journals was repealed in 1855. Since then 'No tax on knowledge' has been a principle agreed by all Governments, even in the darkest days of war.

A free Press is a tax-free Press.

No Government should be given the power to impose financial pressure on a Press it may not like. Tell your MP to say 'NO' to any tax on reading.

PRINTOUT

Your opportunity to ask questions, put us straight, seek advice.



OOPS! — KOALA PAD

Dear Sir,
I would like to point out on behalf of Atari Corp (UK) Limited that there is a misleading statement in your article 'Taking the Tablets' in the December issue of Computing Today.

In the article you claim that: "One of the most remarkable features of the Koala Pad is its price. At £79.95 it is by far the cheapest touch tablet to reach the market . . ."

Atari would like to inform your readers that their Touch Tablet has a recommended retail price of £49.99 and is therefore considerably cheaper.

Kind regards,
Angela Oakes
Public Relations Executive

Thankyou, Angela, for that information. However, your letter was addressed to 'Ms' Jamie Clary. WE would like to inform our readers that Jamie Clary is not of the gender to which you refer. On the contrary he is, in fact, a man.

IN AMSTRAD'S DEFENCE

Dear Sir,
On January 22nd, Chris Curry, in a letter of "comfort" to the press states:

> "The BBC Micro and Electron are the most reliable by far of all the popular micros available. A recent survey by Business Decisions Ltd revealed that only 3% of Acorn's micros are returned faulty in the first six months, compared with 23% of Sinclair Spectrums, 18% of Commodore Vic 20s and 13% of Commodore 64s".

In view of the report carried by the relatively independent Personal Computer News magazine concerning this same survey (a copy of which is attached) we would observe that the Acorn statement again fails to acknowledge that Amstrad computers are equally reliable — yet Amstrad computers comprise a computer system, monitor and cassette recorder.

Draw your own conclusions.

William Poel
General Manager
Amsoft

MICROMART

Dear Sir,
I am a member of the St. Francis Church of England School P.T.A., Teazel Avenue, Bournville, Birmingham. I am investigating the feasibility of holding a Micromart during May 1985. It is planned to invite between 20 and 30 companies/businesses and provide them with appropriate facilities to sell their products — aiming at a home computer market rather than a business market. With these aims in mind I am writing to ask whether you could supply me with the following information:-

- Names and addresses of companies (preferably in the Midlands area) who you believe would be interested in exhibiting.
- Names and addresses of any known computer societies in the West Midlands area with whom we could exchange ideas and advertise the event.
- Details about similar events to be held in the West Midlands area in the near future.

Your faithfully,
AG Wilkins,
Treasurer,
St. Francis PTA Committee,
Kings Norton,
Birmingham, B38 8AQ.

Interested parties are invited to apply to the address given above.

ELITE REVIEW

Dear Sir,
I must admit to a major mistake in my review of Elite (Computing Today January '85). I said that Ian Bell wrote Starship Command on an off day. As someone who has spent many days playing Starship Command I should have known better. It was Free Fall which Ian Bell wrote on a bad day. Starship Command was written on several good days. The future developments with Elite are impressive. The Commodore 64 version is nearly finished and the tube (6502 second processor) version will feature such things as "Witch Space" — an area without a planet — all good stuff. The Z80 rights have been bought by British Telecom for a reputed £25,000 so the Spectrum and Amstrad versions are not far off.

Yours sincerely,
Simon Rockman.

Slapped-wrists, Simon!

WHAT COMPETITION?

Dear Sir,
I claim the prize in your 'odd-one-out' competition as featured on the front cover of the March 1985 Computing Today.

Thge smallest board is an Ambit 70cm to 144MHz UHF convertor board, and as such has nothing to do with baby oak trees (***What do baby oak trees have to do with anything? — ED.***). I suspect also that the two boards under it owe more to arcade video games than to home computers.

Nascome rules and lives on.

Regards,
Richard Smith

Congratulations, Richard, for your astute observations. However, as there was only one entrant to the competition (ie. your good self) your entry has been considered null and void! But thanks for your interest in the magazine.

BENCHMARKS

Dear Sir,
As I have only just bought the February 1985 issue of your magazine, I am sure you must be aware of a major error (apparently) in Benchmarks Revisited' on page 48. In BM8 all values should be multiplied by 10, except for the Amstrad value (I wonder too whether the Sirius 1 is completely wrong). Althogh not strictly comparing like with like, your article now surely reinforces the vlaue of benchmarks to compare 'microcomputers and their BASIC'.

Your sincerely,
Malcolm Chard.

We have always stated that benchmarks should be made over 1000 iterations. It now emerges that in the case of some computers reviewed in the past, our reviewers have taken the benchmarks over 100 iterations, and it is agreed that the 'times 10' suggestion is valid.

GRIFFITH COMPUTER ASSOCIATION

Dear Sir,
I am writing on behalf of the Association to let you know about our organisation.

It was formed in March 1984 and currently has a growing membership of around 50. We cater for all types of computers but the majority of the members own commodore equipment.

Meetings are held twice a month, but are going to be altered this year to cater for the needs of all the members. This year we are also starting our bulletin board to help our members keep up-to-date with all the goings on.

Further information can be obtained by contacting the secretary at the address given or by phoning the number shown.
Yours Sincerely,
Ron Gauci
Secretary.

THE GRIFFITH COMPUTER ASSOCIATION
P.O. Box 525, GRIFFITH. 2680
Ph.A.H. (069) 62-5877.
AUSTRALIA.

RADIO & COMPUTER FAIR

Dear Sir,
I would be grateful if you could print details of our event in the issue just preceeding the 8th April event date.

AMATEUR RADIO & COMPUTER FAIR
Bretton Hall College
Bretton
Nr Wakefield

Organised by the NORTH WAKEFIELD RADIO CLUB
Date: EASTER MONDAY 8th APRIL
FREE ADMISSION
DOORS OPEN 11am (10.30am for disabled visitors)
Refreshments available inc. GOOD ALE Bar.

Radio, Computer & Electronics stands
Bring & Buy
Bookstall selling RSGB Books
Handicraft stalls & film shows etc for the rest of the family.
AMATEUR RADIO TALK-IN ON S22 & GB3WU (RB15)
Located 1 mile exit 38 on M1
4 miles exit 39 on M1
10 miles M1 & M62 interchange
Good vehicular access and car parking
Bretton Hall College is located in 260 acrose of the most beautiful Yorkshire countryside and includes the famous Yorkshire Sculpture Park and Yorkshire Country Park.

For Dealer or Visitor enquiries contact
Steven Thompson, G4RCH, 2 Alden Close, Morley, Leeds LS27 0SG. Tel: (0532) 536633.

Yours,
Steve Thompson
Club Sec & Rally Publicity

A SILLY POINT

Dear Sir,
I write regarding Peter Freebrey's review of the Toshiba HX-10 (Computing Today, March 1985). In his review, Mr Freebrey comments: "One silly point perhaps, is the stress in the manuals on binary or hexadecimal notation — decimal equivalents may be substituted and would often be more readily understood by most readers." True. However, it is worth considering that binary and hexadecimal notations are inextricably linked to the many processes in computing; binary with the logic of the microprocessor and its busses; hexadecimal for its convenient base. To be conversant with these number bases is no bad thing, and to encourage readers' familiarity with such notations is completely justifiable. I would strongly recommended other technical authors to follow suit when writing manuals for other machines.

Yours sincerely,
Thomas Weybridge.

MICROCLUB

Dear Sir,
MICROCLUB is a non profit organisation. It's aim is to create a unique club for microcomputer users, to meet, as well as all those interested in this aspect.

One of our main pursuits is to set up a reference library, including relevant foreign magazines.

The Club's activities expand in several fields one of them is to bring our communiction with other similar Clubs in Europe.

We would be grateful if you could send us a list of addresses of other such Clubs and if you could announce through your magazine our interest in commencing a cooperation with them.

We would kindly request a reinforcement to our efforts by sending us a monthly copy of each of your magazines as an honorary subscriber.
Yours Sincerely,
C. Mantzavinatos
MICROCLUB,

SOFTWARE PIRACY

Dear Sir,
I write on behalf of the application software industry to suggest a permanent solution to the problems of software piracy of commercial applications software.

The uncopyable disk is a complete non-starter in the stakes of software protection. Copy busting utilities abound rendering the whole concept useless and the move towards hard disks renders them inappropriate.

The licensing system which requires voluntary restraint by the end user is a joke — its like leaving a child unattended in a sweet shop.

That leaves the dongle as the only suitable alternative. The only drawback is that dongles cannot at present be staked or chained together so you have to fumble around the back of the machine trying to pull a delicately pinned dongle on and off. If you have more than two or three dongles this is not an enviable task and rapidly becomes responsible for 50% of the "my computer won't work claims".

The solution to this problem is to stack dongles on a dingle tablet in the same way that carriages make up a train. Each tablet would hold half a dozen dongles with the facilty to chain a second tablet onto the end of the first tablet. Each tablet would be on the end of a short fly lead and hence easily accessible. All dongles would be on line at all times. The software simply checks that it's dongle is present and ignores all the others.

The only remaining questions to be answered are should it be RS232 or Centronics. Centronics would be better as the IBM PC has a Centronics port as standard — unfortunately its pin out is not standard! RS232 is standard on nearly all computers but some IBM owners might have to buy another board.

If software houses would agree to dongle standard then the current moans about software priacy would become a thing of the past. If the application software industry seriously wishes to claw back its £150 million of income lost to piracy then it should put its own house in order and not wait for politicians to do it for them. BSF and Dataview kindly take note.

Yours Sincerely,
Spencer Hall
Accounting Software

An interesting thought, but difficult in paractice. RS232 limits the performance of some printers, and owners might want to keep their Centronics ports free. Anyway, a dongle can be copied without undue difficulty.

SINCLAIR 'LESS CONFIDENT'?

Dear Sir,
I think that you should warn your overseas readers, who might contemplate buying a QL from Sinclair while on a visit to England, of the disadvantages and hazards involved.

I attach a (form) letter ***(not shown — ED.)*** received from Sinclair 4 months after I first wrote to them.

Why should one's basic legal rights be abrogated because you live in another country? Especially when, as in South Africa, PSION the Sinclair agents are on the spot.

If you buy a Seiko watch or a Nikon Camera, no matter where, the guarantee is honoured anywhere in the world. Can it be that British manufactuers are less confident about the quality of their products? And why shouldn't Sinclair pull out their finger and save people from abroad about £56 VAT (they probably have to pay tax in their own country anyway) as many thousands of suppliers of other goods in England do for their foreign customers.

To cap it all the computer is faulty which is why I am typing this on my 30 year old Remington instead of with QUILL!

Yours disgustedly,
M.J.F. Dempster.

No comment.

LESSONS OF HISTORY

Bill Horne

This issue, we begin a chronological tour of computing: a journey through almost five decades spent perfecting the Analytical Engine.

During the past forty years, computer technology has developed apace, from vague concept to full implementation, and those directly concerned have usually been too busy to pause and explain to the world at large what has been going on. From time to time there have been 'popular' explanations, written mainly by persons whose knowledge of the subject has been gained at second or third hand, but these have done little to make the subject clear to the lay reader.

Fear of the unknown is a familiar phenomenon, and where computers are concerned it has manifested itself mainly in terms of unjustified criticism. A 'final demand' for £00-00 is received, and the recipient writes to the newspapers deriding these 'newfangled' devices. It is rarely made clear that the fault lies with the computer user, rather than with the equipment itself. Even when this point is brought out, the implication is that a computer, like a gun, can be dangerous in the hands of inexperienced persons.

Yet to those who have been familiar with computers over a period of time, the picture is very different. Like firearms, computers can be misused, but they can also be used to the advantage of the community at large, and the benefits appear, in general, to outweigh the disadvantages. There are, perhaps, some undesirable side-effects. The Dickensian picture of a room-full of clerks slaving away at their desks was not a pretty one, but when a computer replaces the clerks there will be redundancies, and the spectre of unemployment caused by computerisation has been conjured up in some quarters to good effect. There is a sufficient reality behind this to make it a good political weapon, but only when the view taken of the matter is suitably limited in scope.

THE BALANCE OF SKILLS

The real key to the matter is that the introduction of computers must change the balance of necessary skills, and the appropriate adjustment of training and abilities proceeds far more slowly than it should. This is equally true in other fields. Not so long ago, Northern universities provided courses in metallurgy, but the demand for this specialisation has dwindled with the advent of plastics. The answer has been to change to courses in 'Materials Science', but the result has not been entirely satisfactory. One reason is that suitable teaching expertise is in short supply. Another is that employers fail to see how they can make use of the graduates, who are often confused themselves when asked what work they might usefully do.

The process of interviewing graduates for jobs in industry can be very depressing, because of the obvious mismatch between supply and demand. At best, a lengthy period of re-training is needed before useful results can be obtained. There are occasional exceptions, usually students who have had sufficient interest in their subject to carry out their own studies, and these are welcomed with open arms.

It may thus be seen that there are two radically different ways of looking at the effects of compouterisation. On the one hand, it can be contended that a computer can increase the efficiency of its user enormously. On the other hand, it is equally true that fewer employees will be required to carry out the same amount of work, though it is not necessariloy true to say that this must inevitably lead to unempoloyment on a vast scale. A more accurate statement would be that those who are unwilling or unable to come to terms with the changed situation are likely to find themselves in the dole queue.

This fact is commonly linked with a suggestion that only the young and brilliant members of the community can adapt to the needs of computer work. This, in turn, is another manifestation of fear of the unknown. A senior engineering manager who sedulously avoids all involvement with computers is not an unknown phenomenon. One such man, in his late forties, was brought under pressure to change his outlook, and despite firm resistance was given some 'private tuition'. After a while, he remarked, with some surprise, that the subject was really quite simple and straightforward, once you understood the jargon...

TERMINOLOGY

This is, perhaps, the key to the matter. Computers involve certain concepts which do not arise elsewhere, and this necessities the invention of specialised words by which the concepts can be expressed. Having got into the habit of inventing special words where they are necessary, some workers in the field have gone on to invent special words for everything, creating a huge vocabulary that is often coloured by localised variations. This creates an air of mystery, and consequently a shade of trepidation.

Now, perhaps, a change in the situation may be expected. Low-cost compouting equipment is being sold in vast quantities to 'personal users'. The very term 'personal computer', however, is open to misunderstanding. What distinguishes a 'personal computer' from any other kind? The first and most obvious distinction is that it is owned and used by an individual, or by the owner and his family, rather than being owned by an employer for use by his employees. For this kind of application, the personal computer does not need all the complexities of a multi-user machine, and can be built to a limited specification for a relatively low price.

The matter may go further than that, however. In some application, the personal computer may actually compete with its bigger relations in terms of efficiency and utility: Its cost per user may be lower; its convenience of use greater. This is a direct consequence of the way computers have developed over the years, as will appear during the study of computer history which follows.

As yet, the world of personal computers has not freed itself of the hargon syndrome. Indeed, it has developed jargon of its own in some areas. Nevertheless, it has become possible for totally inexperienced users to run purchased programs successfully, while others are learning how to write programs of their own, to their great delight and satisfaction. Some 'home-brewed' programs have even appeared on the market, and have proved to be more satisfactory than other programs produced professionally.

All this has wrought a major change in the outlook of the computer industry. Some companies are attempting to perpetuate out-dated ideas, and are achieving a measure of success with hide-bound purchasers who are equally unwilling to accepot that change is in the air. Others are quietly undermining the established ways with even greater success. It would be rash to try to predict the future, but it is at least possible to look back and see how the present state of affairs has come about.

THE STARTING POINT

Any attempt to define a starting point for computer development is likely to be challenged, because — like most innovations — computers came into being stage by stage, each small advance contributing to the whole. However, it is widely accepted that one of the more important contributions was a paper by A.M. Turing, published in 1937 with the rather daunting title: "On computable numbers with an application to the Entschiedungsproblem."

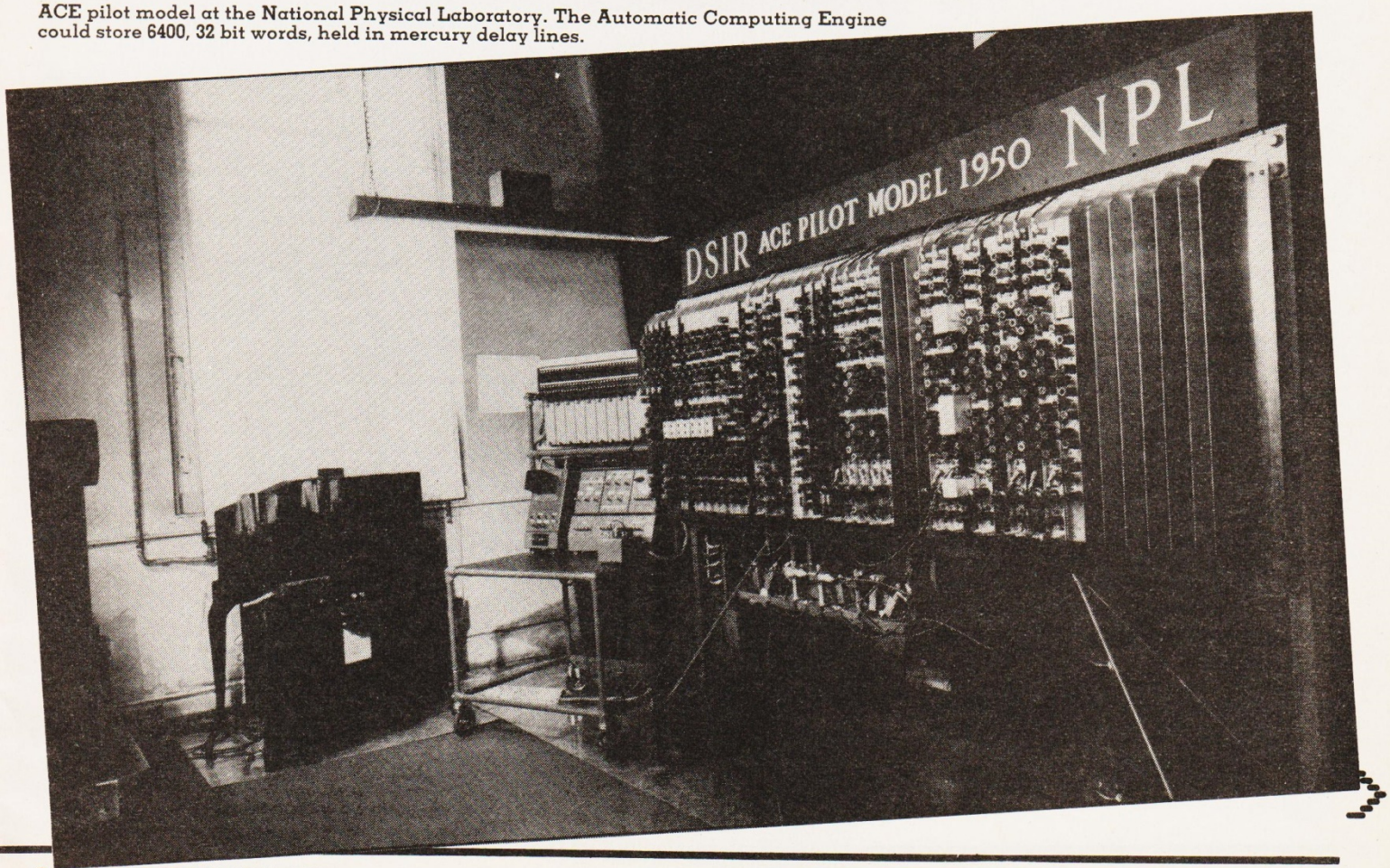
Simple calculating machines, mostly of a mechanical nature, were then commonplace in mathematical circles, but their usefulness was limited. To carry out a complex calculation, it might be necessary to carry out a long sequence of operations, perhaps noting down intermediate results for future reference. A central feature of Turing's concept was the storede program, which could execute the required sequences automatically. That, in essence, is what distinguishes a computer from a calculator.

Turing was then working at the Princeton Institute for Advanced Strudies, and one of his colleagues there, J. Von Neumann, has been credited with the invention of the stored program concept. It would serve no useful purpose to argue which of the two men deserve the honour more. Both made their contributions, Turing mainly in Britain, Von Neumann mainly in America.

In any case, dreaming up the concept was one thing, putting it into practice was another. Some saw a possible answer in an adaptation of the pianola principle, which uses data stored in the form of holes punched in paper. Each hole causes a piano hammer to strike the strings, producing complex tunes. This is essentially a pneumatic system, air pressure being used to create the striking power under the relatively weak control of the roll of paper. Applying the method to computing was far from straightforward, and it provided no useful solution to the problem of storing intermediate results.

Turing evaded this need, since he made no distinction between input data, output data, and intermediate data. His output data was assumed to be in a form that could

ACE pilot model at the National Physical Laboratory. The Automatic Computing Engine could store 6400, 32 bit words, held in mercury delay lines.



subsequently be read as input data, so there was no need to store it in any other manner.

The electronic world could offer little help at the time. The very word 'electronic' was not yet in popular currency, most activities in the related field being classed as radio, television, or — to those few of us who came into contact with the newest development — radar. F.E. Terman's monumental work, the "Radio Engineers' Handbook", published in 1943, contains a thorough-going survey of the techniques then in general use, but is almost entirely concerned with circuitry in which voltages are constantly changing, usually in patterns built up from sine waves. This would nowadays be called Analogue circuitry.

The sole exception is an indirect reference to a circuit known since the early nineteen-twenties. The Eccles-Jordan Bistable is so called because it can take up either one of two stable states, maintaining that state indefinitely while power is applied, but changing to the opposite state under the influence of a control trigger.

In the immediate pre-war years students were taught this circuit, but only as an interesting oddity with no obvious practical application.

CHANGED BY TELEVISION

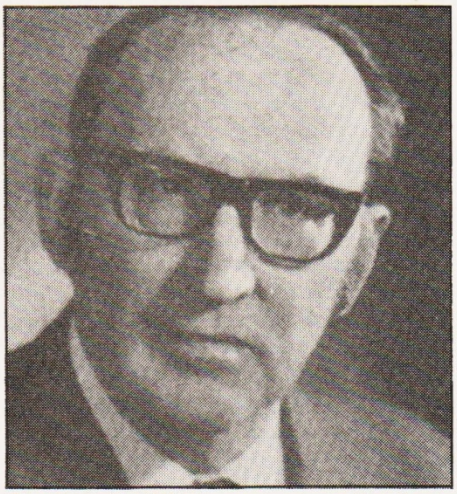
This situation was slightly changed by television. In order to obtain proper synchronism between the horizontal and vertical scan actions by which the picture was created, it was necessary to count down from a master frequency in a very precise manner. To simplify matters, the number of lines forming the picture was made the product of small prime numbers: 405 = 5 * 3 * 3 * 3 * 3. A master frequency of 20250 c/s was generated, this being divided by two to produce the horizontal scan frequency. A series of frequency dividers giving an overall frequency ratio of 405/1 generated the vertical scan frequency of 50 c/s.

The division by two used an Eccles-Jordan circuit in the original form. Division by three entailed adding a third stable state, which meant using three thermionic valves instead of two. As with the original, only one valve was allowed to conduct at a time. Division by five was achieved by the use of two more valves.

These circuits involved networks which were the fore-runners of what we now call Gates. These are multiple-input circduits in which the output state depends on the combination of input states. In an AND gate, the output is 'true' only if all the inputs are true, while a NAND (i.e. Not AND) gate gives a 'flase' output if all its inputs are true. An OR gate3 gives a true output if any one of its inputs is true, and a NOR gate gives a false output in those circumstances. The true and false states are distinguished by different voltage levels.

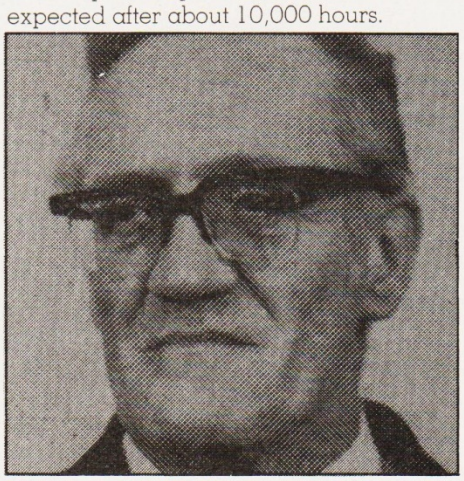
All these circuits and many more are involved in computer hardware, and even where they existed in the years before the Second World War they were not recognised as relevant.

During the war, however, the development of radar and allied equipment proceeded apace. The electronic industry, in the wider sense, blossomed out, and those who maintained and used the new equipment in the services learned new skills which they might not otherwise have encountered. Turing himself, working with the Foreign Office, came into contact with pulse techniques which he saw were applicable to computers.



Professor M V Wilkes was responsible for the construction of EDSAC 1 which became operational in May, 1949. With two colleagues, he published the very first book on computer programming, in 1951.

At the end of the war, however, the electronics industry was thrown into confusion. As military contracts dried up, it became urgently necessary to find replacement markets. Computers were considered, but not very seriously, as it was clear that they would need to use large numbers of electronic valves. It was predicted that these would show an impracticably high failure rate. It was said that every time the American ENIAC was switched on, an average of 35 valves failed. Keeping power on all the time might have avoided this, but a catastrophic crop of failures could then be expected after about 10,000 hours.



Professor F C Williams worked on radar research at Bqwdsey and Malvern during the 2nd World War, returning to Manchester University to occupy the Chair of Electrical Engineering in 1946.

It is scarcely surprising that computer work first began to show results in academic circles, where the prime interest lay in the principles involved, rather than in the production of a marketable device.

Broadly speaking, this work began in 1945, though any attempt to catalogue events on a strictly chronological basis would only lead to confusion. Initially, there were three main centres of activity, the National Physical Laboratory, Manchester University, and Cambridge University, with specialised contributions from elsewhere.

The National Physical Laboratory was first in the field, but that advantage was soon squandered. A Mathematics Division was set up in 1945 with the intention of coordinating developments which had taken place during the war years. A.M. Turing became head of the electronic computing section, working along for some months on the production of a detailed proposal for future activities. This was, in effect, a plan to produce an Automatic Computing Engine, the ACE.

ACE SPECIFICATIONS

The ACE was to have storage for 6400 words of 32-bit length, held in mercury delay lines. Described by a member of the project as 'very large', it would be considered rather limited in scale by comparison with a modern personal computer for home use. It worked Serially, which means that the individual bits of each word emerged from the delay lines in turn, not simultaneously. One bit emerged each microsecond, so it took 32/uS for each word to appear, and with 32 words in each delay line it might be necessary to wait for up to a millisecond for a given word to become available.

Turing overcame this to some extent by 'staggering' the instruction words in the storage system, so that each instruction became available as it was needed. This, in turn, led to the development of a Four-Address instruction format. Addresses A and B defined the location of the course data. After processing, the result was stored in location C, and the next instruction was drawn from loation D. It is said that this made coding of instructions 'very untidy'.

During 1946 and 1947 the electronic computing section slowly grew to about six and a half people, the half being Dr J.H. Wilkinson, who was shared with other work. A number of programs were written, including processes for single and double precision floating point, but there was no hardware on which they could be run. ACE progressed, in concept, from Version V to Version VII, as the theoretical work showed the need for change, but the production of actual equipment was making no headway.

The initial plan was to sub-contract manufacture to some other government department, preferably one with experience of pulse techniques. This was not favoured by some members of the group, who feated that control of the project might thereby be lost, but that was never put to the test, partly because no one seemed interested in taking on the task. Instead, an in-house electronics section was established, mainly by internal recruitment. Unfortunately, there was greater interest in the problems of industrial electronics than in computers, and relations appear to have become a little strained.

LITTLE PROGRESS

In 1947, Turing decided to take a Sabbatical at King's College, Cambridge, where he was a Fellow. Disillusioned by lack of progress, he felt in need of a break, but when he returned in May 1948 he found that little had changed. Construction of a simplified machine, known as the Test Assembly, had been abandoned when it was nearing completion. Seeing no point in staying, Turing joined the Manchester University team.

This was by no means the end of the N.P.L. story, but it had been a sad history of wasted opportunity. Turing's difficulty in getting on with some of his colleagues may have been a contributory factor, but there was a distinct odour of competitive empire-building about the whole affair, and of a lack of leadership from above. It must be suspected that there was some antagonism to computers, perhaps in part because they were unfamiliar and imperfectly understood. Perhaps Turing's ideas were not completely practical in some respects. It would have been better to give him the chance to show otherwise, rather than allowing the issue to be pre-judged.

At Manchester, the work hinged on the activities of Professor F.C. Williams, who was an engineer rather than a theorist, declaring that he never was, never had been and never would be a mathematician. Spending the war years at Bawdsey and the Telecommunications Reserach Establishment at Malvern, he moved to Manchester in late 1946. His immediate task was to develop the cathode ray tube storage system on which he had been working at Malvern, this being an alternative to the mercury delay lines favoured elsewhere. The method had been tried and abandoned in America, but the Manchester team were more successful. Initially able to store just one binary digit, the system was progressively enhanced until it could handle 340,000 digits.

The situation at Manchester was obviously very different from that at the National Physical Laboratory. The N.P.L. team had been primarily theorists. At Manchester, control lay in the hands of engineers, and something was achieved, even if that something was not theoretically ideal. For example, at one stage the equipment was housed in two rooms on different floors, and to transfer data from the cathode ray tube store to the magnetic backing store it was necessary to throw switches manually and then shout instructions up or down the stairs. It may have been rather crude, but it worked...

This photograph, taken in 1947, shows Professor Wilkes inspecting one of EDSAC's delay tube batteries

THE EDSAC

The activities at Cambridge sprang from the establishment in 1937 of a Mathematical Laboratory. It was intended that the old Anatomy School would be refurbished to house the laboratory, but war came before the building could be formally opened, and the staff dispersed to wartime duties. Returning in September 1945, Professor M.V. Wilkes found himself in charge, and he began to examine the computer situation, taking special note of what was being done in America. In 1946, he received a telegram inviting him to attend a course on computer design, the instructors being Presper Eckert and John Mauchly, the designers of ENIAC.

As a direct consequence of this, he set up a project to build the EDSAC. This used mercury delay tube memory, and was otherwise in general accord with Turing's ideas, but the design was conservative, halving Turing's proposed clock rate and leaving wider margins in other respects.

The first of these machines to go into service was the one built at Manchester, which first ran in July 1948. It was very small indeed, with a capacity of 32 words, each of 31 digits, but it worked. By October of the same year, a contract was placed with Ferranti Ltd for the construction of a further machine "to the instructions of Professor F.C. Williams". That must be the briefest specification which ever formed the basis for a computer procurement contract. The machine ran by the end of 1949.

At Cambridge, EDSAC was running by May 1949, at which time the design of the Pilot ACE at NPL had just been started. That bore fruit in May 1950, though public demonstration was delayed until December of that year.

THE FIRST AGE ENDS

So, in fairly brief terms, ended the first computer age. Much detail has been omitted, such as the contribution made by Dr A.D. Booth, working at Birkbeck College, and specialising in magnetic drum storage, but it would be difficult to present a totally balanced picture giving fair prominence to all concerned.

The main point is that three groups had shown that — despite the pessimistic projections regarding reliability — it was possible to build a workable stored-program computer. That, however, was still a long way from building a marketable item for general use.

Photographs reproduced by kind permission of The Institute of Electronic and Radio Engineers

'The Lessons of History continues in the next edition of Computing Today.

PHOTOCOPIES

A COMPUTING Today Reader Service

To take advantage of this simple service, just fill in the required information and send it (or a photocopy) to:

COMPUTING TODAY Photocopies,
No. 1, Golden Square,
London W1R 3AB.

together with your money and we'll do the rest!

Lost and can't be replaced? Well, if you've lost one of the early issues that cannot be replaced from our stocks of backnumbers, all is not quite lost.

If you know the article name and the issue it appeared in, we can supply you with a photocopy for the miserly sum of £1.50 including postage and packing. If you're stuck for the month, try checking in our annual index which is included in the December issue each year.

CT PHOTOCOPIES

NAME .....

ADDRESS .....

.....

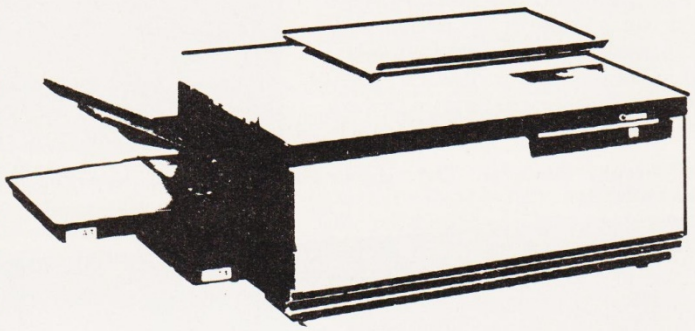
..... POSTCODE .....

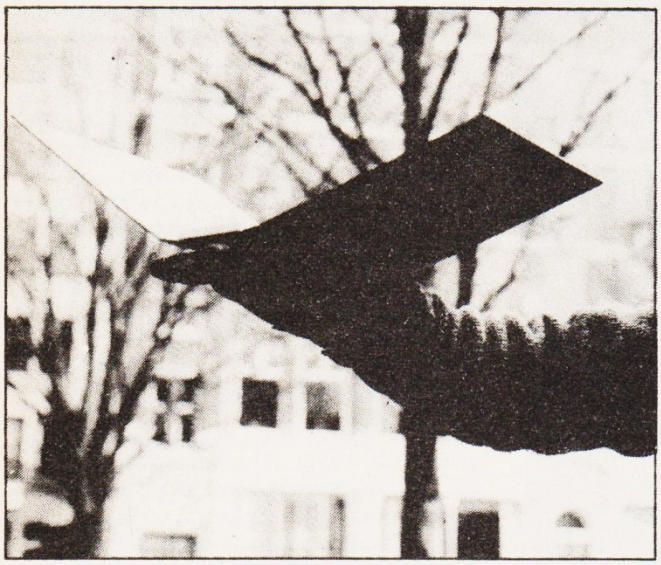
Please send me Photocopies of the following items

ISSUE	ARTICLE	PAGES

At £1.50 each, I enclose £ .....

Cheques and Postal Orders should be made payable to ASP Ltd.





In days gone by, falconry was the sport of gentlemen and kings — this noble and time-honoured tradition is not so prevalent in these technological times, and it is quite a pity, too. Just imagine the pride you'd feel standing in your own back yard while your very own hunting falcon swooped down upon unsuspecting dogs, cats and Ford Sierras.

For a limited time only, Computing Today is offering you the chance to experience the thrill of commanding your own bird of prey, with the new CT Hunting Falcon/Magazine Binder. Swift of wing, sure of eye and made of genuine vinyl and cardboard, the Computing Today Hunting Falcon/Magazine Binder is the spitting image of the hunting birds of old to anybody suffering from cataracts. Release it from your arm, and it dives just like a traditional hawk. If it lands on a small animal, it will probably stun it. Also, when you tire of the sport, and would rather hunt hedgehogs with your Ford Sierra, your CT Hunting Falcon converts into a useful magazine binder that holds a full year's supply of Computing Today. The new CT Hunting Falcon/Magazine Binder will cost you not a farthing more than the old binder alone used to: just £5.00. This includes postage and packing, so your falcon won't have to tire itself out flying to your abode.

Cut out and send to:

COMPUTING TODAY HUNTING FALCON/MAGAZINE BINDER, INFONET LTD, TIMES HOUSE, 179 THE MARLOWES, HEMEL HEMPSTEAD, HERTS HP1 1BB, ENGLAND

I am enclosing my (delete as necessary) Cheque/Postal Order/International Money Order for £ .....
(made payable to ASP Ltd)
OR
Debit my Access/Barclaycard*
(*delete as necessary)

Insert card no.

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss) .....
Delete accordingly

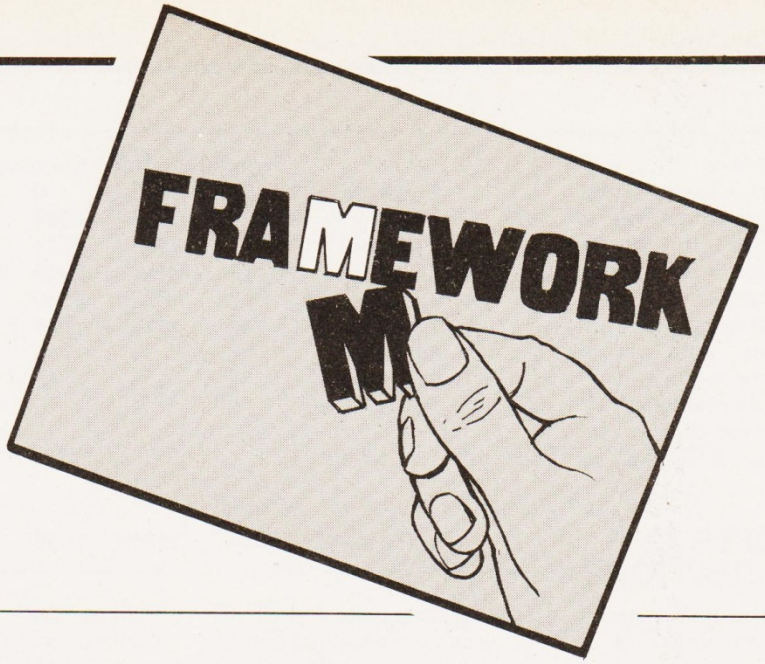
Address .....

.....

.....

Signature .....

Date .....



WRITE-YOUR-OWN SPELLCHECK

J R G Woods

This is our first FRAMEWORK article, designed to provide you with the details and guidance necessary to construct a particular program.

A spelling checker is a most useful accessory to any word processor, not because of any inability to spell words correctly, but because with word processing it seems best to type as fast as possible and correct the mistakes afterwards.

