50p U.S.$2.00

PERSONAL COMPUTER WORLD

APRIL 1979

ARE SMALL BUSINESSES READY TO BUY?

PLUS

CHECKMATE
~MORTGAGES

Europe's first magazine for personal computers for home and business use

Debate of Dinosaurs
Evolutionary Programming
The Learn Machine

Small Systems are Big Business

PERSONAL COMPUTER WORLD

Europe's first magazine for personal computers for home and business use

UK 50p
US $2.00

Vol. 1, No. 12
April 1979

ISSN: 0142 0232

CONTENTS

Editorial and Advertising Office:
62A Westbourne Grove, London W2
Phone: 01-229 5599 (Publisher)
01-727 8758 (Editorial)
01-229 5589 (Advertising & Subscriptions)

Publisher:
A Zgorelec
Editor:
Meyer N. Solomon
Editorial Assistant:
David Bannister
Policy Adviser:
Peter Crofton-Sleigh, FRAS

Due to an industrial dispute at our printers, this issue has been delayed. We shall be on schedule by the June issue. NOTE: Special offer for Seminar Tickets will be extended to April 30th.

Consultants
John Coll, Mike Dennis, Neil Harrison, Charles Sweeten, Patrick Sutton, Michael James, R. W. Davy, David Hebditch, Sheridan Williams, Stephen Castell, Dr. D.J. Hand.

Art: **Sauveur Laurent Sant, Kathryn Hamme**
Secretarial:
Vanessa Blackburn Kiddle
Layout Consultant:
David G. McFarlane
PCW Photography: **Yoshi Imamura, Peter McGee**
Typesetting & Artwork: **Gilfillan**

CONTRIBUTORS:

We welcome interesting articles written simply and clearly. You need not be a specialist to write for us. MS should not be more than 3000 words long, lines double spaced, with wide margins. Line drawings and photographs wherever possible. Enclose a stamped self-addressed envelope if you would like your article returned.

Manufacturers, suppliers and dealers are welcome to contribute technical articles, and send product information, but we are pledged to an independent viewpoint and will publish evaluations and reasoned criticism or praise, space permitting. Naturally there will be right of reply. Views expressed in articles are not necessarily those of Personal Computer World.

We may make arrangements to offer our readers products at special prices, for a limited period, in line withthe policy outlined above.

Published monthly by Intra Press, 62A Westbourne Grove, London W2. Phone 01-229 5599. Contents fully protected by copyright. All rights reserved. Subscription rates: Britain £8 for 12 issues. Prices include postage. USA - $10 for six issues. $20 for 12 issues. Continent and elsewhere £9.80 for twelve issues. Prices include postage.
Sole UK Distributors Seymour Press Ltd., 334 Brixton Road, London S.W.9., England.
Distribution to specialist shops by Intra Press.

Try a comparison test.

Look at the features of Apple II. It still has 8 I/O ports, high and low resolution graphics, a mini assembler, A/D paddles, fantastic Basic, Disks and the best documentation in the business.

Intelligent interface cards.

They're neat. The high speed serial card, printer card and communication card all have elegant documentation. There is also easy-to-use software that mean taking B.C.D. from instrumentation, connecting your favourite printer or talking to a time-sharing computer, is childs play. Apple makes things easy.

Plus:

* Say hello to Apple and let Apple talk back. Explore voice recognition, computer voice recognition and computer voice production. You can now have large vocabularies.
* Perhaps you'd like to be reminded that it is a special day or would like to chase overdue accounts. Apple clock card is simply the best. It is the most advanced on any personal computer.
* Draw graphics with the light pen.
* Make chips for your own software with the Eprom burner. Apple naturally has spare sockets for this.
* Make interfaces with the prototyping boards.
* Use Apple's incredible 'programmers aid' hardware option. More thought and care from Apple for you to use.

And there's more yet more to come from our friends at Apple. Phone Personal Computers' distributors and get to know them. They're busy, successful, highly qualified professionals and the best in the business.

Personal Computers Limited

Personal Computers Limited 194-200 Bishopsgate, London EC2M 4NR. Tel. 01-283 3391

Tim Keen, Keen Computers Ltd., Nottingham. Tel. 0602 583254
Stanton Smith, Microsolve, Edgware. Tel. 01-951 0218
Trevor Brownen, Crystal Electronics, Torquay. Tel. 0803 22699
Trevor Williams, Padmede Ltd, Odiham, Nr. Basingstoke. Tel. 02-5671 2434
Bruce Everiss, Micro Digital Ltd., Liverpool. Tel. 051-236 0707
Claude Cowan, Cambridge Computer Store. Tel. 0223 68155
George Sinclair, G.A.T.E. Microsystems, Dundee. Tel. 0382 74390
Peter McNaughton, Perthshire. Tel. 07388 8267
Robin Woods, Isher-Woods, Luton. Tel. 0582 424851
Tom Piercy, Topmark Computers, Huntingdon. Tel. 0480 212563
Steve Cooper, Creative Vision, Leeds. Tel. 0532 458444
John Page, P.I.P.S. Computer Services, Newcastle. Tel. 0632-482 359
Alan Wood, Apple Corner, London. Tel. 01-580 5304

Editorial

It can be taken as read that **PCW** strongly supports and encourages the formation of clubs and users' groups, but the time has now come for a word of caution. If you start something, you are taking on a responsibility to see it through. A desultory approach brings derisory results. Keep to the schedule of meetings. If you've just received publicity in **PCW** and get enquiries as a result, deal with them. We published James Cunningham's article, **Setting Up a Local Group** (Feb. 79) as a guide. It gives sound advice and anyone intending to organise computing activities in his or her area ought to read it.

Publisher's Letter

Dear Reader,

A comparison between our first issue and recent ones shows that **PCW** has come a long way. I believe that this is because we were determined from the beginning that the magazine would not consist of "in-house" writing – which usually means a gloomy rehash of publicity material and technical specifications. **PCW** created awareness of the market for personal computers for the home and small businesses and, issue after issue, presents to the world the talents of writers who have never before been published. We are proud of that.

Computer Trader

Subscriptions

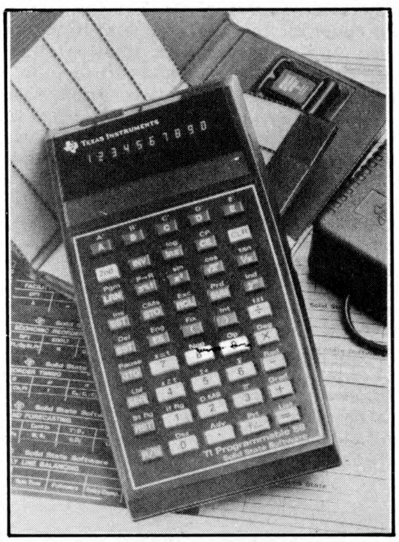
When **PCW** started publication, we had a special six-issue offer. When these subscriptions expired, we sent out reminders.
The renewal rate was 70%!
PCW reader loyalty is becoming a byword in publishing. If you're having difficulty in obtaining **PCW** at your newsagent, take our subscription. You can find the details at the foot of P.3.

Tid Bits PRODUCTS . . . . COMPANY NEWS . . .

PET Bi-directional Interface
This interface package is a combination of hardware and software which provides two-way communication between a Commodore Pet and a Model 33 Teletypewriter. This not only provides the Pet with a printer, but, where the Teletypewriter is suitably equipped, with paper tape input and output and a full-size keyboard. Comprehensive documentation is included.
Details from: Allen Computers, 16 Hainton Avenue, Grimsby, South Humberside, DN32 9AS. Tel: 0472 40568

New NCC Seminar Programme opens on Privacy and the Electronic Office
Implementing the Privacy Legislation, on 11 April, and *What Next in the Office?,* on 24 April, will commence a series of one-day seminars on a wide range of computing and related topics which The National Computing Centre will be running during 1979/80. Held at about fortnightly intervals, the seminars will deal with many of the situations, opportunities and urgent problems to be found in computing today. The seminars will be held mainly in London, but some will be held in Manchester and Birmingham.

Full documentation will be provided for each seminar.
For further information, including resumes of the seminars, contact: Mrs. Val Cording, The National Computing Centre, Oxford Road, Manchester M1 7ED. Tel: 061 - 228 6333.



A Texas Instruments TI58 programmable calculator with the business decisions package housed in a Solid State Software plug-in module. Modules comprise a software business decisions library, and include programs such as debt financing, project planning and budgeting, and demand forecasting. Cost of module is £32.
Full details: Texas Instruments Ltd., European Consumer Division, Manton Lane, Bedford, MK42 7PA.

New Consultants
Authorised MAPCON Consultants and Pet dealers B & B Consultants are opening new showroom premises at 124 Newport Street, Bolton, Lancs. Principal is J. Blackburn, who is interested in handling other products. B & B will also handle the TECS system, which is designed to work with a domestic TV as a Teletext, Computer and Viewdata system.
Preliminary enquiries: 2 Withins Grove, Breightmet, Bolton, Lancs. Tel: 0204 386485.

Micro Publication
Micro-Psych is a bi-monthly newsletter for mental health professionals interested in using small computers. Subscribers are primarily M.D's and Ph.D's and also include (in the USA) numerous libraries and institutions. If you are interested, write to Marc D. Schwartz, M.D., Community Health Care Center Plan Inc., 150 Sargent Drive, New Haven, Connecticut D6511. Subscription is $25. The newsletter seems most useful as a medium for book reviews and contact between professionals.

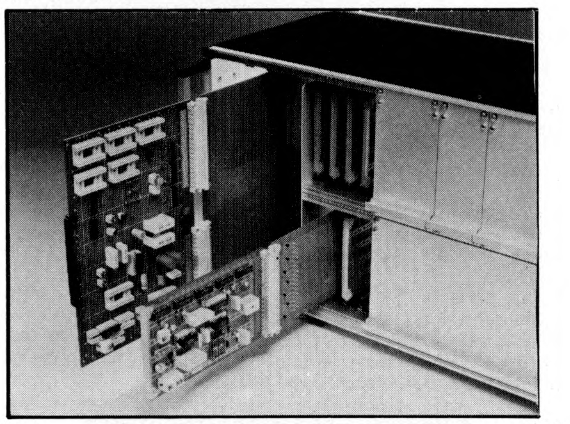
Etcetera includes a lot
T & V Johnson (Microcomputers Etc.) Ltd., are now pursuing a vigorous course in the small computer and add-on market. It stocks and sells the Pet, the TRS80, the Apple II and the Sorcerer, mini floppy disk drives, 16K RAM upgrade for the TRS80, and RS232C/S100 interfaces.
Full details from the company at 78 Park Street, Camberley, Surrey, GU15 3PF.

Contact Datac
A new software interface between the Pet 2001 and the Centronics Standard Parallel Interface is offered by Datac Limited. Using the parallel user port means that the IEEE port is left free for use with other devices. A "package deal" comprises the Centronics P1 Microprinter, Pet Software Interface and the necessary connecting cable/connector assembly for the one-off price of £429.
Contact: Datac at Tudor Road, Broadheath, Altrincham, WA14 5TN. Tel: 061 - 951 2361/2.

Ohio is very Scientific
The Ohio Scientific Instruments C2—4P Minifloppy System (to be evaluated in a forthcoming issue of *Personal Computer World)* has 20K RAM, Basic, Assembler, Software packages — all available now from:
Abacus Computers Ltd., 62 New Cavendish Street, London W1 (Tel: 01 - 580 8841); Mutek, Quarry Hill, Box, Corsham, Wiltshire SN14 9HT (Tel: 0225 743289); Thames Personal Computers, 13 Wilmot Way, Camberley, Surrey (Tel: 0276 27860); Linn Products, 235 Drakemire Drive, Castlemilk, Glasgow G45 9SZ (Tel: 041 - 634 3860); U Microcomputers, PO Box 24, Northwich, Cheshire CW8 1RS (Tel: 0606 75627).

Venture Capital - Risk Finance
PCW Consultant Dr. Stephen Castell has formed a new company — Computer and Systems Telecommunications Ltd., — an independent management and financial consultancy in information technology. Part of its activities is acting on behalf of clients who are prepared to carry out "flotations" of successful private companies active in growth sectors.
Full details can be had through Dr. Castell, at Computer and Systems Telecommunications Ltd., 20 Grange Road, Wickham Bishops, Witham, Essex CM8 3LT. (Tel: 0621 - 891 776).

Extender Eurocards in Two Heights
Extender Eurocard (Ref AB072) boards from Vero Electronics Limited, are now available in 3U and 6U heights and may be used replacing a standard 100mm or 233,40mm Eurocard circuit board for testing the back wiring of a frame, working from the front of the unit.



With the extender board replacing the circuit board, the circuit board itself can be plugged into the extender board for testing under working conditions.

By soldering terminals (20 are supplied with the single height and 40 with the double) into holes provided, scope probes can be attached when required.

Eurocard Extender Boards are available in two versions, either fitted with 64/64-way or 64/96-way indirect connectors to DIN 41612.

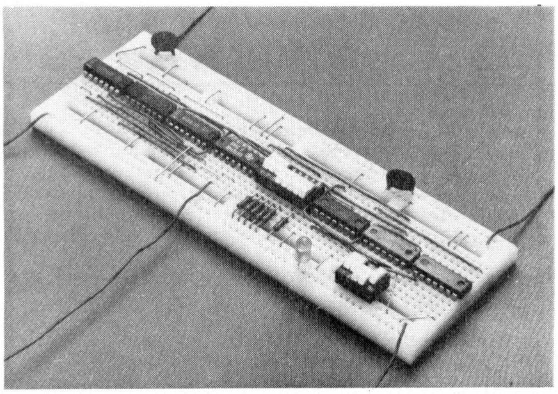
Stop Press: Spring Catalogue (1979) just out.

Contact: Alan Young, Vero Electronics Limited, Industrial Estate, Chandler's Ford, Hampshire SO5 3ZR.

Package Deal
Micro Software Systems can offer the following packages: wages, accounts, stock control, invoicing, word processing, estate agent property handling. The packages are modular and designed to run on systems such as the PDP 11/03, North Star Horizon, Equinox 300, Digital Microsystems DSC.2.
Full details from (new address) 242 Heath Road, Grays, Essex. Tel: (0375G) 41991/2.

BB stands for Breadboard
From Lektrokit Ltd., comes the Superstrip SS-2 solderless breadboard with 840 plug-in tie points. Accepts discrete components

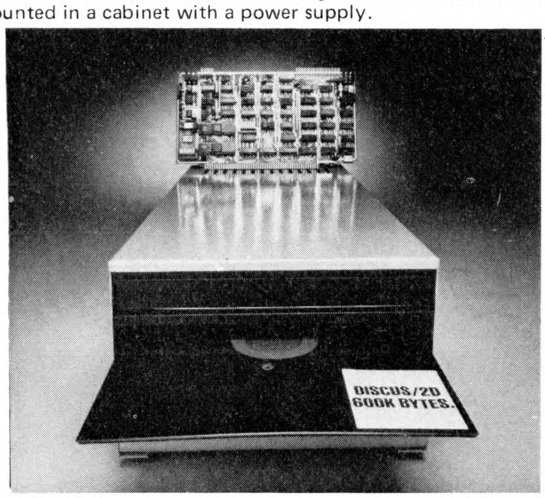
and ICs directly. Up to nine 14-pin dual in-line ICs can be plugged into a single SS-2.



More details from Lektrokit Ltd., Sutton Industrial Park, London Road, Earley, Reading, Berks. RG6 1AZ. Tel: (0734) 669116/7.

Discus 2D – S-100 Single/Double Density Disk System

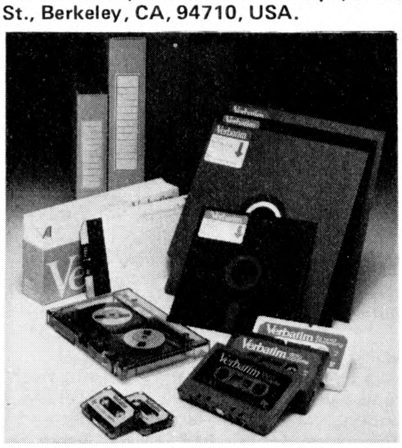
Thinker Toys announce the introduction of DISCUS 2D, a full-size, single/double density disk system capable of storing up to 600K bytes of data on each side of a diskette which is formatted to be compatible with the IBM System 34. Like the original single density DISCUS I, DISCUS 2D comes fully assembled with a controller board and a Shugart SA800R full-size drive mounted in a cabinet with a power supply.



The S-100 controller board is capable of handling up to four drives.

Software includes BASIC–V™ virtual disk BASIC, DOS and Disk/ATE assembler and editor. Extra cost optional software including CP/M™, Microsoft Extended Disk BASIC and Fortran is available.

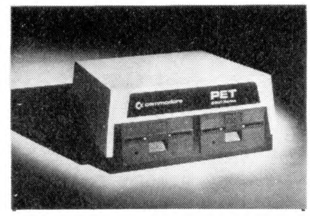
The price is $1149 for the completely assembled single/double density system and $795 for each additional drive. Now available from Thinker Toys, 1201-10th St., Berkeley, CA 94710.
For further information, contact Hilda Sendyk, (415) 524 2101. 1201 10th St., Berkeley, CA, 94710, USA.



Part of the wide range of "Verbatim" magnetic data storage products available from BFI Electronics Ltd. These include discs, cassettes, cartridges, magnetic cards, mini cassettes, mini discs and mini cartridges. In the background are disc/card storage boxes and files.

Details: BFI Electronics Ltd., 516 Walton Road, West Molesey, Surrey KT8 0OF.

The PET Printer. For delivery dates contact CBM, 360 Euston Road, London, N.1.



The PET Floppy Disk. For delivery dates contact CBM, 360 Euston Road, London, N.1.

New Vistas

Vista floppy disc drives available in the UK

*S100 compatible for Z80/8080 systems; *V200 Minidisc 5¼" £562 single and £812 double; *V250 full size floppy 8" £843 single and £1291 double.

VISTA offers either mini disc (200K) or full size floppies (250K). The S100 controller will control up to four drives (full size floppies) or 600 Kb on line respectively.

System start up is automatic from an on board bootstrap ROM. Software included is CP/M BASIC. The computer should have 24K RAM available.

Details from: U-Microcomputers, P.O. Box 24, Northwich, Cheshire CW8 1RS.

Users' Group

Keen Computers Ltd., are hoping to set up the U.K. Apple users group under the chairmanship of Dr. Tim Keen. It is hoped that this group will allow for the interchange of programs and information between Apple users, and the solution of any problems that may arise, both with hardware and software.

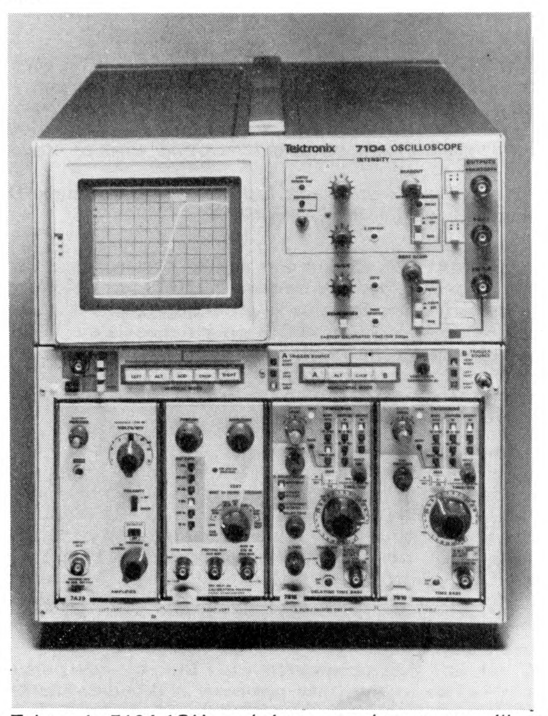
As the organising body of the users group, Keen hopes to provide all members with a regular newsletter, probably bi-monthly and to keep members informed of new developments.

Before the group can be fully organised, Apple users and dealers (and the new ITT 2020 owners) should contact Keen for more information.

Address: 5 The Poultry, Nottingham NG1 2HW.