Having looked at the performance of a 'favoured' spelling checker for the TRS80, I was prompted to write one of my own which could enable me to list errors to the printer, and hopefully operate much faster. This article should enable you to write an equivalent program for your own micro. Mine, incidentally, was written in Z80 assembler, with about 44k of useful memory to work with.

HOW TO...

The first problem that we are up against, is how to store thousands of correctly spelt words against which our text can be checked.

I chose to attack this problem by using the technique of tokenisation. Tokenised words are stored in high memory, serviced by a suite of short programs that reside in low memory.

Tokens are used by many computers to store the 100 or so words reserved for use by BASIC. For the thousands of words used in everyday English, tokenisation can be performed in principle by having a list of the 256 most common syllables and breaking up each word into these syllables; eg com/pet/ition (assuming these three syllables are in the list). Each syllable in a given word is replaced by a number that matches the syllable, to create a string of numbers (or tokens) that is shorter than the original word (3 instead of 11 for 'competition'), and it is these tokens that are stored in memory.

OUR SCHEME

The tokenising scheme I propose is, however, somewhat different. A casual glance at the construction of English words shows that many have the same 'root' but different endings. This makes it worth making the final token operate in a different way: each memory location holds 8 bits, and by making the state of each of these bits signify a particular ending, up to eight words (having a common root) can be stored in one token string.

This still leaves a number of very common syllables (eg 'tion') that occur only at the end of words. Therefore use a different list of common syllables for the end of the 'root' from those used for the rest of the word. In this way a word will be tokenised in a format:- "AAAFE", where each "A" is a syllable from some list 'A', and "F" is from another list 'F', whilst "E" defines the end via its bit pattern. Thus, the shortest words would have the tokenised form "AE" (I later give the "best" syllables to use for each list).

In order to make the search for a word fast, an index is used. This index is simply a list of memory locations that define the start positions of blocks (pages) of memory that hold token strings with the same starting token. These pages are of whatever size is needed to store all the strings, hence the need for an index. Since all words on a page of this kind have the same starting token, there is no need to explicitly store it each time. Thus a token string of length N need only store N-1 tokens! A further subdivision is also worthwhile, which is to keep all strings on one page the same length. This way there is no need to waste memory marking the end of each string, and the searching process is also eased. An analysis of 12000 words (this will eventually be 20000) shows that an efficient structure is: 255 pages for 2 token strings (each 1 token long and so no index is needed — (as a diver-

sion think out how this works) and 255 pages of 3,4,5, & 6 token strings. These short strings have a unique page for each start token but longer strings (which are comparatively rare) are paged only on string length and not starting token. With nearly 1300 pages no page will be very long, and so the search for a match between a new word and one in the list will be very fast.

	"roots"	"words"	Memory used
2 tokens	80	160	0
3 tokens	1432	3613	2864
4 tokens	2257	4925	6771
5 tokens	1479	2444	5916
6 tokens	512	672	2560
7 tokens	104	119	728
8 tokens	14	19	112
9 tokens	5	6	45
10-20 tokens	0	0	0
TOTALS	5883	11958	18966

TRADEOFFS AND CONSIDERATIONS

As well as the tradeoff between speed and memory that is usually present when writing any software, my approach to this program presents yet another tradeoff, between the amount of memory used for code and that required for stored words. The rules of English might be used to make word packing very compact: as a simple example, one might use the fact that q is always followed by u to deal with qu and q simultaneously, but the code might take more space than doing it the simple way. Almost certainly an understanding of the structure of English words would help (adjectives take different endings from verbs etc.), but don't forget that at the end of the day most of the words that you use must be in the list contained in memory. How will you arrange this? The easy way is to take old texts, most of which will have had their spelling corrected, and put them through the spelling checking process, with the 'L' for 'learn' key held down. Doing it this way will allow the program to learn all the words. It would be a very clever program indeed if it could comprehend and take advantage of global structure and relationship to other words and thereby pack more efficiently.

My structure was to allow about 2.5 K of memory for program, followed by index tables for the 'pages' of syllables and tokenised words, with the top of memory holding the tokenised words.

SYLLABLE LIST STORAGE

The manner in which syllable lists are stored is worth comment. For each list, an index is used to select a "page" using the first letter in the syllable (26 pages per list). The first entry on the page is the token value for the next syllable, thereafter the syllables (with the first character removed) are packed sequentially, but with the start character of each having bit 7 set. The last entry on each page is a suppressed single character, and so the entry for this is a space character (with bit 7 set). The "F" syllables are stored backwards. ("ously" is stored as "lsuo"). All this makes tokenising easy and fast. Detokenising, on the other hand, involves a fair amount of code to descramble the above, and is therefore relatively slow: This does not matter since detokenising is only needed for programs that pull the occasional word out of those stored, to be displayed or printed.

I did not use the token 255 (OFFH) to represent a syllable, as it is convenient to use it to mark the end of each page. Since tokens are stored in numerical order a search can exit (with no match) when the stored token string is numerically greater.

CODING AND CRITERIA

Having developed software to the point of being able to tokenise and search the list, some effort is needed to turn it into a useful suite of programs. A modular approach to coding pays off here because many of the programs are simple combinations of the same functions, (eg tokenise, detokenise, search, insert a string in a page, remove a string from a page etc.)

I give below the programs that I found useful, and indicate what they do:-

SPELL reads in a block of data from disc (or from cassette), and converts all u/c letters to l/c and skips any word with u/c content if the user wants. It tokenises each word, and looks for a match. If no match is found, response depends on the option set:

a) the user can either skip the word or add it to the list.
b) The word is simply put to the printer.

TABLE A

```
abs ab acc ac adv ad af ag all al am an app ap ar ass as at  au
bar  ba  be  bi bl bo br bu  can cap ca ce cha ch ci ck cl comp
conc conf cons cont com con cor cou co cre cri cr cu    da  dec
def  dep  des  de  dis di do dr du  ea ec ed el em ent en er es
exc exp ex ev ep et  fa fe ff fin fi fl for fo fre fr  fu    ga
ge  gh  gl  gn  go gra gr gu  ha he hi ho hy  ic ie id if ig il
imp im inc ind ins inter int in ip is it  ju  la le  li  ll  lo
lt  lu man mar ma men me min mis mi mo mp mu  na nc nd ne ng ni
no ns nt nu  ob oc ol om on op or ot ou ov os  par  pa  per  pe
ph  pi pl pos po pre pri pro pu  qua qui qu  ra rea rec rel rep
res re ri rm ro rr rs rt ru  sa sc se sh si so sp  ss  sta  sti
str  st  sub  su sy  ta ter te the th ti to trans tra tri tr tt
tu  um unde un ur us  va ver ve vi   we we wh wi wo
```

Syllables in the "A" list

TABLE B

mb tanc anc atic stic tic ac tric ic enc nc ward ard ad id od ound end nd ud ed ld rd ff ing ag ng rg gg ish sh tch ch th iti si ri ifi ck ak ok nk rabl tabl abl ial sibl ibl tical ical tional onal ral tal tual ual ful ell el il ol val al cl ll pl gl bl am em om im um form rm ption ication gation lation mation tation ration sation ction ition ation ssion sion tion tain ain an ion on ten ven en in ign rn wn nn ap op ep ip mp pp tur lar ar ir our ur ator ctor tor or ter er les nes as es alis is ess ss pos us os ns ememt nment ment inat ulat lat let ight ught ght ort iat ft ient ent tant ant int nt gest est ast ist st erat mat rat nat tat gat at rit ut et ot it lt ect ct pt ert rt tt riou iou ou gu ativ ctiv tiv siv iv ov ev lv ow ew ex vely stly ically ially ally fully bility ility lity ery rly ity ently tely tory ory tary ary ey ry ty ny ably sly bly dly ely ntly ily tly ly ency ay cy dy ify py sy

Syllables used in the "F" list

c) If all words are being put to the printer then it is printed "bold".

In this program questions such as the criteria for end of a word, (space, or. or end of para or page, what about / and — ?), and the manner chosen to deal with paragraph identifiers in the form a) or vi) must be answered. Also any imbedded control characters in your text may need special attention. Features such as Disk and file name and date and time as headers to print out, and an ability to display the directory are worth including.

I though about a mode in which errors could be corrected as they were found. This is messy, probably requiring two disk buffers in memory, and 2 files on disc. My real objection however is the annoyance of being able to correct a spelling error, whilst the screen is showing some other mistake that I can't fix until later.

REMDIC allows words to be deleted or added to the list from the keyboard. It can also create an index to an empty list.

COMDIC adds the new words from a newly created list on a disc to a 'master' list in memory, displaying and possibly printing them. By looking at these very, very, carefully, and removing any rogues with REMDIC, and then saving the updated list to disc, you can keep control of words that have been added to your 'master'.

Some programs are invaluable in development, ie. **DLIST** displays all the words in the list (possibly writing them as an ASCII file to disc, and this is what you would use if you changed the structure of your tokens, but wished to get back all those words that have been learnt).

ANALDIC analyses the list showing page sizes, and frequency of use of each token, along with nearest neighbour.

PERFORMANCE

What is the performance of a spelling checker written in this way? My version, using a Z80 clocked at 1.7 MHz, will check about 100 words per second (half the time is spent reading from the disc). So far I have 12000 words in the list (5883 'roots'). I have to admit that the 12000 words were achieved by looking at about 10000 words that had been learnt, and for those that were 'roots' + ending I added further valid endings (If I had done this completely there would probably be 12500 words). Using this set I typically find about four new words per thousand scanned in new text written by me, and rather more in texts from other authors.

In quoting memory used per word, I quote the effect of adding a word to an existing list (The overhead of the index and page ends is not included). The average memory used per word is 1.6 bytes and with 17000 memory locations free I have no doubt that eventually 20000 words can be stored without difficulty.

SUGGESTED SYLLABLES

The suggestions for best syllables that follow are based on optimisation on my 5883 'roots'.

The distribution of roots and words according to string length is shown in tables 1 and 2.

Each of the lists also contains the 26 single letters of the alphabet.

Since you may wish to use some different syllables from those given I identify below the less useful syllables. The "value" of each is quoted in terms of how much extra memory would be used in storing my set of words if the syllable were not in the list.

"A" list: ap 16 can 17 cap 16 cor 16 fre 16 gh 17 hy 18 if 16 mar 15 mp 17 nc 16 pri 17 sy 16 um 16

"F" list: tric 8 ff 8 tual 9 val 9 form 8 ion 9 ep 8 pos 7 ght 9 gest 7 rit 9 ew 8 stly 8 ility 6 gy 7 sy 9 py 7

"E" list: ing, ers, er, ed, d, es, s, none of these 7

PASCAL COMPILERS

We can advise which is the best Pascal for your needs. Our wide range includes the remarkable TURBO Pascal.

8-bit	Nevada Pascal (JRT4)	£ 40
	Turbo Pascal v2.0	£ 49
	Pascal/MT+	£ 99
	Pro Pascal	£199
16-bit	Utah Pascal (JRT)	£ 40
	Turbo Pascal v2.0	£ 49
	MS Pascal	£ 99
	SBB Personal	£115
	Practical Pascal	£145
	Pro Pascal	£290
	SBB Professional	£335
	Pascal/MT+86	£380

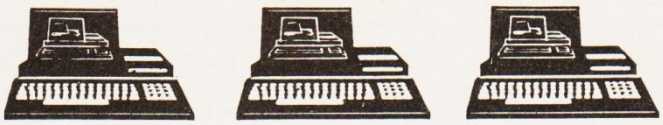
Prices include delivery, but not VAT.

For more information call us.

GREY MATTER

4 Prigg Meadow, Ashburton, Devon TQ13 7DF.

TEL. (0364) 53499



Two new Micro Books

Beginner's Guide to Microcomputing

E A Parr

and

Beginner's Guide to Microcomputer Languages

Michael Duck

Beginner's Guide to Microcomputing is for people who have just bought, or are thinking of buying, a home computer. It gives a serious overall view of their methods and applications with intelligent and informative diagrams and photographs.

192 pages approx *£4.95*

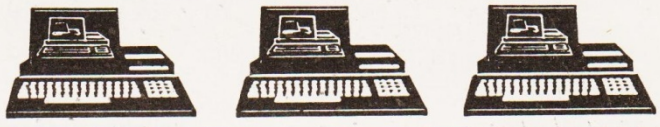
Beginner's Guide to Microcomputer Languages presents to the beginner the six most popular languages – BASIC, Pascal, COMAL, FORTH, PROLOG and LISP.

160 pages approx *£4.95*

Order now from your bookseller or direct from

Newnes Technical Books

Borough Green, Sevenoaks, Kent TN15 8PH



HISOFT

High Quality Microcomputer Software

CP/M PASCAL

When choosing a compiler, support is crucially important. It is vital to ensure that you will receive technical back-up for the package as quickly as possible. This may prove difficult, if not impossible, when the software was designed in another country and you have no access to the authors. **HiSoft Pascal** is a British product, created and manufactured by ourselves and fully supported by our technical team here in Dunstable. We offer inexpensive upgrades and we are continually extending and improving the compiler.

HiSoft Pascal is available in a wide variety of Z80 CP/M disc formats, is very close to Standard Pascal and comes supplied with a comprehensive and sophisticated full screen editor (ED80) which is specially designed for program development; the editor may be easily configured to suit all displays and also for special keyboards and individual keystroke sequences.

All this for a fully inclusive price of **£39.95!**

High Quality Software at a Fair Price.

HiSoft was founded over four years ago and since then has built up a reputation for the quality of its products and the strength of its support. Our utility and language software is available for most of the home computers currently on the market and we are always developing new products: our latest is **Devpac 80**, a powerful assembly language development package for all Z80 CP/M computers. **Devpac 80** is a macro assembler (GEN80), a configurable full screen editor (ED80) and a super front panel debugger (MON80) all in one integrated package.

Devpac 80 makes Z80 assembly language development a joy to do and is incredible value at only **£39.95** inclusive!

Please feel free to write to us or telephone for immediate despatch of full technical details of all our products – there is so much more than we can say in this small space. Our next product for CP/M-80? Wait and **C**!

HISOFT

180 High St. North
Dunstable, Beds
LU6 1AT
Tel. (0582) 696421

ALGORITHM ANGLES

Bill Horne

More algorithms as meat for your micro.

The better microcomputers (which does not necessarily mean the ones that cost most!) provide facilities for bit logic, and when the possibilities of this are understood it may seem that omission of them is idiotic.

For example, here is an oddity you may not have encountered before;

```
B = A AND -A.
```

What would you expect that to do? Well, negating A begins by an inversion, leaving all the bits in the opposite state to those in the original A. In particular, there will be 1 states at the least significant end until the first true bit in the original number. To complete negation, the inverted form must be incremented, and this turns the 1's at the least significant end into 0's, and the subsequent 0 into a 1. That will be the only bit that is true in both A and −A, and will be the only true bit in B. For example;

```
 A = 01010100
 A = 10101011
-A = 10101100
 B = 00000100
```

This little fiddle can be used to scan through the bits of A in turn looking for true states. After responding to a given bit, it can be zeroed for A = A XOR B.

The advantage of this kind of operation is that it allows eight states to be registered within a single byte, or 16 can be recorded in a single-length integer word. That can save a lot of storage space with some kinds of program.

One of the problems which can arise in such programs is that a combination of logic instructions can be difficult to follow, but it is always possible to set up a BASIC simulation. This may not be quite as simple as it sounds. For example, to simulate the Z80 instruction RRC A, you need two BASIC statements;

```
C = A AND 1
A = A\2+C*128
```

C picks up the least significant bit of A before the shift, so that it can be set in the most significant bit after the rotate. Note the reverse slash in the second statement. This should be 'DIV' in some BASICs, standing for integer division. If you lack that valuable facility, you will have to use A = INT(A/2) instead. Incidentally, C also shows the state of carry after the shift.

For RR A the rotate is through the carry bit, which is bypassed in RRC A, so an extra variable is needed;

```
C = C AND 1
A = A\2+CA*128
CA = C
```

CA will have been determined by a previous operation, and it might be necessary to add CA = C to the simulation of RRC A, in case carry is relevant in a subsequent operation.

By setting up a simulation of this kind, and displaying the results at each stage, the action can be following in detail, a 'truth table' being created.

Another use of binary logic is the decoding of bit patterns such as those used for determining character shapes. To display characters in enlarged form, a routine like this is needed;

```
FOR X = 7 to 0
B = (A AND (2↑X))
IF B = 0 THEN PRINT " ";ELSE PRINT
"•";
NEXT
```

A holds the pattern byte, and the bits of the byte are picked out in turn, starting with the most significant.

One advantage of logic functions is that they are inherently fast, no floating point processing being involved. In the above example this is offset by the use of powers of two, and a better version might be;

```
100 X = &80
110 B = A AND X
120 IF B = 0 THEN PRINT " "; ELSE
    PRINT "•"
130 X=X\2
140 IF X < > 0 GOTO 110
```

Worrying about speed in such a small routine may seem a waste of time, but the routine has to be repeated many times to generate a complete character set.

Some systems have an integer number range of + and − 32767, and there are times when this is insufficient, as in store addresses needing to run from 0 to 65535. A neat little conversion is;

```
A = A - 65536*(A↑0)
```

This kind of logical throws some people, so a word of explanation may be useful. The bracketed term is zero if A is positive, −1 if A is negative, so 65536 is added in the latter case, A being unchanged if it is positive.

The conversion is necessary on some machines where A holds an address that is to be split into two bytes to be poked into memory for use by a machine code routine.

Of all the logic function commonly found, perhaps XOR is the most confusing, especially where it occurs in combination, as in;

```
A = A XOR C
A = A AND B
A = A XOR C
```

If the AND operation were omitted, the two XOR C operations would leave A unaltered. The AND operation will zero some bits of A, and these will end up as copies of the corresponding bits in C. The other bits, unaffected by the AND, will be determined by their original state in A. By setting a suitable mask in B, some bits of A can be combined with other bits from C. This is very handy when setting up a complex control byte to be output to a peripheral address.

Processes for updating screen memory can be even more confusing. Here is a real stinker;

```
A = A XOR B OR C XOR C XOR B
```

where A is the existing byte in screen memory, B defines colour, and C is the pixel mask. Why not set up a simulation to show what effect this expression has?

To do that, you will have to give A, B and C all possible bit values. One way is;

```
FOR X = 0 TO 7
A = -((X AND 4) < > 0)
B = -((X AND 2) < > 0)
C = -((X AND 1) < > 0)
....
NEXT
```

You could, of course, use lines of the form;

```
IF (X AND 4) < > 0 THEN
A = 1 ELSE A = 0
```

but that seems relatively clumsy.

These random examples should have shown why any serious computer should implement bit logic. You may not want that kind of function very often, but when you want it you want it badly.

Your computer doesn't come up to scratch? Hard luck. You need something better...

If YOU have an algorithm that you think would interest other readers, send it to:

ALGORITHM ANGLES
Computing Today
1 Golden Square
London W1R 3AB.

LEARN 'C'

Mark Woodley

Part one: Functions, Declarations, and Storage Classes.

It is probable that the majority of microcomputer users either know little about, or have never heard of, C. This is perhaps not too surprising when one considers that it was first implemented on a mini-computer (the PDP 11/45), and in its relatively short life of eleven years or so, has been promoted towards mainframes (such as the IBM System/370) rather than being incorporated in the micros we all know and love.

So what is C?

"C", to quote its creator Dennis Ritchie, "is a general-purpose programming language which features economy of expression, modern control flow and data structures, and a rich set of operators." The aim of this discussion is to illustrate these features in more detail.

C is not a very "high level" language like BASIC, nor is it a very "low-level" language like Assembly Code. Instead, it falls somewhere between these two extremes. It is thus able to model the basic operators of most modern computers, yet can still remain machine independent. As such, C is a highly portable language, it is capable of being translated into efficient and compact code on a variety of machines.

C was originally designed for systems programming and was historically influenced by a language called BCPL. This was perfected at the Massachusetts Institute of Technology in the late 1960's. It had no concept of a variable 'type'; instead of integers, reals or strings, there was just the machine word.

In the early 1970's, Ken Thompson, an employee at Bell Labs, Murray Hill, had developed the first version of the UNIX operating system (in PDP 7 assembly language) and was trying to produce an interpretive version of BCPL, so that UNIX could be improved. The result was B.

The implementation of UNIX in B proved too difficult because the language did not cater for the individual bytes of the machine's 16-bit word. So types were added to B, to produce NB. Now bytes and words were described as separate types which allowed them to be referenced independently. Thompson's colleague, Dennis Ritchie, then began work on a compilor. The compilor was needed to produce the fast code essential for the efficient running of an operating system. UNIX was implemented in C in 1973. C has since survived five generations of UNIX, which has been developed for a variety of machines.

FUNCTIONS

C is a modular, block-structured language, that is composed of functions. The following example program consists of a function called **main** and prints as its output, **It works.** on one line:

```
# include <stdio,h>
main()
[
   printf ("It works. n");
]
```

The function **main** is the first function in every C program.

Functions are used in nearly all programming languages, and in much the same way. For example, in BASIC or Pascal, a function could be called to find the square of a number **n**, and assign this value to a variable **x**. Within a program, the function could be called with one of the following lines:

```
10 LET X=FNSQUARE(N) in BASIC
   X : = SQR(N) in Pascal
   x = sqr(n) in C (Note lower case).
```

In Pascal, the function SQR() is defined as part of the language, but in BASIC and C these functions do not exist and have to be defined by the programmer.

In BASIC, some preceding line would have to be included;

```
5 DEF FNSQUARE(N) = N * N
```

This DEFines the function, so that it 'returns' the square of its argument. (The argument is the number **N** in brackets.) The value returned is found by evaluating the expression that is to the right of the equals sign. (An expression is an algebraic statement, used to represent a value.)

In C, the definition would look like this:

```
float sqr (n)
float h;
/* function to return n/2 *
(
   return (n*n)
)
```

On the first line, the function **sqr** is defined as type float because it will represent a 'real or 'float'ing point number. Its only argument (in brackets) will be held in a variable **n** and used within the function body. When the function is called with a line such as:

```
x = sqr(5)
```

n will be initially set to the value 5 before the function is executed. When there is more than one variable name between the brackets, they are separated with commas.

ARGUMENT DECLARATIONS

After the function heading come the Argument Declarations. These describe the type of each argument. Each entry consists of a type-specifier, a list of variable names (between each name is a comma), and a final semicolon. In this example, there is one argument n, which will also be of type float when the function is called.

The next line is an optional comment. Comments begin with the characters / * and end with the characters * /. Comments cannot be nested, that is, once the compiler has read the opening characters of a comment, it will read the end of the comment as the next occurence of the closing characters. So, for example, the following would be rubbish:

```
/ * / * function to return n/2 * / *
```

The compiler would read the end of the comment as the first occurence of the closing characters and would then complain of an error when it encountered the next pair.

COMPOUND STATEMENTS

The body of a function is called a Compound Statement. A compound statement is a pair of braces, with any number of instructions between them. In the example this occupies the last three lines. The braces could be empty, and this would constitute a perfectly correct syntax, but a function that did nothing at all would be of little use to anyone!

The function body in my example contains a single statement, the **return** statement. It is used to terminate the function and return its argument as the final function value.

The general syntax of a function declaration can thus be summaried:

```
type — specifier function-name
(argument-list)
        argument-type-declarations
        compound statement
```

Although a valid C program could consist of the **main** function alone, it is good programming practice to delegate tasks to other functions. There is a whole programming philosophy towards breaking a program down, into simple, well defined tasks, so that its actions become more obvious. Human beings are only good at concentrating on a few things at a time, so a well broken-down program can be easier to read, easier to understand, and thus easier to work with. Such programs are said to be 'transparent', and details are confined to the functions at the lowest, most basic levels.

C was designed with these criteria in mind. All functions other than **main** are considered to be subordinates of the main function. For this reason, no function can be defined within another. The advantage of this is that libraries of functions can be included within the program and each function can be used by any other. This is another aspect of C's portability.

PRINTF AND SCANF

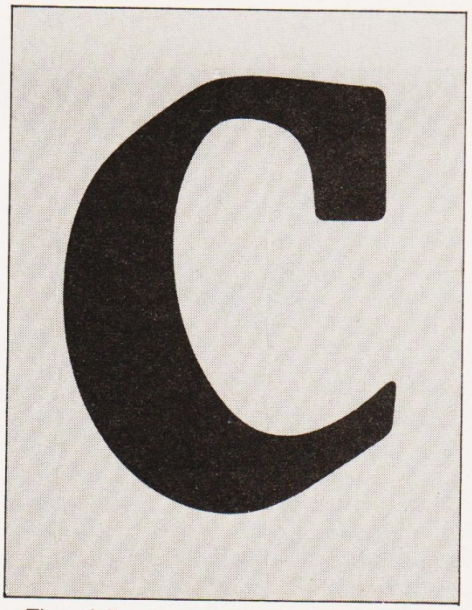
To demonstrate a complete C program, I have incorporated **sqr** into the following section of C code. The functions **printf** and **scanf** are for input and output respectively. These are declared in the standard input/output library which uses the header file **stdio.h.**

```
/* Squares */

/* Reads a real number and prints its
square */
# include <stdio.h>
main ()
(
   float input, output;
      printf ("Enter a number to be
squared      n");
      scanf ("%d", input);
      output : sqr(input);
      printf ("The square of %d is %f.",
      input,output);
   )
   float sqr (n)
   float n;
   /* function to return n  2 */
   (
      return (n*n);
   )
```

The first line contains a compiler directive, # **include,** which includes the specified file in the program. The file **stdio.h** contains function declarations for **printf** and **scanf.**

The first argument of **printf** is always a control 'string', it is made up of characters to be printed (and optional format specifiers that begin with a % sign) enclosed between quotes. Following the control string are expressions, or variables, which must correspond in sequence with the format specifiers.

The first **printf** in the above program contains no format specifiers (n is a special way of reepresenting a carriage return), so the string is simply printed at a terminal. However, the second **printf** contains two numeric format specifiers, which would cause the values of **input** and **output** to be displayed in those places.



The following format specifiers are allowed:

```
%c   character
%d   decimal integer
%o   octal integer
%h   hexadecimal integer
%f   floating point number
%s   string
%%   represents the % character
```

To left justify a number in **n** places, say **n** was 3, we would put the digit 3 between the % sign and the letter of the format specifier. To right justify, the digit will be preceded by a sign.

e.g. 2-4d

Print in decimal, with spaces on the left, to ensure a minimum width of four characters.

Scanf works in exactly the same way as **printf,** but expects characters as defined in the control string, assigning them to the variables listed after it. An asterisk is used between the % sign and the character of the format specifier, to define free format.

```
e.g. scanf ("%*d%*d",a,b)
would give a=2 and b=3 for any of the
data...
    2 2
  2 3
  2       3
```

Now try and work out **exactly** how the above program works.

VARIABLE DECLARATIONS

The following skeleton shows the format conventions within a C program:

```
/* global variable declarations */
main()
(
   /* local variable declarations to main
   */
)
function one()
(
   /* local variable declarations to
   function one */
)
function two()
(
   /* local variable declarations to
   function two */
)
```

A variable declared within a compound statement is 'local' to that compound statement and a variable declared outside any braces is 'global' to the program. A local variable can only be used within the compound statement in which it was declared, but a global variable can be used anywhere within the program. A program can be made more modular, its subprograms being more independent, if there are few global variables. This reduces the interdependence of modules, since variables are restricted in scope.

Notice that once a name has been used for a variable globally, it can not be used again locally, otherwise the compiler would get confused between the two variables. The same identifier can be used more than once however, if it is only declared locally in different functions.

The declaration itself takes the form shown in figure 1, of which we will learn more about later.

SIMPLE TYPES

Float is one of C's simple types. C has a variety of these types including **integers** and **chars.**

Integers can be used by a programmer to represent any ordinal type; in C these

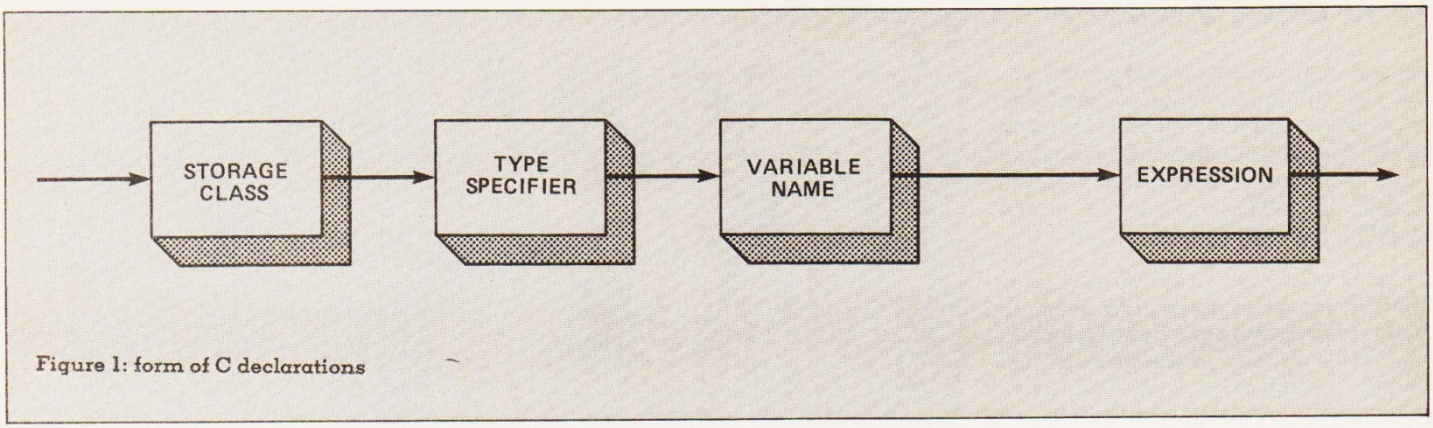


Figure 1: form of C declarations

include Boolean states, whole numbers, and as a special case, characters. There are essentially three types of integer, which vary only in the amount of storage they use. Typically, the following declaration:

```
int i;
```

would allocate an amount of memory equivalent to the size of the processor word, to an integer variable labelled **i**.

If the declaration was preceded by the word **short**, then this is a directive to the compiler to indicate that the range of values an integer can represent will be limited. This is used to save memory.

If the declaration was lead by the word **long,** then this implies that an extended range is required to represent large integers.

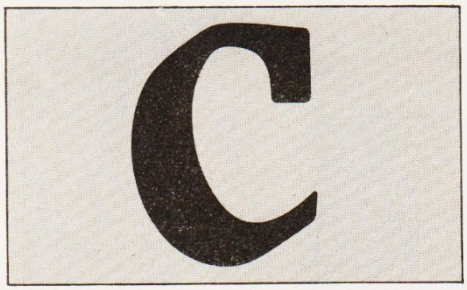
The way in which these various options are interpreted by the compiler is very much dependent on the particular machine. Depending on the size of the processor word, it is likely that either **long** or short will have no effect. For example, if the processor word was long, say 32 bits, then the use of **long** may be redundant on that machine. This is because a long integer is of the order of 32 bits and it is already provided for by the standard **int,** which is as long as the machine word. Conversely, a program compiled on another machine with an 8-bit word, may not provide the **short** facility. Compilers, however, have been designed fairly sensibly so that reasonable ranges are provided for integers, so that they match their declarations. But there are no hard and fast rules.

One other option, that can be used when specifying an integer, is the **unsigned** option. Normally, an integer will be represented within a machine using two's complement notation. This implies that one bit of the integer word is required to determine its sign (+ or −). By declaring an integer as unsigned, the normal range of positive integers can be doubled, (at the cost of having no negative integers). Unsigned integers are also used to simplify bitwise operations.

The **character** type is essentially a very short integer. It is available independentlyu because a character code will only take up a few bits and a separate type will allow for memory economies. To represent strings the character type is then used in an array form.

Float numbers have two levels of precision. To complicate matters further, each level has two different names. Single precision float numbers are normally represented in a single machine word, whereas double precision real numbers require an amount of memory equivalent to two machine words. A single precision float declaration will have either **float** or **short double** as the type specifier. A double precision float variable will have **double** or **long float** as the type specifier.

Float numbers can be represented in scientific notation. Scientific notation of the form **mEe,** where **m** and **e** are numbers, is interpreted as meaning **m** * 10↑**e**. So, for example, to represent the number 432.9674 in scientific notation, one could write 4.329674 E-2. Fractional **e**'s are not allowed.



STORAGE CLASSES

Variables in a C program can also be defined to be of a certain storage class as well as of a particular type. The main storage classes are **extern, static, auto** and **register.** These constitute an additional way of describing the scope of a variable.

Global variables, i.e. variables declared outside functions, are **extern** by default. An **extern** variable is external to the program, it is either shared by another program or it is a system variable. A global variable can otherwise be given the **static** attribute. In the global context, a **static** variable is 'visible' throughout the remainder of the enclosing program. In this sense, it is a means of restricting the scope of a variable to part of a program.

Local variables can be given any of the storage classes.

The **auto** or automatic attribute is the default. Each time a compound statement is executed, in which an **auto** variable is declared, the variable will be reset to the value of the initialiser, or some undefined value if none is given.

The **register** attribute works in exactly the same way as **auto**, except that the compiler will keep the variable in a high speed register (if one is available). This allows commonly used variables to be accessed quicker and thus the resulting program will run faster.

The **static** attribute in this sense, allows variables to hold their values between successive executions of the compound statement. This is particularly useful in maintaining values between function calls which are only used by that function, and so avoids having to unnecessarily declare the variable as global. One example of such a function would be a random number generator, which calculates the next random number in a mathematical sequence from a variable called the seed. Each time the function is called, it calculates the value of the next random number from the seed, and then updates it for the next number in sequence. Normally, the seed would have to be global, but the **static** attribute offered by C offers a much more 'elegant' solution, since the seed remains local to the function in which it is used.

The **external** attribute has the same meaning as in the global context, but it allows the compiler to restrict the use of a particular external variable to a compound statement. Note that extern variables cannot be initialised locally.

SUMMARY

To summarise so far, we have been introduced to the concept of a function — a module that is independent in its own right. We have seen how functions are used as building blocks to construct a program, and we have learnt about the special function main. **We have also looked at the major topic of variable declaration, noticing that a variable can either be global to the whole program or local to a function. The storage classes are used to expand on this concept. The solely global ones — extern and static, and those which can additionally be used locally —** auto **and** register. **We have also seen C's simple types, in all** char, short, unsigned, int, long, float **and** double.

BOOK PAGE

Garry Marshall

Four books for the Apple are the raw material for our reviewer this month.

Having just acquired an Apple IIc, I was naturally delighted to find a number of Apple books awaiting review this month. As we all know, Apple has been around for a long time, and this makes it difficult to come up with something new in the book line. The IIc is new, and it features in the titles of some of this month's books, but it is usually there as an afterthought and then only by courtesy of the compatability of the Apple range.

Another factor, with the IIc at least, is that it comes with a very good set of manuals and, even more importantly, with superb training software to introduce the user to the computer and its uses. As Simon Dismore remarked in his review of the IIc (*Computing Today*, November 1984) Apple has retained an image of goodwill, one of striving for perfection.

The computer speaks for itself, but the software that is supplied with it is strong evidence of Apple's striving to please. This makes it more difficult than usual for the writers of books. There is absolutely no need for manual replacements. And to claim, as does the blurb on the cover of one of the books under review, that 'you're a little unsure of what (the computer) can really really do for you. You tried read the manual that came with the computer and that only got you more confused.' is just plain nonsense.

So, my approach to the books is to see if what they have to offer actually works on the IIc, especially as those that mentioned it in their titles clearly did so as an afterthought, and to look for something a bit different and special in their presentation and content, for to make the publication of yet more Apple books worthwhile at this stage, they must have something out of the ordinary to offer.

Sound and graphics by Jerry and Valerie Abad scores at once with its attractive appearance and production. Also, by concentrating on two of the most attractive features of any micro, it guarantees its appeal. But the book also brings a 'building block' approach to the development of skills in the creation of graphics and sounds with the computer that genuinely requires no prior knowledge. It shows how to create perspective drawings in a simple way that anyone can follow. No perspective formulae here, just common sense. The approach to graphics is developed in such a simple way and yet you feel that you are learning about graphics by using a computer and not, as so often, learning about how a particular computer works when it creates graphics, but in such a way that you will have to start all over again if you use a different kind of computer.

The approach to sound is not quite so friendly, through no fault of the book, but because POKE has to come into play straight away. This problem is circumvented by the presentation of programs that the reader is invited to try, and then to adapt and amend. Music and sound effects are introduced, followed by the small amount of music theory that is necessary to make any progress at all.

The computers mentioned in the title of the book are the II+, IIe and Franklin, but the graphics and sound programs that I tried on the IIc all worked. Anyone wanting a low-level introduction to graphics and sound in the genuine Apple tradition of friendly education and self-improvement could do no better than this.

A byte of the Apple is a guide to help the beginner become familiar with the Apple. Its contents are pretty familiar. It starts by introducing the hardware and the jargon so that the beginner can get the computer up and running. It covers topics such as the keyboard layout, disks, DOS and care of disks. It is clearly American in its concentration on disk as the storage medium: cassettes scarcely rate a mention. But all of this is covered at least as well in the Apple documentation, and a dynamic introduction to the keyboard from a teaching program can never be matched by a book. The second section of the book introduces BASIC programming through a series of 'skill' frames. Each one introduces some feature of BASIC and demonstrates its use in a short program. None of these frames eeemed particularly penetrating or interesting, and the section just amounts to an uninspired catalogue of BASIC features.

After this, we get a section on the uses of the computer in education. This amounts to a list of seven educational computer uses, taking up all of one page. The final section of the book is for reference. All in all, it is rather hard to see any reason for publishing this sort of thing.