Electronics from Tektronix

A new high-performance general-purpose 1GHz oscilloscope from Tektronix, the Model 7104, "incorporates the latest technology in integrated circuits, cathode-ray tubes and interconnection systems" and is compatible with the existing Tektronix 7000 Series of plug-in instrumentation. Among the performance features of the new instrument are a photographic writing speed of 20cm/ns, a risetime of less than 350ps, a horizontal (X-Y) bandwidth of d.c. to 350MHz, calibrated sweep speeds of up to 200ps per division, and a vertical sensitivity down to 10mV per division.



The Tektronix 7104 1GHz real-time general-purpose oscilloscope

For further information contact: Alan E. Hutley, Tektronix U.K. Ltd., Beaverton House, P.O. Box 69, Harpenden, Hertfordshire. Tel: Harpenden 63141 Telex: 25559

Micro-Aid (Lloyds Bank Chambers, Camborne, Cornwall) has a single sheet software catalogue listing useful programs for the MSI and SWTPC computer systems. Programs offered include mailing and Statpack (currently being featured in **PCW**). Micro-Aid also quotes for writing BASIC programs for any purpose, and has a consultancy service for small concerns wishing to computerise. Advice is unbiased. Write to Colin Chatfield.

TRITON expands
The Triton hobby microcomputer now has: a motherboard, with its own power supply (Kit form) and an optional case; an 8 K static RAM board; and 8 K EPROM board.
Full details from Transam, 12 Chapel Street, London N.W.1. Tel: 01 -402 8137. Also, a catalogue and price list is available on receipt of an sae.

SHADE Aid
Shade (Computer Services) Ltd., have developed a low cost V24/20ma convertor allowing V24 devices to be driven from a 20mA line and vice versa. Data rates of up to 9600 baud may be driven by the convertor.
Full details from Shade, 1 Patford Street, Colne, Wiltshire.

The good SEED
Strumech Engineering Electronics Division markets the Midwest Scientific Instruments 6800 system. The MSI 6800 System 12 is a complete system suitable for small business applications. An excellent catalogue sets out in clear detail other products offered by SEED.
Details and catalogue from Richard Hinton, SEED, Portland House, Coppice Side, Brownhills, Walsall. Tel: Brownhills 4321.

Software Architects Ltd. are holding a ten-day (non-residential) course on Systems Design with Microprocessors. Venue is the St. James Hotel, London, from 23rd April – 4th May, 1979. The course is given by Leslie Dewhurst, a leading Z80 specialist. Cost is £430 (excl. VAT). Ring 01 - 734 9402 for more details.

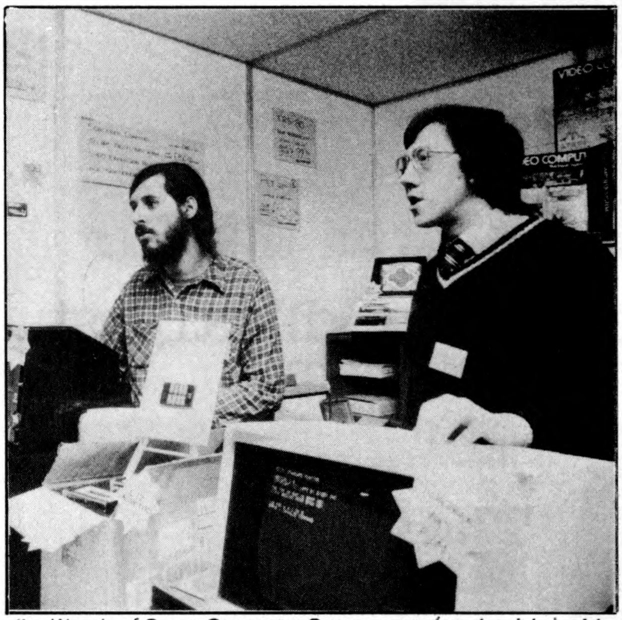
Popular Reprint
Computer Programs That Work, has now been reprinted. Three sections: Mathematics, Science and Games. Cost is £2.40.
Full details from: Sigma Technical Press, 23 Dippons Mill Close, Tettenhall, Wolverhampton WV6 8HH. Tel: (0902) 763152.

MAZUMA – It's no Rumour
A special leasing service for personal computers and associated hardware. Approved small business and personal applicants can lease, from 1 year upwards, the equipment of their choice against monthly rental payments made by banker's order. At the end of the basic lease agreement, rental can be continued indefinitely for a nominal annual charge or purchased outright.
Full details and application form from: Mazuma Ltd., Red House Studio, Ingatestone, Essex CM4 0BL. Tel: 027 - 75 3295.

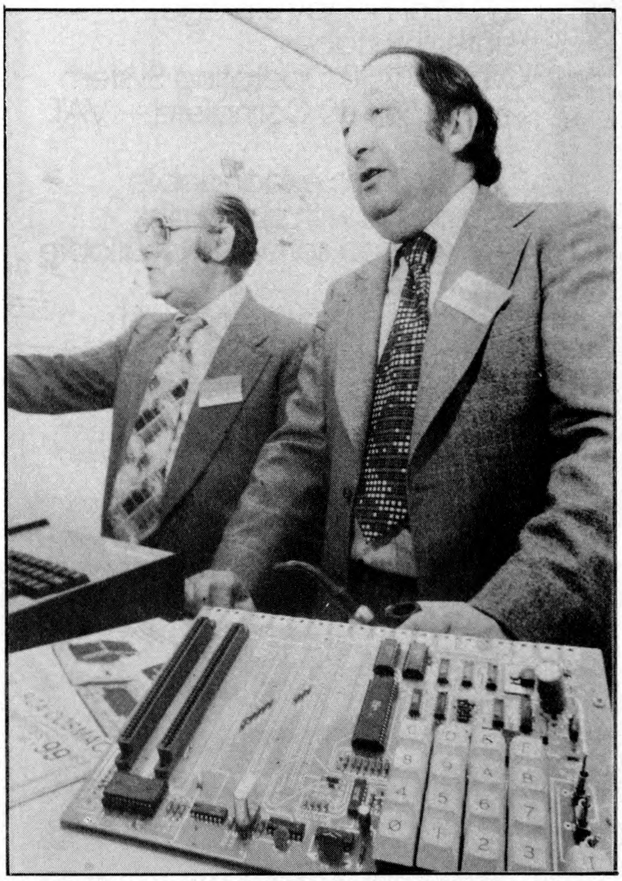
Coddling your PET
Intex (Datalog) Ltd., are offering a dustcover for the PET at £4.95 (inc. VAT, postage 40p extra). A 10% discount is offered to Pet User's Club and IPUG members. The cover is tailored to fit snugly and is strongly made.
**Details from: Intex (Datalog) Ltd., Eaglescliffe Industrial Estate, Eaglescliffe, Stockton-on-Tees, Cleveland TS16 0PN.
Tel: (0642) 781193.**

the Faces Behind the Places

Photographs *by Yoshi Imamura*

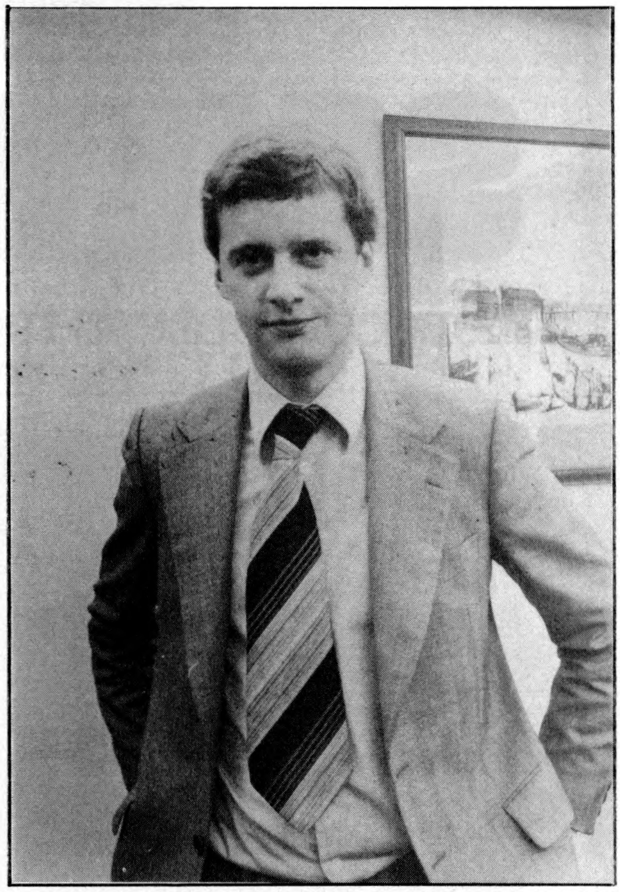


Jim Woods of **Comp Computer Components** *(on the right) with Steven Edelman of Ithaca Audio (an American outfit). The exciting Exidy Sorcerer computer, and the success story of British hobbyist computers, the NASCOM 1, are just two of the systems he deals in. Comp are at 14 Station Road, New Barnet, Herts EN5 1OW.*

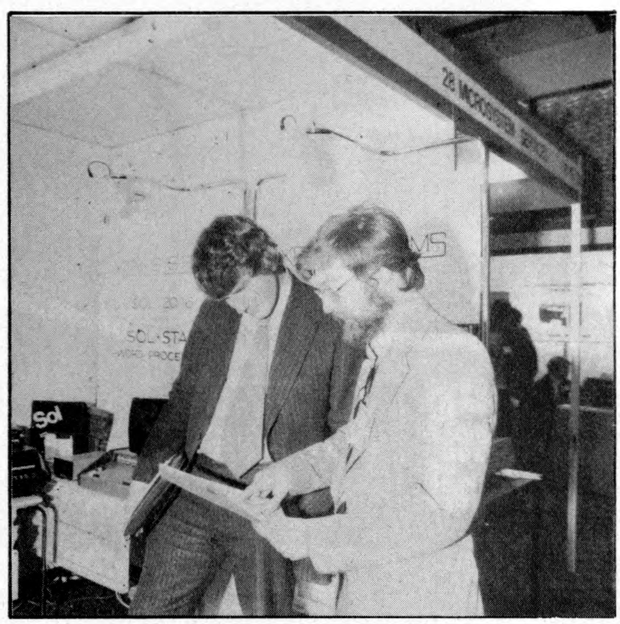


Mr. Harvey Lubin, Managing Director of **HL Audio** *& Mr. S. Larholt, Technical Sales Manager.*

HL Audio *market the redoubtable Elf II Computer. The Elf II has been very successful indeed in the United States and is now being marketed vigorously in Europe. Cost of Kit £109.56 (inc. postage and VAT). Full details from: HL Audio, 138 Kingsland Road, London E28 BY. Tel: 01-739 1582.*



Bruce Everiss of **Microdigital** *seems always to be one step ahead. Offers a big range of systems and books to suit every taste. A businessman with a genuine love of computing.* **Microdigital** *is at 25 Brunswick Street, Liverpool L2 0BJ. Tel: 051 236 0707.*



Jeff Lynch of **Xitan** *with a customer.* **Xitan** *is an authorised* **Comart** *distributor on the South Coast. Jeff Lynch is a* **PCW** *author (see Vol 1, No. 2) and* **Xitan** *offers courses for beginners in microcomputing. A serious dealer for those serious about small systems. At 23 Cumberland Place, Southampton SO1 2BB. Tel: (0703) 38740.*

Letters

A MK14 CLUB

Thank you for publishing my MK14 puzzle in February PCW.

As a result of the correspondence received on the subject I have decided to start up a club for MK14, and other SC/MP machine owners.

Initially the club will consist of a newsletter circulating among the members to which ideas, programs and problems can be added en route.

Geoffrey Phillips, 8 Poolsford Road, London NW9 6HP

OUTFINGER

I have been an MK14 user for some time, writing and running my own programs, I feel sure that many other people are using MK14's and I believe that a need for a users' club has arisen.

I would be willing to print and distribute (for a minimal charge) a magazine, if people could send ideas for programs and applications to me.

So come on MK 14 users, get your finger out and let's get things moving.

D. G. Johnson, 3 Mellor Drive, Worsley, Lancs.

ZAPPING THE Z-80

Your correspondent B. A. Martin (PCW, February 1979) suggests a PCW Z-80 based system. It is a sad historical accident that the Z-80 has gained so much popularity. The expanded instruction-set only slightly compensates for the abominable architecture it has inherited from the 8080. I could only laugh when I read Mr. Martin's comments about its "efficiency when applied to compilers, interpreters, and assemblers, and probably also in operating systems".

There is only one 8-bit microprocessor on the market to which this description can be applied, and that is the Motorola 6809, which is designed with precisely those criteria in mind ("A Microprocessor for the Revolution: The 6809", Terry Ritter and Joel Boney, Byte, Jan thru March 1979).

So my recipe for the system of 1979 would be a 6809 processor; 16K Bytes RAM; ROM BASIC, Assembler, etc; and a video display based on The Motorola 6847 video display generator chip, for those who want colour graphics and normal ASCII displays.

And for the early 80's, how about a 68000 with 64K RAM chips, magnetic bubble mass storage, etc.

But, like Mr. Martin, I do endorse the idea of a PCW system, but designed looking forwards and using state of the art components. Let your competitors publish designs for 8080 and Z-80 machines. I for one don't want to buy or build a design which is already obsolete, be it on one PCB or a dozen.

P.S. In response to your request for information on suppliers, etc., may I suggest you include: Associated Processors Ltd., P.O. Box 1580, Wellington, New Zealand, who produce excellent 6800 — based equipment. And one hobby computing club you won't have heard of, the Wellington Microcomputing Society, P.O. Box 1581, Wellington, New Zealand, which has been around since mid 1977.

P.P.S. I'm sure there are dozens like me who'd be willing to help with the design and software for a 6809 system, so go to it please!

Phil Randal, 22 Oakfield Road, Finsbury Park, London, N.4

PCW *The Motorola 6809 was featured in Vol 1, No. 11* **PCW**

THE PRESSURE GROWS

I was delighted to read B. A. Martin's letter in your February number. Ever since the very first issue of your magazine, in which you featured a Monitor Programme which could hardly be of use except in a home-brew Z80 system, I have been waiting for such a system to appear.

I am not sure it is a good idea to link it with an existing Computer manufacturers. Surely any Companies who *can* produce a viable Z80 system have by now done so, and are really only interested in marketing their own products. If, on the other hand, PCW does come through with a really attractive system, those very companies will be quite happy to offer made-up boards for it!

Please let us have at least the option of a standard bus structure. Much easier, surely, to offer the basic circuitry in a standard form and let those who want to use non-standard boards expand from that than contrariwise.

But please let it be soon; a lot of people, myself included, came into PCs with PCW, and are just about ready for this now. May I predict that the only way to prevent a User Group would be with machine guns.

G. K. Armstrong, 123 Hamlet Gardens, London W6 0TR

PCW *We must once again tell readers that we do not intend undertaking a PCW System* **PCW**

RANDOM HARVEST

*Re Random Writings by Michael James (***PCW** *Feb '79).*

I whole heartedly agree with Michael's comments regarding the annoying feature of 'RANDOMISE', that you cannot control where you start in the sequence of 'random' numbers generated.

As a teacher, when using a simulation program, I want to compare the results obtained when using the same parameter values with different random data, and different values of parameter with the same random data. This enables the pupils to compare their hypotheses, for after all, they could have obtained that one in a hundred result, i.e. Murphy's Law, which must occur if it is a good random number generator.

The BASIC interpreter of the HP2000F has this facility. To start at a definite point in the cycle of random numbers one first calls RND with a negative argument. The value of the argument is then used to calculate the first number of the sequence; so to change the sequence obtained, one changes the value of the argument. For a description of how it is done see John Rickard's article in Computer Education No. 28 or the ILEA's Computer Studies Newsletter No. 18.

Peter Butt, 13 Abercorn Gardens, Chadwell Heath, Romford, Essex RM6 4TA

THE EIGHT BIT MISUNDERSTANDING

With regard to Mr. Finlay's 'Micro Muse' article (PCW Jan '79) the transformation of his "All Quiet on the Western Front' text stems from a misunderstanding. The 'timing-parameter' is actually an offset for a program-counter relative address so the result is not to slow down the scan, but to replace byte OF28 with CO, changing an 'XPAH 2' instruction into the start of a 'LD' instruction. The more wide reaching effects are to prevent pointer 2 being reset to the beginning of the text, and so the attendant effects zoom inexorably down into the program, hence the disruption and the infinite loop. Also the transformed text as stated will not end at OFAO because the program looks for bit 2^7 set and hence the text length will depend on the contents of the RAM used.

The effect *is* well worth seeing and Mr. Finlay is to be congratulated on his series of articles.

W. T. Roberts, 13 Whipton Road, Exeter, Devon EX4 8AT

HIGH AND LOW SPEED BUSES

I was extremely interested to read Mike Lord's comments (PCW Open Page, Jan.'79) about High and Low-speed buses, particularly in connection with peripheral interfacing. Standard computer peripherals seem to me to be fairly well catered for by manufacturers (I mean VDU's, printers, floppy discs etc.) In particular the IEE or HP-IB system is becoming increasingly popular. For 'special' peripherals, such as may be needed in process control, a much simpler — but even more flexible — system is needed. Machine tool control, for example, would cost a large fortune if implemented using HP-IB, if the peripherals were available. A system well known to physicists, but curiously not widely known outside is CAMAC. This is a modular interface system using a standard rack/power supply and plug-in modules. It is capable of high data transfer rates over its two 24-bit data buses (separate inputs and outputs). It works well, but is still very expensive. The rack power/supply will cost around £1200 without modules.

My colleague, A. R. D. Rodriques, and I, have designed a simplified CAMAC-style bus system, together with a small set of modules, which whilst not capable of the data rate of CAMAC, is much cheaper. It has been in use in our laboratory for around 18 months. We originally used it to interface an HP 982S machine to our x-ray diffraction experiments. We have since made a version which interfaces to PET, and I shall shortly do the same chore for an LSI-11. Our bus consists of a 16-bit bi-directional data bus, an eight bit address bus and a few control lines. All peripherals connect to the data bus through 3-state buffers (for outputs) or latches (for inputs). Outputting the appropriate address enables the 3-state buffers to strike the latches or whatever. CMOS is used for inputs, CMOS buffers for outputs so loading is never a problem and you don't need a 10-amp power supply.

Although it uses more wires than HP—IB (just!) the protocol (and hence software) is much simpler. It is physically realised in a 3-U rack, with standard 0.1" 43-way edge connectors and so is relatively cheap. It is most suited to relatively simple peripherals. We have put in stepper motor drives, pulse counters (scalers), D to A converters and A to D converters. Up to 255 functions can be controlled!

In the PET version, the connection between PET memory bus and the interface bus is achieved via 2 of MOS Technology's 6522 interface adaptors which, by the way, are splendid animals. 300 bytes of EPROM handle the PIA conditioning and sequence of I/O operations required by each of the peripherals mentioned above. Parameters are passed using BASIC variables, the parameter in USR being used as an index pointer to access the correct subroutine. This means that the bulk of the control routines can be written in BASIC, with all the consequent ease of editing etc., and a minimum of meaningless POKE's and PEEK's. It all hangs together very nicely; setting up a completely new experiment and writing the control program usually takes only a matter of minutes, or at most hours.

A complete description of the system is to be published in the "Journal of Physics (E)" and if any readers are interested, a pre-print may be had from me by sending a stamped, self-addressed envelope to me at the Department of Physics, King's College, Strand, London WC2R 2LS.

D. P. Siddons, Department of Physics,
University of London King's College, Strand, London WC2R 2LS

THE CUSTOMER OUGHT ALWAYS TO BE RIGHT

There has been considerable comment in your magazine on the subject of the prices charged and delivery times of personal computers. The price gap between here and the U.S.A. has I think been adequately explained as a matter of the differences in economic politics between the two countries. The problem of delivery times is probably due to the fact that many companies in the field of design and manufacture have found it difficult to find retail outlets with such a new type of product. They are not experienced in this themselves, and perhaps overcompensate for the enormous delay between placing an order for advertising and the advertisement appearing. They have to estimate that their product will be available in so many months, and then place the advertisement in the hope that their estimate is correct. If they wait until the product is ready before advertising, then there is the possible loss of customers to someone who hasn't. Obviously it is inexcusable to go on advertising a product when it is not available due to some snag.