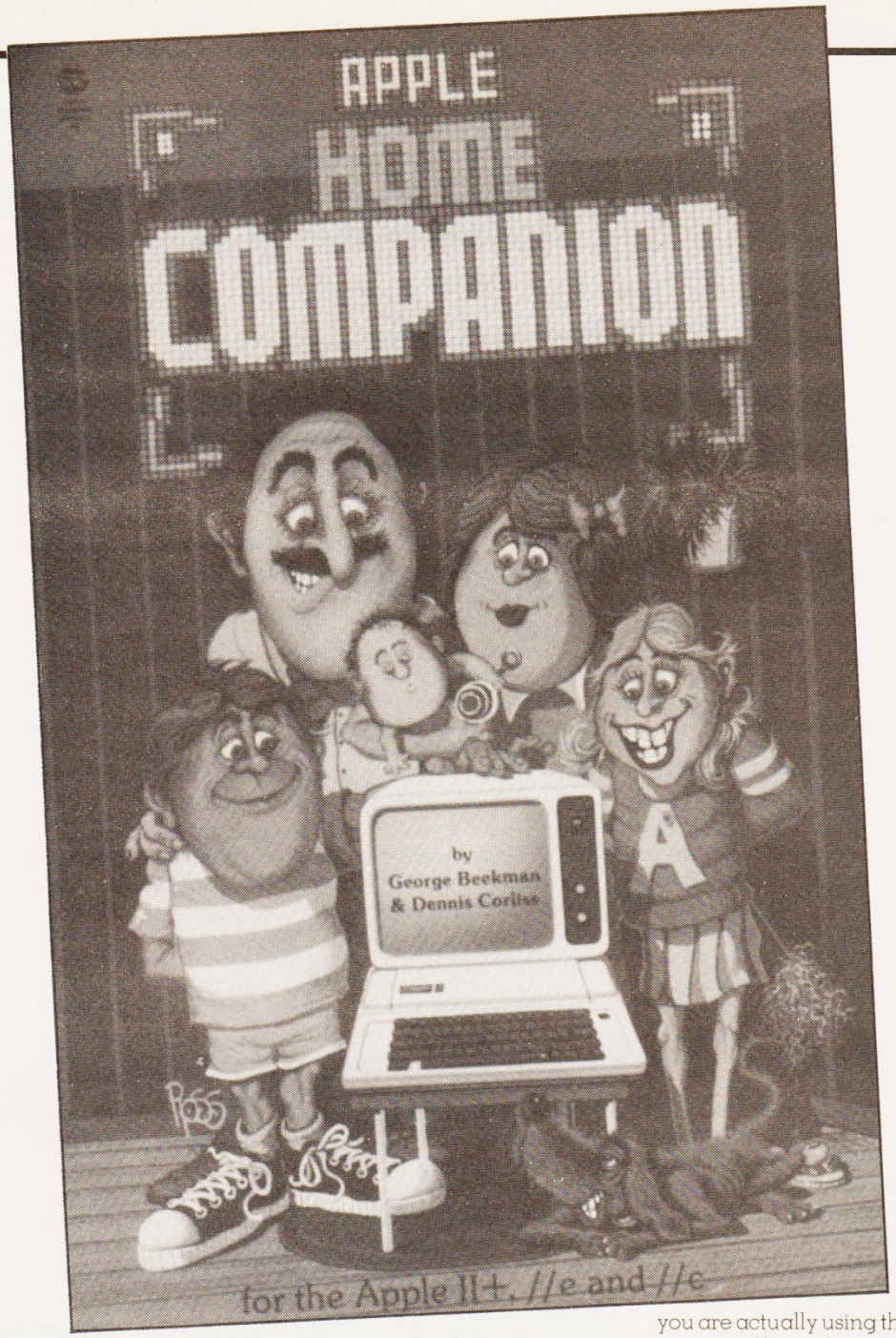
Programming tips and techniques for the Apple II and IIe provides just what its title would lead you to expect. It goes into some topics in con-

siderable depth, and really does contain useful and unusual routines and techniques. It also presents a rather complete collection of reference material on, for example, memory maps and the locations of machine code routines.

A familiarity with BASIC is taken for granted, and its first dealings with programming start with programming methodologies, program structures and writing menu-driven programs. After this it deals with matters such as setting up a clean machine, trapping errors and making the input of data secure. These are quite advanced considerations, but they must be faced and performed satisfactorily by any program that is to be useful as a widely used applications program. Subsequent chapters cover in turn input, processing, output, sound and graphics, and DOS. Each presents a high-level coverage and includes many useful ideas and programs. Again, all the programs that I tried, including some for graphics and sound, worked on the IIc.

Apple home companion by Beekman and Corliss is for the II+, IIe and IIc. Its angle is that it is written for friends of the authors who were not at all familiar with computers, but who wanted to know what they were and what they could do. It succeeds on this level, providing a general, jargon-free account of what a computer is and of what it can do. It attempts to introduce BASIC and BASIC programming at a very low level. The treatment's success may well depend on how you react to the authors' sense of humour. A sample may show you what their humour is like and also indicate the level of treatment. Their first 'BASIC rule of law and order' is that 'when a program is run, the statements are executed one at a time in numerical order'. The second is: 'There are ways of getting round the first rule.' But, humour aside, the account of BASIC is rather ordinary.

The book is much more successful when it moves on to giving an account of what an Apple computer can be made to do by running the appropriate software. By giving accounts of what specific Apple programs can do, it shows very clearly the range of uses for the computers. The list of items of software has an American bias, but much of it is available here. It includes word processors, databases and spreadsheets, and I can endorse its good opinion of the integrated 'Appleworks' package. Games and educational programs are described as well, and 'Rocky's boots' seems to be attracting a lot of attention as the sort of educational program that is at last beginning to take the micro beyond computer-assisted teaching.



The cartoons are very lively and amusing, and I was particularly struck by the one showing a young lady in a computer store being advised by a serpent to take the Apple.

So out of the four books, three have managed a distinctive contribution of their own and, by and large, each contribution is in the Apple tradition. After reading the books, though, I thought that it might be interesting to run some of Apple's instructional software for the IIC just to make the comparison with the books. I'm glad that I did, because Apple's 'An introduction to the IIc' and 'Getting down to BASIC' proved much more fun than the books and, not surprisingly in view of the fact that you are actually using the computer itself, much more effective in introducing the computer and its uses. Of course, 'Getting down to BASIC' won't give you the range of examples and ideas that is contained in the almost 400 pages of 'Programming tips and techniques', but it is unbeatable as an introduction and for any other aspects that do not require deep study.

This month's books are:

Sound and graphics by Jerry and Valeria Abad (Datamost) 141 pages.
A byte of the Apple by L Hyler, S Crowther and M Eltschinger (Reston) 145 pages, £14.50.
Programming tips for the Apple II and IIe by John L Campbell (Prentice-Hall) 403 pages, £19.35.
Apple home companion by G Beekman and D Corliss (Prentice-Hall) 355 pages, £19.35.

INSIDE CP/M

Don Thomasson

Continuing our 'INSIDE' series, we turn to CP/M, one of the most widely applicable disc-based operating systems.

Originally created in about 1975 for use with an 8080-based Intel Microprocessor Development system, CP/M is now implemented on a number of machines using Z80 and 8085 processors, allowing them to run a vast library of programs written to the CP/M standard.

Reviled by some, CP/M goes further than most operating systems towards the ideal of totally compatible microcomputers. Its importance is illustrated by the fact that more than one computer based on a processor other than the Z80 has been extended to include a Z80, almost entirely for the sake of achieving compatibility with CP/M.

Over the years, the standardisation of CP/M has been eroded by changes necessitated by machines designed in an unsuitable way, but the broad principles have remained intact. The present analysis is based on a comparatively early version, which is nearer to the original concept than some of its successors.

THE LAYER CAKE

The broad structure of CP/M is like a three-layer cake. The user communicates with the top layer, the Console Command Processor (CCP). That passes appropriate instructions to the Basic Disc Operating System (BDOS), which calls up sections of the Basic I/O System (BIOS) which in turn talks to the main computer operating system.

To avoid confusion over terminology, it should be said that "Console" means keyboard and display, while a printer is termed a "List Device", and a cassette recorder is a "Punch" and "Reader". These terms reflect the changes in the small computer world that have taken place during the past decade.

The way the layer cake works can be illustrated by a simple example. The CCP wishes to send a character code to the display. It puts the code in the E register and calls BDOS with 2 in the C register. BDOS responds by checking whether control action (e.g. TAB) is involved, and performs any such action. Otherwise, it transfers the code to the C register and calls the Console Output (CONOUT) function of BIOS, which passes the code to the main operating system.

This may seem unnecessarily complex, but there are good reasons for it. The CCP is standardised, presenting the same interface to application programs and user control in all implementations. (Though it will be seen that there are

TABLE I BIOS functions

Entry Address	Name	Function
BASE	BOOT	Cold Boot
BASE + 3	WBOOT	Warm Boot
BASE + 6	CONST	Console Status
BASE + 9	CONIN	Console Input
BASE + 12	CONOUT	Console Output
BASE + 15	LIST	Output to printer
BASE + 18	PUNCH	Output to cassette
BASE + 21	READER	Input from cassette
BASE + 24	HOME	Home drive head
BASE + 27	SELDISC	Select disc drive
BASE + 30	SETTRAC	Select track
BASE + 33	SETSECT	Select Sector
BASE + 36	SETDMA	Set DMA Address
BASE + 39	READ	Read a record from disc
BASE + 42	WRITE	Write a record to disc.

TABLE II BDOS functions

Key	Function	Data Out	Returned Data
0	System Reset	—	—
1	Read Console	—	Code in A
2	Write Console	Code in E	—
3	Read Reader	—	Code in A
4	Write Punch	Code in E	—
5	Write List	Code in E	—
6	Undefined	—	BDOS entry in A/B
7	Interrogate I/O Status	—	Status byte in A
8	Change I/O Status	Status in E	—
9	Print Console Buffer	Buffer addr in DE	—
10	Read Console Buffer	Buffer addr in DE	—
11	Console Status	—	Status in A
12	Lift Disc Head	—	—
13	Reset Disc System	—	—
14	Select Disc	disc number in E	—
15	Open File	FCB addr in DE	Completion code in A
16	Close File	FCB addr in DE	Completion code in A
17	Search First	FCB addr in DE	Completion code in A
18	Delete File	FCB addr in DE	Completion code in A
19	Read Record	FCB addr in DE	Completion code in A
20	Write Record	FCB addr in DE	Completion code in A
21	Create File	FCB addr in DE	Completion code in A
22	Rename File	FCB addr in DE	Completion code in A
23	Interrogate Login	FCB addr in DE	Completion code in A
24	Interrogate disc	—	Disc number in A
25	Set DMA address	—	—
26	Interrogate allocation	DMA Address	Buffer address in A/B
27	Set Read Only	—	—
28	Undefined	—	Read Only vector in A
29	Set DMA address	—	—
30		—	

exceptions to this in practice.) The BDOS may need to be adusted to suit different disc formats, but will retain the same broadly standard form. The BIOS, on the other hand, is tailored to suit a particular machine, and this completes the path from standard to special.

THE BIOS

The BIOS has fifteen entry points, each relating to a particular function, as shown in Table 1. The entries are arranged in a jumpblock at the start of the BIOS, their order and relative addresses being a part of the system standard.

Each entry 'jumps' to an appropriate routine, which may be trivial or complex. Reference has been made to the CONOUT function, the fifth in sequence. This may jump directly to a main operating system entry, or may need to modify the code to be displayed if the machine uses a non-standard character set (CP/M assumes that ASCII is used).

The READ function, on the other hand, is much more complex. It has to read a record from disc, copying the data into a predefined area from a given track and sector, which are also predefined. The data must be checked for read errors, and some general housekeeping is also involved. If the main operating system provides disc control, that may be used, but it is more likely that CP/M will do the whole job, since that simplifies the provision of the standard error checks.

The start of the area in which the data is to be stored is called the 'DMA Address'. DMA stands for Direct Memory Access, which is a method of transferring data to and from memory without using the central processor. The external system puts up a BUS REQUEST (BUSRQ), and the processor responds by releasing its control of the main system bus as soon as it has finished executing its current instruction. It then makes BUS ACKNOWLEDGE (BUSAK) true, and the external system can then take over the bus for the purpose of performing memory transfers. This process can be very fast, as there is not time spent in reading processor instructions, but the method has been little used of late, partly because of the difficulty of ensuring that the process is accurately synchronised with the disc data transfers, especially if long-duration instructions are in use. The term 'DMA address' nevertheless lingers.

A particular feature of the version of CP/M being examined here is the provision for 'cache buffers'. When a record is read in, it is stored in a cache buffer. If the system is instructed to change the record, a change is made in the buffer, but no disc transfers are made. Several records can be stored in this way, and disc transfers are only needed if the relevant record is not in the cache. The oldest cache record is then written to disc before the required record is read out. This saves a lot of transfers, and as each cache record is tagged with its track and sector number, housekeeping is simplified.

THE BOOT FUNCTION

The word BOOT is an interesting example of jargon corruption. In the earliest days of computers, it was axiomatic that no program could be loaded until a loading program had been loaded. The resulting impasse was solved at first by the laborious entry of a simple loading program by means of switches. When a form of automatic initial program load was introduced, someone remarked that it was as if the computer was lifting itself by pulling on its own bootlaces. (In America, the word used was bootstraps.) People began to speak of 'botting' a system and BOOT is the residual of that.

For CP/M, BOOT involves the use of a small ROM holding the initial loader. The ROM is usually superimposed on a RAM area, and is only enabled when the actual BOOT process is taking place. The main program is read from disc, where it occupies a special position, and all is then ready for work to begin.

This process is called 'Cold Boot', and it resets all the system functions and variables. There is an alternative, 'Warm Boot', which is executed by the BIOS. This reloads the CCP and BDOS from disc, but does not destroy all system variables.

OTHER BIOS FUNCTIONS

A simple BIOS need not be too large. In some cases, the whole set of routines is held in ROM, so that there is no need for a loading ROM. About 2000 locations suffice for a BIOS with cache buffers, not including the buffers themselves.

Some of the routines are almost negligible. CONSTAT must return with the A register holding zero if no key is pressed, otherwise returning &FF. On one recent machine, this involves no more than two subroutine calls, followed by SBC A,A,. Other machines need more assistance.

CONIN is normally called only when CONSTAT has shown that there is a character code waiting, but it does not return until a key is pressed, and may therefore be used as a holding function. It returns the character code in the A register, with bit 7 zeroed. CP/M sees this bit as a parity check.

CONOUT, LIST, PUNCH and READER are usually related to main operating system entries. Characters for output are held in the C register, input characters in A.

HOME (the disc head) needs no parameters. SELDISC selects the drive specified in the C register, which also carries the parameter to be used by

SETTRAC and SETSECT. For SETDMA, the address is held in BC. READ and WRITE use parameters already set.

A point to note is that the CCP and BDOS were written in the original 8080 code, but for a Z80 machine the BIOS normally takes advantage of the more efficient Z80 code, and this allows some economy of space.

BDOS

Like the BIOS, BDOS implements a number of functions, but there is only one entry point, the required function being determined by the contents of the C register, as shown in Table II.

Data is passed to BDOS in the DE register, and data is returned in the A register (Low byte) and the B register (High byte, if any).

At first sight, some of the BDOS and BIOS functions look rather similar, but that is deceptive. Subtle differences can involve a lot of code. For example, Read Console does rather more than call CONSTAT and then call CONIN if A = 1. A TAB command code is looked for, and if one is found the cursor is advanced (by output of spaces) until a column is reached that is a multiple of eight. There is also provision for holding a character code so that it can be sent on later. If the input code is Control S, system action stops until another key is pressed, but the identity of the key is irrelevant. This function, all in all, is unexpectedly complex.

It is nothing like as complex as Read Console Buffer, however. This is used to set up a command line for subsequent use by the CCP, or by a 'transient' program, a term which will be explained later. There are a number of special control characters that are ignored by Read Console but which are active when a buffer is being set up. (See Table III).

This is one of the areas where CP/M is unpopular, mainly because of the erase function, which deletes a code in the buffer, but echoes it to the display, making the command line difficult to read. This was done because teleprinters had no backspace function, so there was no point in trying to erase what had been printed. The best that could be done was to indicate that erasure had occurred in the buffer.

Note, however, that Control R will produce a fresh display of a corrupted line, erasure duplications being eliminated. The previous displayed line is terminated with a hash symbol, to show that it was not used. Control U and Control X clear the buffer and start a new line from scratch, again marking the previous line with a hash symbol.

When the line is complete, Carriage Return (ENTER) will send the stored command sequence to the CCP for attention. The buffer, by the way, is physically within the CCP store area, even though its setting up is controlled by BDOS.

DISC CONTROL

BDOS also incorporates the fundamental disc control routines, which must be adaptable to different types of drive. This is largely achieved by a database specifying the drive characteristics. The BIOS also has to be tailored to the type of drive in use.

The details of disc access are controlled by a File Control Block, which is a 33-byte area defined thus:

Byte 0	Entry Type
Bytes 1-8	Filename
Bytes 9-11	File Type
Byte 12	File extent
Bytes 13-14	Zeroes
Byte 15	Record count
Bytes 16-31	Disc Allocation map
Byte 32	Next record

Two default FCB areas are provided. The user, or his program, have the responsibility for supplying bytes 0 to 14, the rest being filled in from disc, on the basis of the directory entry matching the given filename and type. These, in turn, come from the command line. Application programs can set up other FCBs, and for each drive in use there is a 32-byte buffer and a 16-byte buffer. Full data on the file in use and the current record are thus available.

THE CCP

The CCP exercises overall control in response to the instructions given to it by the user. It embodies five 'resident functions', for which code is always available. They are:

DIR Display Director
ERA Erase File on disc
TYPE Display (or print) file
SAVE Create file on disc
REN Rename file on disc

Other functions can be implemented by reading the appropriate files from the disc. For example, PIP (Peripheral Exchange Program) comes into this category, though it might appear to the user to be identical in nature to DIR or ERA.

Such programs are called 'transient programs', since they are only held in store while their services are required. Any program written to the correct CP/M rules and held on a disc compatible with the system in use can be loaded and run on a transient basis.

THE MEMORY MAP

CP/M programs are originally loaded on to discs in such a way that they can be relocated, using the SYSGEN function. This allows considerable flexibility, but the usual approach is to arrange the programs with the BIOS as high as available RAM will permit, BDOS below it and the CCP below that. The transient area lies below the CCP.

But that is not the whole story. Since the routines are relocatable, it is necessary to provide access jumps in RAM which will always be in the same place. The original arrangement was:

000-2	Jump to BIOS Warm Start
0003	I/O Status Byte
0004	Current Drive Number
0005-7	Jump to BDOS entry

This fitted well with older systems, which had RAM in this area. Later systems, noting that program execution after reset always started at location 0000, put ROM at the bottom of the memory map, which upset the Applecard. There was no need for this, since the start address can be changed quite easily by a hardware modification, but ROM at the bottom became a fashion. It also extended up into the transient area, which used to begin at 0100, and that meant relocation of all transient programs.

Some designers have left that situation unchanged, doing sad damage to the universal concept of CP/M. Others have found ways out. One provided an entirely separate 64K of store in which CP/M could run as it wished, but that raised some problems of communication with other store banks. Triumph-Adler provided both ROM and RAM from 0000 upwards, then continuous RAM. By working from a point common to both memory types, it was possible to satisfy the original needs of CP/M. Where a second (Z80) processor was used, with its own memory, the problem did not need to arise.

It could be argued that the original CP/M concept was the source of these troubles, because there are very good reasons for having ROM at the bottom of the address range. The Z80 RST instructions access the area 0000-003F, and the NMI entry is at 0066. Code is needed here from the moment of switch-on. So it is, and the AMSTRAD CPC464 provides it in both ROM and RAM, copying the RAM from the ROM. If these entries are to be used, they can be RAM without harm, providing a simple copying action is executed early in initialisation.

CONCLUSION

The original standardisation concept of CP/M has become a little blown-upon by this conflict of ideas regarding memory usage, but the concept itself remains valid. Versions for the 8086 and 68000 are now in being, but they are not code-compatible with the original. Now, if the Z800 processor materialises, we might see a justification for tidying up the original CP/M, using code that would run on either Z80 or Z800.

TABLE III Control characters

Control C	&03	Warm Boot
Control E	&05	Start new line
Control P	&10	Toggle printer flag
Control R	&12	Display cleaned line
Control S	&13	Hold action
Control U	&15	Abort command line
Control X	&18	As Control U
Control Z	&1A	End console input
ENTER/CR	&0D	Send line to CCP
Erase	&7F	Delete from buffer.

TWO PASCALS

David Scott

If you're considering a change to Pascal, take a look at one of the versions reviewed here, from Hisoft and Grey Matter.

Hisoft Pascal is one of the cheaper CP/M Pascal's that are on the market today. However, although this Pascal is quite cheap at £46.00, it offers most of the facility's of standard Pascal as stated by Jenson and Wirth, as well as several important additions to take advantage of the CP/M operating system.

When you receive your disk you get the program, HP4D16.COM, which is the Pascal compiler. It is up to the individual to use their own CP/M compatible editor such as Ed or Wordstar. Also supplied is a 53 page A4 sizes manual which describes the major Pascal keywords as well as all the non standard keywords in Hisoft's version.

Although the manual is quite small compared to the more expensive Pascals, it has sufficient information to get someone with little knowledge of Pascal, started. This is good in that it contains sufficient technical information for the experienced user to start straight away. Topics covered in the manual are syntax and semantics, predefined identifiers, error compiler options and data representation. At the end of the manual are four sample programs, each demonstrating some different aspects of Hisoft Pascal.

ADDITIONAL COMMANDS

The Pascal offered by Hisoft has the following additional commands to take advantage of the CP/M operating system: INLINE (C1,C2...),HALT,POKE(X,V),USER(V), CPM(V1,V2),PEEK(X,T),INCH as well as ADDR(V). The commands PEEK and POKE are the same as in BASIC; INCH scans the keyboard to see if any key has been pressed; HALT stops program execution; INLINE is used to insert machine code routines into the compiled pascal; USER is equivalent to BASIC's CALL command in that it starts execution of a machine code routine at the called address; ADDR returns an integer value, which is the address of variable V in the command; CPM is used to select the CP/M functions, the command V1 is the function of which there are about 40 different functions available for use, V2 is the call to that function which can be used to select drive numbers etc.

Hisoft Pascal also includes compiler options which, when used correctly can greatly speed up the operation of the compiled program. However, unless these options are used correctly, program execution speed can also be greatly reduced.

In conclusion, Hisoft Pascal is good for the price, but is let down through lack of facilities when competing the more expensive Pascals or even Turbo Pascal which is reviewed below.

FACTSHEET	Hisoft Pascal
Price	£46.00 + VAT
Available from	Hisoft 13 Gooseacre Cheddington Leighton Buzzard Bedfordshire LU7 0SR (0592) 382604

TURBO PASCAL

Turbo Pascal is another of the cheaper Pascals available on the market today, priced at £45.00 + VAT for version 2.0. I have version 1.0 which was obtained on special offer for £35.00 + VAT due to the release of version 2.0. However, it offers several important facilities that Hisoft Pascal lacks. Therefore, let us take a more detailed look at the facilities available under Turbo Pascal.

When you receive your disk you get the program detailed in Table 1.

So, you certainly get considerably ***more*** programs than with Hisoft, but you also receive a 250 page reference manual whih includes details of Turbo Pascal and how it operates under CP/M, as well as details of how it operates under CP/M-86 and MS-DOS. The manual also includes details of how to install Turbo Pascal using the program TINST.COM. Turbo Pascal also includes it's very own operating system, which includes a full screen, Wordstar-compatible, editor. This operating system includes the following commands which are used by pressing the first letter of the given command. The commands are: L for selecting logged on drive; W to select work file; M to select main work file; E to edit file; C to compile; S to save a text file; X to execute a .COM program (only works if TURBO.OVR is present); D gives the directory of the

FACTSHEET	Turbo Pascal
Price	£45.00 + VAT (for version 2.0)
Available from	Grey Matter Ltd. 4 Prigg Meadow Ashburton Devon TQ13 7DF (0364) 53499

TABLE 1
Programs forming the Turbo Pascal package

TURBO.COM	The Pascal compiler
TURBO.OVR	Overlay program, use to execute .COM files
TURBOMSG.OVR	Text file containing error messages (possible to change)
TLIST.COM	Source text listing program
TINST.COM	Terminal and command installation program
TINST.DTA	Terminal installation data
TINSTMSG.OVR	Text file containing messages for TINST.COM
.PAS files	Demonstration spreadsheet program (to be compiled)
ERROR.DOC	Documentation, containing updates on programs and latest errors or omissions.

logged on drive (can be used with wildcards as in CP/M); Q is to quit Turbo Pascal and enter CP/M; O is to select the compiler options.

The Turbo Pascal package also includes a spreadsheet program in Turbo Pascal source-code for you to compile which is very good, as it demonstrates a good number of Turbo Pascal's facilities, and as it is in source-code you can always refer back to it should you have any problems while writing a Pascal program.

Turbo Pascal,like Hisoft Pascal, has included extra commands/procedures to assist in program development. Unlike Hisoft Pascal however, Turbo Pascal includes commands that are similar to BASIC, the advantage being that the programmer need not develop his own routines to do such things as finding a string length, clearing the screen or placing the cursor anywhere on the screen. Turbo Pascal includes commands for these, as well as most of the commands you would meet in BASIC. Of course, the user does not have to use these commands if he does not want to.

Also, you can if you wish, give procedures the names of commands that are already part of Turbo Pascal. However, the original command will no longer work as before.

Also included are commands to access CP/M for random access; serial access; renaming files; finding a file size as well as deleting files. Turbo Pascal does not possess the commands PUT and GET, but has incorporated them into the READ and WRITE commands by having an extension such as KBD, SCR, PTR where KBD scans the keyboard; SCR directs output to the screen (this is the default) and PTR directs output to a printer. This is a very good feature as it means that Turbo Pascal can access any device connected to any CP/M computer by treating it as a device through software control e.g. PTR for PRINTER etc.

Like Hisoft Pascal, Turbo Pascal includes a compiler option which if used correctly can greatly speed up program execution. Turbo Pascal is very good and superior to Hisoft Pascal in terms of the extra facilities that are offered as standard. It can also be compared directly to many of the more expensive Pascals, since Turbo Pascal has incorporated within it, its own type of operating system with its own set of commands, and of course, the Wordstar compatible editor.

CONCLUSION

Both Hisoft and Turbo Pascal offer good value for money, and I would recommend them to anyone who is interested in ***learning*** Pascal. However, Turbo Pascal has several facilities that should be considered if you wish to use Pascal to develop commercial or professional programs, such as the built in, full-screen editor (not as good as the New-Brain's) that is Wordstar compatible or the extra functions and procedures that provide some facilities offered from BASIC, which should make the transition from BASIC to Pascal a lot easier than with Hisoft's version.

BBC and Electron

OXFORD PASCAL

Oxford Computer Systems (Software) Ltd.
Hensington Road, Woodstock, Oxford OX7 1JR, England
Telephone (0993) 812700 Telex 83147 Ref. OCSL

Compilers like these don't grow on trees

Oxford Pascal is Fast

Oxford Pascal compiles down to FAST COMPACT P-code, giving you the real speed and power of Pascal, together with the ability to compile very large programs.

Oxford Pascal is Standard

Oxford Pascal is a full extended implementation of Standard ISO Pascal. This means that you can compile any Pascal program (subject to size), written on any computer, anywhere.

Oxford Pascal is Compact

Because it compiles into P-code, Oxford Pascal reduces programs into the most compact form possible. In fact it allows you to pack more code into your BEEB than any other language, and should your programs become too large, you can still use the CHAIN command to overlay limitless additional programs without losing data.

Graphics & Sound Extensions

In addition to the entire Pascal language, Oxford Pascal features a whole range of Graphics (all modes) and sound extensions designed to make maximum use of the BBC Computer. Oxford Pascal also provides numerous extensions such as hexadecimal arithmetic and bit manipulation instructions.

Oxford Pascal in Education

In Education, Oxford Pascal is fast becoming a *de facto* standard. It is already the most popular Pascal on the Commodore 64, and will soon be released for the Spectrum and the Amstrad. In fact, Oxford Pascal will soon be available for 90% of the computers installed in the U.K., and is already available in German, French, Swedish, and American versions. **Students and teachers alike find that it makes sense to use a standard implementation of Pascal across the whole range of educational micros.** Call us for details of our generous educational discounts.

Manual

Both these compilers come with a manual which has been carefully designed, not only as a quick reference guide, but also as a full tutorial for those new to Pascal.

Resident and Disc Compiler

Oxford Pascal comes in two forms:

For Tape Users...Oxford Resident Pascal. A compiler located largely in ROM which is available at any time. Programs can be written and compiled on the spot without disc or tape access, and compilation is fast enough to make using the compiler much like using the BASIC interpreter. Thus, learning Pascal is a simple interactive process. Some 15K of memory is available for user programs, the remainder being reserved for compiled object code.

For Disc Users...Oxford Disc Pascal offers all the above PLUS...a full disc compiler which is capable of using the WHOLE memory for Pascal object code, it is supplied with a powerful LINKER, allowing you to break large programming tasks down into separately compilable, easily-manageable files.

Friendly Error Messages

Many compilers produce little more than an error and line number to help correct mistakes in Pascal programs. Oxford Pascal however, gives you one of 49 friendly and informative error messages. Messages which not only indicate the reason for an error, but also print out the line in question with a pointer to the exact position where the error was detected.
Run-time errors are reported using line-numbers from the original source-program, with a full explanation of how the error occurred.

Powerful Editor

With Oxford Pascal there is no need for you to learn how to use a new Editor. Pascal programs can be entered in exactly the same way as BASIC programs, without the need to learn any new commands. When you are used to using Pascal, you will find our extensions to the Standard Editor even more useful. What is more, Oxford Pascal allows you to mix BASIC and Pascal together, in much the same way that you can mix BASIC and assembler. In fact you can, if required, mix all three together...BASIC, Pascal and assembler...in one program.

Stand Alone Code

Unlike other compilers, Oxford Disc Pascal allows you to compile on the BBC and then relocate your program so that it will run on the BBC **and** on the Electron. The relocated program will run without a Pascal **ROM** and can be loaded and run from tape or disc just like any other program.

This means that you can distribute or sell your software freely and without the need for **ROMs**, to run either of the above machines.

Price/availability matrix

	BBC 'B'	ELECTRON	C64	SPECTRUM
DISC	£49.95	Not yet!	£49.95	Available April 1st 1985
CASSETTE	£39.95	£59.95 inc. Cartridge	£22.95	

All prices are inclusive of VAT
Please add £2.00 for postage and packing.

Oxford Compilers — The Future

During the next year, we at Oxford will be releasing a series of language implementations such as C, and Modular 2, for the BBC, and other popular micros.

These compilers are being built, using the most modern techniques in **automated** compiler construction, and will bring to the micro-user, a level of robustness and efficiency, only now becoming available to mini and mainframe users.

Oxford...
the **Compiler Compilers.**

Oxford Pascal Order form. Please make cheques payable to OCSS Ltd.
Please rush me my copy of Oxford Pascal. I enclose £..... including £2 postage and packing
My computer is BBC 'B'☐ ELECTRON☐ C64☐ (Please tick as appropriate)
I would like my compiler supplied on DISC☐ CASSETTE☐
Name .....
Address .....
Postcode .....
Telephone .....

RANDOM REFRESH

Bill Horne

A cautionary word for those tempted to generate random numbers using the Z80's Refresh Register.

Generating random numbers in a machine code program is not all that easy, and Z80 users may be tempted to seek a solution based on the contents of the Refresh Register. Before doing so, however, they should read the small print of the processor specification carefully, to make sure that they understand the implications.

The Refresh Register holds eight bits, which can be set from the accumulator contents by the instruction LD R,A. Thereafter bit 7 remains as set, while the binary number in bits 0-6 is incremented after each instruction fetch cycle.

REFRESH CYCLE

In the fetch operation, the processor reads the op-code during clock periods T1 and T2 of the four-clock machine cycle. The op-code must then be identified and implemented, and while this is happening during clock periods T3 and T4 the Refresh cycle is performed. The contents of the Refresh Register are put out on address lines A0 to A7, with the contents of the interrupt register on address lines A8 to A15, and the RFSH signal from the processor is pulled low to indicate that refresh of dynamic memory is to be executed.

That is the intended use of the Refresh Register, but it is possible to read the register contents by the instruction LD A,R. The result, in bits 0-6, will be a number in the 0-127 range. If the read is called at a moment dependent on some action outside the processing system, such as a keyboard action, the number obtained will vary at random.

For most purposes, however, a 0-127 range is excessive. If you want a random number between, say, 1 and 6, further processing is needed. You could use AND 7 to cut the range down to 0-7, and then reject results 0 and 7, looping back to repeat the LD A,R.

You could do that, but it would be unwise. Think about it. The loop to the repeat of LD A,R will contain a predictable number of instruction fetches, so the change in the contents of the Refresh Register will be equally predictable. If it is a multiple of eight, you will always get the same three least significant digits, and that will mean an infinite loop. In any case, the sequence will be fixed, and that is not desirable.

It would be possible to avoid this trap by calling for a further keyboard action if an unwanted number is produced, but that is rather clumsy. Without such an external trigger, the number obtained is not random.

BREAKING THE LOOP

There is a possible exception when a system uses interrupts, since an interrupt can alter the effective length of the loop back to LD A,R, but while that might break the infinite loop the reverse effect might be produced.

This leaves you with one possibility. A number in the 1-6 range can be produced by multiplying the refresh register number by 6, dividing by 128, and adding 1 to the result. A rather complex process? Not really. Listing 1 shows a possible form.

HL is zeroed, and DE is set to the required number range, N. The Refresh Register is copied to the Accumulator, and doubled to discard the most significant bit. A seven-iteration loop is then entered, with the iteration count in B. The most significant bit of A is shifted into carry. If the bit is 1, DE is added to HL. In any case, HL is doubled.

When the loop drops out, the result is taken from H. At this point, HL = 2 * R * N, so H holds 2 * R * N/256 = R * N/128. This will be in the range 0 to N-1, and the range 1 to N can be obtained by an increment.

For maximum generality, the routine could be called with N in A, in which case LD E,A would be added. It would be wise to save BC,DE and HL on the stack, since these registers are corrupted. The result would be a handy general-purpose random number routine.

Figure 1: Two-dice combinations

2nd Dice \ 1st Dice	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

Listing 1

```
START   LD HL,0000  ;N = Required range
        LD D,0
        LD E,N
        LD A,R      ;Read Refresh Reg.
        ADD A,A     ;Double A, discard MSB
        LD B,7      ;Loop count
L1      RLC A       ;MSB of A to carry
        JR NC,L2    ;Skip if MSB = 0
        ADD HL,DE   ;Add if MSB = 1
L2      Add HL,HL   ;Double HL
        DJNZ L1     ;Loop
        LD A,H      ;A = HL/256
```

But what about those who don't use a Z80? Well, if the program is waiting for a key depression, the wait loop could increment a count, which could be used instead of the Refresh Register contents. But care is needed. If the count progresses too slowly, its range may be restricted. Response to a prompt inviting key depression may take no more than half a second, perhaps less, and that means updating the count around once in 2 mS or less.

The actual increment needs only a few microseconds, but the keyboard check routine may take much longer. Ideally, the count should go through its complete range at least once in the minimum time between the prompt and the keyboard response.

This is especially difficult to achieve with systems that have keyboard buffers, since the key can then be pressed in advance by an eager user. However, such systems may have timer counters that can be used instead.

The essential point is that, whatever count is used, the action which triggers the reading of the count must be timed independently of the count increment.

SIMULATING DICE THROWS

A final point about random numbers concerns the simulation of dice throws. If two dice are thrown, it is not correct to generate a number from 1 to 12, with an equal chance of each value. Two random numbers in the 1 to 6 range must be generated, and their sum will give the required value. Why? As Fig 1 suggests, there is only one chance in 36 that two dice will give a result of 2 (or 12) whereas there are six chances that the result will be 7.

Random numbers are very useful, even essential in some cases, but you have to think about them very carefully. Normal methods of checking may be impossible to apply, since there are no concrete figures which can be used as a basis. Even with BASIC, there can be difficulties. Our own pet version of The Valley was almost impossible to play at first, because the initial random number always called up the worst monsters in the first few turns! We found that the number was being initialised from the disc...

INSIDE BUBBLE MEMORIES

Jamie Clary

It is unfortunate, but bubble memories are still an uneconomic alternative to conventional mass storage systems like the floppy disc. However, this article should act as a primer for the day when they become the affordable option.

Imagine a non-volatile mass storage memory system that has no moving parts; which does not require battery back-up to preserve its stored data; that has such high data-integrity that an error is practically impossible. Add to this the fact that the technology upon which the system is based has been developing steadily since the 1950's, and you would be forgiven for thinking that the system to which we refer must be a fearsome competitor to the memory devices which are currently popular.

Yet in spite of these quite remarkable qualities, and after almost TWENTY YEARS of development, the bubble memory has failed to make any impact on a market that is crying-out for a fast, compact, high-density storage medium that does not suffer the mechanical fragility that is virtually a hallmark for the floppy disc.

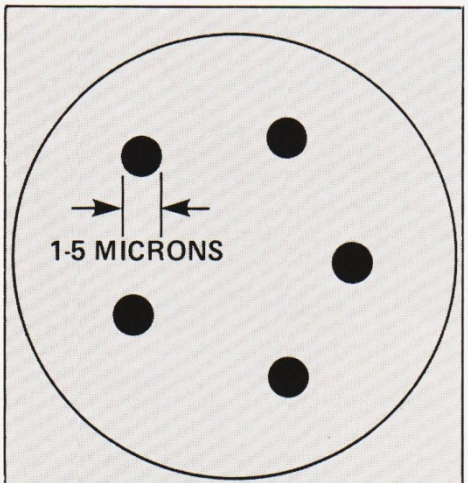


Figure 1: Single bubble domain showing typical diameter

Sadly, there is a straightforward explanation for the bubble memory's lack of commercial success — an explanation that for the present makes the device more of a computing 'curiosity' than a viable alternative to the disc-drive. However, the bubble memory has exciting possibilites, and an understanding of the extraordinary principles that permit it to function may help explain why they are, in some circles at least, such an emotive topic.

PRINCIPLES OF MAGNETIC STORAGE

To extract information from a disc system we rely upon the passage of discretely magnetised areas upon the surface of a disc to 'induce' a current in a single 'Read/Write head'. All magnetic storage systems rely upon a fluctuating magnetic field to transfer stored information, and here the magnetic field is made to fluctuate by moving, with a motor, the disc carrying these (typically 1.5-2 million) magnetic areas past the head.

The magnetic bubble memory also relies upon discrete magnetic areas, called 'magnetic bubble domains', to store information. But what makes these 'bubbles' so different from the discrete magnetic areas found on discs, is that they can, under certain conditions, be made to travel through and along the surface of a stationary magnetic material. It is this very freedom of the bubble to move without mechanical assistance that makes the system immediately attractive, as many of the failures that occur with more conventional magnetic storage media results from flaws in the mechanics, and not the magnetics, of the system.

But what exactly is a 'magnetic bubble domain?'

DOMAIN THEORY

When discussing the physics of magnetism, the term 'domain' is used to describe a small but distinct 'packet' of magnetism that exists within the crystalline structure of a ferromagnetic material. Each domain is considered as having a north and south pole, conforming to our traditional notion of 'The Magnet'.

In a given sample of ferromagnetic material, the total magnetic field is entirely attributable to these domains and their alignment with respect to each other. Hence, in an 'unmagnetised' specimen the domains are oriented in a completely random fashion, with the poles of each domain pointing in no particular direction. However, if we in some way magnetise the specimen, the domains become almost completely aligned, with the respective poles of each domain facing in the same direction. Incidently, this quite conveniently explains why ferromagnetic materials cannot be magnetised beyond a finite limit, the limiting factor being when each domain is in perfect, mutual, alignment.

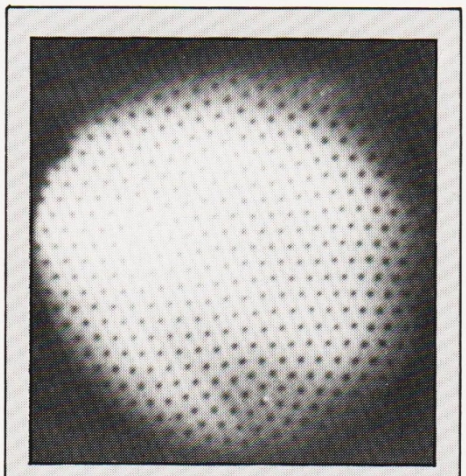


Figure 2: Faraday rotation micrograph of bubble domains in a 10 micron garnet wafer

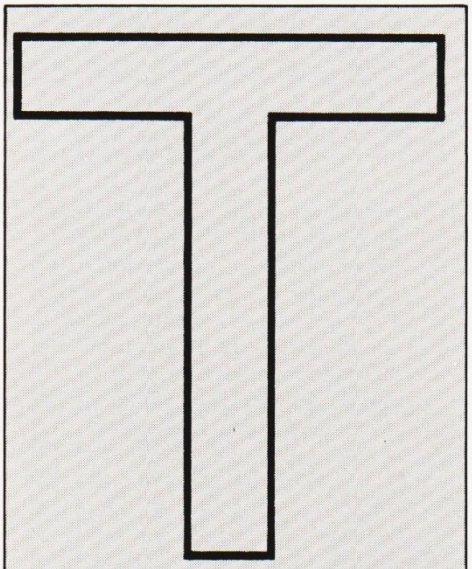


Figure 3: Single T bar propogation element

THE BUBBLE DOMAIN

The term 'magnetic bubble domain' refers to a small, mobile, cylindrical formation (see fig. 1) that exists within a wafer of ferromagnetic material when a constant magnetic field is applied in a direction normal to the surface. Such bubbles are 1 to 1.2 times the thickness of the host wafer in diameter; wafer thicknesses between 1 and 10 microns being typical.

One useful property of ferromagnetic materials such as the magnetic oxides (e.g. garnets), is that thin, transparent, wafers can be cut from the crystal. This permits us to observe the magnetic domains within a wafer using the Faraday effect (fig. 2).

BUBBLE MOBILITY

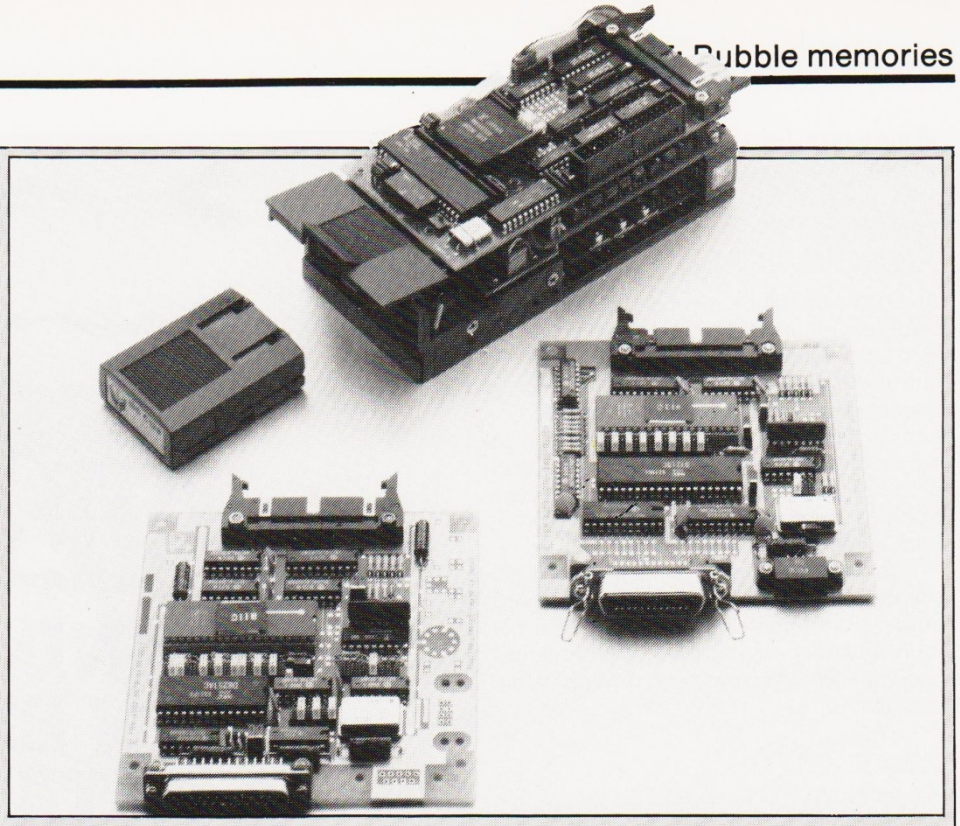
Bubble motion can be seen under a microscope, by placing a small magnet near to the inspected sample. The observed effect is a 'diffusive' one, as the bubbles drift towards regions where the external magnetic field is weakest.

It was this exhibition of mobility, and the fact that binary information could be stored using the presence of a bubble to represent a 'one' and the absence, a 'zero', that first suggested their usefulness as a storage medium. If strings of bubbles could be encouraged to follow a predictable path through the host material, then we would have a serial storage medium not unlike the tape in a digital tape recorder. But, as bubble domains are just a few microns in diameter, and can travel at several metres per second, a data transmission rate of megabits per second is possible, exceeding quite considerably the limits of the tape recorder.

But, consider the requirements of a system which is to accept, store, and regurgitate information, using magnetic bubbles:

- binary information entering the system must be converted into bubbles. To achieve this, we require a method for appropriately generating and annihilating bubbles to represent the 1's and 0's of the incoming bit-stream.
- the information, now in bubble form, must be encouraged to circulate within the host material until it is needed. Hence an accurate propogation technique must be implemented.
- We must also be permitted to 'syphon-off' the information stored in the bubbles when it is needed. Therefore, bubble detection apparatus must also be present.

We have found, then, that there are four main requirements for our theoretical bubble memory ie. a bubble propogator, a generator, an annihilator, and a detector. Let us consider each of these in turn.



The IBS Bubble Memory Cassette System consists of the IBS H100 master unit (£245), the C128 bubble cassette (£278), and the IBS HX1 slave unit (£117). A minimal system consisting of the H100 master unit and a single C128 cassette gives access to 128kbytes arranged as 2048, 64 byte pages. By adding a maximum of three slave units, the system is expandable to 512kbytes.

BUBBLE PROPOGATION

Although it seems reasonable to examine how bubbles are generated before considering how they are propelled, the two features are more easily explained in reverse order.

Bubble motion is not a 'physical' phenomenon; there is no movement of material through the wafer. To appreciate this, consider how wave motion is taught in school physics lessons: a length of rope is fixed at one end, and a shock applied to the other. A wave can be seen travelling down the rope. What is important to recognise, is that alhtough the ***rope*** swings up and down through the vertical axis, most of the ***energy*** is transmitted along the horizontal. Energy, stored in the curve of the rope, is transported by changing the position of the curve with respect to time. Similarly, our small packet of magnetism is swept along a wave, by repositioning the walls of the domain.

In practice, bubble motion is brought about through the use of an oscillating magnetic field, called the Wall Motion Field. As this field oscillates, the bubble wall travels in accordance with changes in the field. This motion can be accurately controlled by depositing a regular pattern of 'propogation elements' (fig. 3) into the ferromagnetic wafer. These elements are made from a Nickel-Iron alloy, which forms a temporary magnet in the presence of a magnetic field, concentrating the field energy into two poles — north and south. By causing the field to oscillate, a proportional change in the strength of the field about the poles will occur. But, rotate the field, and the poles will rotate in sympathy. Thus, a bubble close to a propagation element under the influence of a rotating magnetic field will follow one of the poles as it rotates. Furthermore, by creating a closely-packed matrix of propogation elements, we can move bubbles for a considerable distance with precision, by transferring the bubble from one element to another as the direction

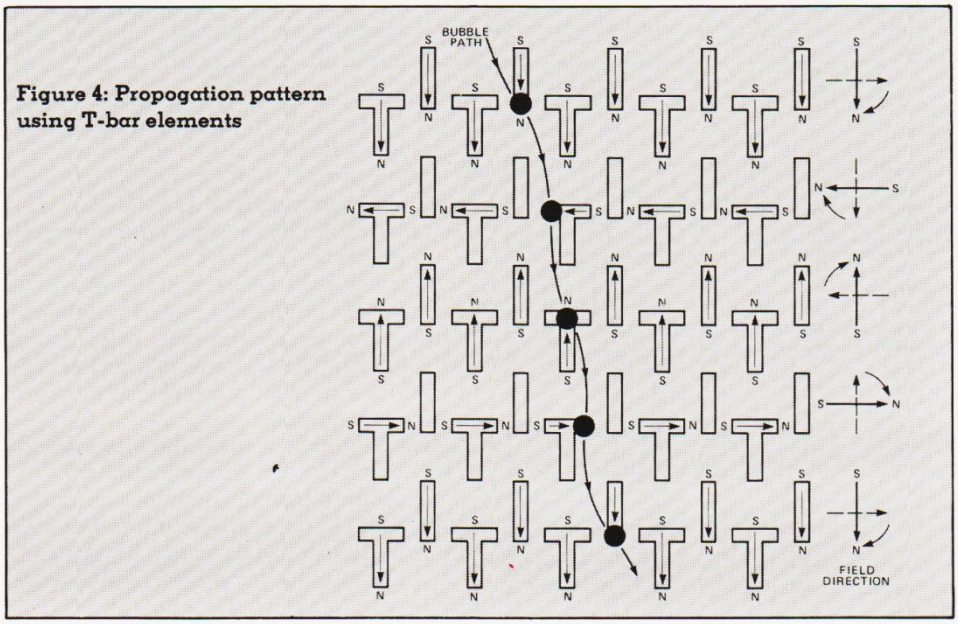


Figure 4: Propogation pattern using T-bar elements

of the motion field changes. This process is shown in figure 4.

NUCLEATION AND ANNIHILATION

As has been briefly mentioned already, a bubble is generated when the strength of a local magnetic field reaches a critical level, called the Nucleation Threshold. In bubble memory devices, data is written into the system by creating a bubble each time a current passes through the conductive part of a special element, the Nucleator. The bubble then passes into the system to circulate until required (see later).

Bubble annihilation is achieved in a similar way, but a higher current is passed through the nucleator until the bubble 'bursts', so to speak, as the domain wall ruptures — effectively removing a memory 'bit' from store.

READ-OUT

Data stored in bubble memory is often read-out using the Magnetoresistive effect. By moving a bubble beneath a line of elements through which a current of a few milliamps flows, the magnetic field associated with the bubble causes the resistance of the elements to rise, generating a detectable change in the voltage across them. This technique is preferred, because it is non-destructive, leaving the bubble intact after a read operation.

Other, destructive, methods are used, such as the Inductive Sensor, and they have certain advantages. But the use of these methods force other considerations upon the designer, such as the necessity for duplicating a bubble prior to a read-out, and this complicates the production of the device.

Optical detection has also been suggested, but by far the most successful detector is the magnetoresistive sensor.

BUBBLE MEMORY ARCHITECTURE

It is not generally known, but the 'Black Box' flight recorder used in aircraft instrumentation is a bubble memory. However, the nature of this application means that a very simple, serial, 'mechanism' (although no moving parts, don't forget!) is quite satisfactory. However, for applications that demand high-speed data retrieval, access to information must be rapid and direct.

To preclude any situation in which a string of files must be examined before the appropriate data is located, a number of bubble architectures have been developed which permit the selective interrogation of serial bubble files, or pages, as they are more often termed.

One such architecture, the Major-Minor loop architecture, is shown in figure 5.

Serial data enters the generator, and a string of bubbles representing the data is launched into what is termed a 'Major loop'. The bubbles are then transferred in parallel to several 'Minor loops' where the bubbles

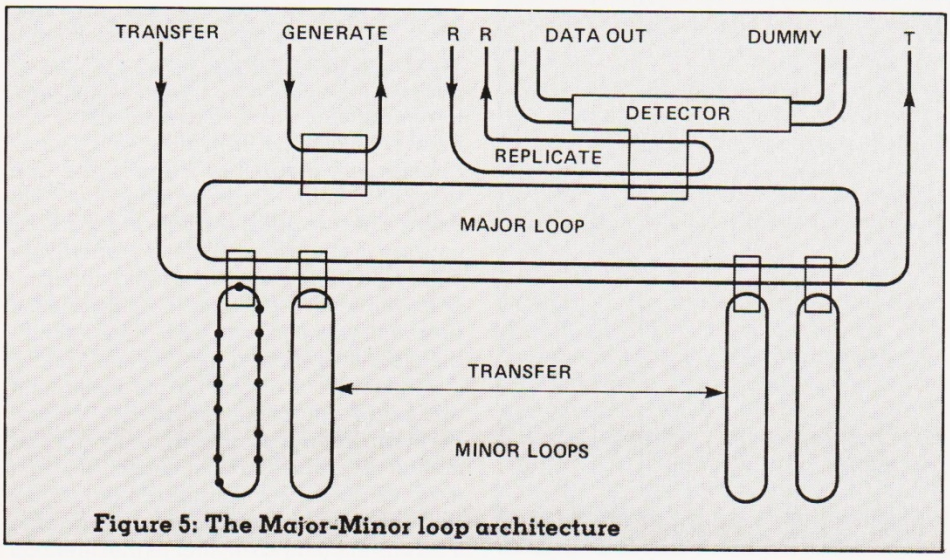


Figure 5: The Major-Minor loop architecture

circulate until they (the loops) are addressed for Read. Once addressed, the bubbles in the Minor loop (or replicas of them if the system employs a destructive readout) are launched into the Major loop where they circulate before passing the detector. Once the data transfer is complete, the bubbles are returned to the Minor loops, clearing the major loop for future operations.

CONSTRUCTION AND PACKAGING

Figure 6 shows an exploded view of a bubble memory package.

A ferromagnetic wafer, into which a propogation matrix has been deposited, is mounted upon a ceramic or resin carrier. Conducting leads are etched into the carrier, connecting the wafer to the outside world. Two bias magnets are positioned above and below the wafer. These provide the stable magnetic field required to keep the domain walls intact. Two coils create the rotating Wall Motion field, and these are mounted above and below the bias magnets. A magnetic screen is fitted — to shield the wafer from stray magnetic fields — and finally, the unit is inserted into a (typically) square, DIL package.

CONCLUSION

Fabrication is a costly process. Although there are fewer and less demanding process-steps in creating the ferromagnetic wafer as compared with the number of steps in MOS memory manufacture, techniques are far from perfect. Yields are poor and significantly low — so low that it is more cost-effective for manufacturers to test each individual wafer, and subsequently encode a ROM to prevent access to faulty loops, than it is for them to discard the imperfect wafers.

This situation will change as yields increase and techniques improve. But for the present, the cost of bubble memories excludes them from use by the general public. Judging by the number of major manufacturers who have decided to leave the bubble market — National Semiconductor, Texas Instruments and Plessey have removed themselves after many years of research — the affordeable bubble memory will be a long time in coming. However, Intel, Hitechi, IBS and Fujitsu are, it seems, prepared to brave the storm — and they are to be congratulated for doing so, we wish them well.

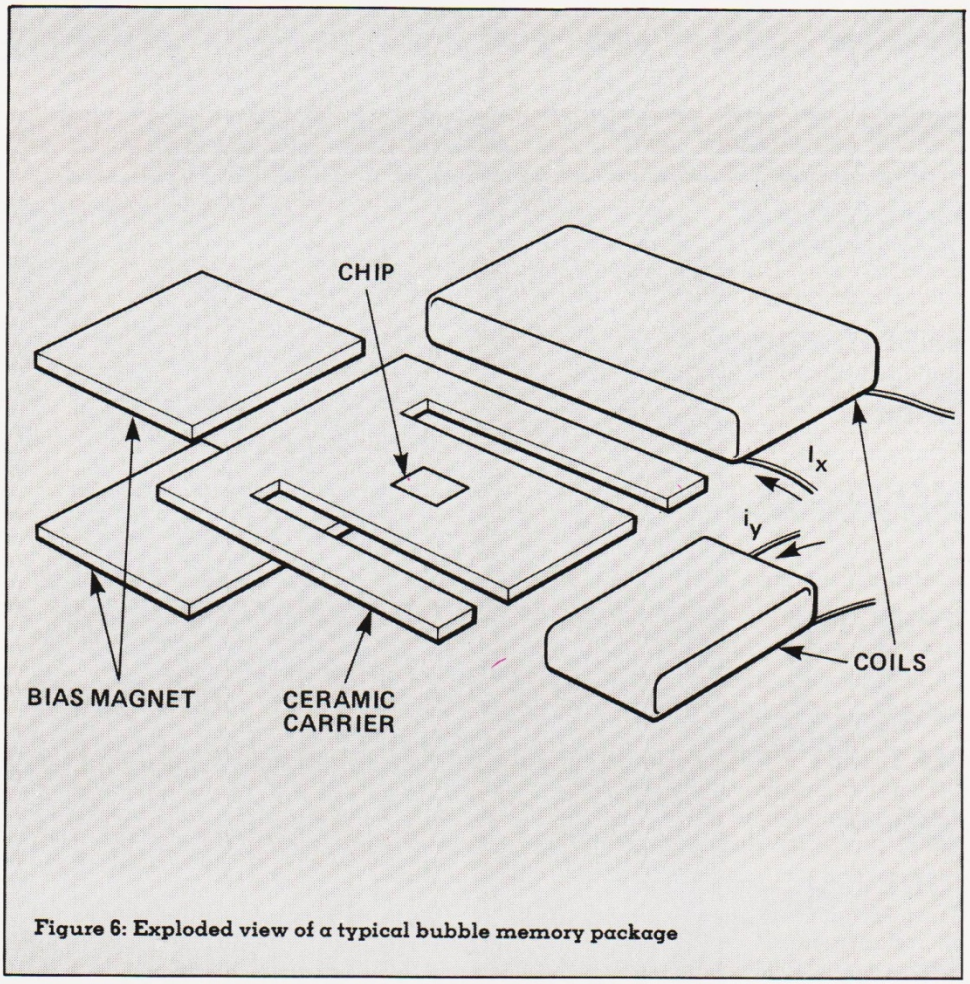
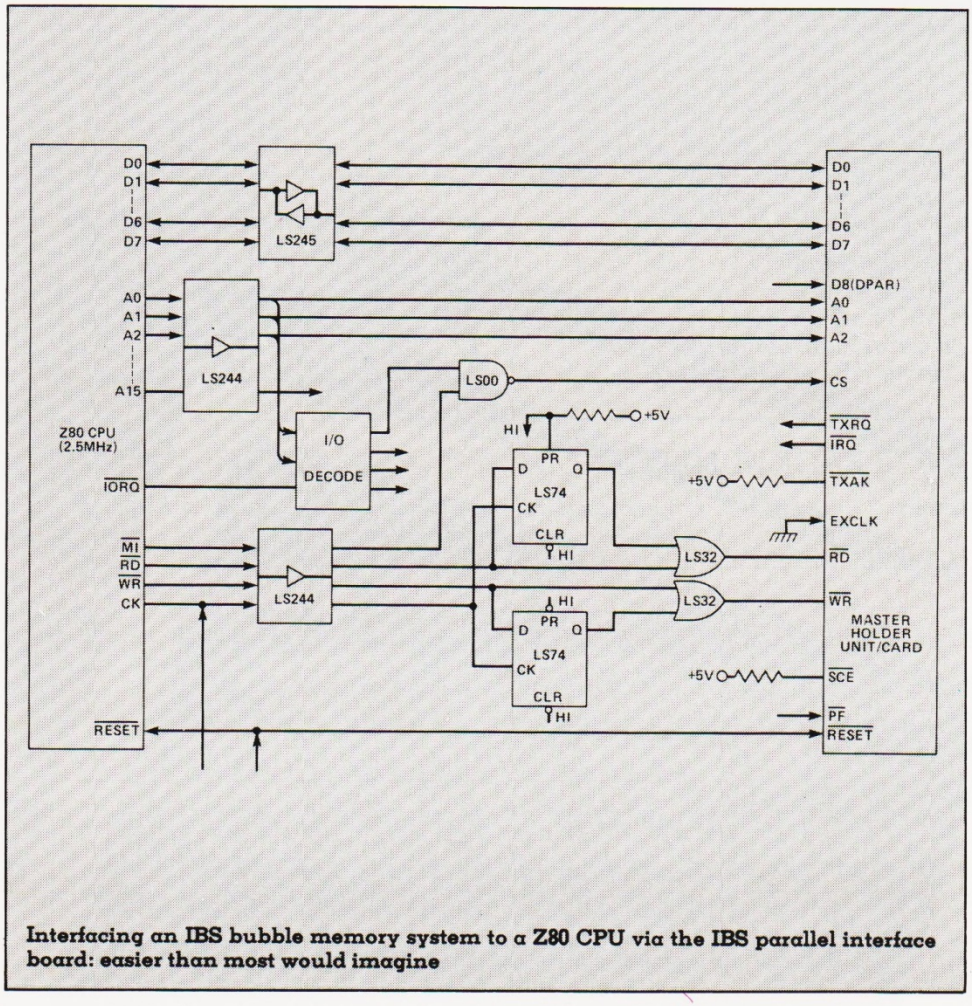


Figure 6: Exploded view of a typical bubble memory package



Interfacing an IBS bubble memory system to a Z80 CPU via the IBS parallel interface board: easier than most would imagine

BIBLIOGRAPHY

Magnetic Bubbles by T.H. O'Dell (Macmillan) SBN 333 1708 5

The Physics of Electricity and Magnetism by William Taussig Scott (Wiley)

Fundamentals of Physics by M. Nelkon (Granada) ISBN 0 7010 0139 9

We would like to thank Immediate Business Systems plc, Intel Corporation and Hitachi Ltd., for their cooperation in the construction of this feature.

BACKNUMBERS

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81 Forth, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
MEMOTECH MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

MAY 1984
Debugging, Spectravideo SV318 review, Extending the Commodore 64's BASIC part 1, Z80 text compactor.

JUNE 1984
Adler Alphatronic review, Digithurst's Microsight review, Commodore search and replace, CP/M directory, Interrupts.

JULY 1984
Commodore BASIC extensions reviewed, The Art of Islam, a fast sort, Brother HR5 review, Random Thoughts, extended palette on the Dragon.

AUGUST 1984
Apricot xi review, BBC Mode 7 screen editor, Genie sprites, Microdrive-file line editor, TRS-80 screen scroller.

SEPTEMBER 1984
CUBE's Beebflex, Electron drawing utility, MTX real time clock, Commodore SX64 review, BBC disassembler, TRS-80 Fastsave.

OCTOBER 1984
AMSTRAD CPC464 review, Dragon sprites, Commodore 64 adventures, BBC Draughts, Nascom screen dump.

NOVEMBER 1984
Apple IIc review, Epson PX8 review, MTX utilities, Z80/TRS-80 memory move routine, 16-page Business supplement.

DECEMBER 1984
Acorn Bitstick package review, Art and the AMSTRAD, BBC Draw, Psion Organiser review, Koala Pad review.

JANUARY 1985
BBC Commodities, Tatung Einstein review, Fujitsu Micro 16 review, Commodore 64 prettyprint, MTX500 Life, Nascom string-save.

FEBRUARY 1985
The Intelligent Computer, Dragon interrupts, BBC Machine-code monitor, Tasword 464 review, Spectrum/BBC cassette volume meter, Sakata SCP800 printer/plotter review, Spectrum ON ERROR, TRS-80 mail list, BBC passwords; Deficiency, Abundance, Perfection.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you £1.40 (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A COMPUTING Today Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ .....

NAME .....

ADDRESS .....

.....

.....

..... POSTCODE .....

Signature .....

I enclose a cheque/PO for £..... (Payable to ASP Ltd)
I wish to pay by credit card

Access ☐ Barclaycard ☐

Buy it with Access

BARCLAYCARD VISA

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form. do not send your card.

Please allow 21 days for delivery.

QL NEWS

News from the world of Sinclair QL computing.

The communications explosion takes shape!

Communications are now the most exciting, essential part of any computer.

In the past six months alone, over 150,000 modems have been sold in the UK.

Now, the QL's own communications explosion is taking shape . . . and it has the potential to make more of communications than any other micro!

Read on and discover exciting new ways to use your QL . . . with the QL modem . . . telephony unit . . . and powerful interface options.

Nº2

DAVID KARLIN

Why Q COM is everything you could wish for in communications.

The QL is now communicating – via Q COM! This exciting three-part peripheral presents QL users with a multitude of ways to exploit the world of communications.

Once connected to the QL, Q COM allows you to access the considerable number of phone-in databases, such as Prestel and QNet.

Q COM enables you to communicate with other computer users. Its facilities include electronic mail, data transfer from Microdrives and bulletin boards.

Through it you can link your QL to larger minicomputers. Q COM has full capability in this area, and allows the QL to talk to powerful mainframes.

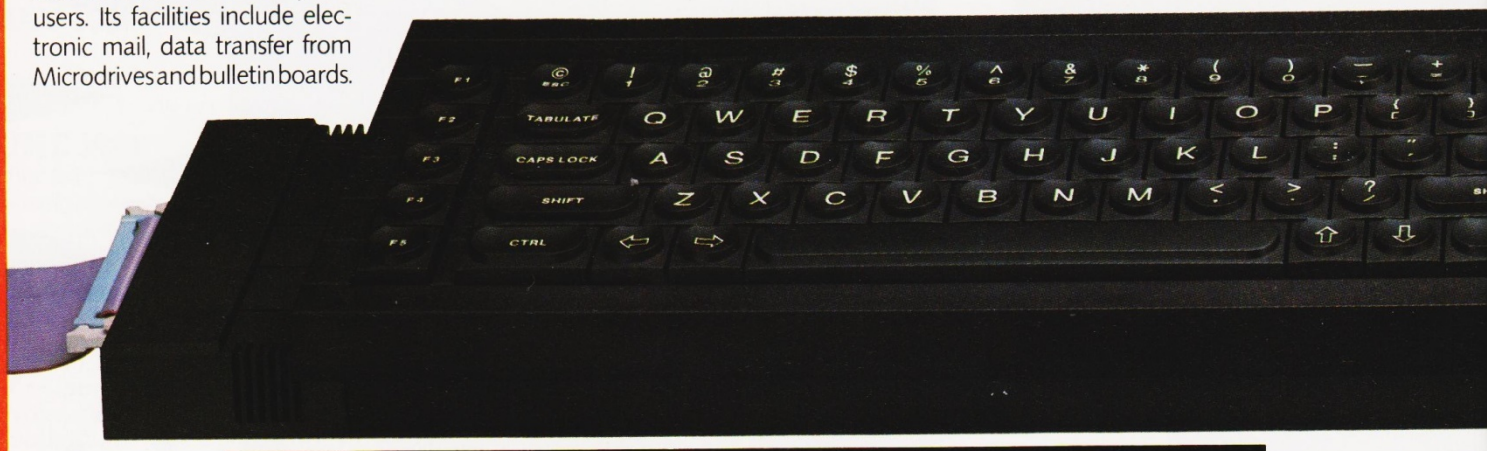
Q COM's automatic dialling and call acceptance facilities, together with the storage of messages from other modems, will revolutionise the way you use your telephone.

The next few pages of QL News tell you much more.

It's enough for me to say here that with the QL and Q COM, you'll be exploring new openings in communications for some time to come!

David Karlin

David Karlin,
Chief Design Engineer.



The Q COM package
Three special parts to stack!

QL communications interface

This multi-speed interface contains the sophisticated software used to set up QL communications – and to control the Q CALL and Q MOD units.

Q CON also comes complete with Microdrive-based software. This enables the QL to link to larger computers using VT100 and viewdata protocols.

The software will also run *any* standard modem – connected via Q CON's built-in RS-232-C port.

Most importantly, Q CON allows the QL to transmit and receive at rates switchable from 75 to 9600 baud (encompassing the widely-used 75/1200 Prestel rates, and 1200/1200 half duplex rates for user-to-user exchange).

Q CON is specially styled to suit your QL – with similar fluting and ribs – and forms the base module of a vertical-stacking system.

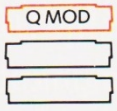
It's supplied with full instructions, software on Microdrive cartridge, and connecting leads.

QL auto dial/answer unit

Q CALL gives every QL user something out of the ordinary.

It's a module which links directly to your telephone, and allows auto-dialling at the push of a single key. In the same way, it will permit incoming calls to be accepted automatically . . . and even trigger pre-programmed activity from the QL!

Q CALL is the central unit of the package. It plugs directly into Q CON – so there are no connecting cables to worry about.



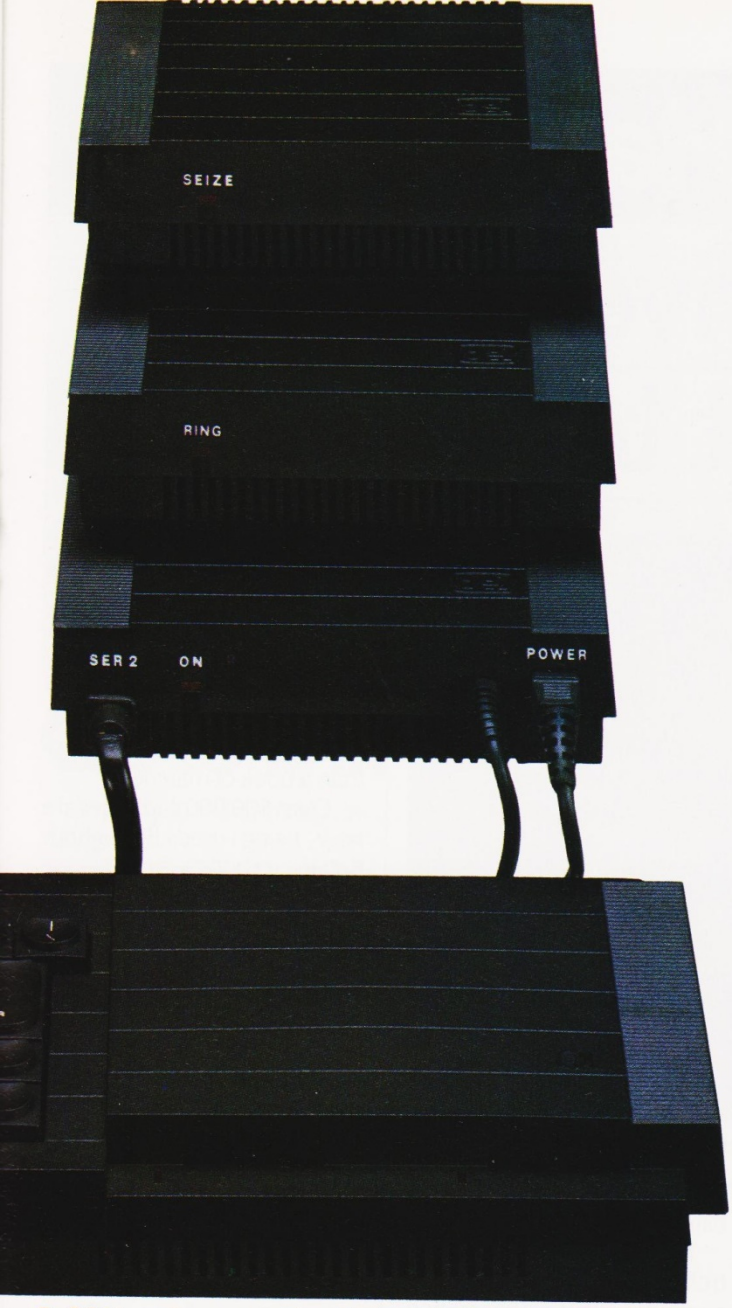
QL modem

Q MOD has all the powerful facilities expected of a modem, in a neat and simple unit.

It uses either V23 75/1200 or 1200/1200 baud rates, for Prestel, Micronet 800 and all the viewdata services described alongside.

It also incorporates a telephone extension socket for manual dialling.

Q MOD is the top unit of



Q COM, and comes with a 9' built-in telephone cable.

All three units are available from OEL on (0276) 66748 and from selected Sinclair stockists.

The QL hooked on voice and data

The QL can now act as your personal address book and telephone operator!

Q COM allows you to store hundreds of personal or business numbers.

You can store lengthy passwords and account numbers – and recall them – at the touch of a single key.

And any information that's sent to you from other modem-owners can be gathered and stored on Microdrive cartridge, or incorporated into your QL Quill documents!

Exploring the world of QNet, Prestel, Micronet and more!

Thousands of QL users already enjoy the excitement of linking to a nationwide mainframe.

Q COM turns your QL into an intelligent terminal, allowing you to access many thousands of pages of information, software and communications facilities.

The services brought to you through Prestel can include Micronet 800, Viewfax 258 and QNet, the new QL database.

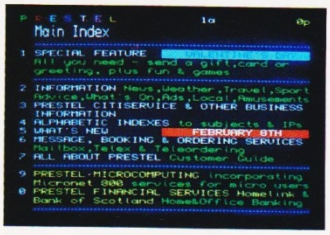
Membership of QNet will bring you free software, QL news and features, and all the wide-ranging services of viewdata!

If armchair shopping is more your style, that's easy too. It's often possible to place a direct order using your QL! For dedicated QL owners, there's a daily selection of software reviews, chart toppers... and all the facts and figures you need to make buying peripherals simple.

With Q COM you can also 'download' software from the system directly into your QL and either use it immediately, or store it on Microdrive cartridge.

In fact the only problem you'll face with a viewdata service is finding enough time to explore its many features!

You can find out how to join QNet by phoning 01-278 3143.



News... information... banking services and QNet. And only a fraction of the QL's new viewdata capability.

QL meets the mainframes!

The Q CON unit of Q COM turns your QL into a VT100 terminal, providing instant access to in-house computing services, both mainframe and mini.

Whether you are using your QL at home or at work, Q COM gives you access to electronic bulletin boards which provide help and advice 24 hours a day. You can leave messages or notices for friends or business contacts and even hold live discussions with them.

Additional benefits for the QL business user include easy access of in-house company software, and the interrogation of other data bases around the country.

There's also the opportunity of linking to British Telecom Gold – the widely-publicised and popular messaging service.

sinclair

QL Hardware

Microdrive cartridge price cut to only £1.99!

Sinclair Microdrive cartridges – up to 100K of programs and data on a medium so compact you can pop it into your pocket.

On February 1, the cost of Microdrive cartridges came down from £4.95 to £1.99 each.

Microdrive cartridges are the QL's own unique storage medium. Each stores up to 100K of information (that's 40 pages of A4 text), on a cartridge no bigger than a book of matches!

Over 500,000 cartridges are now being used throughout Britain.

You can store up to 50 different data files per cartridge, identified by titles of your own choice.

And QL Microdrives themselves are standard equipment on the new ICL One Per Desk micro, and British Telecom's new Merlin Tonto.

IEEE-488... the instrument connection

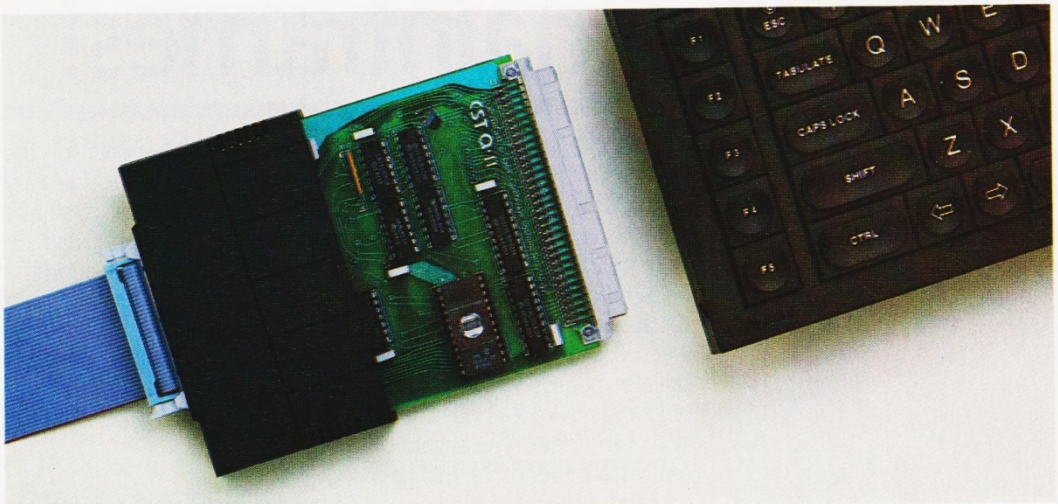
IEEE-488 is the interface standard set by the Institute of Electronic and Electrical Engineers for instrumentation control.