The consumer is best advised to take the following course. Buy from a retailer, and telephone to ensure that stock is present before placing an order. Credit card orders can be placed with the instruction "debit my account when the goods are despatched". In this field, where prices are dropping annually, once a decision has been made on a particular item, it should be obtained fast. If an order is placed, and money sent, only to result in a four month delay, then by that time a similar and cheaper system may well have appeared.

The other problem that suppliers have to face is potential customers' questions. It has been my experience that suppliers are by and large helpful and answer promptly. However obviously once the choice is made some of the suppliers have spent time answering unfruitful correspondence.

Clearly magazines such as yours help here by answering potential queries in review and other articles. Personally I should like to see more articles on the fundamentals of machine language. Although the results of the "Micro Muse" (**PCW** Vol 1, Nos. 7, 8, 9) are hardly as exciting as a chess program or as useful as a business program, this sort of article does give one the possibility of understanding simple machine language programming in easy stages, and with the possibility of "hands on" experiments.

John de Rivaz, BSc(Eng)
West Towan House, Porthtowan, Truro, Cornwall TR4 8AX

PCW ***Look out for an article by Brian W. Haines on just this topic.*** **PCW**

IN MORE PRAISE OF THE PDP11

The articles "In Praise of the PDP11" by Mike Lord which appeared in recent issues of PCW have prompted me to examine the PDP11 a little more closely. There are a couple of points which I believe could be of note:

1. The PDP11 allows operations such as MOVE, ADD etc., to take place between ***any two locations in memory*** and not just a register and a memory location.
   e.g.
   (i) Conventional – e.g. Honeywell 316 (ii) PDP11

   ```
   LDA A
   ADD B      becomes      MOV A,C   on a PDP11
   STA C                   ADD B,C
   ```

   Apart from the saving in lines of source code, another (and more important) feature is that in (ii) none of the PDP11 registers are corrupted. In (i) the main register (the accumulator) is corrupted.
2. PDP11 double operand instructions (like those above) occupy between 1 and 3 words of memory depending on whether the operation is on register – register, location – location etc.
   This would mean that ex. (ii) would occupy 6 words of memory while a more conventional 16 bit minicomputer as in (i) would only occupy 3. The catch of course is that PDP11 operand addresses can be a full 16 bits, i.e. the locations A, B and C may be anywhere in 64KB of memory. To achieve the same in (i), some form of indirect addressing would be needed and more memory would be used in address pointers. This means (i) would also tend to run slower than (ii) as more address cycles would be needed to select the operand address.

The above are just two of the features which make the PDP11 a unique machine and could have contributed to its much deserved success. I am in fact involved in the setting up of a PDP11 users group. I would be pleased to hear from any user of a PDP/LSI11 running under any operating system for any application.

P. C. Harris, PDP11 Users Group,
119 Carpenter Way, Potters Bar, Herts, EN6 5QB

ORIGINAL SIN

I was one of the original subscribers but despite the statement on page 3 of the January '79 issue I have *not* received any notice that my subscription is due for renewal.

I am therefore forwarding of my own volition the sum of £8 and will leave it to you to sort it out as I do not want my PCW to stop.

Whilst writing though I would like to say that whilst I find PCW interesting I could wish it didn't make so many printing mistakes.

I know it can be said in mitigation, that this is only in common with every other computer magazine, particularly the American ones but that is no excuse for a lack of diligence.

It is particularly annoying when an abbreviation, which anyway often has any number of meanings, then gets printed wrong.

M. A. J. Baker, 11 Wilmots Way, Pill, Avon BS20 0JT

AND CRETIN SHALL SPEAK UNTO CRETIN?

I have taken PCW from the first issue and while I generally find the contents interesting it seems to me that there is too much space taken up with frivolous and trivial computer applications e.g. games and short machine language programs of little general application. My main interest in computing is to ease the drudgery of repetitive calculations in such fields as electrical fitter design. Obviously for this sort of work a high level language is needed but why must it always be the cretinous BASIC? I cannot recall seeing one word in your pages about that most elegant of programming languages, APL, one eminently suited to micros and now available on disk for the Z80.

One thing that has long interested me is interfacing a calculator chip to a micro for fast floating point calculations. One or two articles have appeared in various publications on this theme but they all involved programming in machine code. Any ideas for hardware to interface a calculator chip to a micro *which would leave all existing software unchanged,* would be most welcome, as would any hardware constructional articles and articles concerning the writing of compilers and interpreters.

Greg Trice,
Flat 2, 8 St. Mary's Road, Leamington Spa, Warwickshire.

A STERN REBUFF

Re your footnote to my book reviews (Vol 1, No. 10) – it is not whether a computer is personal or not that makes it a toy but the mentality of its owner.

Anthony Aylward, 194 Balmoral Road, Gillingham, Kent.

IN VERO VERITAS

I have been reading with great interest the various articles published on the E78 Buss System.

We, as major manufacturers of Eurocards are very confident that this size of Card is here to stay, in fact we now have another Board being added to our range, 233mm high by 220mm deep, in the Vero High Density Packing Pattern.

With reference to the Bus, no one seems to have considered the problems of tracking. There was one correspondent who was considering a 96-way DIN Connector. These, by the way are available from Vero under special order, but these Connectors could only be used with a Multi-Layer Bus, with all the inherent problems of manufacture.

I feel through talking to many people that two 48-way/96-way connectors on a Double Eurocard would be sufficient. This would enable easy Track Lay-out, and would have a smaller number of ways, i.e. 96 against the 100 of a S100 Board.

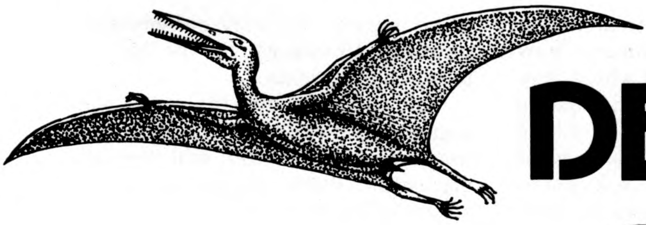
Congratulations on a splendid Magazine, we in fact received in the order of 200 enquiries on the Press Release you did on our prototype Board. Also congratulations on a very good Exhibition, which we will hope to attend in 1979 as Exhibitors.

D. A. J. Smith, Field Sales Manager,
Vero Electronics Ltd., Industrial Estate, Chandler's Ford, Eastleigh, Hants, SO5 3ZR

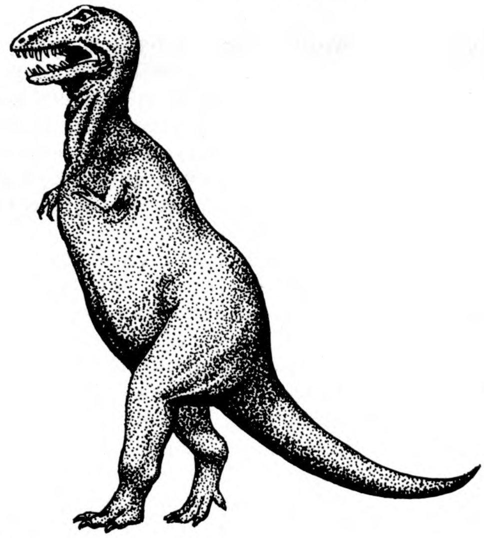
SCHOOLBOY'S REQUEST

Congratulations on an excellent magazine, simple enough for beginners, detailed enough for the experienced. I have a plea to put out through your column to anyone who could supply me with any information whatsoever on microprocessors, which I can put to use in my Computer Studies Project, for which I would be very grateful. Also it would be pleasant to hear from any micro users in the Wolverhampton area, as at Ounsdale High School, the Computing Department, with the aid of the PTA, under Mr. W. A. R. Blackford has just bought a Triton 300 Micro and an information exchange would be a valuable asset, allowing us to pick the brains of experienced micro users.

A. Fletcher (Aged 16)
7 Meadow Lane, Wombourne, Wolverhampton WV5 9BT



DEBATE OF DINOSAURS



PCW Conferences are a little industry on their own. How worthwhile are they? The author gives his impressions of three computers-in-medicine conferences he attended last year. A picture of national attitudes. PCW

Hugh Price

Consultant Physician, Charing Cross Hospital

Germany: Walled In

In my conference wanderings I first stopped off in West Berlin at the AMK annual conference. Berlin is a curious place, almost an abstraction. Though, as one would expect, the conference itself was concrete, well organised; but the sessions followed the now accepted pattern with many of the usual faces. Having attended these almost religious rituals for the last twenty years, I find now that a curious air of stalemate pervades them. Papers are very much the same from one year to another. Yet titles and abstracts try to give the impression that some new vital subject is being dealt with. Never mind that, they are mainly concerned with the old faithfuls; large data banks, networks and regional systems, modelling and simulating; all vain attempts to persuade a somewhat resistant public and the denizens of the corridors of power to make more use of computers, to justify the heavy capital investment.

I remember the days when the majority of people who attended conferences were either the technical people who were designing and building computers or those who were engaged in trying to use them. Today there are the big names, some of whom, from the way in which they answer questions, would appear to have never gazed inside a computer and seen its international guts or the fusty rooms occupied by the systems boys. The big names have always dealt with management theory rather than cybernetics. We had almost identical managers from all the big computer centres. They all look alike presumably because all their machines look alike; like master like dog. The fact that the microcomputer has now reared its tiny head hardly caused a ripple in the smooth course of this conference. Nobody dared mention, except your truly, that microcomputers are going to alter the whole of our concept of computer power, certainly our modes of application in our socioeconomic system.

The standard response to the challenge appears to be based on the pious hope that these new machines can be applied in the same old way. The few papers that dealt with the future seemed somehow to be aware that something new was going on; the possible effects of these changes were treated vaguely. There was nothing about the tremendous potential of microcomputers: their internal strategy, their economics and their pervasive impact not only on society but on all the individuals within it. The inertia is understandable because computers are among the herd of sacred cows kept by the establishment, and because of the considerable capital investment in major systems. These big animals are, with their keepers, a part of the management structure.

But in the future? Then, computers are going to be in the hands of ordinary people, no need for having workers on the company board, production could be controlled from the craft workface. In fact, computers will be able to take up their real role: *not* as expensive desk calculators for government departments, town halls, nationalised industries and the multinationals. They will be available to anyone and will deal with *communications* between human beings and their environment. Not only that, the person of average intelligence and training will very quickly, thanks to the many simple languages such as BASIC, be able to instruct his or her own computer without the aid of the bureau or the local computer manager.

What happens to the sacred cow then? Computers now can cost hundreds of pounds and the price is still falling. What does this mean? In fact, it will make the small business, without an elaborate corporate structure, viable. The shop keeper's till will be his microcomputer; the newsagent will also sell micro-based electronic information.

I read a paper on this subject to the conference and I was interested to observe the response. At previous sessions the papers had dealt with the problems arising from the development and maintenance of large data banks. The Chairman often had to cut the session short, there were so many questions. But on the future of computers as dictated by micros, there were very few questions. And the fact that these computers are likely to remove the necessity for a very large number of managers was not really something any of them would dare to contemplate. So, nothing new in Germany, where the slogan appeared to be: *the past and present are fine, who cares about the future!*

France: Descartian Doubt

Moving from the German conference of AMK and Large Data Bases with its structured formality and dedication to the established order, to the alive stimulating Paris was like passing from the structured technology of IFIP and the ordered academic respectability of the B.C.S. and Datafair to the live area of Tomorrow's

World. The French are aware, and unafraid of mentioning it in public, of their problems of national conservatism and an immobile industrial population. The opening session was chaired *not* by a computer man of the establishment but by, horror of horrors, a journalist.

This annual *Convention Informatique* had a very different atmosphere; the French are certainly with it. Of course there were the usual papers dealing with the past but did the conference open with some dignitary from the City of Paris as we have had in the past? No, we had the highly intelligent woman Minister of Academic Education on the platform with manufacturers, industrialists, manufacturers of computers, users and academics and of course the journalist Chairman, in a round table discussion on what lay ahead in the next five years. The discussion? On existing and future problems rather than the past.

The President of the Republic wrote to the conference chairman: "The development of data processing applications is a factor of the transformation of economic and social organisation, and of the way of life; our society must therefore be in a position to promote and to master it for the purpose of democracy and human development."

And the Minister of Academic Education took an active part in this lively round table discussion on where we are going and what we should do. There was a frank admission that they did not know how many people they would require in Electronic Data Processing, how and where they should be trained, and what should be done to help the population to catch up with the fast moving front of this technology. Discussed were special training for professors in all fields, for technical and professional training, not forgetting the schools; pointing out that there is little point to teaching children in junior school to do mathematics with matches when they are already bored with electronic calculators.

The Minister posed the question: is this Transient Technology? Or a new language or code, a new way of thinking, a new way of doing? She stressed that this concept had not yet penetrated the walls surrounding the applications of existing systems. Mr. Thipauden, an industrialist, pointed out that because of the structured nature of universities today, the computer people of the future should not be trained there, but in industry itself; Mr. Barazer of IBM agreed that a break-down must take place in the barrier between manufacturer and user, especially in the small firms.

This key-note discussion clearly highlighted the obvious: we do not know how we are going to train our computer staff in the future, how many we will require and where they will be. This was the most stimulating session I have been to for a very long time. Of course, the conference went on to deal with the usual subjects of the past. In my own session I dealt again with the socio-economic and technological changes we are now experiencing. There was a much better response than I received in Berlin.

The poor computer has been badly used. It has become a sort of thing in itself, an adjunct of a management taking care that its position in the hierarchy is maintained and making certain that the right amount of paper is vomited out at the right time.

We don't want paper, we don't want management, we want to be able to control what we do, what we provide for society. This, whether we work in industry, in education, in hospital or in the academic or scientific field. We don't want to have to fill in forms, get our secretaries to take things down. When we do our normal work, our actions will in fact be the input to the microsystem. For example, when the millwright at his lathe turns the knobs to the settings required, these actions will be the input to the computer system: the same for the small shop keeper when he uses his cash register. The doctor making demands on the technological backup in the hospital will be served by voice input. We are already carrying out preliminary experiments on this in my hospital. In other words computers are merely a new type of communication channel.

Man has been trying to improve communications since the dawn of history. Previously, all oral and manual communication has been complicated and time consuming. Heavy industrialisation has meant a separate industry of management and data collection. With microcomputers man can go back to the situation where as he does his craft work this function is recorded automatically without further effort on his part.

Japan: Produce, produce!

So finally to Japan over the polar ice to *MEDIS '78*. With unusual honesty, the key-note opening speech succintly described the now well accepted reasons for only 23% of Japanese hospitals having computers. But no mention of micros or the explosion of this technology in a country which is producing most. And no answer to the question: why not give computer power to the doctor? Japan gives more electrical power to the workers and has more technology than elsewhere. But this was not reflected in a conference which, like the basic Japanese society, was very conservative, unwilling to change. So the big systems reign supreme.

Not unexpectedly, this was a well organised medical computer get-together; perhaps too well for real discussion. Polite acceptance of the effects of micros, but mainly in the laboratory and other fringe and back-up areas. My own carefully polite contribution on the economic, socio-political effects of micros — like the micro now helping to guide the automatic cutter at the coalface — was described as heretical by an American. The rest of the sessions were a mirror image of the other European meetings — disappointing, and not unlike the King who could not read the writing on the wall.

What do these musings add up to? Having signally failed to excite the ordinary person and the occupants of the corridors of power that computers are the saviours of our much vaunted socio-economic system, there is a view that perhaps computers, which were developed as management things in themselves, are at fault. Perhaps, dare I suggest, it is not computers that are at fault but the viewpoint. Little or no attempt to understand their potential.

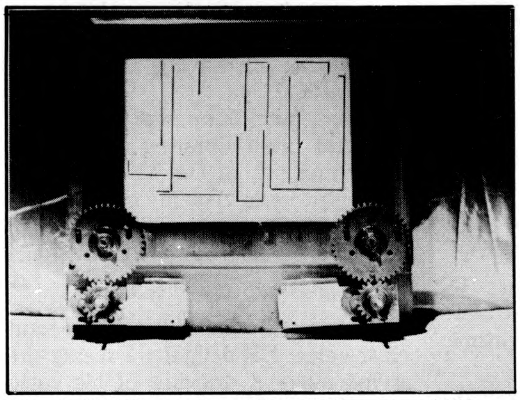
The objective should be to spread computers as widely as possible to improve our working and living standards. In the service industries such as the Health Service this can be the better delivery of health care and, equally, better opportunities for clinical work for doctors and other primary care workers.

In the mode we have employed computers so far, as expensive desk calculators and latter day memory aids, this is not on. As cybernetic assistants — possibly, yes.

Again, looking at the NHS as an example of a service industry we could spread computers to link more doctors, patients and facilities in a new type of functional health or diagnostic information pathway — this is not immediately possible, as the computer establishment would claim, with our existing set up. But have microcomputers given us the key to the pathway that can sidestep the big systems?

The march of this technology has no real terrors: with cheap universally available microcomputers, instead of the neglected end-user being offered what he does not want or need, he can now decide what he wants and go out to the market place and buy it at very low capital and running costs.

The computer is now no longer important in itself. But the established computer world doesn't want to know and is closing its ranks and massing its pikes to defend its honour to its hard core.



Simple X-Y-Plotter from 'Etch-A-Sketch' ca 1969

An Early Steam Computer

Eric Finlay

"... Don't tell me you're one of those guys who did it all in Texas 30 years ago." *W. C. Fields*

"I did it my way ..." *Frank Monaco*

Between 1968 and 1970 I carried out many experiments with electronic circuits to see how far I could progress towards making a computer which would produce drawings. I studied transistor theory (Transistor Fundamentals Vols 1 and 2. Charles A. Pike, Foulsham-Sams 1968; Pulse and Logic Circuits, Angelo C. Gillie, McGraw Hill, 1968; Electronic Computers, S. H. Hollingwood and G. C. Tootill, Pelican, 1965; A.B.C's of Boolean Algebra, Allan Lytel, Foulsham Sams. 1969.). The most helpful book I found to be 'Computers Self-Taught Through Experiments' by Jack Brayton. (Foulsham-Sams 1967).

Brayton's book, a model of clear exposition, connected all the loose ends of information and experiment floating around at that time, and gave precise, well illustrated instructions for making a digital computer from transistors, diodes, resistors, capacitors, switches and lamps. His approach pre-dates RTL,DTL,TTL and CMOS logic advances, but since the basic problems behind all these types of integration (up to, and including the microprocessor) are similar, I still regard it as the book which gave me practical insight into how computers work, as calculating machines.

Transistors were cheap and easy to obtain, and my main problem was to decide which sort of readout machine to construct; this was directly conditioned by what I could afford, and what sort of imagery I wished to produce.

One morning, as I waited for a bus outside Smith's in Sloane Square, my eyes fell on an Etch-A-Sketch drawing toy for children. A moment of inspiration occurred. In the evening, I hurried home with a heap of assorted toys, batteries and connecting wire. Two days later I emerged from my small laboratory with a perfectly reasonable x-y co-ordinate plotter made by attaching two reversible motors with plastic gears (Fischer-Teknik) to the Etch-A-Sketch, the prototype being driven by hand-held switches.

I then began to design a simple system for converting sound-impulses into a graphic analogue. Strictly speaking, to call the final result a 'computer' might cause considerable debate at Cal. Tech., or M.I.T. However, considering modern technology, it might be said to behave very like a microprocessor dedicated to the function required. My excuse for unearthing this electronic fossil is that it may stimulate further experiments with peripheral machines.

I had a cheap, but serviceable tape-recorder with an earpiece output. My next purchase, from Proops, in Tottenham Court Road, was a superb (Sangamo-Weston) D.C. Moving Coil Relay, for the remarkably cheap price of £2. Hardly an advance on electrical theory since the time of Faraday, this beautiful device is capable of switching on an input of only 25 Micro-Amperes, and is like a sensitive meter-coil converted to the job of switching high voltages from low inputs. With the addition of a rectifying diode, to change the alternating current output of the tape-recorder to direct current, sound-peaks from the taped material could switch 9-volt circuits with a minimum of effort. The high inertia of the motors and Etch-A-Sketch made 'contact-bounce' an insignificant problem. This simple approach gave me an analogue output to the plotter.