IEEE-488 – or General Purpose Instrumentation Bus – is a parallel interface specifically designed for high speed data transfer between a number of different types of device.

It is commonly used for controlling instrumentation via a computer, allowing the creation of laboratory data acquisition systems, industrial control schemes, etc.

The QL now has a fully-fledged IEEE-488 interface from CST. It plugs neatly into the QL's RAM expansion port, and can control up to 16 instruments simultaneously.

It's available from CST on (0223) 323302.



An IEEE-488 interface slips discreetly into place.

New inte 3½" or 5¼

With new Q-Disk, you can transform the QL into a powerful small business system – comprising QL, monitor, disk interface, twin disk drives and printer.

Q-Disk upgrades the QL to disk storage. Fitting easily into the QL's left hand RAM expansion port, without the need for a special expansion box, it contains a Western Digital disk controller chip. Software is held in an on-board EPROM (so little of the QL's RAM is used).

Plug in Q-Disk, and the QL accepts one or two disk drives, sized 3 in, 3½ in, 5¼ in, either 40 or 80 track, single or double-sided. Even when two drives are used, they can be different types!

Q-Disk offers up to 1.6 Mbytes of quick, reliable storage with a compatible disk drive.

It's made by Computamate, who also offer a full range of

QL to link students

Strathclyde University, in Glasgow, plans to have a campus network of 7,000 QLs linked to a central VAX minicomputer.

That's one QL for every student . . . a major investment project in a university which is now a leading centre for artificial intelligence work.

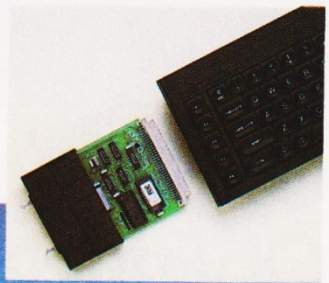
Sinclair is giving support worth £250,000 to the project. And it's likely that QL users everywhere will benefit – the students plan to develop AI programs to run on the QL!

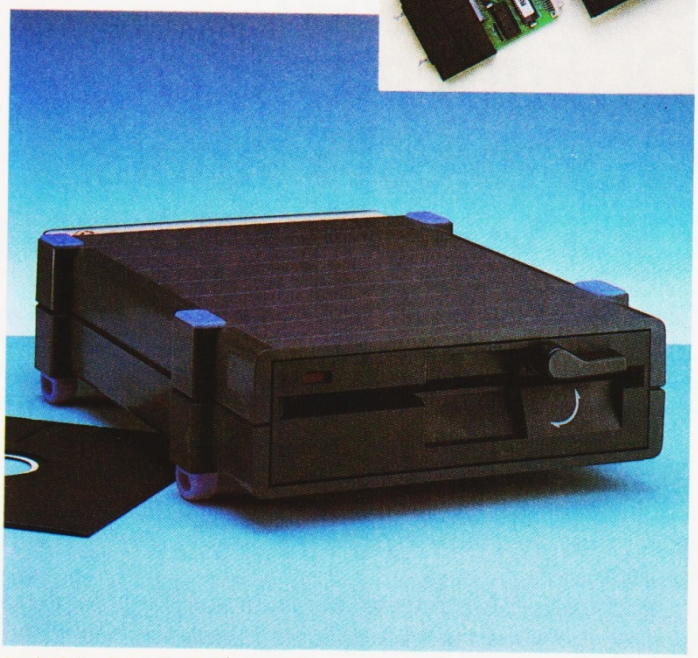
The QL has impressed Prof. James Alty of the University's Computer Science Department, who says *'only the QL could offer the computing power, range of applications, and above all the portability, at a realistic price.'*

face to connect 3½" disk drives

complimentary QL disk drives.

To contact Computamate, phone (0768) 811711.





Single disk unit fitted with 5¼ inch drives and *(inset)* the Q-Disk controller.

The QL's high-tech spec

Dimensions
138 x 46 x 472mm
(5⅜" x 1¾" x 18¾")

Weight
1388 gms (3.055 lbs)

RAM
Massive 128K standard RAM, externally expandable to 640K. Extra RAM is available in 64K, 128K, 256K and 512K units, from third-party suppliers.

ROM
48K, containing Sinclair SuperBASIC and the Sinclair Qdos operating system.

CPU
Motorola 68008 (running at 7.5 MHz) for all principal functions. (Architecturally, the 68008 is a 32-bit processor with an eight-bit data bus. One megabyte of non-segmented address space is available.)

In addition, an Intel 8049 controls the keyboard, generates the sound, and acts as an RS-232-C receiver.

Operating system
Qdos (developed by Sinclair Research) is a single-user multitasking, time-sliced system using Sinclair SuperBASIC as a command language with display handling for multiple screen windows; and device-independent input-output.

Language
Sinclair SuperBASIC, with the advantages of procedure structuring; extendability (including syntax); interpretation speed independent of program size; clean machine code interface; operating system facilities accessible from SuperBASIC; equal capability for strings and arrays; and full error-handling facilities.

Microdrives
The QL incorporates twin QL Microdrives, each with a minimum 100K capacity, 3.5 seconds average access time. Typical loading rate of machine code programs is 2-3K per second.

Video
High resolution graphics capability with colour or monochrome monitor (or TV) in two modes – 512 x 256 pixels (four colours available) and 256 x 256 pixels (eight colours available). Normal character display format of up to 85 x 25 with choice of character sets available (TV format of up to 40 to 60 columns depending on the software).

Keyboard
Full-size, 65-key QWERTY keyboard featuring a space bar, left- and right-hand shift keys, five function keys and four cursor control keys. The keyboard can be angled by means of detachable feet.

Expansion
Excluding RGB monitor, power socket and TV port, eight peripheral/expansion ports are provided – one internal expansion, one Microdrive expansion, one ROM cartridge, two serial and two control channels, and the local area network.

Serial
Two standard RS-232-C communications interfaces for printers, modems, etc. Transmission at rates from 75- 19200 baud or full duplex transmit/receive at seven rates up to 9600 baud.

LAN
For up to 64 QL computers. Data transmission over the net can be achieved at 100K baud.

Power supply
9V DC at 1.8A, 15.6V AC at 0.2A.

Joysticks
Provision for one or two devices for games or cursor control.

Applications Software
QL Quill – word processor
QL Abacus – spreadsheet
QL Easel – graphics
QL Archive – database
All four packages supplied with the QL.

Price
£399 including VAT, QL programs, full A4 manual, power supply, 4 blank cartridges and free Helpline service.

sinclair

QL Software

Updated versions of Psion software now available!

QL Abacus, Archive, Easel and Quill are the four Psion programs supplied with every QL. They're now converted to 100% machine code, and as a result they load from Microdrive cartridge much faster.

The overlays present in Version One software have been removed, resulting in noticably quicker on-screen performance.

With the compactness of machine code, there's a big saving in QL memory too – all four programs now cope with larger, more professional applications!

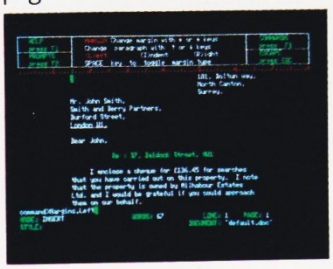
Version Two software is now supplied with every new QL. Existing QLUB members – see back page.

QL·Quill

QL Quill makes it easy to type in, correct and store your letters, memos and reports.

No training is needed – a beginner can be using QL Quill for word-processing within minutes!

QL Quill has the facilities of professional word processing packages: including word wrap, search and replace, justification, page headers and footers.



QL·Easel

QL Easel allows you to create graphs, bar charts and pie charts – at the touch of a key.

The program handles anything from lines and shaded curves to overlapping or stacked bars.

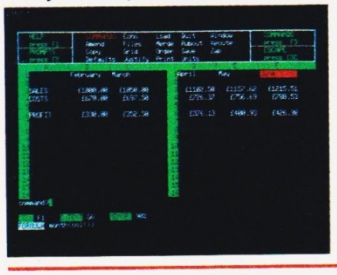
QL Easel designs and scales automatically or under your control. Text can be added and altered as simply as data.

QL·Abacus

QL Abacus is a powerful, yet easy-to-use spreadsheet.

The program allows you to manipulate the contents of whole rows and columns by the names you assign them. There's no need to depend on confusing letters and numbers.

QL Abacus also incorporates a range of functions which let you carry out rapid 'what if' analyses on your data.



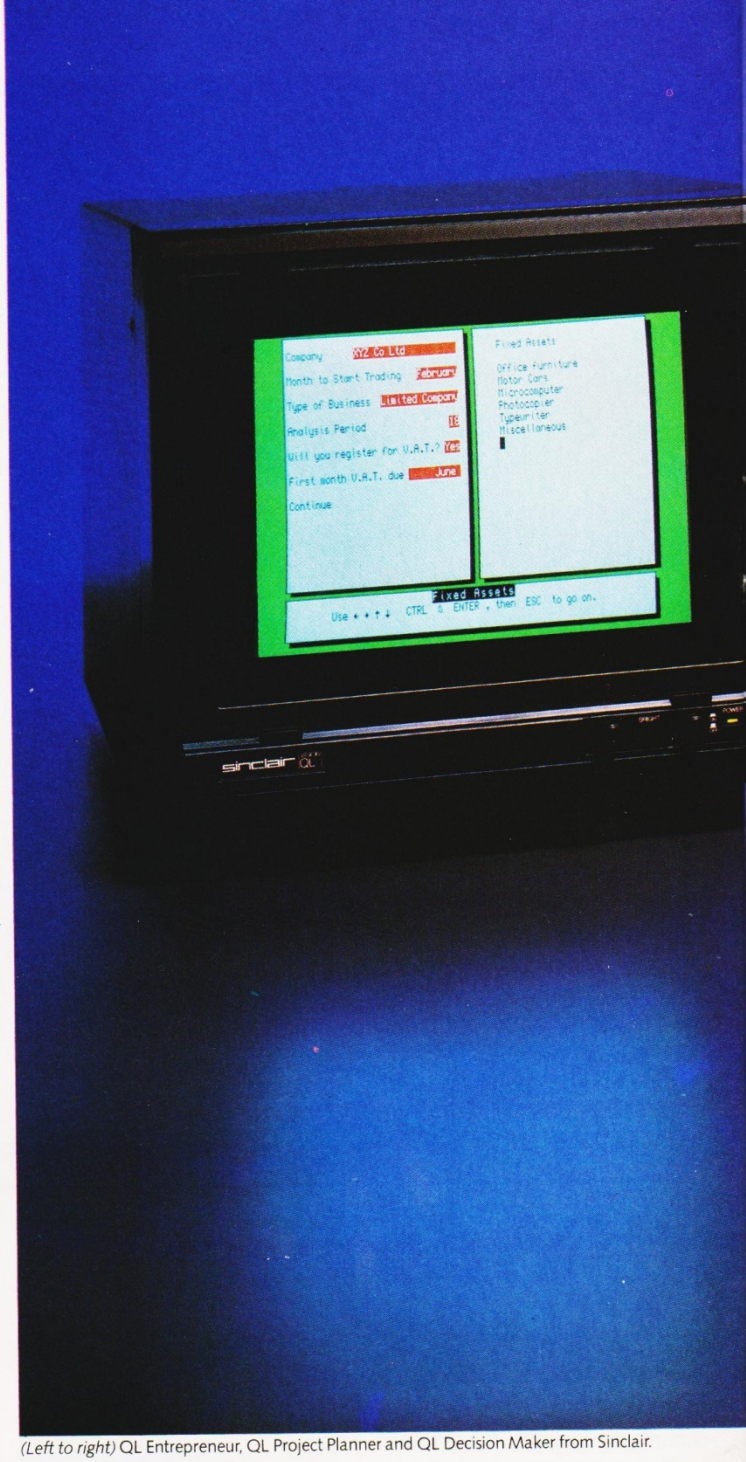
QL·Archive

QL Archive is a sophisticated, powerful database program.

It includes a screen editor which allows you to design your own screen and format your reports, and a procedure editor which lets you tailor QL Archive to your own requirements.

QL Archive is ideal for all database uses, yet it's powerful enough to be used by many software houses to generate specific database applications.

Non-members of QLUB can purchase new versions of the above software for £15 per title, or £50 for all four programs. Phone (0276) 686100 for details.



(Left to right) QL Entrepreneur, QL Project Planner and QL Decision Maker from Sinclair.

Coming soon- QL·Entrepreneur, QL·Project Planner, QL·Decision Maker!

Three new QL business programs – with a difference!

QL Entrepreneur, QL Project Planner and QL Decision Maker train you to apply new and exciting management skills – through original and powerful means!

An interactive teaching program gives you a thorough and enjoyable understanding of each subject – backed by a textbook and 'self-test' facilities - and an applications program helps you to use your new expertise for specific problems and projects.

All three titles will increase your understanding and extend your control – making involved subjects easy, stimulating and useful!

QL·Entrepreneur

QL Entrepreneur is an essential program for anyone preparing to start a new business – whatever it may be!

It uses a 'question and answer' format to help you build a workable business plan.

With the input you give, it works out the break-even point of the business; the first 18 months' cash flow, the type of finance needed; the year end Balance Sheet and Profit and Loss accounts . . . and more!

.QL Entrepreneur builds your skills and techniques.

It's flexible too, so that you can ask complex 'what if' questions at any stage!

The program comes with a third, blank Microdrive cartridge and a comprehensive A5 manual.

QL·Project Planner

QL Project Planner will produce plans you can understand, monitor and more easily achieve.

First, you break the project down into its individual activities, telling QL Project Planner how long each takes and which are inter-dependent.

When you decide on a starting time/date QL Project Planner will tell you when each activity must start and finish and when the project will be completed.

Each activity is divided into its critically important stages – those which can safely be moved around without altering the time taken by the project and those where movement will affect the completion deadline.

Whether or not you've used project planning systems before, you'll be amazed at the difference QL Project Planner can make.

The program comes with a third, blank Microdrive cartridge and a comprehensive A5 manual.

QL·Decision Maker

Whether you're thinking of buying a house, or taking on a new business contract, QL Decision Maker makes the choices clearer!

It lets you look at the possibilities – and their implications – through a decision tree.

Once you've set out the decisions and their probable costs or results, QL Decision Maker shows the outcomes which would occur from each particular route.

You can see how much money a decision could make for you . . . or cost you. Complex 'what if' questions are dealt with swiftly and graphically.

You can depend on the QL to highlight the best possible route!

QL Decision Maker comes with a third, blank Microdrive cartridge and a comprehensive A5 manual.

All three programs are available from Sinclair stockists, price £39.95 each, or Sinclair Research. Tel: (0276) 686100.

sinclair

Now, buy a QL and you're a member of the QLUB-free!

QLUB is the special Users Bureau for Sinclair QL owners.

Already, there are well over 10,000 QLUB members . . . enjoying a whole range of information and advisory services.

Until now, joining QLUB cost £35 per year. From March 4, every new QL owner can become a member – free of charge!

With your new QL, you'll find a post-paid form. Complete and mail it, and you'll soon be a member of the fastest growing computer club in the country.

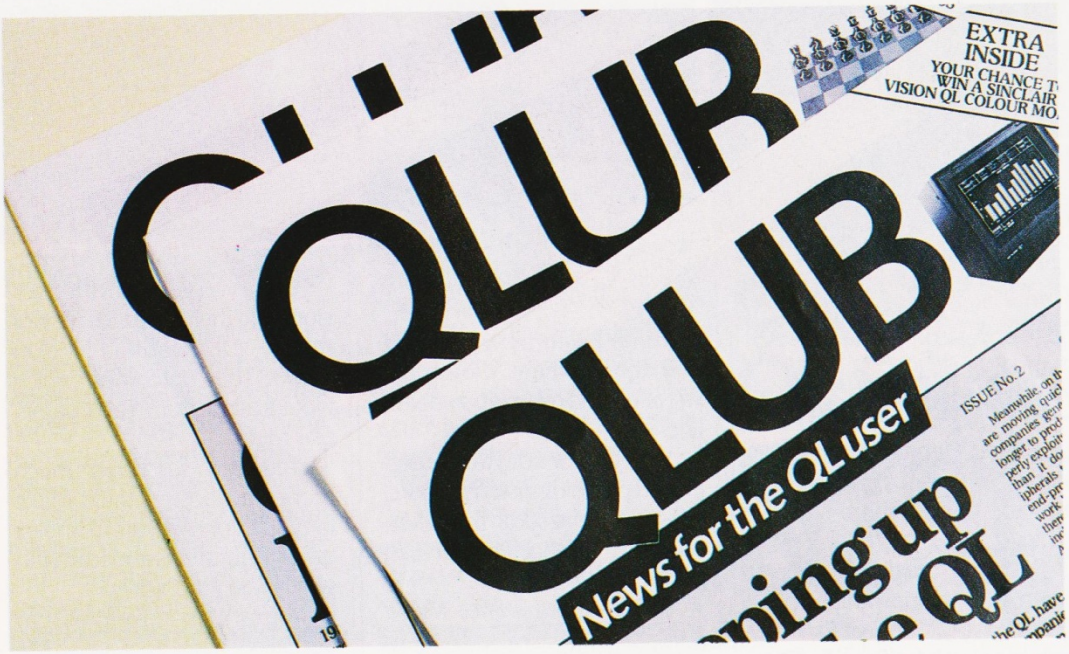
And you'll enjoy all the helpful services listed here!

What QLUB membership offers you

Regular newsletters delivered to your door

One of the most important QLUB benefits is the special news magazine, appearing six times a year. The magazine provides a forum for QL owners to exchange views and keep in touch with all the latest developments.

Each issue is packed with updates on QL hardware and software, tips on applying the four QL programs, and news of how other people are using the QL.



Special discounts

QLUB members also receive a range of special discounts, with savings of at least 20% on selected software products.

There are also special subscription rates for Personal Computer News and QL User.

Free Helpline service from Psion

All QLUB members are entitled to 12 months special assistance from Psion.

They're at the end of the telephone to answer any questions on using the QL Abacus, Archive, Easel and Quill programs supplied with the computer.

Help is also available on any aspect of using Sinclair SuperBASIC, Qdos, or linking your QL with major peripherals.

Psion will normally answer any queries within 48 hours.

QL program updates are no longer available free to QLUB members. They will be sold separately.

Good news for existing QLUB members too!

As one of the first members of QLUB, you should already have received one *free* update of each of the four QL programs – and a letter with your new membership details.

If for any reason you haven't, you should ring (0276) 686100.

You're a QL owner, but not a QLUB member?

Then joining QLUB is easy and free! Ring (0276) 686100 for full details. You can be a full QLUB member within a few days.

Where to find the QL

The Sinclair QL is available at selected branches of Dixons, WH Smith, John Lewis Partnership, Currys, Greens in Debenhams and Ultimate, and larger branches of Boots, John Menzies and specialist computer stores nationwide.

Sinclair Research Ltd
Camberley, Surrey, GU15 3BR.
Tel: Camberley (0276) 686100.

sinclair

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, news and reviews – all to help committed micro users make more of their microcomputers at home or at work.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,

Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1985 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
INFONET LTD,
TIMES HOUSE,
179 THE MARLOWES,
HEMEL HEMPSTEAD,
HERTS HP1 1BB.

Please commence my subscription to Computing Today with the ..... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)

£16.20 for 12 issues UK	☐
£18.70 for 12 issues Overseas Surface	☐
£51.20 for 12 issues Overseas Air Mail	☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £.....
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

ACCESS BARCLAYCARD VISA

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) .....
delete accordingly

ADDRESS .....

.....

..... POSTCODE .....

Signature .....

Date .....

CT April '85

MACRO 2

Farouk Elhiddiny

We tie up our preliminary macro assembler project this issue with some examples demonstrating its use.

In the last issue, I simply outlined the purpose and gave some general guidelines for the use of the MBASIC-80 CP/M Preliminary Macro Assembler. This time I intend to demonstrate the use of the assembler by using a demonstration macro library typical of the sort that would be constructed for general use. It contains some instances of imbedded macros and multiple LOCAL statements. Listings 3 and 4 are two different examples of the same routine, but there are important differences: Listing 3 contains some instances of real parameters replaced by the mark (#), thus bringing into action the IF NOT NULL condition, whereas the version in listing 4 contains literal parameters. Listing 5 is the routine which is generated by our assembler, replacing all of the macro calls as appropriate before treatment by the CP/M assembler. Listing 6 is the output from the routine in listing 4 after treatment by the assembler.

THE ASSEMBLER AT WORK

To engage the services of our assembler, issue the command

MBASIC MAC/F:9

The assembler will call for the name of the main file which can be entered with or without extension. The extension, if present will be deleted. MAC will be added to form the main file name, ASM will be added to form the output file name.

The assembler will next ask for the library file name. If there is more than one library in existence, you will be permitted to use one library only — MBASIC will search for the main file and library. If either the file or library is undetected, a corresponding message will be displayed and the program will be aborted. If a file is found to have the same name as the output file, it will be erased and a new, empty, file will be created.

The assembler then starts its work, albeit slowly. Although BASIC is notoriously slow, frequent calls are made to the disc in order to fetch macros.

Eventually, the 'OK' message will be displayed and the user will have an output file (identified by the extension ASM) ready for treatment by the resident assembler.

Incidently, the routine used as a demonstration here can be used as a substitute for the CP/M command <TYPE>. However, the routine can be saved and called from within another program such as Wordstar whereas <TYPE> could not.

Listing 1: demonstration macro library

```
;;TEST MACRO LIBRARY
;;28-SEP-1984
;;MACROS IN THIS LIBRARY
;;START
;;OPEN
;;READ
;;PDMA
;;CLOSE
;;SFUN
;;
START   MACRO   TXT
;;THIS MACRO DEFINES IMPORTANT
;;LABELS AND STARTS A PROGRAM
;;TXT IS WRITTEN WITHOUT QUOTES TO DEFINE
;;A VERSION NUMBER, NAME, OR COPYRIGHT.
;;IF TXT IS NOT REQUIRED PUT  &  INSTEAD
;;
BOOT    EQU     0
FALSE   EQU     0
TRUE    EQU     NOT FALSE
NULL    EQU     FALSE
BDOS    EQU     5
LF      EQU     10
CR      EQU     13
EOF     EQU     1AH
SPC     EQU     32
FCB     EQU     5CH
BUFF    EQU     80H
TPA     EQU     100H
        ORG     TPA
        SPHL
        PUSH    H
        JMP     START
        DB      'TXT'
START:
        ENDM
;;
OPEN    MACRO   FCBX
;;TO OPEN A FILE WITH A PARAMETER
;;DEFINING THE FILE CONTROL BLOCK
;;IF FCBX IS NOT REQUIRED PUT  #  INSTEAD
;;THE STANDARD FCB AT 5CH WILL THEN BE USED
;;
        LOCAL   OPENO
        IF      NOT FCBX
        XRA     A
        STA     FCB+12
        STA     FCB+32
        LXI     D,FCB
        JMP     OPENO
        ENDIF
        XRA     A
        STA     FCBX+12
        STA     FCBX+32
        LXI     D,FCBX
OPENO   MVI     C,15
        CALL    BDOS
        INR     A
                ENDM
;;
READ    MACRO   BUFFX   FCBX
;;TO READ THE FILE IN THE BUFFER
;;BUFFX FROM THE FCB=FCBX
;;IF EITHER BUFFX OR FCBX IS TO BE
;;OMITTED USE  #  INSTEAD TO GET THE
;;STANDARD CP/M VARIABLE
;;IN THIS MACRO WE DEMONSTRATE THE USE OF
;;MULTIPLE LOCAL STATEMENT
;;
```

```
        LOCAL   READO
        IF      NOT BUFFX
        LXI     D,BUFF
        JMP     READO
        ENDIF
        LXI     D,BUFFX ;SET DMA ADDRESS
READO:
MACRO   SFUN    26
        LOCAL   READ1 READ2 READ3
READ3   IF      NOT FCBX
        LXI     D,FCB
        JMP     READ1
        ENDIF
        LXI     D,FCBX
READ1:
MACRO   SFUN    20
        ORA     A       ;IS END OF FILE?
        JNZ     READ2   ;YES    END
MACRO   PDMA    BUFFX
        JMP     READ3   ;REPEAT
READ2:
        ENDM
;;
PDMA    MACRO   BUFFX
;;TO PRINT A BUFFER CONTENTS
;;IF BUFFX IS NOT REQUIRED USE  #  INSTEAD
;;THE STANDARD BUFFER AT 80H WILL BE TAKEN
;;
        LOCAL   PDMA PDMB PDMC

        IF      NOT BUFFX
        LXI     B,BUFF+80H
        LXI     H,BUFF
        JMP     PDMA
        ENDIF
        LXI     B,BUFFX+80H
        LXI     H,BUFFX
PDMA    MOV     A,M
        CPI     EOF     ;IS END OF FILE?
        JZ      PDMB    ;YES END
        MOV     A,H
        CMP     B       ;IS END OF EXTENT
        JNZ     PDMC    ;NO CONTINUE
        MOV     A,L
        CMP     C       ;IS END OF EXTENT
        JZ      PDMB    ;YES END
PDMC    MOV     E,M     ;NO GET ONE CHARACTER
        INX     H
MACRO   SFUN    2
        JMP     PDMA
PDMB:
        ENDM
;;
CLOSE   MACRO   FCBX
;;TO CLOSE A FILE
;;IF FCBX IS NOT REQUIRED USE  #  INSTEAD
;;THE STANDARD FCB AT 5CH WILL THEN BE TAKEN
;;
        LOCAL   CLS1
        IF      NOT FCBX
        LXI     D,FCB
        JMP     CLS1
        ENDIF
        LXI     D,FCBX
CLS1:
MACRO   SFUN    16
        ENDM
;;
SFUN    MACRO   NUM
;;TO USE A STANDARD BDOS FUNCTION
;;NUM MUST NEVER BE OMITTED
;;
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,NUM
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        ENDM
```

Listing 2: main routine version 1

```
MACRO   START   F.ELHIDDINY-1984
MACRO   OPEN    #
        JZ      FNL
MACRO   READ    # #
MACRO   CLOSE   #
        JMP     FIN
FNL     LXI     D,MSG
MACRO   SFUN    9
        MVI     E,CR
MACRO   SFUN    2
        MVI     E,LF
MACRO   SFUN    2
        JMP     FIN
MSG     DB      'FILE NOT FOUND$'
FIN     RET
        END
```

Listing 3: main routine version 2

```
MACRO   START   &
MACRO   OPEN    FCB
        JZ      FNL
MACRO   READ    BUFF FCB
MACRO   CLOSE   FCB
        JMP     FIN
FNL     LXI     D,MSG
MACRO   SFUN    9
        MVI     E,CR
MACRO   SFUN    2
        MVI     E,LF
MACRO   SFUN    2
        JMP     FIN
MSG     DB      'FILE NOT FOUND$'
FIN     RET
        END
```

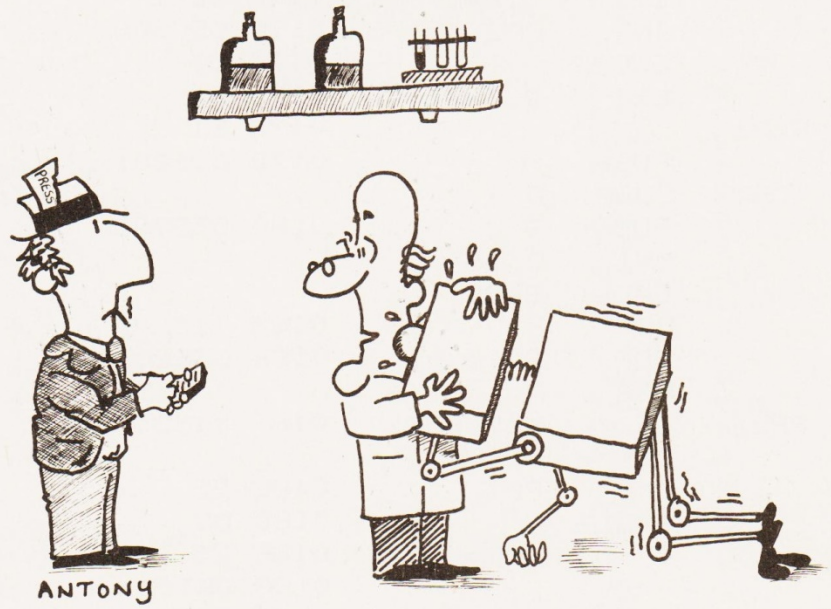
Listing 4: generated code from version 1

```
BOOT    EQU     0
FALSE   EQU     0
TRUE    EQU     NOT FALSE
NULL    EQU     FALSE
BDOS    EQU     5
LF      EQU     10
CR      EQU     13
EOF     EQU     1AH
SPC     EQU     32
FCB     EQU     5CH
BUFF    EQU     80H
TPA     EQU     100H
        ORG     TPA
        SPHL
        PUSH    H
        JMP     START
        DB      'F.ELHIDDINY-1984'
START:
        IF      NOT NULL
        XRA     A
        STA     FCB+12
        STA     FCB+32
        LXI     D,FCB
        JMP     PP1
        ENDIF
        XRA     A
        STA     NULL+12
        STA     NULL+32
        LXI     D,NULL
PP1     MVI     C,15
        CALL    BDOS
        INR     A
        JZ      FNL
        IF      NOT NULL
        LXI     D,BUFF
        JMP     PP2
        ENDIF
        LXI     D,NULL
PP2:
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,26
        CALL    BDOS
        POP     B
        POP     D
        POP     H
PP5     IF      NOT NULL
        LXI     D,FCB
        JMP     PP3
        ENDIF
        LXI     D,NULL
PP3:
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,20
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        ORA     A
        JNZ     PP4

        IF      NOT NULL
        LXI     B,BUFF+80H
        LXI     H,BUFF
        JMP     PP6
        ENDIF
        LXI     B,NULL+80H
        LXI     H,NULL
PP6     MOV     A,M
        CPI     EOF
        JZ      PP7
        MOV     A,H
        CMP     B
        JNZ     PP8
        MOV     A,L
        CMP     C
        JZ      PP7
PP8     MOV     E,M
        INX     H
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,2
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        JMP     PP6
PP7:
        JMP     PP5
PP4:
        IF      NOT NULL
        LXI     D,FCB
        JMP     PP9
        ENDIF
        LXI     D,NULL
PP9:
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,16
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        JMP     FIN
FNL     LXI     D,MSG
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,9
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        MVI     E,CR
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,2
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        MVI     E,LF
        PUSH    H
        PUSH    D
        PUSH    B
        MVI     C,2
        CALL    BDOS
        POP     B
        POP     D
        POP     H
        JMP     FIN
MSG     DB      'FILE NOT FOUND$'
FIN     RET
        END
```

Listing 5: generated code from version 2

```
0000 =          BOOT    EQU     0
0000 =          FALSE   EQU     0
FFFF =          TRUE    EQU     NOT FALSE
0000 =          NULL    EQU     FALSE
0005 =          BDOS    EQU     5
000A =          LF      EQU     10
000D =          CR      EQU     13
001A =          EOF     EQU     1AH
0020 =          SPC     EQU     32
005C =          FCB     EQU     5CH
0080 =          BUFF    EQU     80H
0100 =          TPA     EQU     100H
0100                    ORG     TPA
0100 F9                 SPHL
0101 E5                 PUSH    H
0102 C30501             JMP     START
                        DB      ''
                START:
                        IF      NOT FCB
0105 AF                 XRA     A
0106 326800             STA     FCB+12
0109 327C00             STA     FCB+32
010C 115C00             LXI     D,FCB
010F C31C01             JMP     PP1
                        ENDIF
0112 AF                 XRA     A
0113 326800             STA     FCB+12
0116 327C00             STA     FCB+32
0119 115C00             LXI     D,FCB
011C 0E0F       PP1     MVI     C,15
011E CD0500             CALL    BDOS
0121 3C                 INR     A
0122 CA9A01             JZ      FNL
                        IF      NOT BUFF
0125 118000             LXI     D,BUFF
0128 C32E01             JMP     PP2
                        ENDIF
012B 118000             LXI     D,BUFF
                PP2:
012E E5                 PUSH    H
012F D5                 PUSH    D
0130 C5                 PUSH    B
0131 0E1A               MVI     C,26
0133 CD0500             CALL    BDOS
0136 C1                 POP     B
0137 D1                 POP     D
0138 E1                 POP     H
                PP5     IF      NOT FCB
0139 115C00             LXI     D,FCB
013C C34201             JMP     PP3
                        ENDIF
013F 115C00             LXI     D,FCB
                PP3:
0142 E5                 PUSH    H
0143 D5                 PUSH    D
0144 C5                 PUSH    B
0145 0E14               MVI     C,20
0147 CD0500             CALL    BDOS
014A C1                 POP     B
014B D1                 POP     D
014C E1                 POP     H
014D B7                 ORA     A
014E C28301             JNZ     PP4

                        IF      NOT BUFF
0151 010001             LXI     B,BUFF+80H
0154 218000             LXI     H,BUFF
0157 C36001             JMP     PP6
                        ENDIF
015A 010001             LXI     B,BUFF+80H
015D 218000             LXI     H,BUFF
0160 7E         PP6     MOV     A,M
0161 FE1A               CPI     EOF
0163 CA8001             JZ      PP7
0166 7C                 MOV     A,H
0167 B8                 CMP     B
0168 C27001             JNZ     PP8
016B 7D                 MOV     A,L
016C B9                 CMP     C
016D CA8001             JZ      PP7
0170 5E         PP8     MOV     E,M
0171 23                 INX     H
0172 E5                 PUSH    H
0173 D5                 PUSH    D
0174 C5                 PUSH    B
0175 0E02               MVI     C,2
0177 CD0500             CALL    BDOS
017A C1                 POP     B
017B D1                 POP     D
017C E1                 POP     H
017D C36001             JMP     PP6
                PP7:
0180 C33901             JMP     PP5
                PP4:
                        IF      NOT FCB
0183 115C00             LXI     D,FCB
0186 C38C01             JMP     PP9
                        ENDIF
0189 115C00             LXI     D,FCB
                PP9:
018C E5                 PUSH    H
018D D5                 PUSH    D
018E C5                 PUSH    B
018F 0E10               MVI     C,16
0191 CD0500             CALL    BDOS
0194 C1                 POP     B
```

```
0195 D1                    POP     D
0196 E1                    POP     H
0197 C3D401                JMP     FIN
019A 11C501      FNL       LXI     D,MSG
019D E5                    PUSH    H
019E D5                    PUSH    D
019F C5                    PUSH    B
01A0 0E09                  MVI     C,9
01A2 CD0500                CALL    BDOS
01A5 C1                    POP     B
01A6 D1                    POP     D
01A7 E1                    POP     H
01A8 1E0D                  MVI     E,CR
01AA E5                    PUSH    H
01AB D5                    PUSH    D
01AC C5                    PUSH    B
01AD 0E02                  MVI     C,2
01AF CD0500                CALL    BDOS
01B2 C1                    POP     B
01B3 D1                    POP     D
01B4 E1                    POP     H
01B5 1E0A                  MVI     E,LF
01B7 E5                    PUSH    H
01B8 D5                    PUSH    D
01B9 C5                    PUSH    B
01BA 0E02                  MVI     C,2
01BC CD0500                CALL    BDOS
01BF C1                    POP     B
01C0 D1                    POP     D
01C1 E1                    POP     H
01C2 C3D401                JMP     FIN
01C5 46494C4520MSG         DB      'FILE NOT FOUND$'
01D4 C9          FIN       RET
01D5                       END
```



"YOU SHOULDN'T HAVE CALLED HIM 'TINHEAD', HE'S VERY SENSITIVE"

THE SOUND OF MUSIC

Bill Horne

The science behind an art.

Music can be treated as an art or as a science, and both aspects need to be taken into account when music is created on an automatic basis. In principle, a musical box is a very simple mechanism, but the tiny pegs set in the cylinder or disc must be positioned with mathematical precision if they are to pluck the reeds at exactly the right moment, and the reeds must be of the exact size needed to produce the required tones.

By comparison, a system generating sounds under computer control is complex, with many additional variable factors to be taken into account. Yet user manuals for some recent computers suggest that music can be created quite easily. This might be acceptable if the word music were enclosed in inverted commas, suggesting something broadly related to music, but the sounds produced only too often emphasise the scientific aspect of music, lacking an adequate leavening of art.

Of necessity, the manuals must assume that their readers have some understanding of music, and must concentrate on the workings of the computer system, but for those who have limited musical knowledge a little more help is useful. These notes seek to provide that help.

PITCH

Musical pitch is a more complex subject than is immediately apparent. It need scarcely be said that pitch is related to frequency, but it may be less obvious that frequency ratios are of more importance than absolute frequency values. Some people can recognise the absolute pitch of a tone, but many more can only recognise pitch intervals.

As far back as the days of Pythagoras it was recognised that tones with frequencies in simple ratios could be combined to give pleasant concords. Particular attention was paid to the 'tetrachord', which consisted of four tones with frequencies in the ratios 4:5:6:8. Later, it was seen that three tetrachords could be interlinked, forming what we now know as a musical scale (Fig. 1).

If this process is extended further, in an attempt to define a tetrachord based on D, two things happen. The second note is found to have a ratio equal to 27*30/24 = 33¾. That lies between F and G, and is a 'black note' identified as F#. A more difficult situation arises with the third note, which is 27*36/24 = 40½, where A has already been defined as 40.

If all twelve tetrachords are defined in this way, it will be found that each note has two different values. For a given key signature, it is necessary to select eight of the twelve available notes and allocate the correct values to them. This is possible for a string quartet, the instruments of which can be made to give an infinite gradation of pitch, but a keyboard instrument is a different matter. There was an attempt to build one with fifty-six notes to the octave, the notes being arranged in rows appropriate to given keys, but players gave that up as impractical. The solution usually adopted was to tune for key signatures having a small number of sharps or flats, which gave good results in the scale of C, but increasingly bad results as sharps or flats were added, until the remote signatures became known as the 'wolf keys' from the way they howled.

Art could offer no solution, but science could. To the logical mind, there were twelve intervals of a semitone in an octave, and between them they spanned a ratio of two to one. What would happen if all the intervals were made equal to 12/2 ? The answer is shown in Fig 2, which compares the ratios based on tetrachords with the ratios given by 'equal temperament'. C is taken as a base, and remains at 24, but all the other pitches are slightly in error.

Bach evidently thought the result acceptable, since he composed his Forty-Eight Preludes and Fugues for the Well-Tempered Clavier, using all the twelve major and minor keys. There have been suggestions that the intention was to advertise the merits of the new system, in which case it was the most notable commercial of all time!

Purists still preferred the perfect intonation of a string quartet, but even that could run into problems with music that confused sharps with flats, moving artfully from D sharp to E flat. These notes are identical on a keyboard, but not in a perfect pitch situation...

But what does this all mean to those who want computers to sing? The usual approach adopted by recent systems is based on equal temperament, which makes for simplicity, but why not aim higher? If we can recognise the appropriate base note for a tetrachord, we can generate perfect concords.

This process needs to be approached with care. Apart from the simplest of tunes, notes foreign to the tetrachord are likely to arise. The beginning of 'Three Blind Mice' uses the notes, C, D and E. C and E lie in the tetrachord CEGC', but D does not. It is a 'passing note', foreign to the main chord but acceptable because it is used transiently between two notes of the chord. However, the notes C and E identify the accompanying tetrachord.

Experiment in this area is quite rewarding, because it can show up the implied structure of the music, but for those who merely want to produce musical sounds the equal temperament basis is best. Some systems incorporate a table of equal temperament pitches. The BBC Computer goes further, allowing quarter-semitones to be defined by the use of a difference table, but some tables in other systems

C . D . E F . G . A . B C'
24 30 36 48
32 40 48
27 36 45

Figure 1: Using the tetrachords CDEG', GBD'G' and FAC'F' to form the scale of C

C	24	24.00
.		25.43
D	27	26.94
.		28.54
E	30	30.24
F	32	32.04
.		33.94
G	36	35.96
.		38.10
A	40	40.36
.		42.76
B	45	45.31
C	48	48.00

Figure 2: The true ratios, based on the tetrachord, compared with the 'equal temperament' ratios.

are less than ideal. The usual procedure is to define one octave and multiply or divide by powers of two to obtain other octaves.

PITCH GENERATION

Having decided on the pitch of the required note, it is then necessary to work out the number to be passed to the sound generating system. Most sound generator chips are driven by a clock signal, which is divided down internally to produce a Master Frequency. The data fed to the chip to define pitch is set in a down counter driven by the Master Clock, and is reset when the count reaches zero. The data thus refers to cycle duration, rather than frequency, and the required value is the Master Frequency divided by the required pitch.

When the down counter reaches zero, the audio output level changes between low and high, so the cycle period defined leads to a half-cycle time. This needs watching. Some years ago, two electronic organ builders put their instruments side by sidee, and found that one had a pitch an octave higher than the other. An octave difference in pitch is not always obvious in isolation from a standard, which may explain why a certain computer appeared to be giving notes an octave lower than the user manual stated...

Mention of electronic organs is a reminder that 'tremolo' was introduced to disguise errors of pitch, some of which arose from the use of a single chip to generate all twelve tones in the octave by dividing down from a master frequency. The tone envelope principle serves a similar purpose. Quite apart from the errors introduced by equal temperament, there are additional errors caused by the dividing down process, and these errors do not always tend to cancel each other out.

For experiment with musical sounds, it is useful to set up a table of equal temperament ratios as a starting basis, and derive pitch control values as required. It is also useful to set up a table giving the pitch intervals for a given key signature. These ideas are illustrated in Listing 1, which is for the AMSTRAD CPC464, but can be adapted for other computers.

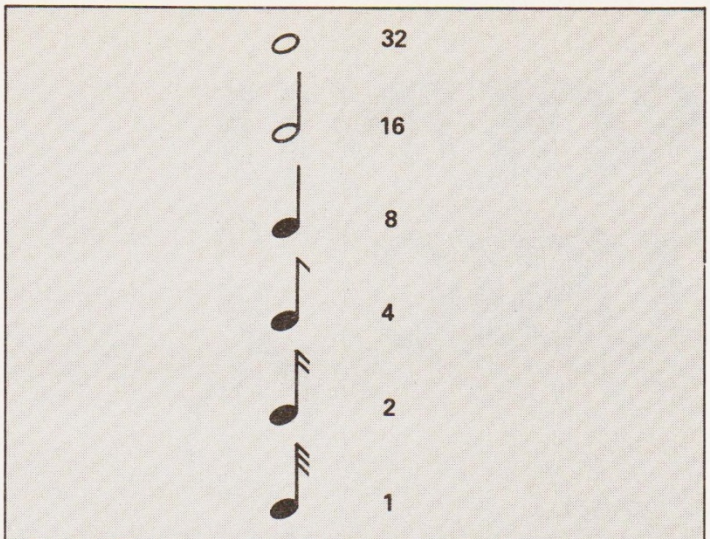


Figure 3: Relative durations for musical note symbols

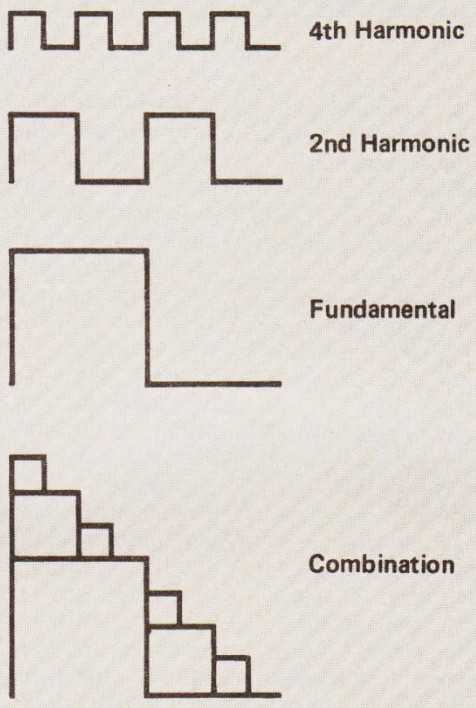


Figure 4: Building a sawtooth from square-waves

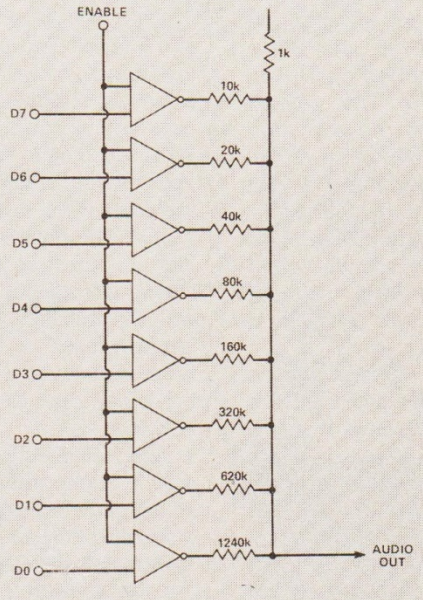


Fig. 5. Simple Digital/Analogue Converter for Music Generation, using eight open-collector NAND gates.

Line 90 sets up an envelope, which can be varied without affecting anything else. Then a = 3322, the master 'period value', while b is set to the twelfth root of two, the semitone interval. The FOR loop in lines 130-160 sets up the equal temperament ratios in array d, and the second FOR loop sets the data in array E. We will return to that in a moment.

The main running loop begins at 240 with an invitation to define an octave by input of a number (0 to 8). An inner loop through 250 is then entered, and pressing keys CDEFGABH will generate a corresponding tone (H is C').

The ASCII code for the key pressed is converted to a number by subtracting 65. If a numeric key is pressed, the invitation to define an octave is repeated. If the key is not otherwise in the A-H range it is ignored.

The point of the array E can now be seen. Pressing key A calls for the ninth semitone in the octave. The pitch data for that is looked up in array d, rounded, divided by 2^X, where X is the required octave, and incorporated in the SOUND command.

It would be possible to add another array for interpreting a row of keys to supply the input numbers, but as it stands the program illustrates a number of points about pitch, and the imperfections thereof. Octaves 0 and 1 are unsatisfactory. Octave 8 is partly inaudible. The routine is a useful experimental tool.

DURATION

Some tunes used in computer games are barely recognisable, mainly because the duration of the notes is all wrong. Take 'Three Blind Mice'. The first bar has four beats, in the ratio 1:1:2. No problem there, but in the third bar these beat durations are subdivided, and the durations are in the ratio 3:2:1:6. It could be said that each bar really has twelve beats, and the pattern should be

3:3:6
3:3:6
3:2:1:6
3:2:1:5:1
2:1:1:1:1:2:1:2:1
1:1:1:1:1:1:2:1:2:1
1:1:1:1:1:1:1:1:1:2:1
3:3:6

Listing 1

```
90 ENV 1,15,-1,15
100 DIM d(12)
110 a=3322
120 b=2^(1/12)
130 FOR c=0 TO 12
140 d(c)=ROUND(a)
150 a=a/b
160 NEXT
200 FOR x=0 TO 7
210 READ E(x)
220 NEXT
230 DATA 9,11,0,2,4,5,7,12
240 INPUT "octave";o
250 a$=INKEY$:IF a$="" THEN 250
260 b%=ASC(a$)-65
270 IF b%<0 THEN 240
280 IF b%>7 THEN 250
290 f%=e(b%)
300 SOUND 129,ROUND(d(f%)/(2^o)),0,0,1
310 GOTO 250
```

Perhaps the terminating 6 should be 3:3, with the second period silent, and the 5:1 termination should be 3:2:1, but the written music says otherwise.

The point is that most music has a standard beat pattern repeated in each bar. The time-signature at the start of a piece only gives the major beats. For Three Blind Mice it is 4/4, four beats in a bar, but each beat has up to three sub-beats, giving the overall twelve. Until you have worked out the overall beat, you may not be able to determine the individual note and rest durations.

For those who are less familiar with music, the duration table in Fig 3 may be useful. Each note symbol defines a pitch and a duration, while the rest of the symbols define duration. A dot following either type of symbol increases the duration by 50%.

TIMBRE

The timbre or tone of musical notes is a matter of waveform. The type of generator commonly provided in a small computer producers square waves, which have a timbre broadly related to that of the clarinet. Unfortunately, the clarinet — while a beautiful solo instrument — does not blend well with other clarinets, and this effect is emphasised when an attempt is made to produce chords with the square-wave generator.

However, electronic organs use square waves as a basis for synthesising various sounds, so a similar process may be applicable here. A low pass filter which attenuates the harmoncis of a square wave (they are the odd harmonics, 3, 5, 7 etc) can produce something approaching the flute sound, a simple sine wave. A high pass filter emphasises the harmonics, and approaches a string sound, because the violin, for example, is really too small to resonate properly with the fundamentals of its lower notes. This effect is improved if the filtered tone has all the harmonics, not just the odd ones.

No problem. Mix a fundamental tone with another tone an octave higher and of half the strength, and the 2nd, 4th, 6th and so on harmonics are filled in. Fig 4 shows how this process can be continued, synthesising an approximation to a saw-tooth wave.

For brass instruments, a band-pass filter is needed, preferably with a saw-tooth input. For the trumpet, differentiating the sawtooth may help.

So quite a variety of tones can be generated by combination and filtering, using the computer sound output as a basis, rather than using it directly. Your computer only feeds sound to an internal loudspeaker? Bad luck. Unless you can bring out the sound chip output, your options are limited.

However, there is another aspect to timbre: The overall sound envelope. Some envelope control systems are of enormous flexibility, but are also so complex that using them is a pain. Nevertheless, they can do much to impart character to a note, and are worth a little experiment.

The main objection to envelope control is that some versions produce quite audible clicks as fresh data is fed to the sound chip. For a musical purist, this can be the last straw. He turns his back on the sound system provided, and looks for something better.

A SOFTWARE APPROACH

An American called Howard Arrington showed the alternative path. First, he set up a table of sine wave values over the amplitude range 0-&3F. Then he set up a variable — call it X. By adding X repeatedly to a pointer to the sine table, he was able to read off sample values at varying rates. The bigger he made X, the more quickly the pointer scanned.

The next stage was to make a very simple digital to analogue converter (Fig 5), which could be driven by a printer port. The values from the sine table were fed to the port, and the converter provided a sine wave tone. So simple. It only needed a table of X values and durations, and he could play a tune. For best results the routines needed to be in machine code, but that was not impossible.

That was only the beginning. Before long, the sine table was replaced by a table based on SIN(A) + ½*SIN(2A), giving a second harmonic content. This could not be taken too far, since with the higher notes only a few samples were taken during each pass through the table, and the output waveform was not too detailed, but the tone was a lot more pleasant than a square wave.

The labour of producing the data tables for a lengthy piece came under attention next, and the answer was a basic routine which displayed musical staves and all the usual note values. By moving pointers which picked out a given note duration, and picked out the require note position on the stave, it was possible to set up the database quite rapidly, and there was provision for playing sections of the tune at any time to check progress and accuracy.

But why was the original sine table limited to 0-&3F in amplitude? Well, that meant it was possible to pick up four samples at different rates, add them together, and generate four-part harmony without exceeding &FF. The hardware generator, limited to three tones, cannot compete with that!

In all seriousness, the Arrington method works and works well. It appeared in 1980, and has since been copied to some extent by others, but the originator deserves the credit.

CONCLUSION

Computer-generated music is a great field for experiment, but only if some restraints are avoided. Keep your programmable sound generators for space-battle effects, but try a wider range of ideas if you want real music.

This has necessarily been a fairly superficial tour of the subject. We might have considered some of the fascinating aspects of harmony, like the seventh interval (e.g. G to F'), which is not quite perfect, pulling the upper note down to E, and so giving the full close that is an inherent part of classical music.

But enough is enough. Let's leave it there, and hope that we may have encouraged others to experiment. If you have bright ideas that work, let us know, and we'll continue the subject.

ORANGE WITH STABILITY

Mick Seymour

A problem associated with putting anything onto a Dragon 32's text screen, is that it always reverts to a green display. We describe a way around this.

An annoying problem with putting anything on the DRAGON's text screen is that it always reverts to a green display. This is because prior to printing a character the machine is reset to text mode by a call to $A93A from $B58F. Part of this routine resets all but bits 0,1 & 2 of the contents of $FF22 (65314), which is the address holding the graphic mode selection data. Bit 3 governs the colour set to be used. For the text screen, 0 denotes green and 1, orange, therefore the screen always returns to green.

One way around the problem is to insert a SCREEN 0,1 statement after each PRINT statement but this causes the screen to flicker as it reverts to green momentarily.

A more satisfactory method is to prevent bit 3 at $FF22 from being reset in the first place, except by a SCREEN 0,0 statement. It is this solution I present here.

Most DRAGON users are by now probably aware that a number of ROM routines pass through RAM in order to redirect the routine through a disc operating system if one is fitted. If there is no DOS these vectors contain the machine code RTS instruction. We can, using these vectors, redirect a ROM routine through one of our own so that the machine does what we want in certain cases, rather than what the designers intended.

The vector called by the PRINT routine from $B54A lies in three bytess starting at $0167. Storing a jump to my routine here will pass control to my version of the DRAGON's code, which will leave the colourset bit intact. Control is then passed back to the ROM to an address after the call to the DRAGON's version.

The code is fully relocatable and may be either POKEd into position using the BASIC loader in listing 1 or assembled using listing 2. Either way, the start address should be $7FAA (32682) or lower, and a suitable space reserved for it using CLEAR. Note that the BASIC loader CLEAR's from $4E20 (20000) but this may be altered if required.

SAVING THE SOURCE

It is always good practice to save the assembler source code or BASIC loader prior to running, and the object code, once installed, before it is executed. This is especially important with machine code as it can do very strange unexpected things if typed in wrongly. To save the code to cassette use CSAVEM "ORANGE",start, end,exec where start is the loading address, end=start +85, and exec=start as this is the executable address for switching the routine on. If you use DRAGONDOS 1.0, save by SAVE "ORANGE",start, end+1,exec following the same rules as above.

Once saved, if you are using discs, type CLOSE and remove them from the drives. This will protect them from possible corruption should the program

have been incorrectly typed in. Once the routine is tested and working you can of course use it with discs in the drives.

Type EXEC and the start address. It will appear that nothing has happened but in fact the data at $0167-$0169 will have been stored in data bytes within the program. In its place will be the jump to the program. Now type SCREEN 0,1 (or SCREEN,1 which is also accepted by BASIC). The screen will change to orange and stay that way. Type SCREEN 0,0 (or SCREEN,0) and it will revert to green. EXEC and the start+25 will reinstate the original contents of $0167-$0169 to enable the vector to be used by another user program if required.

The program first checks that it is the screen which is being printed to and not the cassette or printer. If this condition is not met control is passed straight back to the ROM, via the previously stored data bytes. If however the condition is met a branch to the subroutine originally called by the ROM is performed, which will return to the routine.

On returning, the machine is set to text mode, but leaving bit 3 at $FF22 intact. The stack pointer is then increased by 2 to discard the original return address of $B54D and a direct jump is made to the ROM at $B592 to enable the character to be printed.

It is worth noting that if a return to text mode is made from a high-res mode, by printing to the screen, the colour set (0 or 1) will remain as it was set for the high-res mode unless the alternative is specified before or after printing.

If you are running my Screen Print Reversal program (Computing Today, Oct. '84) which also uses the vector at $0167, it is possible to use both programs together to obtain an inverse orange character set. This must be carried out as follows:

- Using a start address of ay) $7ECA, CLEAR 200, &H7ECA
- Cassette users CLOADM "SCREEN" from $7F20 or use its BASIC loader with that as the start address. DRAGONDOS users LOAD "SCREEN.BIN"
- CLOADM or LOAD this program from $7ECA by CLOAD "ORANGE" or LOAD"ORANGE.BIN" and the offset is necessary, or again, use the BASIC loader with $7ECA as the start.
- EXEC &H7ECA will point the PRINT routine to the ORANGE program, which terminates in the jump to the ROM.
- EXEC &H7F20 will now point the PRINT routine to SCREEN which will now terminate in a jump to the ORANGE program.

You now have the facilities of both available to you. They must, of course, be taken out of the PRINT vector in reverse order to restore the jumps correctly, ie. EXEC &H7F52 then EXEC &H7EE2. I will leave it to the adventurous among you to integrate the two into one program if you wish. This is a relatively easy task as the main assembler labels, checks, and jumps are similar in both cases.

```
1 '          DRAGON 32
2 '   SCREEN 0,0 OR 0,1 HOLD
3 '      FULLY RELOCATABLE
4 '
5 '    WRITTEN BY M.A.SEYMOUR
6 ' VERSION 1:13 JANUARY 1985
7 '
10 CLEAR200,20000:CLS
20 INPUT "START ADDRESS";ST
30 IF ST>&H7FAA THEN PRINT "TOO HIGH!":GOTO 20
40 IF ST<20000 THEN PRINT ST"IS NOT PROTECTED!":GOTO 20
50 READ D$:IF D$="END" THEN 100
60 D=VAL("&H"+LEFT$(D$,2))
70 POKE ST,D
80 C=C+D:ST=ST+1
90 IF LEN(D$)>2 THEN D$=MID$(D$,3):GOTO 60 ELSE 50
100 IF C=8870 THEN 180 ELSE PRINT "CHECKSUM ="C"WHICH IS WRONG"
110 PRINT "PLEASE CHECK THE LISTING":END
120 DATA B60167,A78C4D,FC0168,ED8C48,308C16,BF0168,867E,B70167,39
130 DATA A68C38,B70167,EC8C33,FD0168,39
140 DATA 0D6F,262A,8D28,3412,8EFFC8,A70A,A708,A706,A704,A702
150 DATA A701,A71E,A71C,A71A,A718,B6FF22,840E,B7FF22,3512
160 DATA 3262,7EB592,39,3939
170 DATA END
180 PRINT"END  ADDRESS ="ST-1,"EXEC ADDRESS ="ST-86
190 PRINT"EXEC OFF     ="ST-62
```

```
                                        LISTING2
7FAA  7FAA              ORG    $7FAA
7FAA                  **********************************
7FAA                  *                                *
7FAA                  *    SCREEN 0,0 OR 0,1 HOLD      *
7FAA                  *      FOR THE DRAGON 32         *
7FAA                  *      FULLY RELOCATABLE         *
7FAA                  *                                *
7FAA                  *    WRITTEN BY M.A.SEYMOUR      *
7FAA                  *  VERSION 1:13 JANUARY 1985     *
7FAA                  *                                *
7FAA                  **********************************
7FAA B60167           XECON  LDA    $167      ;PRNT VECT
7FAD A78C4D                  STA    STR1,PC
7FB0 FC0168                  LDD    $168
7FB3 ED8C48                  STD    STR2,PC
7FB6 308C16                  LEAX   ONOFF,PC
7FB9 BF0168                  STX    $168
7FBC 867E                    LDA    £$7E      ;JMP CODE
7FBE B70167                  STA    $167
7FC1 39                      RTS
7FC2                  *
7FC2 A68C38           XECOFF LDA    STR1,PC
7FC5 B70167                  STA    $167
7FC8 EC8C33                  LDD    STR2,PC
7FCB FD0168                  STD    $168
7FCE 39                      RTS
7FCF                  *
7FCF 0D6F             ONOFF  TST    $6F       ;CHANNEL
7FD1 262A                    BNE    STR1      ;NOT 0:OUT
7FD3 8D28                    BSR    STR1      ;DO BUSNES
7FD5                  *
7FD5                  * PUT INTO TEXT MODE
7FD5                  *
7FD5 3412                    PSHS   X,A
7FD7 8EFFC8                  LDX    £$FFC8
7FDA                  *
7FDA                  * WRITE ANYTHING TO RESET IT
7FDA                  *
7FDA A70A                    STA    10,X
7FDC A708                    STA    8,X
7FDE A706                    STA    6,X
7FE0 A704                    STA    4,X
7FE2 A702                    STA    2,X
7FE4 A701                    STA    1,X
7FE6 A71E                    STA    -2,X
7FE8 A71C                    STA    -4,X
7FEA A71A                    STA    -6,X
7FEC A718                    STA    -8,X
7FEE B6FF22                  LDA    $FF22
7FF1                  *
7FF1                  * DO NOT RESET BIT 3 (ROM=£$07)
7FF1                  *
7FF1 840E                    ANDA   £$0E
7FF3 B7FF22                  STA    $FF22
7FF6 3512                    PULS   X,A
7FF8                  *
7FF8                  * DISCARD OLD RTS ADDR.
7FF8                  *
7FF8 3262                    LEAS   2,S
7FFA 7EB592                  JMP    $B592     ;PRINT
7FFD 39               STR1   FCB    $39       ;WILL HOLD
7FFE 3939             STR2   FDB    $3939     ;PRNT VECT
8000
```

HOW MANY USERS ?

Bill Horne

Arguments for and against stand-alone micros in large organisations will continue for some time to come. Here are the pro's and con's.

In the early nineteen-sixties, a well-known pundit pronounced his gospel thus: 'No computer can be an economical proposition unless it is very large and very fast.' What he did not say was that this assumed that there would be multiple users, multi-stream programming, and other features essential to his contention.

The advent of the microcomputer appeared to squash such theories, since systems that were far from large, and sometimes not at all fast, had become a feasible proposition. There was usually only one user, and only one program could be run at a time, but for many kinds of work the cost per user was very low compared with that for a multi-user mainframe. There were things which micros could not do, of course, but there were a surprising number of things which it could handle with comfort, especially if the user knew how to get the best out of it.

Then the old contention began to rear its head again. The coming of enhanced processors allowed store to be extended considerably, and this allowed the bigger micros to undertake additional kinds of work, but it also made them an overkill for many routine tasks. So someone decided that multi-stream programming should be re-introduced in thsi context, allowing multiple users, each with its own dumb terminal.

Meanwhile, the same objective had been approached in a different way, by linking small micros into networks. This allowed communication between users, access to common programs, and other interesting possibilities. The cost per user was slightly increased by the networking equipment, but not to a disproportionate extent.

- The independent microcomputer, for a single user, though perhaps to deal with multiple programs. Cost: Perhaps £200 - £800 would be a useful bracket, with an increase for discs and printer.
- The networked microcomputer, multi-user, multi-program, with data exchange between users. Cost per user perhaps £200 - £300 more than that of independent units, with extra if any units were remote.
- The multi-user microcomputer, running a large operating system designed to emulate mainframe facilities. Cost per user of the order of £4,500.

In comparing these options, we need to consider their limitations. As anyone who has used a large system knows, there is always liable to be a period in which all users but one have to wait, because a large high priority program is being run. Yet in many cases it would be possible to run that program on a separate, dedicated, system, or transfer other users to such system. The cost per user is already high, but is effectively made higher because some users are not getting full value for money.

It could be suggested that part of the problem is that the kind of operating system needed to handle multi-user/multi-programming is so complex that it erodes machine capacity to a significant degree. A single readily-identifiable interrupt can be processed quite quickly, but if there is a need to check priorities and work out a calling sequence the interrupt handler may mop up quite a lot of machine time.

WHAT WORK?

So do we really gain an advantage by setting a small computer to work for several users? We may be able to justify a larger memory, which in turn needs an enhanced processor. That — in theory, but not always in practice — should give increased program execution speed, but against that must be set the time taken by the executive.

The answer must, inevitably, be conditioned by the type of work to be done. If a vast database is involved, the small independent machine will be useless, though it is sometimes worth checking whether the database might be subdivided.

On the other hand, it is worth starting at the bottom, and asking whether a single independent machine can cope with the work to be done, or at least a share of the work. If it can, then — end of story. Why look further?

The next stage crops up when the small machine can do most of the work, but a larger machine is needed for other tasks. It might seem reasonable to get the larger machine to take over some of the other work, but that may degrade the capabilities of the bigger machine for its special task. It must also be noted that a breakdown of the big machine will cause more chaos.

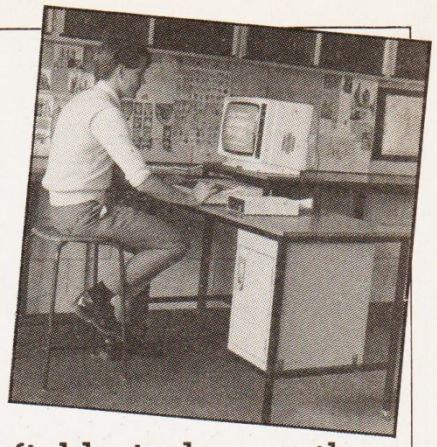
This question of 'down time' is rarely given enough consideration. A larger furniture store had a breakdown on its busiest day, and the sales system fell apart. Makeshift measures were brought into play, but the resulting muddle took ages to sort out. Had it been a system of linked independent computers, the disruption would have been reduced. It is true that modern systems are usually quite reliable, but they are still vulnerable to 'fingerpoken' and mental lapses on the part of the operator. Recovery may be possible if the software has got into a twist, but hardware failure may be more difficult to trace.

Altogether, there is a strong case for keeping simpler computer operations separate. If they are put into the same box, they cease to be simple. Not everyone would agree with this. At a recent TABS National Dealer Conference, Stephen Vitadis, managing director of Raindrop Computers, spoke out forcefully in favour of multi-user machines with up to six terminals, decrying networked single-user machines as unsuitable for the market needs. The multi-user system, he claimed, was faster and more cost-effective.

At the same time, he suggested that a combination of multi-user systems coupled into a network was worth consideration, allowing a useful expansion route from a single multi-user system.

To put these views into context, Mr Vitadis was speaking of a market for systems of limited size for first-time users, particularly users that were small to medium-sized companies. Such customers rarely have a clear idea of what they want their computers to do, and often prefer to approach computerisation with caution. The thought of having up to six trainee operators learning to use a brand new system may not encourage them. In the long run, the multi-operator approach may suit them very well, but it can be implemented in the networked form as well as in the multi-user form. For some kinds of application, Mr Vitadis is undoubtedly right, but we would not back his views unconditionally on a wider basis.

HIGH-TECH HIGH SCHOOL



Richard Porch

The current skill shortage in the computer and related fields, is due partly to a school curriculum that has not changed to meet the demands of its society. Here, Richard Porch offers his predictions on the future of our education system.

Those of us who are content to use a microcomputer privately may see no reason to worry about what micros may do to the world in general. Others take a different view. Their predicdtions may be off target, but they serve to show some of the reactions which the computer can produce. They may some times look like science fiction, but SF has a disconcerting way of anticipating fact...

TAKE A GOOD look at the schools around you now, because the chances are that they are going to radically change over the next 10-15 years. Schools will become factories for the job market, and may to some extent begin to look like factories. Computers will play a larger and larger role, and while robot-tutors may belong more the realms of science fition, teaching via computers will become a reality. The blackboard and desk learning situations will diminish, and in will come instead the video lecture and the VDU console with pupils working on computerised lesson-sequences. With the teachers contribution being one of advice and guidance, rather than tiring — and ultimately repetitious — set-piece lecturing day in, day out, on a given subject.

"The schools of the next 10 to 20 years will really be in the business of preparing their pupils for the job-market and its rigours."

GRADUAL CHANGE

The change from seeing home-micros simply as superior toys, to seeing them as teaching vehicles, will be a gradual one. It is perceptibly under way now, look at how complicated and involved are the latest fantasy and adventure games. Compare them to the early 'spae invader' and tennis games, and the new levels of game complexity are not just down to programs and equipment becoming more sophisticated either. People will steadily demand that home micro's do more for them than just amuse them, in other words, computers will stop being a passing entertainment fad. I feel sure that as programs evolve, education will play a steadily larger role.

For example, everyone is familiar with the car-owner manuals that you can buy to service-check your car with. Imagine if someone could write a program which projected onto the monitor screen a 3-D computer graphic of your car and all its components. You then explained the nature of the fault to the computer via the 'mouse' and cursor, pointing out the defective area/sequence on the appropriate 'menu'. The computer might then respond by producing a graphic printout, giving you a selection of potential troublespots, and suggesting courses of action. Think how this would cut down on service time at the garage...but remember the old programmer's proverb 'Rubbish In — Rubbish Out!' Now take that idea and expand it into education generally. And you would have a tremendously stimulating and interactive way of teaching children how ie, a car, train, plane, nuclear reactor, space shuttle, economy or solar system works. Programs and computers will never replace human teachers, nor should they. But the vast amount of facts and figures they can store in their memories and the visually stimulating graphic presentations they can produce, can only greatly assist the talented teacher.

BEYOND 1990

In the schools of the 1990's and beyond, I feel sure that preparing the pupils for prospective careers will start much earlier. The schools of the next 10-20 years will really be in the business of preparing their pupils for the job-market and its rigours. Large-scale unemployment will be very much of a problem. I also tend to think that the whole structure of the school day might change and more involvement be made of the many home-micro's that will be in almost all homes by then. Like the Open University, schools could broadcast lessons via the television, with a transmission of questions fed into the home micro's memory, to be answered later as homework. The pupil would then produce a response, which his own micro might answer or correct, or he/she might have to print out some hard copy, to be checked by a teacher. Who knows? In a decade's time the home-micro may have been superceded and in its place will be the portable micro. Made to fit inside a satchel (advances in flat-screen technology could soon make this possible). We might then see pupils arriving at school and simply plugging in their portable micro's into desk mounted powerpoints. This is not such a far-fetched idea. All it takes is the required vision, and a desire to make it happen. Using the portable micro scenario, fixed site schools could disappear altogether, and students might move around picking and choosing the power points in the educational network, that they want to plug into (and in theory change school every week!) They would have access to a cybernetic network whose aim was education, except that they could be anywhere in the country, itinerant student wanderers, abroad in a post-industrial culture. The current pressures that the economy is exerting on many people's lives may make this scenario very plausible. The continued run-down of manufacturing industry and the rise of service industry (burger bar employment, office work, factory assembly line employment etc) will mean jobs will be less permanent and vulnerable to fluctuations in what the market wants. People will have to be more mobile (remember 'On your bike?'). If people want jobs, they will, out of sheer

"The pupils who will leave our high tech high school will be very independent creatures and street-wise in the best possible way."

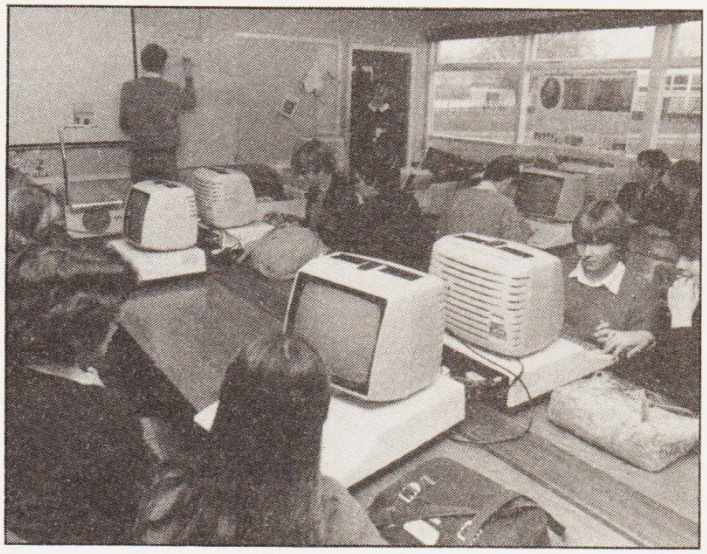
necessity have to accept being more rootless.

The facility therefore of being able to carry all your text books (on floppy disks) and writing material (keyboard and printer) in a convenient form (some sort of plastic satchel) will be invaluable. All you need then is a convenient power source (or batteries) in order to access the educational network, and away you go. A highly mobile workforce, able to roam the country searching for employment. Student wanderers going where they will, to find the best teachers of video lecturing — tutorial service. Their P45 in their back pocket, a tube bag full of spare clothes, with their portable micro slung over their shoulders...the way they carried the old 'ghetto-blaster' stereo's of yesteryear.

THE SHORT TERM

However, for the short term at least, I foresee the school continuing as a fixed institution in the landscape. But even these will not be able to resist the flow of change. They will hum and purr like high-tech factories, as pupils sit at banks of micro computers situated in almost every classroom. Workshop areas that formerly taught woodwork and metalwork will be given over instead to 'dummy' electronics' assembly lines, manufacturing and assembling computers instead of bookcases. When the software houses realise just how much money there is to be made in the educational programs market we'll see another 'boom' period in the home computer field and when schools have more than just the one solitary overworked micro we'll be well on the way to fulfilling this scenario. Pupils will be taught a basic 'core' of subjects such as English language, maths, computer science, a choice from chemistry, physics or biology, and a further choice from history, geography, art of a language. With a typical week comprised of three days compulsory attendance (for the 15-18 year olds) and two days of home-based, micro-taught lessons, I am quite sure that 'modems' will have been superceded in the 1990's by some superior device. Perhaps by a piece of additional equipment that would enable schoolmates (or whole class groups) to 'converse' and bounce around ideas and information via their own home micro's? Who knows? This sort of link-up idea may well be the logical conclusion of what started with the car-bound citizen band radio equipment. To take this one step further, if a satisfactory 'Open University' type video lecture-lesson scheme could be developed, and at the same time pupils could form small group computing networks, they might only need to go to school to arrange tutorial sessions with their appointed teacher. This sort of notion throws an enormous amount of responsibility onto young shoulders, but if tangible success (in terms of future job hunting, as well as academically) can be shown to be the end product of accepting this additional responsibility I think pupils would buckle down. In this scenario, schools start to look more like high technology factories and office developments, and less like local authority hospitals, as they do now.

A modern classroom: changes on the way?



The all-pervading atmosphere would be one of brisk efficiency, with the building reflecting this in the no-nonsense style of its construction. Made of glass and steel and similar industrial finishes, numerous floor levels would be accessed by a mixture of escalators and spiral staircases. The whole structure would ring with the clatter of pupils between classes, its internal lighting would be a diffused warm glow coming from banks of fluorescent lights, recessed flush with the metal ceiling. In the background would be a distant, persistent hum from the schools large power generator needed to keep all the schools hardware functional. This generator would be located deep in the school's basement, near to the fall-out shelter.

STREETWISE

Pupils of the high tech school would expect to gain and lose quite a few jobs in the course of their working lives. And furthermore, will be expected to chase around the country after them, in the face of fierce competition from their contemporaries. They will be skilled in programming, know possibly 2-3 computer languages and pick up a useful small-scale income writing the occasional game program of exam revision notes. To these youngsters, computers will be invisible, a familiar and natural part of their lives, not the high tech 'genie of the lamp' that we see them as. Very powerful portable micro's will be to the sixth formers of our high tech high school, what pocket calculators, portable stereos and sophisticated hi-fi's are to us...commonplace. The pupils who will leave our high tech highschool will be very independent creatures and street-wise in the best possible way. This is because the post industrial society that they will be going out into will be a demanding and competitive one. They will not wait until they actually leave school to start looking for a job, but will be commandeering the school's word processor and job database printout, in order to fire off rapid job applications, the moment they appear.