My next problem was to design a regular output to the second motor, and be able to reverse the direction of the plotter-stylus when the stepped output reached the edge of the plotter screen. I used a variable multivibrator as the 'clock-pulse', and to isolate this from stray feedback I made it drive a reed-relay coil by way of two diodes. I added some signal lamps to the system as a direct indication that the various sections were in working order. Then I switched on the tape-recorder, and with very considerable interest watched my 'computer' drawing a visual analogue for the acoustic output fed to the micro-relay.

Finding the task of manual polarity-reversal beneath the dignity of an

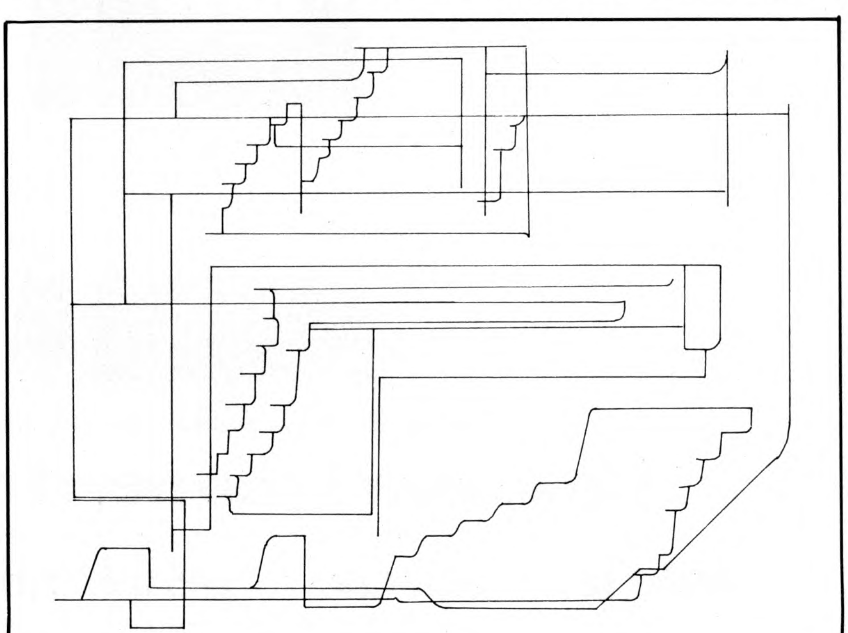


Fig. 1 *Typical Sound-Analogue Image. ca 1970*

electronics pioneer (I liked to think!) I then designed an automatic polarity-reversal system timed by another multivibrator.

The tape-recorder had a microphone monitor which enabled me to use direct speech to drive the plotter. I could either photograph the resultant drawing (using a polaroid filter on the camera to cut out reflections from the screen), or, as I decided, I could just trace the images carefully if they seemed interesting.

I announced the completion of my computer to George Fullard, who was then head of Sculpture at Chelsea. George's eyes narrowed visibly.

"Would you like to demonstrate it?" he asked.

"Sure," I replied, "I'll wire it up and have it working by Thursday."

I brought the components to school packed neatly in a portfolio. To the background of a lunchtime game of darts in the sculpture — school staff room, I assembled my digital-analogue-drawing-machine in an annexe.

When everything was connected up, and multivibrating away like something out of Dr. Who, I invited the sculptors to view my masterpiece. Their incredulity was most gratifying. I handed the microphone to George.

"Sing something" I said.

He sang the Marseillaise in a loud clear voice. His delight on seeing the stylus make a drawing of his voice must have matched the historical enthusiasms at Menlo Park. It was an instant success.

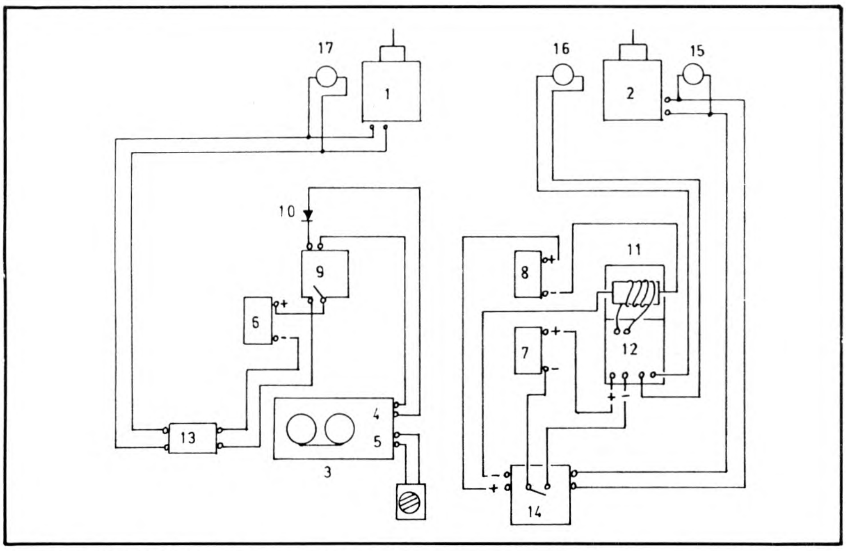


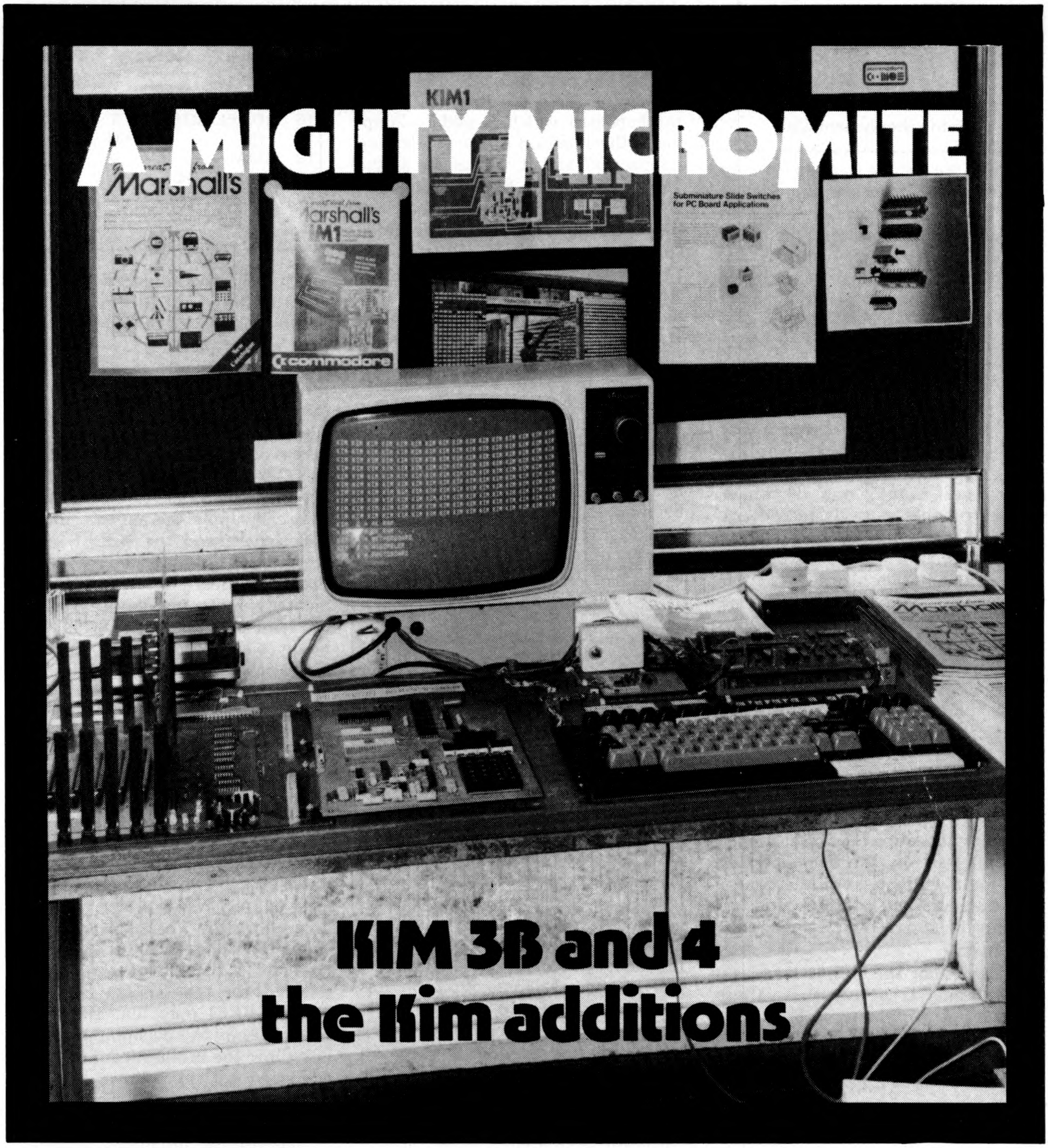
Fig. 2. *Basic Digital-Analogue machine*

Legend

1 & 2. 6 Volt D.C. Motors.
3. Tape-Recorder.
4. Recorder Monitor Output.
5. Microphone Input.
6. Micro-Relay (25μA Sensitivity).
7,8,9. 9 Volt Batteries.
10. Diode.
11. Reed-Relay.
12. Multivibrator.
13. Polarity Reversal Switch.
14. Multivibrator On Switch, and Reed-Relay Polarity Reversal Switch.
15,16,17. 6 Volt Lamps.

Multivibrator Values.

Transistors. NKT 153.
R1, 3,300 Ohms.
R2, Variable 100K to 1K
Capacitors Microfarad, Electrolytic. 100



A MIGHTY MICROMITE

KIM 3B and 4 the Kim additions

Jim Morris
Queen Mary College, Dept. of Computer Science

PCW *See "A Mighty Micromite", Vol 1, No. 4* **PCW**

Introduction

This is a review of two, currently available, additions to Commodore's Kim—1 microcomputer board. For those who already know and love the Kim—1 this will provide some information on expanding the system and for those who are still looking for a microcomputer system it should outline the features of the expanded Kim system (no review of the Kim—1 is undertaken here.)

As you may, or may not, know the Kim—1 is a functional microcomputer on its own with I/O, cassette and TTY interface and memory on-board, with the addition of the Kim—3B is an 8K byte memory expansion module that can be used with or without the Kim—4 motherboard.

The motherboard provides the interface between the Kim—1 and up to 6 system expansion boards, doing all the decoding and buffering that is needed. It also has two regulators which supply the +5 volts and +12 volts required for the Kim—1. Only a power supply is needed which can handle +8 volts, and +15 volts (and optional —15 volts) raw D.C., with a current rating of 1 amp for the Kim—1 and 2 amps per 8K memory board.

For the purposes of this article I was loaned three boards: the Kim—1, 3B and 4 by A. Marshall (London) Ltd. my own Kim—1 being involved in the control of an experiment.

Unfortunately no documentation at all was available on the motherboard; so all the information in this review was obtained by using the system rather than by copying the technical manuals, which I was told would be available in this country soon, and if the Kim—1 and 3B manuals are anything to go on, it promises to be fairly good.

Firing the System Up

Luckily the whole set-up is very simple to put together; the Kim—1 plugs into two horizontally mounted 44-pin, female connectors, the memory plugs into any one of the six vertically mounted female connectors (with card guides), and the power plugs into a separate socket altogether, which accepts the +8v, +15v, optional —15v, and earth leads, from a raw D.C. P.S.U (e.g. just a transformer, rectifier and smoothing capacitor.) The only adverse comment that sprang to mind at this stage was that the module could be plugged in either way round, one of which would probably cause instant death to the 2114's etc; the memory board had a polarizing slot on the connector, but the mother board had no bar in which it could locate. I presume this will not occur on any future boards being sold, as the one I had was somewhat of an exception being described, at the time, as the only one in the country and therefore not typical. However, in case the same does happen to anyone, it goes in with the component side away from the Kim—1, as the manual points out.

After connecting the power supply and switching on, the Kim operates as it would without the motherboard, with the important addition of 8K of memory on top of the 1K already on the Kim—1.

Other than the power leads, no other wiring is necessary, as those with Kim's will know a few other connections were needed to get the Kim—1 alone running, like the decode enable being earthed etc., but the motherboard handles all that for you.

The Motherboard

So for the person who is interested in a microcomputer system of 8K or more (48K max. using 6, 8K boards) with little fuss, this is comparatively simple to get up and running, so long as he, or she, can get a power supply.

He, or she, can also plug in the other Expansion modules which will be available with no extra trouble, no bugging problems or decoding, as the motherboard seems to have it all sorted out.

Connectors

For the other person who already has a Kim—1, and has presumably built the interface for the cassette, TTY, I/O ports etc., (my position) he will want to know if this all has to be rewired; the answer is *no*. There is a connector" and it will as far as the motherboard which is said to "duplicate the function of the application connector" and it will as far as the above mentioned facilities are concerned. Being inquisitive (and not having access to the manual) I decided to check the connections, and found that most of them are connected straight to the Kim—1 application connector, except pins B, C, D, E, F, H, J and K. On the Kim—1 application connector these are the K0—K7 and decode enable lines, which are memory decode signals and are dealt with by the decoding logic on the motherboard, and thus being obsolete the relative pins on the motherboard's application connector have been reassigned as follows:—

- B is a buffered IRQ line and is connected to pin 4 of the Kim—1 expansion connector via a buffer.
- C is a buffered NMI line, connected to pin 6 of the E—C on Kim—1 via a buffer.
- D is the RST line connected directly to pin 7 of the E—C on the Kim—1.
- E is a buffered RDY line connected to pin 2 of the E—C on the Kim—1.
- F, H, J and K are not connected at all.

So long as the above connections have not been used, (unlikely, and un-necessary when using the motherboard) the original applications connector will plug straight onto this connector, keeping all the functions the same. Of course the original +5v and +12v power connections on pins A and N will have to be removed, as the power is now being supplied by the motherboard, and appears on these pins. It does not matter if the link between the old decode enable (pin K) is removed or not as pin K on the motherboard is not connected.

There are six female 44-pin connectors, with card guides arranged vertically on the motherboard for the expansion modules, and the bus is also brought off the back of the motherboard.

Standard bus configuration

1	GND	A	GND
2	B SYNC	B	BAB 0
3	B RDY	C	BAB 1
4	B IRQ	D	BAB 2
5	—15v	E	BAB 3
6	B NMI	F	BAB 4
7	B RST	G	BAB 5
8	BDB 7	J	BAB 6
9	BDB 6	K	BAB 7
10	BDB 5	L	BAB 8
11	BDB 4	M	BAB 9
12	BDB 3	N	BAB 10
13	BDB 2	P	BAB 11
14	BDB 1	R	BAB 12
15	BDB 0	S	BAB 13
16	BD SELECTED	T	BAB 14
17	+15v	U	BAB 15
18	DMA	V	B 02
19	+8v RAW DC	W	B R/W
20	+8v RAW DC	X	B 02
21	+5v	Y	+5v
22	GND	Z	GND

The signals prefixed with B have been buffered by the motherboard before going into, or coming out of, the Kim—1 board.

Two signals here are not found on the Kim—1 expansion connector and they are: the DMA signal, which allows the motherboard bus to be taken off the Kim—1 bus, allowing an external device to access the memory on the motherboard bus, so long as it can supply the address and control signals; and the BDSEL signal, which does not have a connection on the motherboard bus, although documented, but could be used to take the same signal from the 8K memory board to a DMA controller, the signal indicating that the board is being addressed.

Decoding Logic

The motherboard provides full decoding for the address space on the Kim—1; taking into account the on-board memory and I/O of the Kim—1 board. It will also allow memory on the motherboard bus to reside at Hex address 0400—13FF, (the 4K hole above the 1K RAM and below the I/O addresses on the Kim—1.) However, if the 8K memory board is selected for the bottom 8K block (0000—1FFF) only that 4K will be accessible, as the decoding logic will not allow the motherboard bus to be addressed by any address that conflicts with used memory space on the Kim—1.

Memory can be located anywhere between Hex 2000 and FFF7, it stops at FFF7 as the motherboard relocates FFF8—FFFF to 1FF8—1FFF in the Kim—1 ROM, which contains the reset and interrupt vectors.

8K memory board Kim—3B

This board comes assembled and tested, with good documentation which describes how to connect up a single board to the Kim—1 expansion and application connectors, via a cable; it explains how to use the selector switches and supplies a chapter on the theory of operation with a schematic diagram. It also gives memory test programs to run with a TTY or the keypad, in case of suspected faults.

The board has its own 5v regulator and can be run off 8-10 volts dc, there is also a jumper that can bypass the regulator and will allow the memory to be run off an existing 5v supply. It uses 16, 2114 hispeed, static RAM chips and is said to draw 2.1 amps maximum but the one I had drew around 1.5 amps.

The board is fully buffered and will present one buffer load to the bus, thus explaining why only one board can be connected directly to the Kim—1 bus. (Remember that the motherboard can take six.)

It has 5 DIP switches, one of which acts as a memory protect by disallowing a write to the memory, and one of which logically isolates the board (effectively switching it off) but keeping the memory contents intact. The remaining three select which 8K block the memory will occupy out of the possible 64K memory space.

Selecting 0000 as the base is not recommended as this will coincide with the on-board Kim—1 memory, although as mentioned earlier the motherboard logic will deal with this problem.

The switch settings of the address select lines will locate the memory board at the following locations:—

A15	A14	A13	Base addr.	Top addr.	
0	0	0	0000	1FFF	(not recommended)
0	0	1	2000	3FFF	
0	1	0	4000	5FFF	1 = open
0	1	1	6000	7FFF	0 = closed
1	0	0	8000	9FFF	
1	0	1	A000	BFFF	
1	1	0	C000	DFFF	
1	1	1	E000	FFFF	

When wired directly to the Kim—1, the signal BDSEL on pin 16, is connected to the decode enable on the Kim—1, and goes high when the board is addressed, inhibiting the address decoder U4 on the Kim—1 board. This signal is not used when plugged into the motherboard. A problem that may arise, and is not dealt with by the manual, is decoding FFF8—FFFF to 1FFF—1FFF in the Kim ROM for the reset and interrupt vectors, when an 8K board is residing at E000—FFFF; this *is* taken care of by the motherboard.

Summary

The 8K expansion board is very convenient for simple add on memory to the Kim systems, and the motherboard makes this even easier. The motherboard, at £69.95 (ex. VAT), takes care of all addressing and buffering, and provides convenient interconnecting between the various modules available.

The 8K board, at £129.95 (ex. VAT), is expensive compared to some of the S—100 memory boards, but then the time involved in designing an interface for the Kim to S—100 bus, must be taken into account, where the Kim—3B provides all signals necessary (even if the motherboard is not used), and requiring no extra circuitry.

The memory expansion is necessary for any advanced work on the microcomputer, and with say an 8K Basic, (available shortly) or the Kim—5 resident Assembler/Editor (also available shortly), the Kim system can be built into a powerful system comparable to many of the systems currently available. I am told that boards to fit the motherboard include a TV interface card, and prototyping board as well as the memory expansion cards and assembler cards. Using these expansion modules, the Kim can be made into a system comparable to Commodore's Pet, with the advantage of being far more flexible when used to control external devices.

Personally, I have found the Kim a useful tool as well as a development board, uses so far have included a speech analyzer, Neurophysiological-experiment data analyzer and controller, and a fully automatic sheet layout control unit!

PCW *A very helpful book is "Programming the 6502", by Rodnay Zaks, published by Sybex. The basic KIM 1 is £99.95 (ex. VAT).* **PCW.**

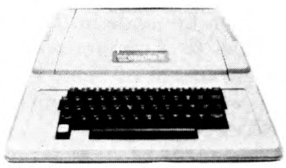
MICRODIGITAL

25 BRUNSWICK STREET
LIVERPOOL L2 0BJ
Tel: 051-236 0707

The Microcomputer only shop providing a complete service from a single chip to a commercial data processing installation. Well worth a visit for a look around and a chat.

OPENING HOURS: *9-5.30 Monday to Saturday. Friendly, expert staff always on hand!*

APPLE II



Still the best machine on the market with elegant hardware design and powerful systems software. Apple is a developed product with unmatched flexibility and versatility. Made to the highest professional standards. Apple brings commercial computer quality at the price of a good hi-fi system.

Apple is a completely assembled and tested computer system. It includes 8K bytes of ROM, rugged plastic moulded case, typewriter-style keyboard, high efficiency switching power supply, two games paddles, line cord, tape recorder connection cables, operators manual, plus two demonstration cassettes.

Usually ex stock.

Apple II 16K bytes RAM	£1063.80
Apple II 32K bytes RAM	£1279.80
Apple II 48K bytes RAM	£1409.40
8 x 16K RAMS 16K bytes	£ 216.00
Applesoft II in ROM	£ 118.80
Vinyl carrying case	£ 27.00
Joystick	£ 21.60
Prototyping board	£ 19.44
Printer controller	£ 118.80
Communications card	£ 118.80
High speed serial card	4 118.80
Voice recognition	£ 178.20
Floppy disk system	£ 159.00
Extra drive	£ 405.00
Diskettes	£ 3.24
Centronics 779 printer	£ 809.92
Tractor Feed on printer	£ 70.20
British ASCII on printer	£ 23.76
AXIOM micro printer	£ 376.92
AXIOM graphics printer	£ 754.92

BOOKS

Apple operations manual	£5.50
Applesoft extended Basic man.	£4.00
Basic tutor manual	£4.00
6502 programming manual	£6.50
6502 hardware manual	£6.50

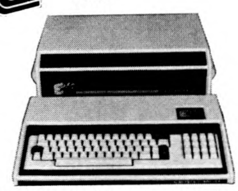
CLEARANCE OFFERS

Whilst stocks last, money refunded if all gone. Chess Challenger level 3 .... **£100.00**

U.K. Monitel. Monitors phone call costs **£20.00**

Computer Battleships. The seek and destroy computer game with live action and sound **£20.00**

Exidy inc. SORCERER



The subject of rave reviews this new computer offers a combination of the best features of each of its competitors. The full size keyboard has a seperate numeric pad and gives upper and lower case characters, graphics symbols and user defined graphics. 8K Microsoft basic comes with the machine as a plug in cartridge with cartridges for other high level language and word processing in the pipeline.

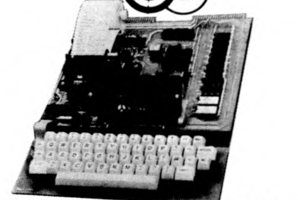
The Sorcerer is ahead of comparable machines when it comes to interfacing with external devices. The following are standard features:— 2 cassette interfaces with motor control and baud rate selectable 300 or 1200. RS232 serial interface, parallel port with handshaking, unmodulated video, modulated video and an S100 expansion connector.

Usually ex stock.

Exidy Sorcerer 16 bytes RAM **£820.80**

Exidy Sorserer 32K bytes RAM **£927.72**

AIM 65



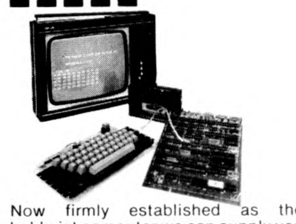
Exceptional value for money the AIM 65 is a fully built and tested single board computer based on the 6502. Standard features include a full keyboard, 20 character alphanumeric display and 20 column printer. Memory consists of an 8K byte monitor, either 1 or 4 K bytes of RAM, optional 8 K byte BASIC and optional 4K byte assembler all on board, with external expansion up to 64 K bytes of memory.

Interfacing is by three 8 bit bidirectional ports (two parallel, one serial), two 16 bit interval timer/event counters, a teletype connector and two cassette interfaces.

Usually ex stock.

AIM — 65 with 1K RAM	£269.46
AIM — 65 with 4K RAM	£340.20
8K Basic in ROM	£75.60
4K Assembler in ROM	£64.26

nm



Now firmly established as the hobbyist computer we can supply you Nascom 1 in kit form ex stock or built by our engineering staff to your specification.

The Nascom 1 is the best possible introduction to the world of personal computing, yet it has the power and flexibility to be expanded into a full data processing system.

The specification includes powerful Z80 processor, parallel 1/0 controller with two 8 bit ports. UART driving cassette interface or most serial peripherals, video output to plug in the ariel socket of your T.V., 2K bytes of RAM (1K user and 1K video), proven 1K byte monitor program in EPROM and a spare EPROM socket.

The kit is complete, all that is required is a power supply a domestic T.V. and a domestic cassette recorder.

Nascom 1 kit	£178.20
3A PSU kit	£26.46
8A PSU kit	£64.80
Buffer board kit	£35.10
Mother board	£10.26
Mini Mother board	£3.13
Vero frame	£31.86
8K RAM kit	£91.80
16K RAM kit	£151.20
32K RAM kit	£216.00
8 x 4116	£75.60
2708	£11.34
I/O board	£37.80
Tiny basic	£27.00
Super Tiny basic	£37.80
8K Basic ROM	£43.20
Eprom board	£43.20
Graphics board	£102.60
Nasbug T4	£27.00
Zeap (tape)	£32.40
Zeap (eprom)	£48.60
Letter Editor	£75.60

BOOKS

Nascom hardware manual	£1.50
Nascom Software manual	£1.50
Seminar notes	£1.50
Z80 Programming manual	£4.50
Z80 Microcomputer handbook	£7.95
Z80 Programming for logic design	£5.95
Practical microcomputer programming for the Z80	£23.96

SERVICE

Kits built, tested, burnt in and guaranteed	£54.00
Standard repair charge	£27.00

BITS AND PIECES

UHF modulator with full instructions	£2.70
Keyboard bleeper kit	£3.00
Keyboard ribbon cable, 8", with plugs	£4.00

CASSETTES

Despite a 12½% increase in manufacturing costs we are able to continue offering our special own brand data cassettes at the old price because of buying in enormous quantities. Tried and tested by thousands of users including education and government departments. We believe our cassettes to be the best for the job.

Box of 10 C15 Quality Cassettes **£4.75**

100 C15 Quality Cassettes **£40.00**

We also sell science of Cambridge MK14, Acorn, 77/68, KIM, PET and North Star HORIZON to make the widest range of microcomputers available in the U.K. There are working demonstrators of most of these in our Liverpool shop and we can arrange hire of machines so you can try before you buy!!

Newbear

Now we are stockists of the full range of bearbags. The two best sellers are:—

Bearbag 4. Vero racking system	£29.92
Bearbag 18. Cottis Blandford cassette interface	£18.63

VDU — model 700 upper case ASCII 64ch x 16 lines scrolling. Full cursor control RS 232C/V24 110 Baud, 300 to 1200 Baud 12 inch display separate keyboard. Full or Half Duplex Quality Professional keyboard (Hi-Tek mechanism.)
British designed and built **£332.92**

CHIP SHOP

A selection from our range of semiconductor devices:

Z80 CPU	£16.20
SC/MP II CPU	£9.72
8080 CPU	£9.72
6800 CPU	£9.27
6502 CPU	£16.12
6802 CPU	£10.80
6850 ACIA	£7.78
6402 UART	£5.94
3881 PIO	£8.64
3882 CTC	8.64
5204 UVEPROM	£7.56
2708 UVE PROM	£10.80
8154 RAM I/O	£8.82
2111 RAM	£3.19
1103 DRAM	£1.19
4116 DRAM	£13.50
96364 VDU	£12.69
6820 PIO	£4.59
6821 PIO	£4.59

TIL 311 dot format hexadecimal display, fits 14 pin DIL socket, incorporates TTL compatible four bit latch, decoder and display driver **£6.75**

ALL PRICES INCLUDE VAT & CARRIAGE

BOOKWORM

We sell the widest range of Microcomputer books in the country, backed up by a return of post service.
We accept Access and Barclaycard and welcome telephone orders.

Microcomputer Programming:
6502 Zaks — £7.95
A superb introduction to the techniques and tricks of programming the 6502 micro, used in the APPLE, PET, ACORN, SYM, etc. Completely self-contained, it can be used by the novice to learn programming, or anyone who wants to make more effective use of the 6502.

Basic, Basic — J. Coan — £6.50 and Advanced Basic — J. Coan — £6.50
These to books give you the complete picture of the Basic language. Both texts begin with short, complete programs and progress to more sophisticated problems.

Basic with Business Applications — Lott — £8.40.
An excellent introduction to Basic for the small businessman.

The First Book of Kim-Butterfield, etc. ad — £7.16
Games system programs, how to expand you Kim — essential reading for the KIM freak.

Programming Proverbs Ledgard — £5.56
An excellent little book. The principles of good programming with numerous examples to improve programming style and proficiency.

The Design of Well-Structured and Correct Programs — Alogic and Arbib — £10.24.
This text synthesizes ten years of research in top down program design and verification of program correctness. It shows how these techniques may be used in day to day programming with the Pascal language.

57 Practical Programs and games in Basic-Tracton — £6.36
Programs for your APPLE, SORCERER or PET, including an excellent Star Wars game.

Computer Lib — Nelson — £5.95
The classic work on all computers big or small, it brings the reader through the past ten and into the next ten years of computer science.

Chess Skill and Machine — Frey — £11.84
The classic work on computer chess. Includes the intimate details of one of the Worlds strongest programs — Chess 4.7.

How to Build a Computer Controlled Robot — J. Loofborrow — £3.36
This book details the step-by-step directions for building a computer controlled robot, named "Mike", controlled by a Kim-1 microprocessor, photographs, diagrams and tables help to direct you in the construction. "Mike" moves under his own control avoiding all objects placed in his way . . . stops, starts, and changes direction on voice commands.

Scientific and Engineering Problem Solving with the Computer — Bennett — £15.96
A superb book — the exercises run the gamut from random processes to the dynamics of motion, from entopy in language to the Watergate problem, you'll discover BASIC applications in lasers, and in the Fourier Series, and the law (!)

Star Ship Simulation — Gorret — £5.10
Everything you need to program the ultimate star game complete with a control console connection to your computer.

The First West Coast Computer Faire—£9.56
A mine of hard to find information on such diverse subjects as computer music, bit-slice systems, to implementing high level languages.

Practical Microcomputer Programming: the Z80 — Weller — £23.40
A much needed text on the techniques and trick of programming the Z80. Includes complete listings of a powerful Editor/Assembler and debugger.

A Guide to SC/MP Programming — Drury — £4.00
Essential reading for the MK.14 owner! Deals with programming the SC/MP at a level anyone can understand.

Microcomputer Problems Solving using Pascal — Bowles — £7.84
Pascal is rapidly becoming the language of the future! The author pioneered its implementation on a Microcomputer and as such is uniquely qualified to write this text.
A superb book!

Pascal user Manual and Report Wirth — £5.52
The "official" reference manual on Pascal.

APL—AN Interactive Approach — Gilman — £9.50
An industry text on this powerful, interactive language.

Sargon— Z80 Chess, Program — D and K Spracklen– £11.96
An excellent, well documented 8K chess program, it won the 1st microcomputer chess championship with 5/5. The book contains a source listing and quite incredible detailed annotation.

Essential reading for anyone planning to write their own program.

Computer Data Directory — £3.98
Hard to find information on the Stateside scene.

SMALL COMPUER SYSTEMS HANDBOOK — 50L LIBES — £7.16
SOL LIBES
A practical introduction to the personal computer by one of Americas leading authors.

Basic and the Personal Computer Dwyer — £10.36
An excellent introduction to the programming language Basic as it applies to the personal user.

Basic Computer Games — AHL £5.50
102 Games for your APPLE, SORCERER or PET complete with listing and sample runs.

Game Playing with Basic D. Spencer — £5.56
In game playing with Basic, the author writes in a non technical style allowing almost everyone to understand computerised game playing.
He includes the rules of each game, how each game works, illustrative flowcharts, and diagrams, and the output produced by each program.

Game Playing with Computers D. Spencer — £13.56
Sharpen your programming skills with this collection of over 70 games, puzzles and mathematical recreations. The book includes over 25 game playing programmes.

The cheap Video cookbook D. Lancaster — £5.10
Techniques of cheap video with complete hardware designs and debugged software for a 7 chip V.D.U.

T15 Workbooks 1-5 for PET.
The information (Commodore) does not tell you. Absolutely essential for the PET owner.

1. Getting started with your PET	£3.00
2. PET strings and arrays	£3.00
3. PET graphics	£3.00
4. PET cassette	£3.00
5. PET miscellaneous	£3.00
The best of Byte Vol. 1	**£8.95**

Reprints from the first year of Byte magazine — hardware reviews, hardware designs, and 'how to do it', software.

What to do after you hit return — £7.00
PCC's first book of computer games in BASIC.

The best of Micro Vol. 1 — £6.95
Reprints of the first six issues from MICRO: The 6502 Journal. Hard to find information on KIM—1, PET, APPLE, AIM—65, etc.

Software Tools—Kernighan Plauger—£7.20
This book teaches good programming style using complete programs such as a full text editor, macro processor, text formator, etc. Highly recommended.

Phone in your Access/Barclaycard Number on
051-236-0707
or complete this order form

BARCLAYCARD — Buy it with Access

Prices all include V.A.T. and Carriage.

PLEASE SEND ME:
....
....
....
....
....
....
....
....

COMPLETE AND POST TO THE ADDRESS ABOVE

I ENCLOSE:
CHEQUE/POSTAL ORDER NO...........................
BARCLAYCARD NO...........................
ACCESS CARD NO...........................
NAME..........................
ADDRESS..........................
....
....
....
....

NASCOM 1:

BREAKING THE SOFT BARRIER

Alan Secker

PCW *This article will help you get to grips with your NASCOM 1* PCW

Once my machine was operational, my first desire was to use it. I loaded a few pre-recorded games but remained unimpressed. Because of lack of knowledge of programming there was little I could do.

It seemed clear that an understanding of the monitor programme embodied in the Nasbug would be helpful but the Software Notes and listing proved to have been written for experts rather than novices.

There was however, the little programme enabling the display of the contents of the character generator and this was accompanied by notes illustrating its construction although insufficient to expose the basic manipulation of the programme instructions, enabling the user to secrete a 'cache' of instructions whose purpose and function he understood and upon which he could build as his skills developed.

I decided to go through the notes and programme, analysing each line in detail:

1. The first note of substance on page 80 reads:

   "The obvious write instruction . . . . . is" LD (HL), A".

   Not to me it wasn't. I couldn't make sense of any of it. I looked it up in the Mostek Z80 CPU Manual. On page 26 I spotted HL in brackets. It was under a heading marked "Register Indirect Addressing". The nitty-gritty boiled down to this:

   (i) This type of addressing specifies a 16 bit CPU register pair. In this case it is registers H and L forming HL.

   (ii) When a mnemonic includes the name of a register in brackets, the instruction doesn't so much deal with the register contents themselves but with what they indicate because they contain not data but the address of a memory location.

   (iii) In short, the HL Register contains a 16 bit number. That number is the address of a location and it is the contents of that location that is the subject of the instruction.

To put it another way, the HL register was being used as *a pointer to memory.*

On page 27 I found an explanation of LD (LOAD & EXCHANGE GROUP). LD is the Z80 mnemonic used for all write instructions and from the table titled EIGHT BIT LOAD GROUP on page 29, I worked out that LD (HL), A, means load the contents of A into the memory location pointed to by HL, (HL containing that address.)

Consequently, if A contained the character required on the screen and HL contained the address of anywhere in the visible video RAM, the character would appear on the screen at that location. Great stuff!

The character generator ROM contains 128 characters. The programme specified that only the top half of the screen is to be used, so we have to fill 8 out of the 16 available lines. That means 16 characters per line.

The way the display is organised by the hardware is into 16 lines of 48 characters. If we wish to display 16 characters per line evenly spaced, we have to arrange for two spaces between each character.

Thus from position one on the screen at the extreme left, we must jump three locations to the right for the next character. As HL points to the first memory location, all we have to do is increment it three times and 'Bob's your Uncle!'

However, in the second location we wish to display the second character. Fortunately the machine code for each successive character is the incrementation, by one, of the former value, we can signify 'next character' simply by incrementing the register that holds the code for the first digit, namely register A.

Now the explanation in the Software Notes was clear:

"Thus the core of the programme is:

```
LD (HL), A    Write character on screen
INC A         Code for next character
INC HL
INC HL        Move pointer to next
INC HL        location"
```

So now we understand how to write one line, but how do we get from the end of it to the beginning of the next? The NASCOM notes tell us.

First, it suggests the above 'patels' is a bit of programme to be used for each line — a subroutine. After each occasion when it is executed a test has to be made to establish whether or not the whole line of 16 characters has been displayed.

Suppose we make a test after each character is written to see if a certain something has happened and to act in a certain way if it has. For example, if we load a register with the number sixteen and decrement it after each character is written, it will reach zero once a whole line has been written. The Z80, sophisticated machine that it is, has a double byte instruction in its armoury that fulfills the function neatly.

DJNZ, e

If the multiplicity of registers contained in the Z80 has been puzzling you then you're in for a partial answer. DJNZ, e uses the B register.

The actual instruction is:

(i) Decrements the contents of Register B; and

(ii) If the contents of B are *NOT* zero, jump back to a memory location, displaced from the memory location of the first byte of the instruction by a number contained in the second byte.

The displacement contained in the second byte is a 'signal two's complement.'

In this case we wish to jump back to LD (HL), A which is back five bytes from OCEF (the address of the memory location containing the first byte of DJNZ, e).

However, there is a snag.

When using DJNZ, e the Programme Counter (Register PC) advances two counts prior to the execution of the jump instruction.

This means that to go back 5 bytes, we have to take into account a forward jump of 2 first. We therefore have to cater for a backward jump of 7.

Right. Let's check it:

1 Mnemonic	2 Symbolic Operation	3 Flags	4 OpCode Binary	Hex	5 No. of Bytes	6 No. of Machine Cycles	7 No. of T States	8 Comments
Add HL,SS	HL←HL+SS		00SS 1001		1	3	11	SS Reg 00 BC 01 DE 10 HL 11 SP

Column 1. Gives the mnemonic for the name of the instruction
2. Gives an abbreviation of what takes place
8. Explains the code to be included in the binary version of the opcode, dependent upon the register selected. In this case DE was chosen. The code is 01. The opcode is therefore 0001 1001. The hex is therefore 19.

Figure 1

The binary for decimal 7 is	0000 0111
The complement of the above is	1111 1000
Add 1 for the signed two's complement	1
	1111 1001
The denary (decimal) for 1111 1001	249
The hexadecimal is	F 9
As the first byte of DJNZ, e is	10
the code for the whole instruction is therefore . . .	10F9
which is precisely what Nascom wrote.	

Now why didn't they explain it!

Clearly if DJNZ, e is going to jump while B is decrementing to zero, B ought to have the right starting number loaded into it in the first place, otherwise instead of jumping while the contents of B are not reduced to zero, the programme will carry on at the wrong moment.

What we actually want is for DJNZ, e to allow the programme to continue once 16 characters are written on the screen of the VDU. Thus if we load 16 (10 in hex) into B before we start, this will decrement at each evaluation of DJNZ, e until it reduces to zero.

The instruction is therefore:

LD B, 10 (10 being the hex equivalent of 16 decimal)

Once we've reached the end of a line, B will be zero and DJNZ, e won't jump. The programme will continue to the next instruction.

The next instruction must relate to the need to start a new line. As HL is pointing to the visible RAM locations, we must find a way to increment its value to the start of the next line. The two 8 bit Registers H and L form HL, a 16 bit register. If we could put a number into another pair of 8 bit registers and add it to HL at this point, HL might then point to the new start address.

This is what Nascom have done. They have used registers D and E which had hitherto been unused, as DE.

The gap between the end of one visible line and the start of the next is 16 Bytes. The number 16 (10 in hex) is loaded into DE near the beginning of the programme.

LD DE 0010

The contents of DE are added to HL at the end of each line by the instruction

ADD HL, DE

This instruction, taken from the 16 Bit Arithmetic Group. If you look it up in the CPU Manual, you will find (Fig. 1).

So now we have the machine writing out 16 characters per line, line after line and running like a Swiss watch. However, we only need to write 8 lines. How do we instruct it how and when to stop?

Clearly there are numerous ways (if only we were more experienced) Nascom chose to compare the ASCII code for a character with the character code sitting in register A (the accumulator). The character code Nascom chose for comparison is 80 which is in fact a non-character. It would have been the 129th character but there are only 128.

There are several 'compare' instructions. Nascom chose

CP, s

The s relates to the figure.

CP,s comes from the 8-bit Arithmetic and Logic Group. The code used is a derivation of the two-byte instruction

ADD A, n

The code for the first byte is 11 XXX 110. The XXX being bits 5, 4 and 3 are defined by the appropriate derivation, in this case CP,s.