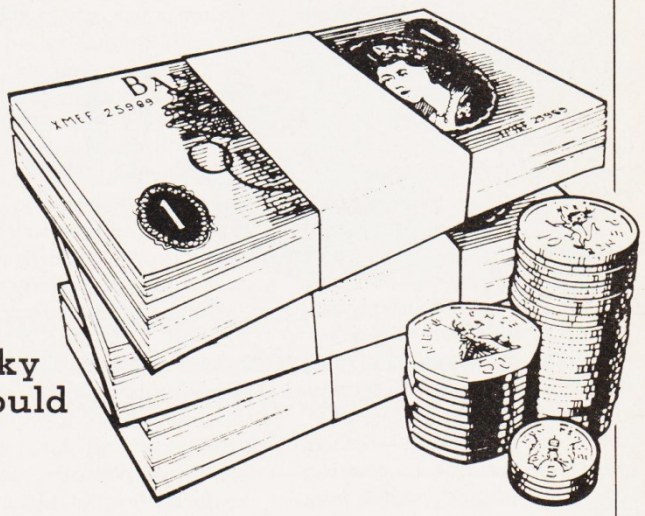
Remember, that microcomputers and their accessories will still cost money, even in the year 2000 plus. And young people will still want to set up 'state of the art' home micro systems to suit their own needs. Just as the young today inevitably personalise their cars or bikes etc. Pupils will eagerly watch the requirements of the jobs market, looking for that first job, to pay for the latest system. To help pupils out financially, the school itself might provide facilities to help cultivate money-making ideas. An area similar to a large video arcade might be given over to students as some sort of 'computing laboratory', where they can de-bug programs, write new ones or just brush up on their programming. With the school perhaps taking a small royalty on sales of software/hardware, with which it might purchase new and even more powerful equipment for students to develop further ideas. Thus the school might actually have a business side to it, in addition to an educational one! This would then free them from being too dependent upon local authority finance. Thus a school with a really bright selection of pupils could find itself a profitable business, via the sales of software... 'High Tech Highschool Ltd'! Schools such as this could then spend more on attracting even better teachers and still more computer expertise. You might then have a situation where certain schools develop a highly successful reputation for producing 'whizz kids' and the whole atmosphere of school breaks down completely. With school seen as a place to enhance your computer skills, and a place to make money. The pupils good at art could design the packaging, with the pupils good at maths taking care of the accounts etc etc. Post industrial society, if the politicians are to be believed (!) is going to bring about the 'enterprise culture' ...The era of the small business. The pupils who will graduate from the 'high tech high school' around the 2000 plus, will be able to confirm or deny this scenario in approximately 16 years.

Some of the predictions and concepts mentioned here have already come into being, but their future acceptability is as yet uncertain — Ed.

CONSIDER THE COST

Kurt Fleischmann

Purchasing a computer system can be a risky business. We consider what should and should not be known before buying.



Is cost the only consideration when buying a computer? If you only have a certain sum, be it fifty, or five hundred pounds; if you MUST have a computer, and you must have it now; and if there is only one computer on the market at YOUR price, then the answer to the above question ***may*** be Yes.

I emphasise ***may***, because even in the above circumstances it is my opinion that cost should NOT be the only consideration, if this can be at all avoided.

Why do I try so hard to discourage this excessive concern with cost? The answer could be given briefly with one well-known adage: "Cheap is dear — at least in the long run". But to be of real help to prospective purchasers of computers it is necessary to be far more specific. And so let us consider some of the reasons for the need NOT to buy on price along.

Why then is there a MUST for a computer? And why NOW? there isn't, is there? Considering that you and/or your business have been going on without the aid of a computer all this time, the need, or at least the immediate need, may suddenly seem not quite as urgent. So much so, that you may now be in the mood to consider many of the not completely-cost-related reasons for choosing and buying a computer.

WHY BUY?

Let's consider some of them. Perhaps you have a business, and this business is expanding. Much needed cash cannot be expected unless the invoices reach the customer. Will the computer of your contemplated choice satisfy this need? If it doesn't a low cost will be of little advantage.

Moreover, an unrealistically low price COULD be a positive reason NOT to entice you to purchase! Or at least not to purchase until you are satisfied that the price tag is not just a means to induce you to buy a computer range that has not yet proved itself. But how do you find out?

One way is to read carefully the terms of the guarantee. These can be so worded that they seem to promise everything, yet offer nothing.

Or ask the dealers how long they have been in business and how many of these machines they have sold. These may seem impertinent questions but it's YOUR money; and if the dealers have only started in business a short time ago and have sold relatively few of these computers, they may not have the experience to advise you; or give you the service the guarantee promises. It may be advisable to go further than this. If a claim is made that 'this machine is a good seller', ask for the names, addresses, and telephone numbers of a few 'satisfied' customers. To contact them and ask their opinion in the light of their experience may be time-consuming, but it may pay dividends, whatever it reveals.

If the machines have caused much trouble to their purchasers, you save YOUR money, by deciding NOT to buy. If on the other hand, the opposite is true, yours MAY truly be a bargain. Once more I say MAY, rather than will. For the computer may have been ideal for them, for their specific needs, but may not do for yours.

A SPECIFICATION

And so it cannot be emphasised too strongly that a thorough analysis of what you consider YOUR computer's tasks will be of paramount importance. The reason? Computers, like many other devices, and even human beings, tend to excell in particular tasks; may perform others tolerably well; but are useless when it comes to anything else.

At this point a list of uses to which computers, in many cases different computers can be put to should be of interest:

> Record-keeping, business or personal; diagnosing disease; for chess and other games; controlling production processes; aiding education and learning; planning financial or scientific activites, such as automating processes, be they in the home, factory, or laboratory.

To know what you need your computer for helps. It is not, however, enough to actually make your choice of computer, at least unaided. And so perhaps at least some knowledge of the specific qualities for special and specified tasks, is necessary. Thus, and perhaps surprisingly, computers to aid education may need very little memory.

In contrast, word processing, financial planning and record-keeping consume large amounts of memory.

To accomplish any task a computer needs programs. But again, the number and type of programs needed, varies according to the job the computer is expected to perform.

There is another consideration with regard to programs, apart from mere numbers. Certain tasks need not only larger, but more complicated programs. This affects the cost of accomplishing what has to be done.

There is another quality that varies in importance with the task a computer is expected to accomplish: reliability. Great reliability is needed for critical applications. It is thus essential that any maintenance contract that goes with the computer is water-tight, as otherwise breakdown costs could be prohibitive. Again, reliability is not the most important quality for games and learning activities.

Another important point in considering cost, is how much it would cost to start the sys-

tem. This varies, too, according to the particular job a computer is expected to tackle. With this in mind it might be possible to start on a more limited budget in the education field or laboratory automation. Not so in word processing or business data processing.

There is yet another cost-related consideration: the possibility of future expansion. A computer system may be started within the budget available. But can it, when the time comes, be expanded? If it cannot, then, truly, cheap is dear — in the long run. Once more, the anticipated use is the key to decision. Or should be. Not just price. For while classroom related tasks and control applications seldom need much expansion, business applications grow as the business gets bigger.

But if the minimum system costs more than you can afford, be not dismayed. It may be a blessing in disguise. For computers, unlike most other commodities, not only tend to go down in price, but also go up in quality.

BUYING SECONDHAND

One way of getting NOW what you MUST have, is to buy secondhand. There are, however, many pitfalls to watch out for, especially for the first time and less experienced buyer. Let's discuss some of them:

For one thing, the service, offered, if any. This is often less satisfactory if you buy from a dealer, than that which goes with new equipment. It is non-existent as a rule, if you buy privately. Then there is the matter of update. The difficulty of adding peripherals to equipment outdated when bought. Once more, the intended use is important. Computer systems destined for the classroom needing fewer peripherals, may be more usefully bought secondhand.

There are on the other hand some advantages, buying secondhand, even apart from price. One is that 'teething troubles' may have been largely overcome.

Also secondhand can be bought from 'stock' so there is, as a rule, no waiting.

There is little doubt that buying secondhand may be best for some price-conscious purchasers. In this way they may well get what they want at a price they can afford or are willing to pay. Yet the savings may often be minimal. This is the case with well-reagrded models. These often command 80% or more of the original price. On the other hand, average savings for secondhand equipment average from 30% to 50%. These must then be considered quite substantial. And more experienced buyers may save even more.

But there are risks, and the adage Buyer Beware is even more applicable in the secondhand than in the new market, for obvious reasons. Unless you are willing to take exceptional risks or are something of an expert in the field, cost should certainly not be the only consideration for choosing a secondhand computer, rather than a new one.

Micros in the shops: prices have fallen, but the decision is as hard as ever.

GRAPHICS, TURTLES, RECURSION

Garry Marshall

The Sinclair QL possesses several unusual features and at least one that is unique for computers in its price range. This article aims to show something of the capabilities of these features, as well as indicating ways in which they may be used to advantage.

Turtle graphics provide an alternative system to line graphics for the creation of pictures and drawings and are probably most familiar from Logo. The turtle can take the form of a small wheeled vehicle that can be attached to a microcomputer and controlled from the computer by giving it turtle graphics commands or it can take the form of a 'screen turtle', which is a simulation of the turtle on the screen of the computer, but which can be moved around the screen in the same way as a real turtle can be moved on the floor.

The turtle that is supported in SuperBASIC, the QL's dialect of BASIC, is a screen turtle. Much the same results can be achieved with the turtle graphics commands of SuperBASIC as with Logo.

Unlike most dialects of BASIC, SuperBASIC supports recursion. This is a form of repetition in which each repetition depends on the outcome of the previous one. In particular, it allows us to write procedures which call on themselves. If this seems to be circular, it should be added that a procedure that calls itself must also contain a stopping condition so that it can decide when it should stop doing so. When allied to turtle graphics, recursion allows interesting and unusual designs to be created. And, further, because the results of the recursion are visual, it is possible to 'see' how the recursion works.

One of the innovative features of SuperBASIC is its set of commands for creating and manipulating windows. A window is a rectangular area of the screen in which text and images can be created and displayed independently. It is unaffected by happenings elsewhere on the screen and can behave in all the ways that the usual display area can. Several windows can be present on the screen at the same time, and we shall take advantage of this to show to advantage the ouput from our recursive turtle graphics routines.

TURTLE GRAPHICS COMMANDS

When writing a turtle graphics program in SuperBASIC, the screen turtle is initially situated at the origin of the screen coordinates, facing to the right, and with its pen up as illustrated in Figure 1. In this situation established by the default SCALE command, which establishes the scale for the displayed graphics, the turtle will be at the bottom left corner of the screen. The SCALE command applies with turtle graphics just as id does with line graphics, though, and if a turtle graphics program starts with its own SCALE command, the turtle will be positioned at the origin established by this command, and not necessarily at the bottom left of the screen. the distances that the turtle will move are specified in the units of the current scale.

The command for moving the turtle is MOVE, rather than the more familiar FORWARD of Logo. It must be followed by a number, and the effect of the command is to cause the turtle to move the given number of units in the direction in which it is facing. If the number is positive and turtle moves forwards, and if it is negative the turtle moves backwards. To make the turtle move forwards a hundred units, then, the necessary command is MOVE 100, while MOVE −50 will cause it to move fifty units backwards.

There are two commands that can be used to make the turtle face in a new direction. One is TURN which, when followed by a number, causes the turtle to turn through that many degrees starting from the direction in which it is currently facing. If the number is positive it turns anticlockwise, and if the number is negative it turns clockwise. While the TURN command causes the turtle to turn relative to its current heading, the other command for making it turn, TURNTO, causes it to turn to a given heading. The numbers associated with the different directions on the screen are shown in Figure 2 with, for example, indicating the direction to the right on the screen and 90 indicating the direction up the screen. So the command TURNTO 0 causes the turtle to turn to face to the right, and TURNTO 180 causes it to face to the left.

The remaining turtle graphics commands are PENUP, which causes the turtle to leave no trail as it moves on the screen, and PENDOWN, after which the turtle leaves a trace in the current ink colour to mark its path.

By using three commands the turtle can be driven around the screen on a particular route, leaving, or not leaving, a trail behind it and in this way creating a drawing. Any turtle graphics program should begin by initialising the turtle, placing it in a known position and facing in the required direction. It is, of course, essential to remember to give the PENDOWN command if a program is to produce any visible results!

The attraction of the turtle graphics is that it provides a much more natural way of creating a drawing than does line graphics since the commands could be given to a person who, by following them, could then create a drawing in exactly the same way as a turtle. The natural correspondence between the actions of turtle graphics and human activities makes it easy for us to relate to turtle graphics, and also helps to account for the success of Logo as an educational aid for children.

The turtle graphics commands of SuperBASIC are summarised opposite. They can be mixed with line graphics commands. In particular, the commands for colour work with turtle graphics just as they do with line graphics, and the FILL command can be used to fill a shape created with turtle graphics.

RECURSIVE DESIGNS

The classic example of a recursive definition is that of the factorial function. This can be written in SuperBASIC as:

```
100 DEFINE FUNCTION factorial(n)
110 IF N=0 THEN
120 RETURN 1
130 ELSE
140 RETURN n*factorial(n-1)
150 END IF
160 END DEFINE factorial
```

Here, line 140 contains the recursive definition of factorial, defining it in terms of itself. The stopping condition is in line 110. Once it is defined in this way, the function can be called in the same way as any other function, for example by 'PRINT factorial(5)'. The function appears much as it would in a structured language such as Pascal. It is more compact than it appears on the page due to SuperBASIC's insistence on the layout that must be used when writing the IF-THEN-ELSE structure.

Recursion is an important aspect of the use of Logo and, correspondly, it is an important aspect of turtle graphics in SuperBASIC. Here, it allows the creation of recursive designs, that is, of designs which are described in terms of themselves, in a way similar to that in which factorial has just been described in terms of itslef. But recursive designs are particularly attractive because their results are visual, and, with the careful use of colour in their presentation, the recursive processes can actually be watched as they unwind themselves.

For our first example of a recursive design, we can write a program to draw a simple branching process which the turtle draws by first drawing a trunk of a certain length and then turning through 30 degrees, ready to draw the entire branching process again, but reduced in scale by a half. To prevent the program from continuing for ever, we can tell it to stop when the length of a trunk it is about to draw falls below some small value. The resulting procedure is:

```
10 DEFINE PROCEDURE branch(trunk)
10 IF trunk<2 THEN RETURN
30 MOVE trunk
40 TURN 30
50 branch trunk/2
60 END DEFINE branch
```

This procedure is called by the following program 3.

```
200 CLS
210 POINT 50, 5: TURNTO 90
220 PENDOWN
230 branch 50
```

Actually, this branching procedure just produces spirals. It can be made to give a more interesting pattern by making the turtle turn either to the right or the left at random before drawing the next branch. The amended procedure is:

COMMAND	PURPOSE OF COMMAND
MOVE m	To move the turtle by m units in the direction in which it is facing. Positive values for m cause forward movement, and negative values cause backward movement.
TURN a	To turn the turtle through a degrees from the direction in which it is facing. If a is positive it turns anticlockwise, and if a is negative it turns clockwise.
TURNTO a	To turn the turtle to face in a given direction. The direction represented by a=0 is to the right of the screen, by a=90 is up the screen and by a=180 is to the left of the screen.
PENDOWN	To cause the turtle to draw a trace as it moves.
PENUP	To prevent the turtle from leaving a trace as it moves.

```
10 DEFINE PROCEDURE branch2
   (trunk)
20 IF trunk<2 THEN RETURN
30 MOVE trunk
40 IF RND(1 TO 2)<2 THEN
50 TURN 30
60 ELSE
70 TURN -30
80 END IF
90 branch2 trunk/2
99 END DEFINE branch2
```

Different results of calling this procedure can be shown with the following program.

```
200  CLS: RANDOMISE
210 POINT 50, 5: TURNTO 90
220 PENDOWN
230 branch2 50
```

A more complicated branch procedure that gives two branches at the end of the trunk and at the end of ever branch is:

```
10 DEFINE PROCEDURE tree(trunk,
   level)
20 IF level=0 THEN RETURN
30 MOVE trunk
40 TURN 45
50 tree trunk/2, level-1
60 TURN -90
70 tree trunk/2, level-1
80 TURN 45
90 MOVE -trunk
99 END DEFINE tree
```

The following procedure uses tree to create a binary tree with a value of 5 parsed for k:

```
200 INPUT k
210 CLS
220 POINT 50, 5: TURNTO 90
230 PENDOWN
240 tree 40, k
```

The form of the tree that is created by this proecedure can be described by saying

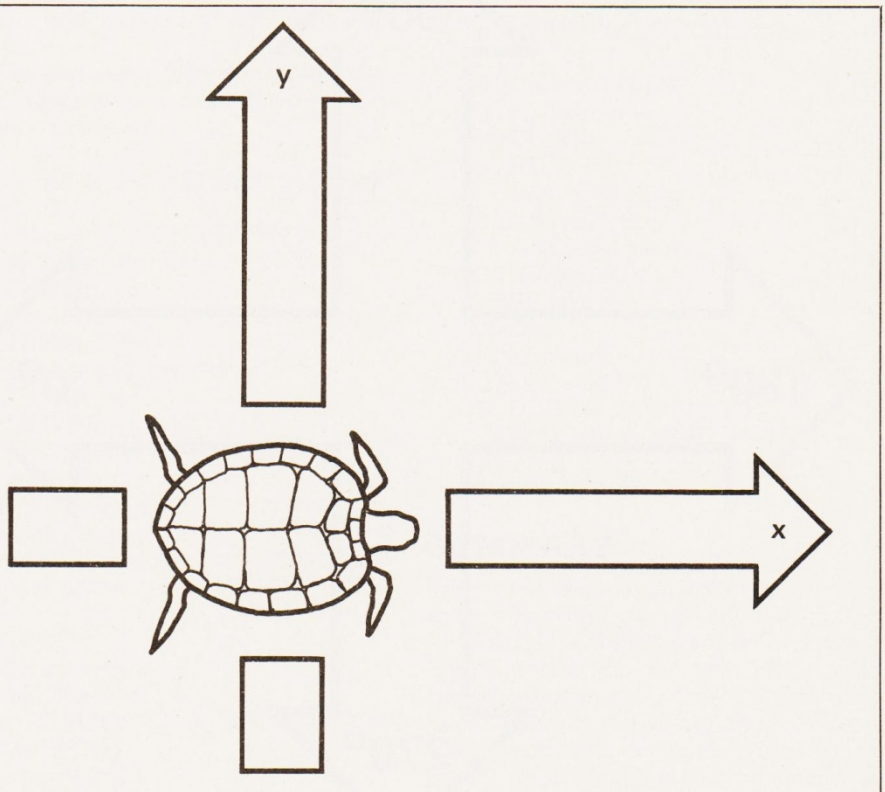


Figure 1. The turtle's initial position

that at the end of each branch two more branches start, and the length of the new branches is half that of the branch they spring from. The parameter 'level' gives the number of times the tree has grown either a trunk or branch of a new length. It can also be seen as the number of different lengths for branches in the three, or the number of branches counted in passing from the trunk to the tip of an extreme branch of the tree. In any event, a tree of this kind with level, 'level' can be seen as consisting of a trunk of length 'trunk' with a tree of level 'level'−1 growing from it at 45 degrees to the left and a second tree of level 'level'−1 growing from it at 45 degrees to the right. The lengths of the trunks of the latter two trees are half that of the original trunk. The form of the procedure 'tree' corresponds to this description.

Adding the line

```
25 INK level
```

to the procedure 'tree' (and ensuring that the procedure is never called with a value of more than seven for the parameter 'level') will cause the plotting at the different levels to be done in different colours, so allowing the progress of the recursion to be seen.

As a further example of a recursive design, consider the pattern created by starting from the large triangle, but drawing a smaller triangle at each of its corners, then doing the same with each of these triangles, and so on until the process is halted at the point when the triangles are reduced to a specified size. This gives the procedure:

```
10 DEFINE PROCEDURE triangle(side,
   min)
20 IF side<min THEN RETURN
30 triangle side/2, min
40 MOVE side
50 TURN 120
60 triangle side/2, min
70 MOVE side
80 TURN 120
90 triangle side/2, min
100 MOVE side
110 TURN 120
120 END DEFINE triangle
```

An interesting pattern results from the program:

```
200 PAPER 7: INK 0: CLS
210 PRINT 80, 20: TURNTO 0:
   PENDOWN
220 triangle 60, 3
```

A further example of a recursive design is provided by the C-curve. The level zero C-curve is just a straight line, while the level n C-curve consists of two level n−1 C-curves at right angles. This simple definition gives the intricate C-curve produced by:

```
10 DEFINE PROCEDURE c(side, level)
20 IF level=0 THEN
30 MOVE side
40 RETURN
50 END IF
60 c side, level−1
70 TURN −90
80 c side, level−1
90 TURN 90
99 END DEFINE c
200 CLS
210 PENDOWN
220 POINT 40, 25: TURNTO 0
230 c 3, 8
```

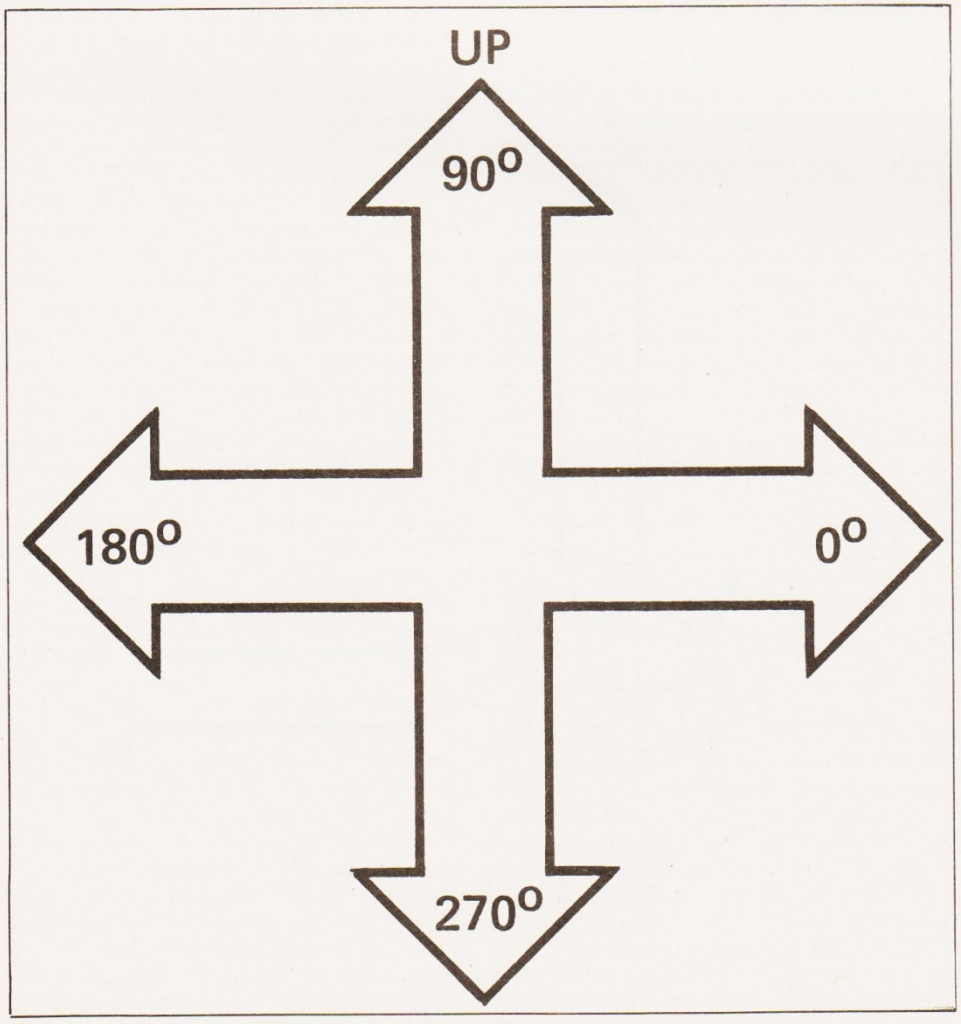


Figure 2. Absolute directions on the screen

WINDOWS

Now, the recursive designs produced by the programs of the previous section will be appreciated more fully if we can see several members of a family of curves at the same time. If we can see how one member of a family relates to another, then we can understand the family relationships better. By opening several windows on the screen, we can place a different member of a family of curves in each. Before doing this, we must generalise the recursive procedure so that they leave their output in a particular window, but this can be done quite easily.

In fact, the QL maintains windows at all times, and they are present from the time that it is powered up. Window 0 forms the bottom part of the display, and windows 1 and 2 form the upper part, either overlapping or side by side, depending on whether the monitor or television mode of display is selected. This makes the window numbers that are available to the programmer start from 3.

A command is associated with a window by placing the hash sign (#) after the key word of the command, then the number of the window and a comma, but otherwise writing the command in the usual way. As an illustration, whereas the command PRINT "Window" leaves its output on the display screen (actually, this is the default window, window number 1), the command PRINT#3,"Window" will leave its output in window number 3.

In SuperBASIC, a window is created with the OPEN command. This command first associates a number with the window so that different windows can be identified, and the results of other instructions, such as PRINT and LINE, can be directed to a particular window by incorporating its number. The remainder of the OPEN command gives the size and position of the window: it does this with reference to the pixel numbering system in which the pixel in row 0 and column 0 is at the top left corner of the screen. A window on the screen can be opened as window number 3, which is 300 pixels wide and 150 pixels high, positioned with its top left corner at the pixel 80 columns across and 40 columns down, by the command:

```
OPEN#3, SCR_300x150 a 80x40
```

Once a window has been opened, almost all the instructions that can be used for creating any display on the whole screen can be used to create a display in a window. A command is associated with a window as we have just explained. Just as the whole screen is cleared by the command CLS,

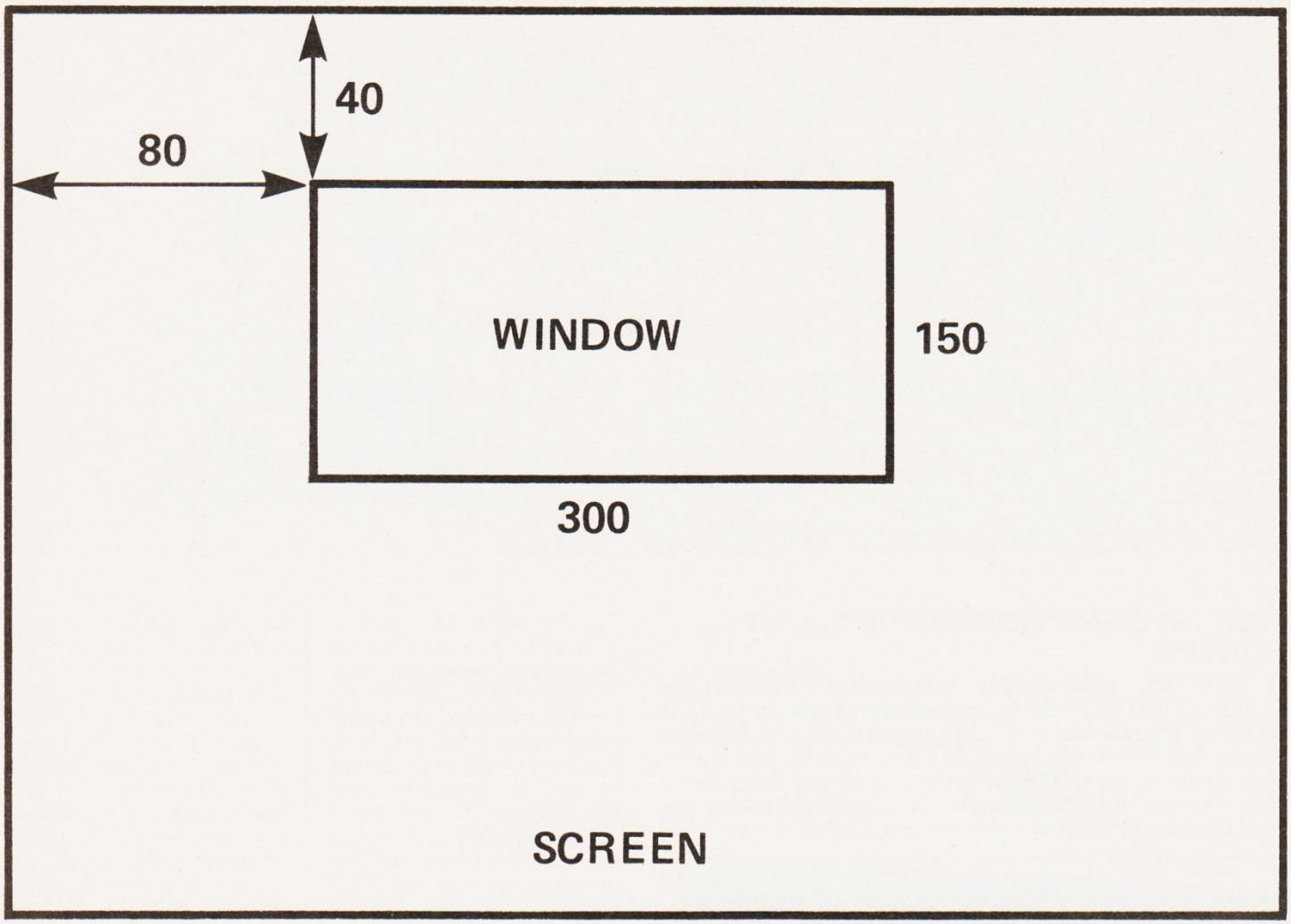


Figure 3. The location of the window

window number three is cleared by CLS#3 (no comma is needed in this case because there is no part of the command following the channel number from which the rest of the command must be separated). Each window can have its own scale, established with a command such as SCALE#3, 100, 0, 0. A line can be drawn in a window, for example, by LINE#3, 20, 20 TO 50, 50.

the next procedure shows how several windows can be supported at once by opening four of them. It is written as a procedure so that it can be incorporated in the rest of the programs of this section.

```
100 DEFINE PROCEDURE fourwindows
110 OPEN#3, scr_120x80a80x30
120 OPEN#4, scr_120x80a80x120
130 OPEN#5, scr_120x80a280x30
140 OPEN#6, scr_120x80a280x120
150 FOR n=3 TO 6
160 PAPER#n, n: INK#n, 0: CLS#n
170 NEXT n
180 END DEFINE fourwindows
```

DESIGNS IN WINDOWS

We can now show how four members of the same family of designs can each be placed in their own window so that we can see them all at once. The procedure itself must be generalised slightly to place its output in a given window, but otherwise the procedures and programs are much as before.

The procedure for drawing a binary tree can be generalised to draw it in a window number n as follows.

```
10 DEFINE PROCEDURE tree(trunk, level, n)
20 IF level=0 THEN RETURN
30 MOVE#n, trunk
40 TURN#n, 45
50 tree trunk/2, level-1, n
60 TURN#n, -90
70 tree trunk/2, level-1, n
80 TURN#n, 45
90 MOVE#n, -trunk
99 END DEFINE tree
```

When called by the following program, which also incorporates the 'fourwindows' procedure, it creates four displays in different windows.

```
200 CLS
210 fourwindows
220 FOR k=3 TO 6
230 POINT#k, 50, 5: TURNTO#k, 90: PENDOWN#k
240 tree 40, k, k
250 END FOR k
```

In a similar way, the four C-curves illustrated in Figure 9 were produced by:

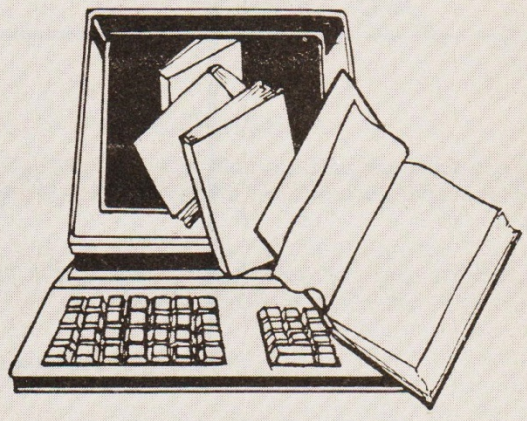
```
10 DEFINE PROCEDURE c(side, level, n)
20 IF level=0 THEN
30 MOVE#n, side
40 RETURN
50 END IF
60 c side, level-1, n
70 TURN#n, -90
80 c side, level-1, n
90 TURN#n, 90
99 END DEF c
200 CLS
210 fourwindows
220 for k=3 TO 6
230 POINT#k, 70-5*k, 25:
    TURNTO#k, 0: PENDOWN#k
240 c 9-k, k+2, k
250 END FOR k
```

SUMMARY

This article tries to demonstrate practically some of the power of the BASIC dialect of the Sinclair QL. Its ability to support recursion is demonstrated with the help of its turtle graphics commands. This not only produces attractive results, but also provides a visual aid to the understanding of recursion itself. The QL's ability to support windows is also harnessed to the presentation of the results and, in a small way, this demonstrates how windows can be used to advantage.

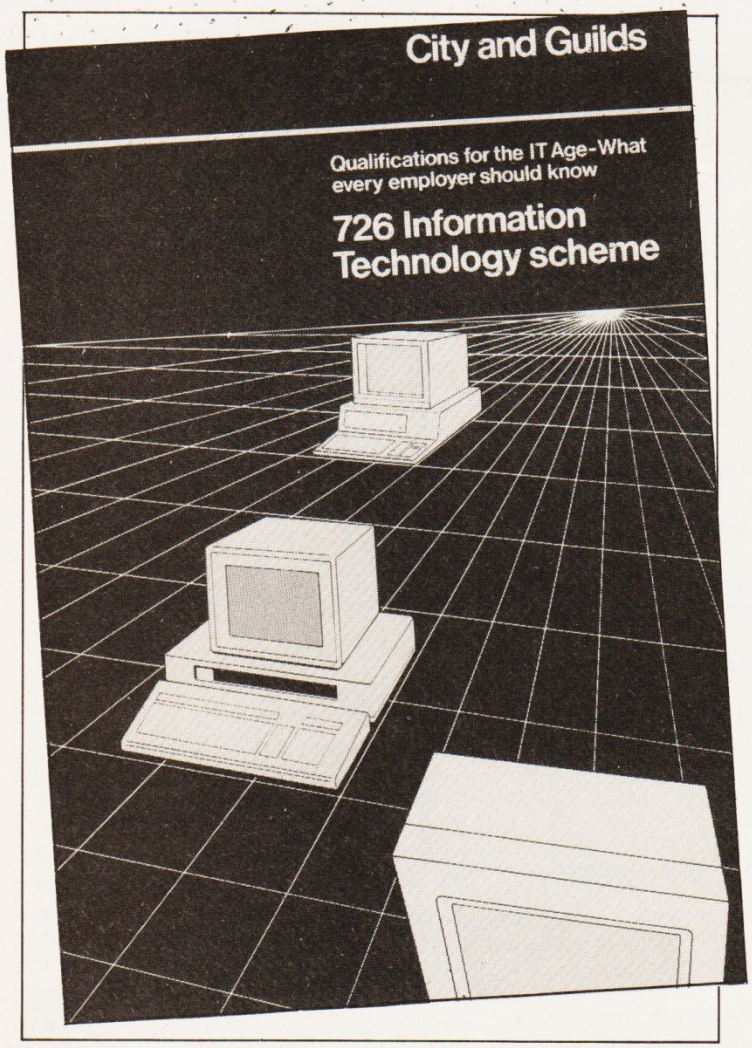
COURSES AND TRAINING

Education around the U.K.



NEW INFORMATION TECHNOLOGY SCHEME

A new City and Guilds Information Technology scheme (726) is to be launched on 18 January. This scheme represents a double breakthrough: it takes an entirely new approach to educational and training methods as well as a uniquely broad consideration of vocational training in the rapidly developing and changing field of Information Technology.



The 726 series is intended to provide for very flexible study or training methods; it is entirely pupil paced and the criterion referenced assessment ensures that successful candidates demonstrate an ability to do the job competently.

The approach is modular, but there is a limited number of modules and each is nationally devised so that employers will be able to assimilate and assess the content of each as well as deciding which combinations might best meet the needs of their workforce. Modules will relate to the three subject disciplines of:

- Programming and software
- Electronics and hardware
- Computer applications and operation.

Module levels are defined by the terms

- Introductory
- Elementary
- Intermediate
- Advanced.

These levels relate to the subject matter and are not necessarily indicative of ability. Many of the modules will be equally suitable for a very wide range of candidates; some will ideally suit YTS trainees; and others will be pitched at supervisory and management personnel.

Thus the main features of the new scheme will be:

- Criteria based assessment. Candidates prove their ability to do the job in a work related environment.
- Learning objective syllabus format. Thus the candidate knows precisely what level of performance is required for success.
- Candidate paced learning. This allows candidates to progress at their own individual pace.
- Regular updating of module contents. This is particularly important in an area where the technology is altering so rapidly.

The scheme may be offered by any centre approved by the City and Guilds. This means that centres could be set up in schools, F.E. colleges, industrial premises, ITeC's or Skills Centres, etc — anywhere where the necessary hardware, training personnel and accommodation can be provided.

The scheme is initially released at the Introductory and Elementary levels, but Intermediate and Advanced level modules will be available very soon.

Scheme notes are now available and further information can be obtained from:

Section 18
City and Guilds of London Institute
46 Britannia Street,
London WC1 9RG
01-278 2468

P & P ENTER TRAINING FIELD

Major UK distributor of IBM and Apple related products, P & P Micro Distributors, announce the introduction of their new Training Services Division. Titled simply "P & P The Trainers" the division is headed by recently appointed Peter Dalton, formerly Senior Lecturer for the Inner London Education Authority.

Dalton is well qualified in the skills of microcomputer education/training, having designed and presented microcomputer training courses in further education and industry, in particular for the S.W. Area Health Authority and for Fleet Street (Newspaper Publishers Association, News Group Newspapers and Newspaper Society).

The courses start on March 1st with an "Introduction to Multimate", and the other eight courses scheduled for March and April include "Introduction to the IBM PC", "Getting Started with Lotus 1-2-3" and "An introduction to the Apple Macintosh and its capabilities".