CP, s requires that 111 be substituted for XXX making the first byte 1111 1110 or FE.

The second byte (n) contains the code for the number being compared. In this case it is 80. The whole opcode is therefore:

FE 80

If the result of the comparison leaves zero in the Accumulator (Register A), then one of the bits in the flag register — the 'zero flag' goes to a one (sometimes described as "goes high" or "the flag is raised") and this event can be detected by an appropriate following instruction

In this case the next instruction is

JRNZ, e

JRNZ, e is a two byte instruction. It causes the following to occur:

(i) The Programme Counter (register PC) increments by 2
(ii) The comparison is made:

a) If the zero flag is raised (viz the condition is *NOT* net), the programme continues; or

b) If the zero flag is 'O' (the condition *IS* met) then the programme counter jumps to a location specified by a combination of two figures.

The two figures consist of the value in the Programme Counter plus a displacement defined by the *second* byte of the instruction.

The displacement is a 'signed' two's complement number in the range

– 126, 129 –

In this case, if the condition is met (that is, the zero flag is not raised), it means the comparison between character 129 (code 80) and the contents of the accumulator do not produce a zero and so there is a relative jump back into the programme to the point where "LINE" begins and another line is written before the test is made again.

Because JRNZ, e moves the PC to OCF5 and 'LINE' starts at OCE8, we have to jump back the difference — 14 steps (in denary that is).

Fourteen in binary is	0000 1110
The complement is	1111 0001
The 'I' for two's complement	1
	1111 0010
Which in hex is	F 2

The second byte of JRNZ, e is therefore F2 being the code for the relative backward jump. The tables give the code for the first as 20 so the whole opcode is 20 F2.

Nascom however, say that the source code that programming step is:

JRNZ – CH (the H merely referring to hex)

Now as I calculate it C is 12 not 14. Is it a mistake or have I missed an important clue?

Once the JRNZ, e condition is *NOT* met, meaning the zero flag indicates a zero in the accumulator, the programme will move to the next instruction.

The programme has fulfilled its object and is ended. We must however instruct the computer as to what to do next. Nascom offers two solutions. One is to finish with a HALT instruction, the other executes a jump to the monitor programme contained in the ROM. The second enables control to be effected from the keyboard without having to RESET THE CPU.

PCW SEMINAR

an introduction to
Personal and Business Computing

PCW in association with **SYBEX** presents **Rodnay Zaks** speaking on the various aspects of small computers: understanding them and using them.

Rodnay Zaks is the author of best sellers "Introduction to Personal and Business Computing", "Microprocessors: from chips to systems".

As a speaker, he is in great demand in the United States and the Continent.

Now **PCW,** in keeping with its high standards, has secured his services as a speaker.

BOOK YOUR TICKETS BY PHONE 01 229 5589

Date: Saturday, May 12th

Venue: The West Centre Hotel, Lillie Road, London SW6

Time: 9.30 a.m. to 5 p.m. (Registration at 9 a.m.)

Personal Computer World aims to give a definitive introduction to, as well as a refresher in, personal and business computing with this Seminar. Few people are better equipped to do this than **Rodnay Zaks**. The opportunity to take advantage of his first appearance in Britain should not be missed!

Price of tickets is £12

Special offer: Readers who purchase tickets by the 30th of April are entitled to a 10% reduction in the price of tickets.

A note to CASBA members: Membership of CASBA entitles you to a 12½% reduction in the price of tickets.

How to obtain your tickets: Simply fill in the coupon below and send it together with a cheque or postal order payable to Intra Press, or enclose payment with a letter.

To: **PCW Seminar,** 62A Westbourne Grove, London W2.

Please send me . . . . tickets at £12 each*. I enclose a cheque/P.O. for £ . . . . made payable to Intra Press

Name ______________________________

Company/Organisation ______________________________

Address ______________________________

______________________________ Tel No.: ______________________________

**Less*

12½% reduction for CASBA members. (Send £10.50 – Save £1.50)

10% reduction for all other tickets bought by the 30th April 1979. (Send £10.80 – Save £1.20)

THE RESOURCEFUL DISASSEMBLER

A Labelling Disassembler

John A. Coll

And now for the good news: Programming in assembler language on M6800 based computers is about to become a lot more fun. How often have you wanted to modify a commercial program to make it do something special? The new Labelling Disassembler from "Source" makes that quite an easy job.

6800 users have had an excellent Text Editor, Assembler and Text Processor available for some time; I refer to the TSC (Technical Systems Consultants) versions, and these have set new standards of quality and reliability. However the available Disassemblers have all had snags.

A couple of years ago Tim Routsis started work on a labelling Disassembler which would take a disk file and transform it into fully labelled source code. A number of people used this version for some time, but it was never sold commercially. One copy found its way to Computer Workshop and, after further modification, they are now selling it. However this version had a number of drawbacks and a couple of months ago Tim (now a director of his own company "Source") set about writing a quite new type of Disassembler. What he has produced is an absolute pleasure to use and enables one to pull apart anyones software in a couple of minutes. If you want to add another feature to BASIC it is simplicity itself to Disassemble it, modify it and reassemble it. If you want to move BASIC to another section of memory you just change one line of code — the ORG statement.

To illustrate the Source Disassembler's power have a look at this piece of code, which was produced directly by the Disassembler. *(next Col)*

```
                NAM     BUILD
        DEOL    EQU     $7082
        WARMS   EQU     $7103
        INBUF   EQU     $7115
        PSTRNG  EQU     $7118
        NXTCHR  EQU     $7121
        FILSPC  EQU     $7127
        SETEXT  EQU     $712D
        RPTERR  EQU     $713C


                ORG     $7600
7600    L9      BRA     L11
7602            FCB     1,0
7604    L10     FCB     0
7605    L11     LDA A   DEOL
7608            STA A   L10
760B            CLR     DEOL
760E            LDX     #FCB
7611            JSR     FILSPC
7614            BCS     L17
7616            LDX     #FCB
7619            LDA A   #1
761B            JSR     SETEXT
761E            LDX     #FCB
7621            LDA A   #2
7623            STA A   X
7625            JSR     FMS
7628            BNE     L14
762A    L12     LDX     #L20
762D            JSR     PSTRNG
7630            JSR     INBUF
7633            JSR     NXTCHR
7636            CMP A   #$23
7638            BEQ     L16
763A    L13     PSH A
763B            LDX     #FCB
763E            JSR     FMS
7641            PUL A
7642            BNE     L15
7644            CMP A   #$0D
7646            BEQ     L12
7648            JSR     NXTCHR
764B            BRA     L13
764D    L14     LDA A   1,X
764F            CMP A   #3
7651            BEQ     L18
7653    L15     JSR     RPTERR
7656    L16     JSR     FMSCLS
7659            LDA A   L10
765C            STA A   DEOL
765F            JMP     WARMS
7662    L17     LDX     #L21
7665            BRA     L19
7667    L18     LDX     #L22
766A    L19     JSR     PSTRNG
766D            BRA     L16
766F    L20     BRA     L23
7671            FCB     $3D,4
7673    L21     FCC     /ILLEGAL FILE NAME/
7684            FCB     4
7685    L22     FCC     /FILE EXISTS/
7690            FCB     4
        L23     EQU     $7691
        FCB     EQU     $7740
        FMSCLS  EQU     $7803
        FMS     EQU     $7806
                END     L9

      DEOL    7082    WARMS   7103
      NXTCHR  7121    FILSPC  7127
      L9      7600    L10     7604
      L13     763A    L14     764D
      L17     7662    L18     7667
      L21     7673    L22     7685

      INBUF   7115    PSTRNG  7118
      SETEXT  712D    RPTERR  713C
      L11     7605    L12     762A
      L15     7653    L16     7656
      L19     766A    L20     766F
      L23     7691    FCB     7740
```

As you will see the output looks as if it has been produced by an assembler! The labels are even in order but best of all the assembler has substituted standard names for locations that it recognises. One can-not overestimate the value of that facility. It quite often happens that once you have Disassembled a program you recognise many of the jump locations; for example $E1D1 means OUTCH to all Motorola users! Source's Disassembler searches for a disk file called EQUATES.DSM and reads that file to try and find the names of as many of the labels as possible. The file itself just contains a list of locations and their names in this format:

```
7127   FILSPC
7082   DEOL
712D   SETEXT
7115   SETEXT
7103   WARMS
E1D1   OUTCH
E1AC   INCH
```

The Disassembler pulls in to memory only those equates that it needs for that particular Disassembly.

Quite often one is provided with a source listing of a program and one wants to produce a source file on disk. Typing in 600 equates (the number in a typical 10k program) is, shall we say, somewhat boring. Source have included another feature which makes this process a lot easier. If you were going to create an EQUATES. DSM file with all 600 entries you would have to type in the 600 four character addresses and spaces, well these 3000 key strokes are avoided by getting the Disassembler to prompt you with each address which is still unresolved after it has checked the standard EQUATES.DSM file. As a result all you have to type in is the name of the label — if you know it.

The output of the Disassembler can be directed to the terminal, or the printer if using the P command, or to a named disk file. The format can include the compression of all tabs to single spaces so as to save memory space; the expansion of all labels with leading zeroes to enable each label to be uniquely referenced by the editor; the printing of the address of each instruction, the printing of the Hex codes of each instruction; the printing of an ordered symbol table; pagination of output and finally the option of keyboard input of label names as outlined above.

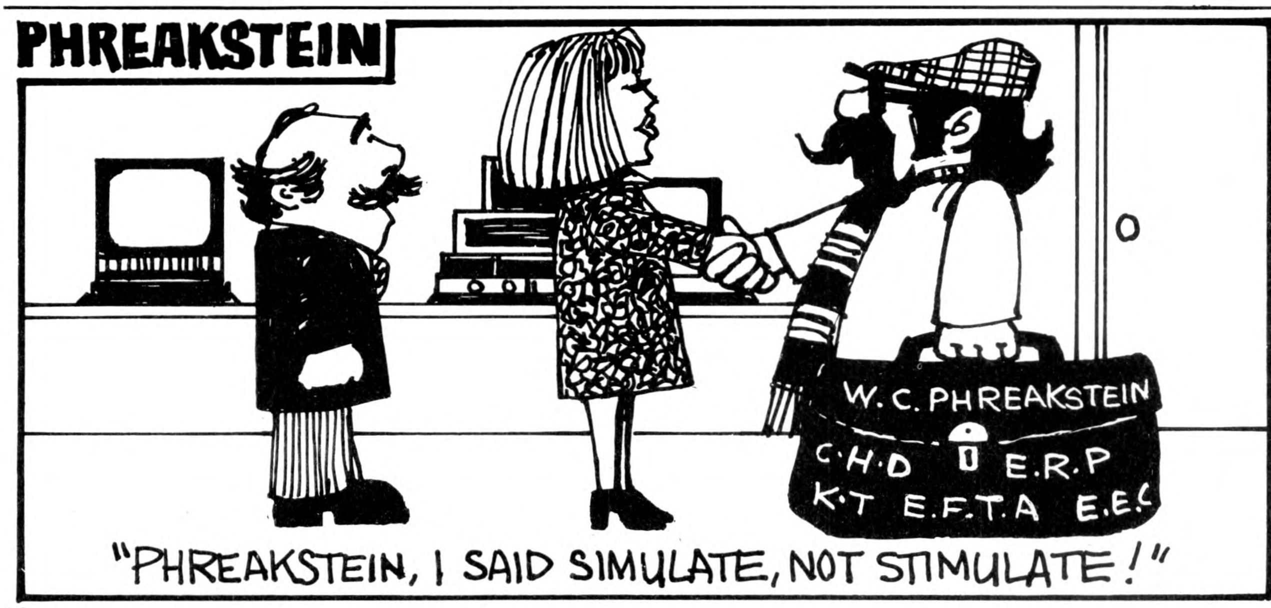
The Disassembler is designed for disk to disk operation, that is, it reads a binary file in (for example BASIC or EDIT) and normally produces a text file on disk. The binary file can be Disassembled between specified address limits if required, and the file can, generally, be immediately re-assembled. The only limitation seems to be that some assemblers get confused if all the labels start with an "L" and are rather short. This poses no problem in practice if the "expanded label" facility is used. It is just so simple and it works. The insides of Altair BASIC are an absolute delight, Microware BASIC is an awful mess in places, but if you want a real horror have a look at some of Motorola's early Editor/Assembler tapes!

This Disassembler is an absolute must for anyone who works in machine code — it really does everything that you could ask of a Disassembler — except make the coffee. The trouble is that there isn't time for coffee: it only takes 4 minutes to pull apart 10k BASIC! Try to do that with any other disassembler and you will find it takes a "little" longer *and* doesn't do nearly such a good job.

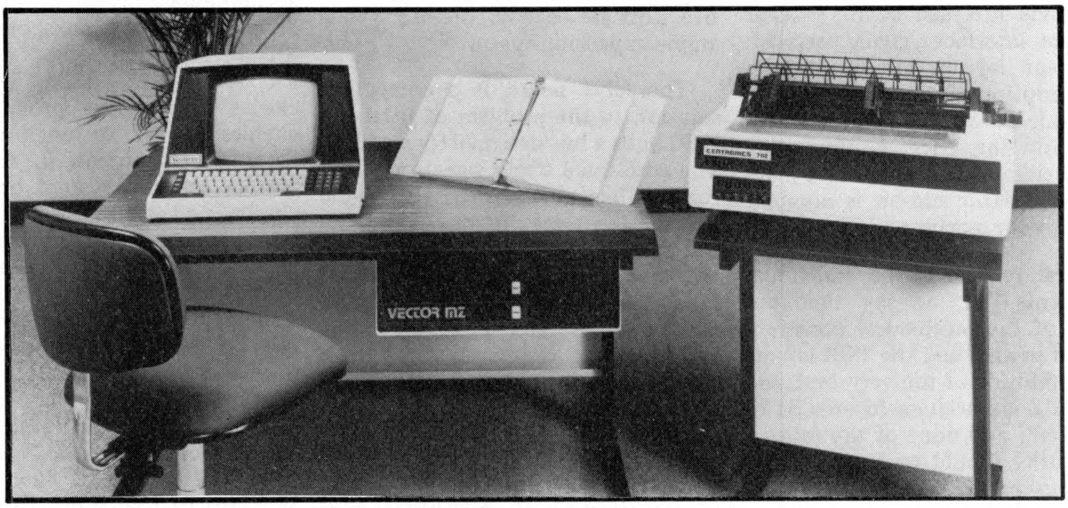
This version of LABDIS is available only from

SOURCE
12 Vivian Road,
Wellingborough,
Northamptonshire

for £45.96 including the disk, VAT, postage and packing, and I don't have shares in the Company (that's a thought!)



Vector Graphics MZ



Sheridan Williams

The Vector Graphics MZ is an American system selling for £2,300 + VAT for the system as tested. The MZ comes complete with built-in twin 5.25 inch floppy discs, and a Z-80 CPU of 32K bytes which runs at 4MHz. Software supplied with the MZ for test was the Micropolis disc operating system (MDOS), BASIC, editor and assembler. Also supplied with the Vector MZ was an ELBIT terminal on loan from ELBIT DATA SYSTEMS of Maidenhead.

Vector Graphics was incorporated in August 1976. The people involved in its conception were Carole Ely, Lore Harp, and Barton Gordon. The company was started with around £3,000, and their objective was to market a memory that Robert Harp had designed. The board was so successful that Harp decided to design some additional products, culminating in the Vector 1 in January 1977. As a company Vector started with 5 employees and a 1200 sq ft premises; now they have expanded to 60 employees and 16,000 sq ft of premises. By January 1978 they were able to offer a fully fledged systems configuration, and they refined their marketing approach to concentrate mainly on the small business and O.E.M. market; for this reason they discontinued selling kits to the hobby market.

STANDARD HARDWARE AND SOFTWARE

Standard hardware and software supplied with the system are:

1. Chassis with power supply and an 18 slot motherboard.
2. 4MHz Z-80 CPU board.
3. Two quad-density Micropolis mini-floppy disc drives.
4. Micropolis disc controller board.
5. Bitstreamer I/O board with one serial and two parallel ports.
6. Flashwriter I/O board (if needed)
7. Two 16K static RAM boards.
8. One 12K PROM/RAM board.
9. Vector Graphics extended systems monitor on PROM.
10. Two copies of the system disc, each contains:—
    - (a) Micropolis disc operating system (MDOS), including Assembler Editor, and other utilities.
    - (b) Micropolis BASIC.
    - (c) Games and video displays.

It is interesting to quote the disc specifications as some of the figures are quite impressive. Apart from the actual parameters of the disc system which are fixed, it is interesting to note their estimate for the Mean failure rate and media life.

Capacity	315K bytes formatted
Transfer rate	250K bits/s
Ave rotational latency	100 ms
Access time (track-to-track)	30 ms
Access time (settling time)	10 ms
Head load time time	75 ms
Drive motor start time	1 s
Rotational speed	300 RPM
Recording density	5248 bits/in
Track density	100 tracks/in
Media life	3×10^6 passes on a single track
Head life	10,000 hours
Mean time between failures	8000 hours

It is also interesting to note that the Micropolis disc drives once engaged are permanently rotating; this has the advantage of making the response and access times quicker when accessing a file; but if the drives are not disengaged manually then the life of the discs themselves will be seriously limited. I remain unconvinced of the need to have the discs constantly rotating. I was, however, very impressed by the sheer capacity of the discs, at 630K for two drives, single sided discs as well, this makes the system a true business system, capable of handling quite sizeable files.

Before mentioning and reviewing the software and hardware individually I shall give below the memory map for the system.

Hex address	
0000	RAM available to the user
006A	Used by the bootstrap loader to load MDOS or BASIC. Afterwards space is used for the system stack
01A0	RES module – printer, console and disc I/O routines and buffers
146B	MDOS or BASIC
2A00 if MDOS 5700 if BASIC	Users programs
C0000	Monitor and console I/O
C400	MZOS PROM
C800	EVIOS PROM
CC00	Unused
D000	Used by flashwriter as a video buffer
D800	Bootstrap ROM
DA00	Micropolis disc controller
E000	RAM used by the systems software
FFFF	

On the hardware side the Vector Graphics is very tidily put together in a robust and attractive cabinet. All you need to make it go is a serial video terminal, and Almarc offer the stylish but large ELBIT for around £500; again, value for money.

The handbook includes a separate user's guide for each board, (except the floppy interface), giving parts list, component layout, circuit diagram and description of operation. This information, as well as being an education for the user, is invaluable when deciding whether another manufacturer's S100 add-on is compatible with your machine.

Before reviewing the individual components, let me say that the quality of the mechanical construction and wiring, and the PCB layout and assembly is of the very best; the Vector MZ is a well engineered S100 bus system, and none of the following remarks should be taken to detract from that.

The chassis is a wrap-around upper and lower steel tray, quite strong enough to support a VDU placed on top, with slightly recessed front and back panels. The fan, fuse, and all connectors are on the rear panel, while the front houses the twin mini-floppy drives and the power and reset buttons; beware of accidentally knocking these buttons or you will abort your program, and possibly corrupt a disc too. Almarc are considering modifying them similarly to the 38OZ key system to prevent an unfortunate accident. Inside a power supply unit occupies the space on the right hand side behind the discs, directly in front of the quiet cooling fan. The massive mains transformer promises plenty of power for expansion. On the left is the S100 motherboard. The card frame runs from front to back, and has a stiffening tie half way. The motherboard is terminated, and the edge connectors have such a good grip that the extractor cams provided on all boards (except the floppy interface) are really needed. Only five slots are used by the MZ, giving ample expansion space.

The CPU board is a competent solution to the problem of fittting a Z-80 onto a bus designed for an 8080. The additional status signals required for the bus are synthesised on the board to give maximum compatability. The Z80, with its 158 instruction set, is operated a full 4MHz clock rate. This can be reduced by a jumper to 2MHz, and a corresponding logic level is output on S100 stack (A Z80 cannot provide stack status).