The courses will take place at three venues overall — two in London and one in Rossendale, Lancashire, at the company's head office. The cost of attending a course is very competitive and with others in the same or similar fields. A one-day course in word processing on the IBM PC in London will cost you £80 including all necessary course materials and a good lunch but excluding VAT. A two-day course on programming the IBM PC in BASIC will cost £190.

Discussing "The Trainers" Peter Dalton said, "In running the courses we shall be using well-qualified lecturers who also have skills in their own specialised fields — for example in communications, spread sheets, word processing, or database management systems. My own field covers programming, spreadsheet modelling (particularly using Lotus 1-2-3) and dBase II, stemming from my lecturing and consulting background."

Peter Dalton can be contacted at P & P Micro Distributors' London office at 1 Gleneagle Road, Streatham — phone 01-677 7641.

SUMMER SCHOOL 1985

In July 1984, the South Warwickshire College of Further Education ran its first Summer School. Over 80 people attended five different courses and such was the success of the occassion that a second Summer School will run from 22 July to 26 July, 1985.

There will be an increased number of courses including Computing.

In addition, there will be a range of social events and outings. The courses are designed to appeal to those who wanted an activity holiday in Stratford upon Avon, perhaps the most famous town in England.

Apart from the well-known Shakespeare connections, Stratford is ideal as a centre for exploring the beautiful and historic heart of England, including the Cotswolds, and full advantage of this environment will be taken during the five days of the School. The College does not possess its own living-in facilities, but lunch, refreshments and some social activities will be provided at the College, inclusive of the course fee. Guidance will be given on the range of accommodation to be found in the area.

Free brochures are available from Graham Winton, Summer School Director, South Warwickshire College of Further Education, The Willows North, Alcester Road, Stratford upon Avon, Warwickshire, CV37 9QR. Tel: 0789 66245.

SYSTEM DESIGN AND STRUCTURING

'System design and structuring' will be the subject of the third Jubilee Lecture, to be given by Professor Brian Randell on 13 February 1985.

Professor Randell is Professor of Computing Science at the University of Newcastle-upon-Tyne.

He says" the subject of computing system design methodology first rose to promence in the late 1960s — a time of growing concern in the profession about cost and schedule over-runs on large computing projects, and about the inadequate performance and reliability of many of the resulting systems — also a time when terms like "structured programming" and "software engineering" first started to be bandied about. Our aim, of course, then as now, was to find better methods of designing and implementing sophisticated computing systems.

"One of the major difficulties involved in designing these large and sophisticated systems is that of minimising their internal complexity.' Professor Randell will discuss the problems of choosing appropriate means of structuring systems, with particular reference to distributed computing systems and the achievement of high reliability and security of operation.

He will illustrate his talk using Unix. He explains 'all our systems have been based on UNIX — for a particular reason. This is that UNIX, though far from perfect, has, we have discovered, a rather special property: although it was designed as a multi-programming system, running on a single computer, its functionality is equally appropriate for a distributed system'.

"It should be possible to produce, very easily, an apparently ordinary UNIX system which provides the twin abilities of high reliability and high security, without modifying either type of mechanism, or having to re-validate the security mechanism" he continues.

Brian Randell worked on operating systems, the design of ultra-high-speed computers and systems design methodology with IBM, from 1964 to 1969, mainly at the IBM Research Centre in the United States. He then joined the University of Newcastle-upon-Tyne where in 1971 he initiated a programme which now encompasses several major research projects sponsored by the Science and Engineering Council and the Ministry of Defence.

The third Jubilee Lecture, sponsored by Computer Weekly, will be held at the Royal Society, 6 Charlton House Terrace, London SW1 on Wednesday 13 February 1985 at 7.00pm. A sherry reception will preced the lecture at 6.30pm.

Among the expected guests will be not only eminent computer people but senior representatives of government, education, commerce and industry, who should provide a stimulating discussion after Professor Randell's presentation.

Tickets, at £11.00 (£8.00 to BCS members) inclusive of VAT, are available from

British Infomatics Society Ltd
The British Computer Society
13 Mansfield Street
London W1 0BP
Tel: 01-637 0471

BIS APPLIED SYSTEMS COURSES

• **Structured Programming Workshop** is a five day non-residential workshop that provides practical training in the methods, techniques, documenation and team control which form the basis of structured programming.

It is aimed at experienced programmers, designers, programming managers and team leaders working with — or considering — structured techniques

The course costs £565 plus VAT.
MARCH 5-7

• **Program Design Techniques** is a three day non-residential course for programmers who wish to complete their induction to the full range of methods, techniques, documentation and team activities which are demanded within the structured programming environment.

The corse costs £400 plus VAT and will be held at the Park Court Hotel, London. Details from Cherry Bigmore on 01-261 9237.
MARCH 6-7

BACKNUMBERS

JANUARY 1984
TRS-80 programmer's aid, Apple music, Electron review, TRS-80 screen editor, calendar program.

FEBRUARY 1984
Using MX-80 graphics, Colour Genie monitor, non-random random numbers, ZX81 Forth, Program recovery on the Commodore 64.

MARCH 1984
Easycode part 1, BBC poker, Spectrum SCOPE review, Genie utilities, Spectrum Centronics interface.

APRIL 1984
MEMOTECH MTX500 review, Genie BASIC extensions, Brainstorm review, Disassembly techniques, Recursion.

MAY 1984
Debugging, Spectravideo SV318 review, Extending the Commodore 64's BASIC part 1, Z80 text compactor.

JUNE 1984
Adler Alphatronic review, Digithurst's Microsight review, Commodore search and replace, CP/M directory, Interrupts.

JULY 1984
Commodore BASIC extensions reviewed, The Art of Islam, a fast sort, Brother HR5 review, Random Thoughts, extended palette on the Dragon.

AUGUST 1984
Apricot xi review, BBC Mode 7 screen editor, Genie sprites, Microdrive-file line editor, TRS-80 screen scroller.

SEPTEMBER 1984
CUBE's Beebflex, Electron drawing utility, MTX real time clock, Commodore SX64 review, BBC disassembler, TRS-80 Fastsave.

OCTOBER 1984
AMSTRAD CPC464 review, Dragon sprites, Commodore 64 adventures, BBC Draughts, Nascom screen dump.

NOVEMBER 1984
Apple IIc review, Epson PX8 review, MTX utilities, Z80/TRS-80 memory move routine, 16-page Business supplement.

DECEMBER 1984
Acorn Bitstick package review, Art and the AMSTRAD, BBC Draw, Psion Organiser review, Koala Pad review.

JANUARY 1985
BBC Commodities, Tatung Einstein review, Fujitsu Micro 16 review, Commodore 64 prettyprint, MTX500 Life, Nascom string-save.

FEBRUARY 1985
The Intelligent Computer, Dragon interrupts, BBC Machine-code monitor, Tasword 464 review, Spectrum/BBC cassette volume meter, Sakata SCP800 printer/plotter review, Spectrum ON ERROR, TRS-80 mail list, BBC passwords; Deficiency, Abundance, Perfection.

If you've lost, lent or had stolen one of those precious back copies of Computing Today then now is your chance to fill the gap in your collection. The list of issues given here represents the few remaining copies that we have available to help complete your library of all that's good in features, programs and reviews.

If you want one of these issues, it's going to cost you £1.40 (including postage and packing) but we think that's a small price to pay for the satisfaction you'll get. Ordering could hardly be made simpler — just fill in the form, cut it out (or send a photocopy) together with your money to:

Backnumbers,
Infonet Ltd,
Times House,
179 The Marlowes,
Hemel Hempstead,
Herts HP1 1BB.

If you wait until next month to do it, the chances are that we'll have run out of the very issue you wanted!

A COMPUTING Today Reader Service

BACKNUMBERS

Please send me the following Backnumbers

ISSUE

At £1.40 each. I enclose £ .....

NAME .....

ADDRESS .....

.....

.....

..... POSTCODE .....

Signature .....

I enclose a cheque/PO for £ ..... (Payable to ASP Ltd)

I wish to pay by credit card

Access ☐ Barclaycard ☐

Buy it with Access

BARCLAYCARD VISA

Insert Card No.

If you wish to pay by Access or Barclaycard, just fill in your card number and sign the form. do not send your card.

Please allow 21 days for delivery.

Suzanne and Tim — engaged

'I thought Suzanne was too attractive to be a member of Dateline. I thought the only people who used dating agencies were the real lonely-hearts types, but not at all!'

Sue and Martin — married

Sue says: 'I met genuine men who honestly and truly wanted to meet new people as much as I did; who wanted to make friends, and have a good time — and I met Martin!'

Margaret and Louis — married

'When you meet other members of Dateline you know they have gone into it with the same attitude as yourself — they want to meet people. Some people think Dateline is cold but you are meeting men who have something in common with you, and quite frankly how else do you meet them?'

Dateline

— for friends, love or marriage

WHO JOINS DATELINE?

People just like you! Currently 2,000 people a month join Dateline, so you probably already know people who have used the Dateline service, or possibly met through Dateline.

People who join Dateline come from all over the country — from farms, villages, market towns and cities. The problems of meeting people are not confined to any particular locality. The backgrounds and occupations of Dateline members are equally varied — MP's, surveyors, solicitors, engineers, doctors, journalists, teachers, secretaries, stockbrokers, nurses, chemists, receptionists, bus drivers and even zoo keepers. Loneliness is classless. What all Dateline members have in common is an optimistic, positive attitude in tackling the problem practically. Meeting someone special is difficult, but after that the romance is easy.

WHY?

If the disco is not for you, your friends are all married, you don't meet anyone through work, you have just moved to a new area, or you are newly single again — just where do you begin? There really are very few ways you can meet the kind of people you want to meet. Dateline is simply the most honest and realistic way of being introduced to a wide choice of compatible partners. It also saves a lot of wasted time and expense!

HOW DOES DATELINE WORK?

When you join Dateline you will complete a very comprehensive questionnaire, information from which will be confidentially locked into the memory of our computer; the matching process, in which your data is compared with every Dateline member of the opposite sex, will then begin. Finally the computer will have found the most compatible matches for you, and their details will be forwarded to you. You will also be matched to other compatible people living in your area, and they will contact you. And so, with your year's membership, a whole new social life begins.

IS DATELINE SUCCESSFUL?

Yes, in many different ways! Some people simply have a wonderful time, meeting as many new people as possible, sharing new interests, going out, just enjoying themselves. Some are looking for the companionship, the love and romance of one or two relationships, and others are looking for a special love with marriage in mind.

Many thousands of couples have met and married through Dateline, as the Dateline questionnaire is designed to match couples through many facets of physical type, personality and life-style; not just matching those who want to meet people similar to themselves, but those who also want to meet someone different. With Dateline you meet the people you want to meet, who also want to meet you.

If you would like to be one of the many thousands of people nationwide who have been enjoying a new social life, and finding love and happiness through Dateline, complete the simple questionnaire below. We will send you confidentially and completely free, full details about Dateline and how it works, and details of just one of the Dateline members who are compatible with you. Send to:

Dateline Computer Dating, 23 Abingdon Rd., London W8. Tel: 01-938 1011

CONFIDENTIAL **FREE**

Computer Test to Find Your Ideal Partner.

START HERE

1 Do you consider yourself:

- ☐ Shy
- ☐ Extrovert
- ☐ Adventurous
- ☐ Family type
- ☐ Clothes-conscious
- ☐ Generous
- ☐ Outdoor type
- ☐ Creative
- ☐ Practical
- ☐ Intellectual

2 Indicate which activities and interests you enjoy by placing a '1' (one) in the appropriate box. If you dislike a particular activity, write a '0' (nought) in the box. If you have no preference, leave the column blank.

- ☐ Pop music
- ☐ Fashion
- ☐ Pubs
- ☐ Sport
- ☐ Pets
- ☐ Folk music
- ☐ Jazz
- ☐ Travelling
- ☐ Cinema
- ☐ Good food
- ☐ Politics
- ☐ Classical music
- ☐ Art/Literature
- ☐ 'Live' theatre
- ☐ Science or technology
- ☐ Creative writing/painting
- ☐ Poetry
- ☐ Philosophy/Psychology/Sociology
- ☐ History/Archaeology
- ☐ Conversation

☐ I am over seventeen and would like you to send me completely free and without obligation, a description of my ideal partner. Plus a free full colour brochure and lots more information about Dateline. I enclose two first class stamps.

3 Your sex ___ put M or F Your Height ___ ft ___ ins

Your Age ___ yrs Age you would like to meet Min ___ Max ___

Christian Name _____

Surname _____

Address _____

_____

_____

Nationality _____ Religion _____

Occupation _____

Send today to:
Dateline, Dept CTB
23/25 Abingdon Rd.
London W.8.
01-938 1011

Dateline

TEN COMMANDMENTS

K Y Wong

For those as yet unfamilier with the AMSTRAD CPC464 sound commands, we present ten rules which should be observed when attempting to get some noises from it.

Owners of the Amstrad CPC464 probably have litle doubt as to the quality of this machine. But those new to computing who have purchased an Amstrad as their first machine, may have found the description of the sound commands in the User's Manual a little, well, complicated. This author found the commands to be initially rather daunting, to the point where the specification for the GI sound chip used in the Amstrad was examined in fine detail, and the information subsequently divined used as a basis for some experimentation on the Amstrad itself.

After a time spent producing all sorts of very curious and unpredicatble noises, and having them HELD, RELEASED, and FLUSHED, and eventually deciphering various writings on hardware envelopes and absolute and relative software envelopes, the following truths, what I call my Ten Commandments of Amstrad sound, became obvious:

1. A sound note can only be generated by the SOUND command. There are 3 independent frequency generators, one for each of the channels A, B and C. A SOUND command can activate any 1, 2 or all 3 generators to produce the note.

2. The time when a SOUND command is executed to produce a sound note is governed by its associated channel RENDEZVOUS and HOLD-RELEASE (see channel status G* of the SOUND command) conditions.

3. The volume of sound is normalised into 16 levels, from 0 (no volume) to 15 (maximum volume). The final speaker volume is further governed by the volume control knob on the computer. The stereo output is arranged so that one half of the stereo outputs is equal to sound from channel A plus half of the sound from channel B, and the other from channel (C+½B).

4. The tone period range is from 0 (no tone generated) to 4095 giving a frequency range of30. 5Hz to 125KHz. Tone period = 125000/ frequency.

5. The volume value in the Volume Envelope (ENV) command dictates the actual (absolute) volume value in the SOUND command that specifies the ENV number (N* in the ENV command). Having specified an ENV, the volume K* gives only the initial volume value of the sound note upon which the ENV will alter.

6. The tone period or frequency of the Tone Envelope (ENT) command effects the absolute frequency but not the volume of the SOUND command which specifies that particular ENT.

7. The duration of a sound note is always that given in the SOUND command. Another sound note cannot begin until the current sound note's speified duration is over (even if the volume has reached zero or no sound well before the current sound command duration is finished).

8. There is only one noise generator. A SOUND command can produce a pure noise, or mix noises from 1, 2 or all 3 of the frequencies in the 3 channels A, B and C.

9. For each channel there is a queue storage capable of accommodating 5 SOUND commands specifying that channel. The SOUND command at the head of the queue will be played if its RENDEZVOUS or HOLD-FLUSH-RELEASE (or the ***lack*** of all of these) conditions are satisfied.

10. The ON-SQ(y)-GOSUB command generates an Interrupt to the program when there is free space in that queue, allowing automatic 'topping-up' of SOUND commands. It can be reset by either the SQ(x) or the SOUND command.

DEMONSTRATION

In order to demonstrate just some of the uses of the SOUND command, the following four short programs have been devised:

European Siren

```
,10 ?" THIS EUROPEAN SIREN IS PRODUCED FROM LOOP 30-60"
20 ?"PRESS ANY KEY TO SWITCH SOUND COMMAND SET"
30 SOUND 12, 284, 35, 15
40 SOUND 33, 1, 35, 0
50 SOUND 1, 670,35, 15
60 A$=INKEY$ : IF A$="" THEN 30
70 ?"THIS SIREN IS PRODUCED FROM LOOP 90-100
80 ?"PRESS ANY KEY TO SWITCH SOUND LOOP"
90 SOUND 1, 284, 35, 15
100 SOUND 1, 670, 35, 15
110 A$=INKEY$ : IF A$=""THEN 90 ELSE 10
```

GUNSHOT

```
10 ENV 1, 15, -1,4
20 SOUND 1, 0, 60, 15, 1, 0, 1
```

WHISTLING BOMB

```
10 ENV 2, 30, 2, 3
20 SOUND 7, 60, 90, 15, 0, 2
30 ENV 2, 15, -1, 13
40 SOUND 7, 0, 195, 15, 2, 0, 15
```

WOLF WHISTLE

```
10 ENT 1, 12, -3, 2
20 ENT 2, 9, -2, 4
30 ENT 3, 21, 3, 1
40 SOUND & X10001,72,24,7,0,1
50 SOUND &X1010,72,24,4,0,1,1
55 '
60 SOUND 1, 0, 25, 0 : REM PAUSE PERIOD
65 '
70 SOUND &X10001, 72, 36, 7, 0, 2
80 SOUND &X1010, 72, 36, 3, 0, 2, 1
85 '
90 SOUND &X10001, 53, 21, 7, 0, 3
100 SOUND &X1010, 53, 21, 3, 0, 3, 1
110 '
120 INPUT "PRESS ANY KEY TO REPEAT";A$:GOTO40
```

MORE EXAMPLES

The following, rather shorter, examples should give you some idea of what is possible with a bare minimun of coding. Try entering them in direct mode and see, or rather, ***Hear*** what you get:

- SOUND 1, 0
- SOUND 1, 100, 200 = SOUND 1, 100, 0 = UND 1, 100, 200, 4 =
- SOUND 1, 100, 0, 4
- SOUND 1, 200
- ENV 1, 0, 8, 19, 1, -8, 1:SOUND 1, 200, 200, 13, 1
- SOUND 1, 100, 100, 7, =ENV 1:SOUND 1, 100, 100, 15, 1
- ENT -1, 5, -2, 1 :SOUND 1, 10, 10, 8, 0, 1
- SOUND 7, 0, 100, 7

Readers who feel capable of putting names to these sounds are invited to write to us with their suggestions. We would be delighted to hear from anyone who felt that they had really put their machine to the test in any way regarding sound or music.

***See Chapter 6, pages 6, 9, and 10 of the User Manual for the notations N, J, K, etc..**

COMPUTING Today

Liquid Crystal Display panels are just one of the core features of next month's Computing Today. Peter Gee of Epson UK explains the operational principles of LCDs, and discusses the advantages and disadvantages of modern display panel technology.

PLUS

- The National Computing Centre — just what goes on behind all that concrete and glass? You can find out by reading the next edition of Computing Today.
- INSIDE Wordprocessors — how they work, and the details of what a good wordprocessor should offer.
- Training computer staff — how is it done? Why is it necessary? All is revealed in the next edition of Computing Today.

All this plus our regular features in the next edition of

Computing Today

THE EVOLUTION CONTINUES..

THE ASSOCIATION OF LONDON COMPUTER CLUBS
PRESENTS

RESERVE THE DATES NOW!

THE CULMINATING EVENT OF THE

The Personal Computer Show
for EVERYONE
Further information from
PATRICIA SPILSBURY
01 - 303 8849

COMPUTING FOR LONDON.

SHARP MICRODEALER

SHARP MZ-3541

CPU	Z80A (two), 80C49
MEMORY	128K RAM, 8K ROM
LANGUAGE	Sharp BASIC
MASS STORAGE	Twin integral 5¼" floppy disk drives, total capacity 1.28 Mb
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, interface for extra external floppy disks
DISPLAY	Monochrome monitor, colour optional
GRAPHICS	80 by 25 text, 640 by 400 high-resolution graphics
SOUND	Single channel

Notes: The Sharp MZ-3541 is aimed at the businessman. RAM is expandable to 256K, while two disk drives may be added externally to complement the integral pair. Colour is only possible with the optional graphics expansion RAM. One Z80 handles the main CPU activities while the other handles peripheral activities. The third processor handles the keyboard. The availability of CP/M means a ready supply of business software.

LONDON

SHARPSOFT LTD.
With the Sharp user in mind.
Sharp MZ 700 including cassette
£199.99
Sharpsoft Ltd, Crisallen House, 86-90 Paul Street, London EC2. Tel: 01 - 729 5588.

LANCASHIRE

STATION ELECTRICAL
Dept CT, Coastal Road, West Bank, Lancs LH26HN. Tel: 0524 824519
Large range of software, books and peripherals for SHARP MZ 700. 2nd hand computers bought, sold and part exchange. Also repairs.
SAE for lists

MICRODEALER — ORDER FORM

Please include my business details in the next available issue of Computing Today:

Business Name: .....

Address: .....

.....

Details: .....

Tel. No.: .....

Open Hrs: .....

Contact (Office Use Only): .....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

ONLY £20.00

ACT MICRODEALER

xi APRICOT

CPU	8086
MEMORY	256K RAM
LANGUAGES	Microsoft BASIC, Personal BASIC
MASS STORAGE	No cassette drive Integral Sony 3½" 315K microfloppy disk drive Integral 5 or 10 Mb hard disk
OS	MS-DOS 2.11 with GSX bundled CP/M-86 (not yet available) Concurrent CP/M-86 (not yet available)
KEYBOARD	QWERTY, cursor, numeric pad, function keys
INTERFACES	RS-232C, Centronics, Microsoft mouse
DISPLAY	Monitor (supplied)
GRAPHICS	80 by 24 text with block graphics 800 by 400 high-res graphics under GSX
SOUND	No

Notes. The Apricot xi is a development of the award-winning Apricot, and replaces one of the latter's disk drives with an integral hard disk, providing vastly increased storage with faster access. Memory may be expanded in 128K increments to a maximum of 768K. The languages and operating systems mentioned above come bundled (except for Concurrent CP/M) and four software tools are also bundled, including an asynchronous package for use with the optional modem card.

HAMPSHIRE

TIMATIC SYSTEMS LTD
The Market, Fareham.
Tel: (0329) 239953

For the complete range of Apricot hardware and software. Also dealers for Zenith, Memotech. For future information call or ring anytime.

LONDON

PROGRAMS UNLIMITED COMPUTER CENTERS

35 Baker Street, London W.1.

MIDDLESEX

SIRIUS AND APRICOT
IN STAINES
Micronomy Ltd., Unit 18,
Central Trading Estate, Staines,
Middlesex TW18 4XE.
TEL: STAINES 63651

SCOTLAND

SIRIUS
is alive and well and supported at
ROBOX
(Office Equipment) Ltd,
The Scottish Computer Centre
Anderson Centre, Glasgow
041-221 8413/4
34 Queen Street, Edinburgh
031-225 3871

PROGRAMS UNLIMITED COMPUTER CENTERS

17 Queen Street, Edinburgh EH2 1JX.

SURREY

PROGRAMS UNLIMITED COMPUTER CENTERS

60 George Street, Croydon, Surrey CR10 1PD.

WALES

SIGMA SYSTEMS LTD
266 North Road, Cardiff
Tel: 0222 621414
Main dealer and Service for ACT, SIRIUS, APRICOT, IBM, COMMODORE & DIGITAL

WEST MIDLANDS

Q data limited
The Black Country's specialist in micro-computing. Full range of ACT Apricots and IBM personal computers.
The Limes, High Holborn, Sedgley, West Midlands.
Tel: Sedgley (09073) 62331

CBM MICRODEALER

Notes: The Commodore 64 is a popular micro with a great deal of games software available. There is also some business software available.

The Commodore 715B is the top model in the 700 range of business machines.

YORKSHIRE

YORKSHIRE ELECTRONICS
Commodore Appointed
Commerical Systems Dealer
Full range of peripherals and Software available.
Caxton House, 17 Fountain St., Morley, West Yorkshire.
Tel: 0532 522181

NASCOM MICRODEALER

NASCOM 3

CPU	2 MHZ Z80
MEMORY	8K or 32K inbuilt RAM (expandable to 60K)
LANGUAGE	Full Microsoft BASIC
MASS STORAGE	Single or twin 5.25" disc drives 350K capacity per drive
OS	NAS-DOS or CP/M 2.2
KEYBOARD	Full size QWERTY
INTERFACES	RS232 and 16-bit parallel
DISPLAY	40 or 80 column 25-line display
GRAPHICS	High resolution graphics with 8 foreground and 8 background colours (400 x 256 pixels) Double density graphics with 2 colours (800 x 256 pixels)
SOUND	No

DEVON

PLYMOUTH'S DEALER nascom
S & R BREWSTER LIMITED
86-88 Union Street, Plymouth PL1 3HG
Tel: 0752 665011 Open: 6 days

GLOUCESTERSHIRE

ZETA COMPUTERS
For Kaypro, Lucas, Nascom, Assorted Business Software.
66 HIGH ST., STONEHOUSE, GLOUCESTERSHIRE GL10 2NA
TEL: 045 - 382 2444

LANCASHIRE

EV COMPUTING
700 Burnage Lane,
Manchester M19.
Tel: 061-431 4866
80-BUS SOLUTIONS

COMPUTING TODAY

Lineage: 40p per word.

Semi display: £9.00 per single column centimetre
Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

01-437 0699

Send your requirements to:
CAROLINE
ASP LTD, 1 GOLDEN SQUARE,
LONDON W1

HARDWARE

NEWBRAIN & SANYO

Professional Micro Computers for the price of hobby machines.

NEWBRAIN ON SPECIAL OFFER

With over £200.00 free software. (Accounts, databases, etc. etc)
Limited Offer-ring now!

SANYO 550/555 COMPUTERS

Micropro Wordstar, Calcstar etc at no extra cost! **Printers:** Epson, KDC, Juki, Shinwa, Daisystep 2000 etc. **Sanyo Monitors & Recorders.**

Call STEVENAGE (0438) 812439 anytime for hardware/software lists

Mail Order and **Access** facilities.

ANGELA ENTERPRISES
4 Ninnings Lane, Rabley Heath, Welwyn, Herts AL6 9TD.

ALARMS

BURGLAR ALARM Equipment. Please visit our 2,000 sq. ft. showrooms or write or phone for your free catalogue. CWAS Ltd., 100 Rooley Avenue, Bradford BD6 1DB. Telephone: (0274) 731532.

FLOPPY DISKS

AAAA authorised **APRICOT** and **SANYO** dealers

FLOPPY DISKS

	DYSAN*	3M	NASHUA	SONY
SS40 :	17.00	14.50	13.15	3.5"
DS40 :	24.50	18.00	16.50	↓
SS80 :	24.50	21.00	17.45	34.50
DS80 :	29.50	23.50	18.95	45.50

*Special offer **DYSAN** branded disks deduct £1.50. Prices per box of 10 excluding VAT. **POST FREE** discounts for **any** larger quantity. £2.00 extra in "SEE 10" Library case. Send C.W.O. or write for full list of supplies to:

A. M. A. COMPUTER SYSTEMS AND SUPPLIES
Dept F. 8 Glebe Street, Beeston, Nottingham NG9 1BZ.
Tel: (0602) 255415.

DON'T LEAVE IT TO CHANCE!

GIVE YOUR BUSINESS A BOOST BY ADVERTISING IN COMPUTING TODAY

PHONE 01 - 437 0699 FOR DETAILS

COMPONENTS

68000 SYSTEM KITS

A range of high quality, double sided, 8 by 8 PCB's which form the core of a 68000 system.

CPU contains 68000 L8 micro with a mHz crystal clock, dynamic refresh sequencer, bus buffering etc. PCB £80. Component set £110.

B10 provides 8/16 K Eprom, Ram, 20 I/O lines, RS232 interface, counter-timer, interrupt handling etc. PCB £80. Component set £98.

RAM 256K dynamic. Page/address selectable. PCB £85. Component set £245.

A boot loader/monitor is available in EPROMS at £30.

Add VAT to above prices. Delivery 14 to 28 days.

P. L. Watson, 101 Village Road, Bromham, Bedford MK43 8HU.
Tel: 02302 2867

COMPUTING TODAY
CLASSIFIED ADVERTISEMENT — ORDER FORM

If you have something to sell now's your chance! Don't turn the page — turn to us!
Rates of charge: 40p per word per issue (minimum of 15 words). Please state classification and post to **COMPUTING TODAY, CLASSIFIED DEPT., 1 GOLDEN SQUARE, LONDON W.1.**

Please use BLOCK CAPITALS and include post codes.

Name (Mr/Mrs/Miss/Ms) .....
(delete accordingly)
Address .....
.....
.....
Signature ..... **Date** .....
Daytime Tel. No. .....

Please place my advert in **COMPUTING TODAY** for . . . . . issues commencing as soon as possible.

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

CHESHIRE

Computer Junk Shop

We Buy, Sell, Break Computers & Peripherals.
10 Waterloo Rd, Widnes, Halton. Tel: 051 420 4590

HERTFORDSHIRE

NEWBRAIN & SANYO
HARDWARE & SOFTWARE
Printers, Epson, KDC, Juki etc. Monitors, Tape Recorders, Books, Expansions, CP/M. **Sanyo 550/555 Computers.** Access/Mail Order. Ask for details.
ANGELA ENTERPRISES
Tel: Stevenage (0438) 812439 anytime

LONDON

A. J. Duchesne
(Computer Consultants) Limited
Specialists in Small Business Computing
10-12 Creechurch Lane, London EC3A 5AY
Telephone: 01-621 0433

SOUTH LONDON

CROYDON COMPUTER CENTRE
Authorised Acorn Service Centre
29a Brigstock Rd., Thornton Heath, Surrey. Tel: 01 - 689 1280
BBC, Acorn, Electron, Genie, Oric, Kaga Microvitek Zenith Monitors. OKI 80, 82A + 84 Printers. Paper, Ribbons, Software etc. BUY-HIRE.

MIDDLESEX

SCREENS MICROCOMPUTERS
6 Main Ave., Moor Park, Northwood, Middx.
Tel: Northwood (09274) 20664
Telex: 923574 ALACOL G.
Official Dealers for: Acorn, Atari, Amstrad, Apricot, Commodore, Dragon, Einstein, Memotech, Oric, Psion, Sirius, Sanyo & Sinclair.
Open 6 days per week

NORFOLK

ANGLIA COMPUTER CENTRE
88 St Benedicts Street, Norwich.
Tel: (0603) 29652/26002.
Open: 6 days 9am-5.30pm.

SCOTLAND

VICTOR MORRIS GLASGOW
TANDY TRS 80, VIC 20, VIDEO GENIE, APPLE PANASONIC, CUMANA, EPSOM ETC.
340 Argyle Street, Glasgow G2: 041 221 8958

SUSSEX

GAMER
24 Gloucester Road, Brighton.
Tel: 0273-698424.
Open: Mon-Fri 10am-5.30pm, Sat 9am-5.30pm.

TYNE AND WEAR

HCCS ASSOCIATES
533 Durham Rd., Low Fell, Gateshead. Tel. Newcastle 821924.
Open: 6 days 9am-5.30pm (Sat 10am-5.30pm). Specialists in: Acorn, BBC, Video Genie, VIC 20.

CLASSIFIED ORDER FORM
COMPUTAMART

Please include my business details in the next available issue of Computing Today:

Business Name: .....

Address: .....

.....

.....

Details: .....

.....

.....

.....

Tel. No.: .....

Open Hrs: .....

Contact (Office Use Only): .....

Post To: Computamart, Computing Today, 1 Golden Square, London W1.

ONLY £17.50!

Software News

INNOVATIVE SOFTWARE

from the professionals

SANYO 550/5 SOFTWARE

Cashman — Very colourful arcade game for Sanyo.	£ 26.00
DC-10 — Flight Simulator. Over 30 radio aids.	£ 26.00
Demon Seed — Another excellent arcade game.	£ 26.00
DS-DOS — Double sided DOS (1.25)	£ 37.00
DS-DOS80 — Quad DOS 80 track 9 sector (1.25).	£ 39.00
DS-DOS PLUS — Quad DOS 80 track 10 sector (1.25 or 2.11 state)	£ 39.00
Emperor — A full size wargame from Roman times.	£ 26.00
Freeze Frame — Text & graphics screen dump, colour too.	£ 32.00
French Tutor — Just that.	£ 26.00
Grafiti — Computer assisted graphics. 256K.	£ 35.00
Joystick — de Luxe quality — 2 button	£ 37.46
— 3 button	£ 46.18
Joystick Extender Cable	£ 14.00
King Arthur — 20 minute wargame.	£ 18.00
Major Motion — A fast road battle arcade game. 192K.	£ 26.00
Master Graph — Even plots 3-D curves.	£ 62.00
Maz — A fantastic four part maze game.	£ 20.50
M-DISK — Extra disk in RAM. 256K.	£ 26.00
MI-KEY — Re-define the whole keyboard or part of it.	£ 27.00
MI-TERM — Communications terminal program.	£ 44.00
Mighty Mail — The best mailing list handler available.	£ 82.50
Mysterious Adventures — A series of 10 classic Adventures. From damsels and dragons to space travel!	
Any 1	£ 18.00
Any 3	£ 45.00
All 10	£120.00
Personal Money Manager — Personal finances handler.	£ 40.00
Picasso — The ultimate Paint program.	£ 77.00
Quick & Simple — Very simple database.	£ 39.00
Soft Spool — A software print spooler. 192K.	£ 26.00
Solitaire — Patience, Poker Squares, Blackjack, etc. Terrific graphics. 192K.	£ 29.00
Super Calc III Zaps — Convert the spreadsheet to the Sanyo.	£ 24.50
Super DS-DOS — DS-DOS plus Utility Disk No. 1.	£ 62.00
Super Zap — Full disk access and edit.	£ 37.00
Thunder Chief — A shooting, bombing arcade game. 192K.	£ 26.00
Utility Disk No. 1 — Collection of utilities.	£ 37.00

All programs supplied on a single sided formatted disk for 128K, unless mentioned, and compatible with colour monitor. DOS's require ownership of either MS-DOS 1.25 or 2.11.

All prices are VAT exclusive. P&P 75p per program.

TEL: [0424] 220391 / 223636

MOLIMERX™ LTD

A J HARDING (MOLIMERX)

TELEX 86736 SOTEX G

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

Send 24p A4 sized SAE for full list.

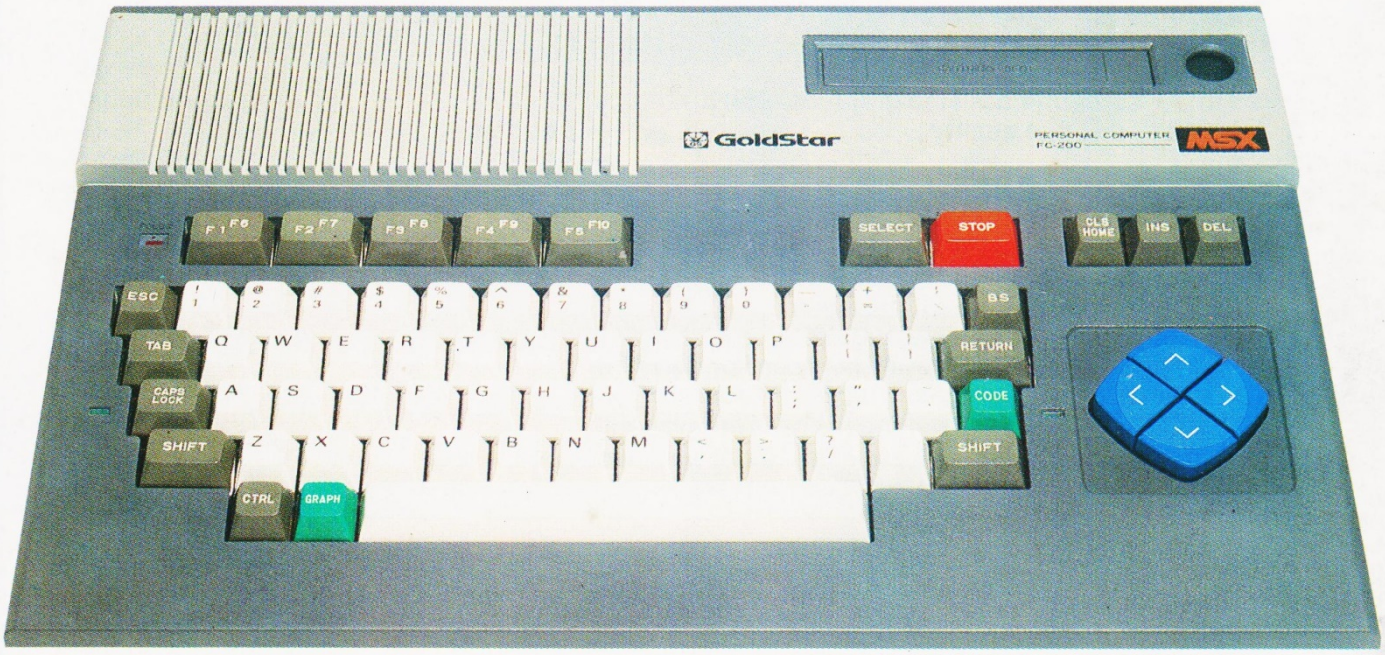
The price tag. It's something rather special.

When you take a look at the new Goldstar MSX you'll find an asking price of around £199

Quite a bit less than most of the others. And since you're choosing a micro that's designed and built to one standard, that'll leave you a whole lot more for the super new MSX games or business software.

What is standard, of course, is the superb MSX specification.

There's one thing about this MSX that isn't quite standard.



State-of-the-art feature like 64K of 'user' RAM and 32K of ROM and 16K Video RAM place the Goldstar right at the top of the MSX league.

Sixteen vivid colours and eight octaves of sound make it a great games player's micro.

And there's a powerful Z-80A processor to take on a world of home office tasks.

Goldstar MSX. The brightest new star among micros that's unbeatable value for money.

You'll find it at above standard computer dealers now.

GoldStar

Distributed by:-

MICRO DEALER UK Ltd

29 Burrowfield, Welwyn Garden City Herts AL7 4SS

Tel: (07073) 28181 Telex: 23674 MDUKG