Wait states to accommodate slow memory or peripherals, can be inserted automatically by a jumper option, or requested via the bus PRDY line; this is a better method than slowing the clock, since all other CPU activities still run at maximum speed. One compromise has been made for compatability which will dismay Z80 assembler programmers; although both the Z80 and 8080 use only an 8-bit peripheral address during I/O instructions, some peripheral boards decode the high order address bits, so the MZ duplicates $A_0 - A_7$ on $A_8 - A_{15}$ This means that the A or B register contents which the Z80 outputs to $A_8 - A_{15}$ are not available on the bus.

The 16K static RAM board is capable of 4MHz without wait states; it is organised as two 8K blocks, and each block has 8K boundary address select and write protect switches. An important memory expansion feature is bank select; up to 8 boards may share the same address space, each set to a different bank number to port 40_H. This board also responds to S100 PHANTOM.

The 12K PROM/RAM board will accommodate 2704s or 2708s. It is organised as two blocks, one of up to 8K PROM, the other up to 4K PROM and 1K RAM. This board carries the monitor program which enables the MZ to function without front panel controls. It has several jumper selected system options, including power-on clear and jump-on-reset. During these functions, other memory boards are disabled by the PHANTOM line. If you use expansion memory which does not recognise PHANTOM, you must not assign it address 0000_H, or both boards will attempt to drive the data bus simultaneously! The BITSTREAMER is the system I/O board. It provides a *serial* port using an 8251 USART, and two parallel ports implemented in TTL. There is nothing inspired in its design.

The two Micropolis mini-floppy drives and their interface board did not have any documentation, but were obviously to the same high standard as the rest of the system. Quadruple density gives the Vector MZ 630K bytes on-line storage — a big advantage for a small machine.

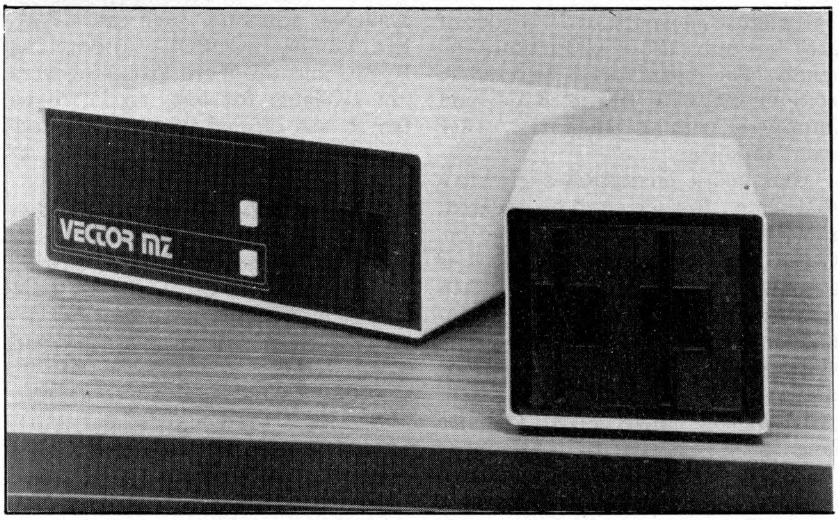
The video terminal by ELBIT, an Israeli subsidiary of Control Data, is a display console with separate keyboard. The circuit is designed almost entirely in TTL, hence it occupies several boards and contributes to the large dimensions of the console unit. Numerous options may be selected by jumpers, which are all described in the Elbit manual. The housings are in the "office status symbol" style to minimise the bulk of the unit. The keyboard has a good feel to it.

Software

The first program that the user comes into contact with is the 'systems monitor'. This is stored in a PROM and is divided into two parts:

1. A group of programs used for display and manipulation of memory data.
2. Code used to control console input and output.

After the 'systems monitor' the next program encountered is the disc operating. Micropolis Disc Operating System (MDOS) consists of an executive program, a group of shared subroutines available to user programs, and an assembly language development package. Applications packages available in MDOS are:—



ASSM a two pass 8080/8085 disc to disc assembly program.
LINEEDIT a line number orientated assembly language text editor with 'character within line' editing, and global 'search and change' facilities.
FILECOPY to copy disc files.
DISKCOPY to copy entire discs.
SYMSAVE creates a source file of symbol equate statements from the symbol table left in memory after an assembly.

Note that MDOS and BASIC overlay each other, but facilities exist to leave one and call the other.

My reference system for this article is a Research Machines 380Z, which although not up to the specification of the MZ holds its own in many respects, except that I have no disc system to compare.

Basic Interpreter

MICROPOLIS BASIC Version 3.0 (1978) is an interpreted BASIC with most of the facilities expected of a recent version. A brief summary of available commands is listed below:

Arrays with up to 4 dimensions.
Integer variable mode.
NUMERIC FUNCTIONS: all usual ones + MAX,MIN,FRAC,FIX,LN,LOG
STRING FUNCTIONS ASC, CHAR $ FMT,INDEX,LEFT$,MID$,RIGHT$, LEN,MAX,MIN,REPEAT$,STR$,VAL VERIFY
SPECIAL FUNCTIONS: IN,PEEK,POKE, PGMSIZE,SPACELEFT
BASIC STATEMENTS: EXEC,FLOW/NO FLOW,MEMEND,SIZES,STRING
DISC COMMANDS: DISPLAY,LOAD, PLOADG,SAVE,SCRATCH,LINK, CHAIN,OPEN,PUT,GET,CLOSE, ATTRS,EOF,FREESPACE,PUTSEEK, GETSEEK,RENAME
FILE FUNCTIONS: ATTR,ERR,ERR$, NAME,RECGET,RECPUT,SIZE, TRACKS,FREETR

I enjoyed programming in BASIC on this system, and found the system quite tolerant. I especially liked the ability to use expressions and variables in GOTO and GOSUB statements. Another statement that I liked was the 'EXEC' function; this allows a string variable to be given a BASIC statement and then executed.

Example
```
10 A$ ="PRINT B"
20 EXEC A$
```

This facility has a use in that a function may be input and then executed, this is only possible by parsing the function onto disc if the 'EXEC' facility doesn't exist. Lines may be up to 250 characters long and the maximum line number is 65529. Some of the other unusual routines are worthy of a mention; SIZES allows you to select the precision of numeric variables up to 60 digits for simple arithmetic, or 20 digits for functions. FMT(X,Y$) is Micropolis' equivalent of PRINT USING, and allows the output to be formatted according to the image of Y$. CHAIN passes variables from the current program segment to the next one loaded from disc.

File handling in this BASIC is exemplary, and I had very little trouble using both serial and direct access files. Files (up to 10 at a time) can be opened simultaneously for both direct and serial access in both read and write modes. File copying using the supplied software is very easy and fast. A BASIC program that I wrote to copy files was rather slow, however. I encountered slight problems with the file pointer in direct access mode, but these were relatively simple to overcome; I suspect that it was my unfamiliarity with both the manual and the machine. One final comment on the discs is that having 630K available on the two drives was amazing. I never managed to fill even one disc in the three weeks that I had the system, and I created a *lot* of programs and data files. I am told that one megabyte will soon be available.

In order to plot graphics on the MZ a rather tedious routine had to be performed. First one had to remove a board from the Elbit terminal (supplied with the machine) and remove a 'jumper' which disenabled screensrolling. MDOS then had to be entered and location 0540 altered from 18 to 00. ASCII character number 24 would then act as a clear screen character. All this could have been achieved with the relevant POKE instructions in BASIC itself. When these two tasks had been performed it only remained to use a statement thus:—

```
PRINT CHAR$(31)+CHAR$(Y)+
      CHAR$(X)+"*"
```

and a graphical output could be obtained. Note that the (0,0) co-ordinate was at the top left corner of the screen. Other snags arose which I will not bore you with; suffice to say that I managed some interesting plots, and the graphics do work.

Micropolis BASIC was not particularly fast — but for those who really want speed, interpreted BASIC is not the appropriate software to use. I am informed that Vector have a compiling version of BASIC available. Here are the benchmark tests for comparison with other machines listed previously in John Coll's articles. (**PCW** Vol 1, No 1 & Vol 1, No 8)

Benchmark Tests	VECTOR GRAPHIC
BM 1	6.2
BM 2	13.5
BM 3	34
BM 4	36
BM 5	39
BM 6	74
BM 7	99
BM 8	70
BM 7+8	169

Function evaluation on the MZ is very slow, I know that the Research Machines 380Z is particularly good in this respect but there are no excuses for such slow and inefficient algorithms.

LINEEDIT

I enjoyed trying out the line-edit package, but would have been much more interested in the WORD PROCESSOR that is available in the USA, but not here yet. The line-editing package has all the usual facilities of deleting and changing all references to a particular string. I started writing this article using line-edit to test its facilities. I found it quite a useful package, but rather limited compared to a word processor; however, line-edit does not profess to being a word processor. I never managed to find the lower case characters that are supposed to be available.

Machine Language

Micropolis disc operating system (MDOS) can be entered by typing

"B" from monitor or by LINK "MDOS" from BASIC. MDOS is as good as most other disc operating systems that I have encountered, and has a good display mode. Monitor commands are the letters A-Z and include facilities such as:—

N-Non-destructive memory test. It can be used at any time as it is non-destructive. Memory locations 0000 onwards are read and stored temporarily. The location is tested to see if 00 and FF can be written and read correctly. This is continued until an error is detected.

F- Finds all locations with a common address.

M-Move memory block.

L-Load and go.

— and 22 others.

I found them very easy and useful on the limited machine language programming that I did.

Graphics Board

The Vector Graphics high resolution graphics board excited me to such an extent that I asked for a demonstration of its capabilities. The specification is very good and I wanted to see how it would turn out in practice.

The graphics board works in two modes, digital output with only black or white dots, this gave 256 x 240 picture elements; or 16 shades of grey but only 128 x 120 picture elements. The board must be used in conjunction with 8K of RAM, and interfaces with a standard "raster scan" monitor.

The board incorporates circuitry that allows the screen to be updated; since the memory is multiplexed between bus and graphics board, it is still available for general use when not using the graphics facility. Included with the board is a software alpha-numeric character generator program, and x-y plotting program providing a high level graphics language.

The demonstration given did not do the graphics board justice, and I would like to have had longer to get to know it. Detail obtainable in either mode was remarkable; in fact it was not possible to see the dots from further than 1 metre.

Conclusion

One gripe about the MZ. Why put the adjustable Baud rate switches inside the cabinet? I was constantly taking off the lid to change the rates. To be fair, in a business environment one is very unlikely to be using the system in different configurations. It is a pity also that some of the other available software such as CP/M, FORTRAN, COBOL, Compiling BASIC and the Word Processor were not available for test. Again, to be fair, I was offered CP/M to evaluate but declined because of shortage of time.

I had a demonstration of another system marketed by ALMARC. This was the Educational Timesharing System, which will handle up to 4 users simultaneously on *any* I/O devices (even ones with different Baud rates). I was very impressed with its versatility. There are advantages to both single user micros and time-sharing micros, the advantage of a multi-user is the ability to reference common files. I would suggest that educational establishments ask for a demonstration of this single disc, 4 user system. Its price is around £2000.

Almarc Data Systems are a company whose joint directors seem accommodating, friendly, and knowledgeable. I think that they have chosen a good system in the MZ, and feel confident that they will do Vector Graphics justice here in Britain.

Finally, may I say that I am indebted to L. S. Warner who greatly assisted me with his in-depth hardware knowledge.

ARE SMALL BUSINESSES READY TO BUY ?

D. R. Worsley
Wye College, School of Rural Economics and Related Studies, University of London

With the recent breakthrough in technology, computers are now very small and relatively speaking very cheap. For the price of a car, a small businessman can purchase for himself a micro computer capable of handling all his paperwork, leaving him to get on with running his business. Although the micro computers and their manufacturers may not yet be ready to sell the benefits of computer operation to the average sceptical businessman, it is only a matter of time before both attractively designed equipment and simple, easy to understand and operate application programs are available. However, there remains the question 'are small businesses ready to buy computers?'

The answer almost certainly is 'no'. Although at the Personal Computer World Exhibition a number of businessmen bought computers, these men were undoubtedly enthusiastic about owning a computer and were probably pleased that they had a business excuse for buying one. Only the already converted purchasers could have bridged the gap between the ordinary business world and the computer world with its excessive jargon in evidence there. The visitors to that exhibition were not typical of ordinary businessmen.

There are three apparently unconnected clues to the answer to this question.

1. *Primary versus secondary activities*

Small businesses are set up by individuals who have a particular skill which they enjoy and which they believe they can exploit:— the car repairer starts a garage; a pilot starts an air-spraying company; the insurance salesman sets up as a broker. Each of these businesses has a primary activity which is the one that the proprietor enjoys and is skilled in.

However, he soon learns there are many other activities that he has to perform that are usually much less enjoyable and often very hard to learn. The difficulty of learning tasks that are often alien to the proprietor, together with a lack of enthusiasm combine to leave these tasks unlearned or misunderstood and therefore undone, leaving a continuing feeling of dissatisfaction, irritation or even guilt. Only those secondary activities that have a powerful bearing upon the primary activity are performed, often with the aid of some patented system that helps the task to be done, but without any understanding and with the minimum of effort. Business readers will be familiar with special cashbooks, wages, stationary, etc., and there is even a system which is supposed to avoid trouble with The Industrial Tribunal if followed faithfully.

2. *'Small' Businesses?*

One specific identifying feature of small businesses is that their size does not warrant systems in the sense of bureaucratic procedures such as stock indenting, cheque requisition forms and the like. The importance of this feature arises not only from the absence of systems, but from the virtually inevitable lack of thought given to the development of such systems.

Experienced readers will remember, perhaps with embarrassment, their own early attempts at systems analysis and will surely recognise the great divide that separates those of us with successful systems development experience from the inexperienced. Such development experience applies equally to the development of manual systems and it therefore follows that many proprietors of small businesses have little concept of the rigours of any kind of systems development. Indeed, developing systems is probably one of those tasks referred to in 1. above.

3. *First-time users*

Anyone who has been associated with the installation of a computer system to first-time users, will know the problems associated with misunderstandings, changing needs, lack of co-operation, and impatience or lethargy. These problems can be summed into the two phrases; "lack of discipline", and "lack of well defined needs". It is seldom that a first-time user knows the necessary data concerning the number of customers, etc., and even though his office may be organised with some set routines and systems, these are often different to the systems the proprietor thinks are in operation, and always have a surprising number of exceptions.

Until now first-time users have been businesses that could afford computer systems costing many thousands of pounds. They have therefore been large enough to warrant office routines and the like with consequent experience of their development. Yet they still present the analyst or consultant with a continual stream of problems as they learn to accommodate the computer in their midst.

Now that micro computers are available to smaller businesses, first-time users will now include ill-informed proprietors with little or no staff having experience of the secondary administrative tasks, having no systems, but possessing a desire to see the end of these tasks, and a small amount of money. Before they can buy and operate a micro computer system to the satisfaction both of themselves and the suppliers, they will have a great deal to learn.

There is a fourth clue pointing to a 'no' answer to this question. 'Necessity is the mother of invention' is a phrase which applies to small businessmen. Resigned to the knowledge that they could never afford the resources that large businesses have to make analyses of sales, margins, stock trends etc., or even to keep adequate stock records, they improvise, making speedy changes to jobs, hours of work etc. etc., to accommodate changes as they happen rather than trying to anticipate them through analysis. Indeed their flexibility is one of their greatest strengths. Lacking formal training they gain experience over the years of those elements of their business that must be watched carefully and dismiss all others. Whilst the advent of micro computer power that is now within their financial reach opens the door to these analyses, few may be in a position to take advantage of the better information:

(i) because they have not had them, they would not know what to do with them

(ii) the imposition of a computer system could deny them the flexibility that is one of their assets.

The terse conclusion of all this is that the small businessman, who through *no effort of his own* is now in a position to purchase computer power, is singularly ill equipped to gain any benefit from its operation. Dogged by increasing government bureaucracy, discrimination through prices and credit terms, with desperately scarce resources, and with hearty mistrust of computers built up over the years through the experiences of himself or others, most small businessmen just don't want to buy a computer. What they may be persuaded to buy is an *end to administrative drudgery;* or, *the benefits of being better informed;* or, *the kudos that comes from operating the very latest equipment.*

But before this can happen, the onus is fairly and squarely on the shoulders of the micro computer manufacturers and distributors to develop suitable systems for even the dullest to understand, and then to engage in an aggressive marketing policy to show that the benefits of computer operation can far outweigh the costs and penalties.

To undertake this role, there is a need for a thorough understanding of what small businesses are and their variety, why they are still small, their needs both overt and covert, and particularly their limitations.

Summary

1. Although some micros have apparently been bought by small businesses, the purchasers were most likely to have been enthusiastic enough about computers to make the time, money and energy available both to program the machine and fit the result into their businesses.

2. The vast majority of small businessmen will be particularly poor first-time users:

   a) the machine will be expected to deal with secondary activities, about which they have little knowledge and even less enthusiasm
   b) they have virtually no experience of systems development — even manual systems.

3. Lacking the forecasting capacity provided by analyses, they learn to react quickly to events — which they can through the absence of systems and other bureaucratic overheads — and thus not only avoid penalty from the absence of these analyses, but develop an asset, namely flexibility.

4. The installation of a computer based administrative system can tend to place a real or imagined burden upon the business through its restrictions, thus reducing the capacity of the business to respond to new circumstances.

5. Although good marketing may place the micro so that it is 'visible' to the businessman's eye and "mind's eye", he may have them shut for any number of reasons, not least of which is the poor press computers get generally.

Finally, there has always been a gulf between the "computer-people" and others who work for an organisation. The hours of work, dress, salaries and jargon have all helped to maintain this. Improvements in business education for the young together with the introduction of computers to secondary schools are preparing a generation that is capable of the analyses referred to earlier, but more importantly, one incapable of preparing these analyses without computers. With twenty or thirty years to wait before they are in positions to specify company policies, this generation will suffer more or less frustration and difficulty during these years and the words "generation gap" are likely to take on a new and dangerous meaning in our businesses.

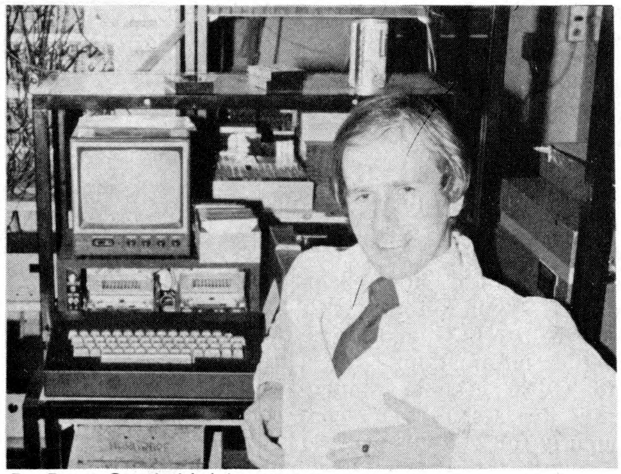
Reducing this generation gap is probably one of the greatest challenges of educationalists. I could see this gap developing at the Exhibition. I had just come from a conversation with a bank official who had demonstrated a planning program (via a terminal) to farmers at agricultural shows up and down the country this summer. He despaired not only because farmers were stunned by the program's speed and presumed accuracy, but both they and one or two of his colleagues believed everything it printed as though the product of some higher and omniscient intelligence.

Perhaps readers who are in the education field would care to comment on this question.

PCW Interview

Dr. Quy and his Research Machines Computer

David Bannister



Dr. Roger Quy in his laboratory

Doctor Roger Quy (pronounced Kwy) is a Research Fellow of the National Hospital's Institute of Neurology. He studied Psychology and Law at Keele University and is currently investigating methods of continously monitoring the brain and heart activity of patients with certain types of epilepsy. Since this produces a large amount of data for analysis, Doctor Quy decided to use a Research Machines 380Z Microcomputer, interfaced to a Datac printer. PCW spoke to the Doctor about his system:

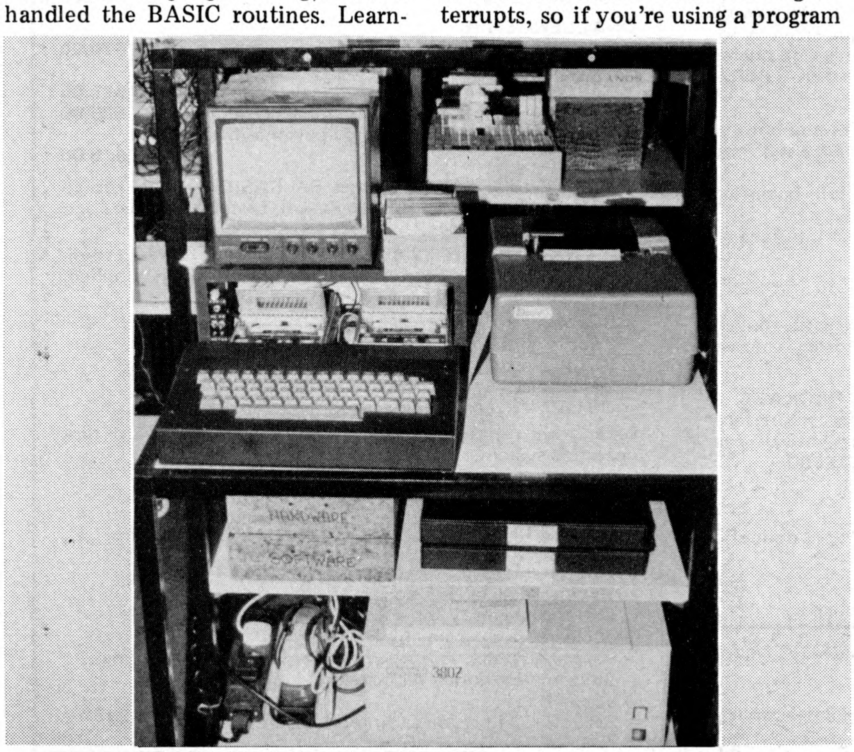
"The traditional method of analysis (i.e. Electrocardiogram or Electroencephalogram recordings) is to simply 'eyeball' the traces. But for twenty-four hour recordings this is tedious, and thoroughly boring, so we decided to get the computer."

Why the 380Z? "Mainly because David Small, who works here in the hospital happens also to be a software consultant to RML. But the machine itself — as the name implies — seemed more geared to being a research tool than something like, say, the PET".

Doctor Quy had studied FORTRAN when at college but BASIC and machine code were new to him. "At the beginning David Small did the machine code programming, while I handled the BASIC routines. Learning BASIC wasn't difficult: learning about the actual system and its control options was more of a problem.

"Machine code programming I found fairly easy to learn. Finding a decent text-book on machine code wasn't easy, however. I used the Adam Osborne "Z80 programming for logic design", but even that only restates the Z80 op-codes. The RML manuals are rather closely written and badly need an index, because you have to keep flicking backwards and forwards, trying to find things. One useful source of multiply and divide routines, I found, was PCW.

"One of the current problems with RML's option ROM is that the fast scroller insists on re-enabling interrupts, so if you're using a program which calls up interrupts it's impossible to disable them using BASIC, which means that if you don't reset, the interrupts are still ticking over when you load other programs, which causes lots of headaches.



The System

"Apparently this problem will go away when we get a disc-operating system, so we're learning to live with it. The whole kit is supposed to be going back for upgrading, so one's rather reluctant to change anything.

"One of my grumbles with RML is that they tend to promise things and keep you waiting for them. We've had this disc system on order for over a year, and the PIO took a similar time, which meant that we couldn't run, for example, our heart-rate analysis program during that period.

"The system itself works well. We had some initial trouble with mains transients, which it was very sensitive to. In a hospital there is a lot of heavy electrical equipment — electron microscopes, x-ray machines, and so forth — which cause huge transients. In fact, the solenoids on our old printer were blown by mains transients.

"I find the graphics limited, particularly when drawing graphs. It's very difficult to line up characters with axes.

"The front panel mode of the 380Z is good, it's very easy to change the contents of registers and addresses, which makes it very nice to use for modifying programs. The assembler is also nice, though I've not had experience of any others to compare it with, but it's nice to work with.

"Writing machine code to be patched in with BASIC is incredibly tedious when using cassettes, because of the number of operations. But this is something else which will disappear when we get the disc system.

"RML, though, were *very* helpful generally. So too were Datac, who supplied us with information on how to convert their printer for graphics. We chose the Datac, incidentally, because it was the only cheap matrix printer with software programmable step size."

One of the first uses the computer was put to was in the analysis of ECG recordings for abnormalities in heart rate.

Twenty-four hour ECG recordings are made from patients wearing a miniature cassette recorder — the 'Medilog', developed by Oxford Medical Systems Ltd. Four channels of data are stored on this; two EEG channels, one ECG, and a time track or other information. Using a standard C-120 cassette running very slowly this system can record continuously for 24-hours without interfering with the patients' freedom of movement unduly.

The cassette is replayed at twenty-five times the original speed and analysed by an analogue detection unit. Each ECG recording is converted into a series of pulses by a schmitt trigger unit and fed to the 380Z. The interval between each beat is then measured by a machine code routine and returned to the main program which is written in extended BASIC. The intervals are sorted for histogramming and a scan is made for abnormalities in the heart rate (arrythmia).

The VDU displays seventy successive intervals, calibrated in beats per minute.

A histogram of the heart rate can be displayed on command, together with the mean and standard deviation. It is also possible to obtain a print-out of the times of occurrence of four different types of abnormality; tachycardia (abnormally fast), bradycardia (abnormally slow), dropped beats, and premature beats.

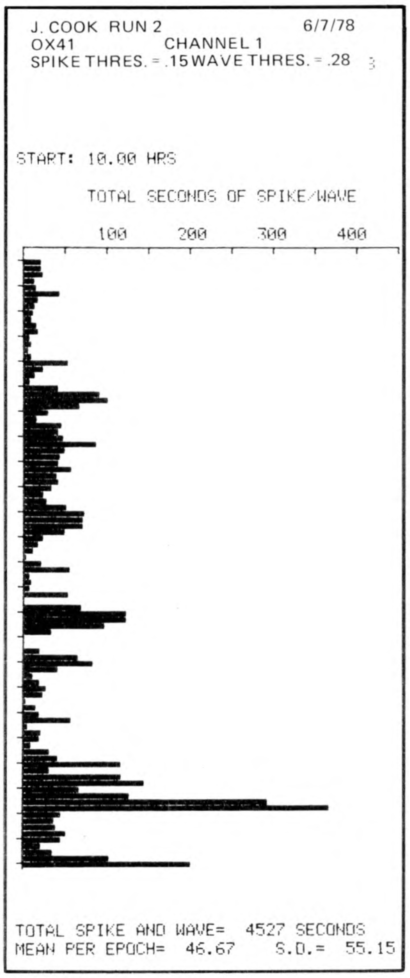


Figure 1

By using cassettes a permanent record of each patient is obtained.

"We were particularly interested in epileptics suffering from absence seizure, where the person will simply go blank for up to thirty seconds. This is normally very difficult to detect. The only satisfactory method is to monitor the brain activity for the wave form characteristic of this seizure, known as a spike and wave pattern. Using the 24 hour recording method and the computer we can obtain accurate analysis.

"The spike and wave is converted to a TTL pulse by the analogue and again fed to the computer. A graph (see Figure 1) is obtained showing the number of seizures in fifteen minute intervals, with the mean and standard deviation."

Doctor Quy sees this as only the beginning of the use of micro's in his work. "What we hope to do now is to use the Z80 as a micro-processor proper, as a dedicated, hard-wired unit. One for spike-and-wave analysis, and another for ECG analysis. That's the goal."

nputing Store

77-68

K3 compatible.
+8v 850mA
+16v 55mA
—16v 150mA
£18.75
£108.75

e or double sided

j power down.
'en K2 operating

£26.25
£131.25

stem

SGEN).

sockets.
ets.
ail.
ulators: 1. off 5v
2. off 5v or 12v

boards.

'ith full

£1.00
£1.50
£0.25
£0.75
£1.00
£1.00
£5.00

★

ΞROR

Cartridges
ion Unit
face

y graphics

★

'ARDS

e touch keyboards

*BRITISH DESIGN

£43.50

★

ıl offer

t only **£48.00** (cash

rd **£27.50**

TRS 80 16K upgrade kit **£95.00**
77-68 — The best supported hobbyist 6800 system in the U.K.! Available in 'Bearbags' or as individual components.
***77-68** Manual gives full assembly and test details for the CPU, 4K TAM and MON 1 boards. Price includes 1 year's free membership of the 77-68 user group.
£7.50

*Expandable — new boards are coming all the time

*Fully supported — **77-68** user group has over 400 members

*British design

BEAR BAG NO. 1 77-68 P.C.B., COMPONENTS & EDGE CONNECTOR

This contains all the parts which fit on this low cost start to a large 6800 microcomputer system. (Ref. Personal Computer World Vol. 1 No. 1 page.) To use this computer the parts in Bearbag No. 2 and 3 are needed if the constructor does not already possess them. For a full description of this system the design manual is necessary and recommended because the manual also includes membership to the invaluable 77-68 user group.
£45.00 + 8% VAT

BEAR BAG NO. 2 77-68 LED'S AND SWITCHES

The 77-68 CPU board has a toggle switch and LED interface and enough memory on board to run programs. This kit contains high luminance 0.2" LED'S and high reliability gold plated subminiature toggle switches suitable for low voltage low current contacts, to match Bearbag No. 1.
£14.95 + 8% VAT

BEAR BAG NO. 3 77-68 POWER SUPPLY*

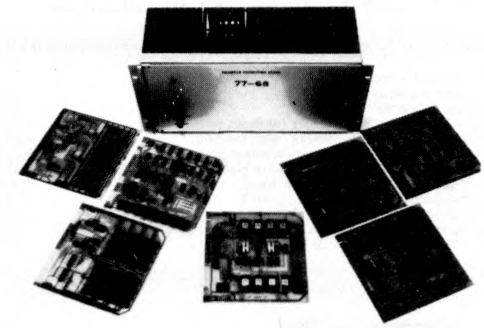
77-68 CPU needs a 5V stabilized 1 amp power supply. The supply in this bag is conservatively rated with a 5V 3 amp regulator which can adequately support two or three 77-68 bards (depending on the use of standard or low power memory I.C.'s).
£17.95 + 8% VAT

BEAR BAG NO. 4 77-68 u RACK AND BACKPLANE*

The major cost of a large computer is not the m.p.u. IC, the card frame or power supply but it is the memory, which can amount to several hundred pounds. This card frame is not the cheapest way of mounting 77-68 p.c.b.'s but it is the recommended one. (The lowest cost alternative is to use a wooden box and plastic double glazing channel for guides).
£27.70 + 8% VAT

BEAR BAG NO. 5 77-68 4K RAM P.C.B. AND COMPONENTS

This is the vital element for any computer. This design is fully buffered and address decoded. The 4K block can be positioned anywhere within the 64K memory allowable.
£75.00 + 8% VAT



BEAR BAG NO. 6 77-68 Mon 1 P.C.B. AND COMPONENTS

It is impossible to run a 'Basic' interpreter on a computer with only LED's as a read out! Sooner or later a V.D.U. or teletypewriter is essential. This also implies that a small operating system of "Monitor" is within the computer to control the V.D.U. interface. This board can support two V24/RS232C interfaces (or 20 MA current loop (only one supplied)) and a method of "bootstrapping" a kilo byte of operating system into protected memory. This is a very elegant and low cost way of maintaining a flexible operating system that can easily be modified for the users own set of peripherals. A listing of an operating system is supplied and Kansas City cassette or Paper tape is available. Beware to load the operating system a Kansas City cassette interface is required (see Bear Bag No. 10) or a terminal with a paper tape reader.
£50.70 + 8% VAT

BEAR BAG NO. 12 77-68 V.D.U. KIT

This is a Direct memory accessed type of Visual Display Unit which gives a very fast writing speed. Its format is 40ch by 24 lines e.g. Teletext. Its output is composite video and it also has a keyboard interface which expects parallel 7 bit ASCII and strobe.
£69.50 + 8% VAT

BEAR BAG NO.13 77-68 Mon 2 KIT

A convenient way of starting up a microcomputer is to have an operating system already in ROM. This board can support "MIKBUG" (with a hardware single step), "SWATBUG" of a 2708 Eprom. The V24/RS232C ports can be via one of the two 6821 PIA's or the ACIA port. The kit is normally sold without this ROM so the user has a choice of operating systems.
£64.10

BEAR BAG NO. 16 77-68 EPROM BOARD (ROMA)

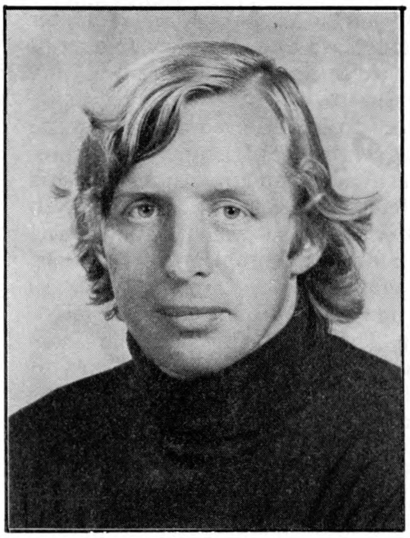
This board holds up to 8K bytes of 2708 Eproms or 16K bytes of intel 2716 which are fully address relocatable in 4K byte segment. This bag does not contain the EPROMS. Please note that Bear Bag No. 14 does a fine job programming 2708's.
£29.50 + 8% VAT

BEAR BAG NO. 17 77-68 INTERFACE BOARD (PIO)

This is a utility board which helps make 77-68 the most complete hobbyest system available, supplying a 6840 Timer and enough 6821 I/O ports to keep your peripherals busy.
£45.00

★★★

WIDE RANGE OF 6800 SOFTWARE ON CASSETTE



THE LEARN MACHINE

Richard Hooper

Richard Hooper is Managing Director of Mills & Allen Communications Ltd., a company specialising in electronic publishing, viewdata, and computer assisted training software. From 1973-1977 he was Director of the Government-sponsored £2.5 million National Development Program in Computer Assisted Learning.

New Idea?

The idea of using the computer to teach people usually arouses surprise, enthusiasm and hostility in roughly equal proportions. It's not a new idea. It was born in the USA in the early 1960s, arising out of work in programmed instruction, and christened CAI — computer assisted instruction. (The hostility towards CAI can on occasion be caused by typographical errors in publications on the subject, which refer to the CIA!)

Programmed instruction, the brainchild of Harvard Professor B. F. Skinner, involves the design of self-instructional teaching materials with three main features:

- the author/instructor/teacher must define the objectives of the lesson material in behavioural terms, i.e. what can the student do to show that he or she has learnt the material.
- the lesson material is divided up into a logical sequence of chunks or 'frames'
- each frame concludes with a question or questions to the student to test mastery, and a response to the student's answer (so-called reinforcement).

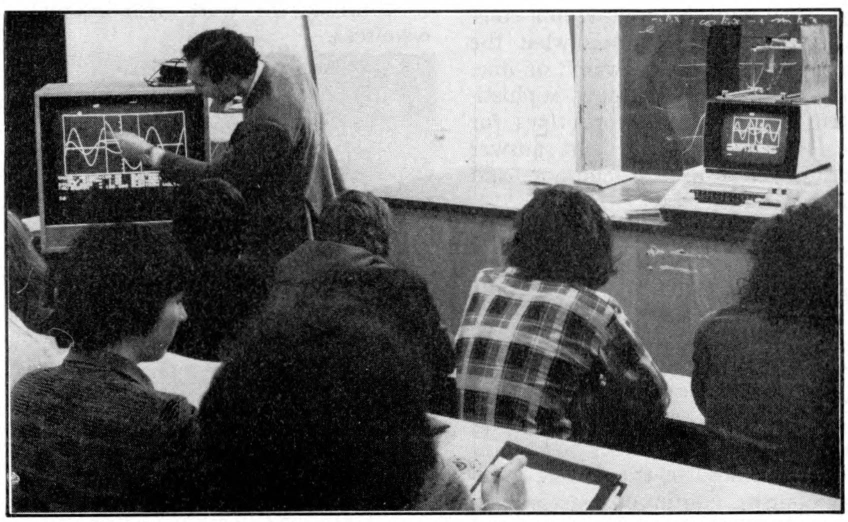
Special 'teaching machines' were invented and aggressively marketed, which stored and presented the programmed instruction lessons to the student. The teaching machines were rather cumbersome and limited in the amount of branching they could permit. Also teaching machines were restricted to multiple choice questions. The electronic digital computer, by contrast, was potentially more flexible, offered infinite branching capability, and could be programmed to 'understand' a wider range of responses from the student at the terminal. In short, the computer would soften the resistance to programmed instruction techniques which was becoming noticeable in both the education and training markets.

By the middle 1960s, two major CAI developments were underway, both of them receiving federal funding. Patrick Suppes' team at Stamford University on the West Coast was developing CAI materials in maths and reading for the elementary school classroom. Don Bitzer's team at the University of Illinois was creating the PLATO system, with the emphasis on advanced graphics and a low-cost terminal based on plasma screen techniques. In 1966 Suppes wrote:

"One can predict that in a few more years millions of school children will have access to what Philip of Macedon's son Alexander enjoyed as a royal prerogative: the personal services of a tutor as well informed and responsive as Aristotle."

But by the end of the 60s the euphoric balloon of CAI enthusiasm was beginning to be pricked by harsher realities. The educational system was not rushing into the 'educational revolution' which Skinner a few years ago had described as 'inevitable'. Computer manufacturers like IBM got into CAI (the IBM 1500 system) and burnt a few fingers and millions of dollars. The claims regularly made by the Illinois team about low costs per student terminal hour were not being realised in practice, partly because the 'economies of scale' argument is very difficult to implement in the educational system. In addition, it was proving more difficult than predicted originally to write good tutorial teaching materials for the computer. The simpler CAI techniques — called somewhat ominously 'drill and practice' — were proving less acceptable to teachers, particularly in Britain, who were moving away from that kind of approach to mathematics.

But the 1970s has seen a gradual resurgence of interest, at a much more realistic level, in computer assisted learning ideas. This has come about for a number of different reasons. In the educational system, particularly in higher and further education, more slowly in secondary ed-



Dr. J. W. Humberston, University College London, using a CAL package to demonstrate the variation in wave function obtained in solving the Schroedinger equation.

ucation, computers have begun to take a significant place — not specifically to do computer assisted learning, but to help with the teaching of computer science and computer appreciation, and to help with the teaching of increasing numbers of subjects which require a computing component. Computers are academically respectable because of their central contribution to research in the quantitatively based disciplines. This is no longer just true of physics and mathematics — computers are now involved in social science research, linguistics, art and design.

The Revival

A further reason for renewed interest concerns technological changes which have taken place in the world of computing at large. The American CAI developments of the 1960s took place on and for large main-frame machines. With the arrival of interactive, multi-access computing, minis and micros, not only have hardware costs declined,. but a psychological barrier to acceptance has been largely removed. Teachers like to have the educational technology under their control, and tend to dislike large-scale centralised facilities beyond the school walls. Headmasters can get more mileage in the prestige stakes from a small computer in-house, than with a terminal with a telephone line. Educational technologists tend to underestimate the political, human and psychological factors that assert influence upon decisions about innovations.

Improvements in software have also made a contribution. A range of so-called author languages has become available in the 1970s to help teachers and instructors design course material on the computer. These author languages, for example PLATO's TUTOR, IBM's IIS, Sperry Univac's ASET, Hewlett Packard's IDF, Leeds University's LAL, are high level enough to allow non-computing specialists develop what the Americans call 'courseware', on-line. They contain increasingly sophisticated answer-matching routines for the tutorial question and answer sections, and also calculation and simulation facilities.

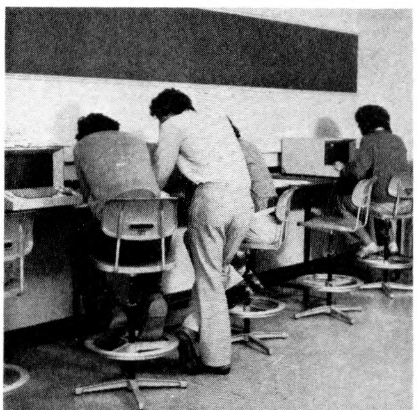
A final reason for the resurgence of interest in computer assisted learning techniques is the changing climate of opinion in education and training about the desirability of individualised instruction. In the 1960s programmed instruction came up against the brick wall of traditional *group*-paced and *group*-prescribed classroom methods. In the 1970s, modular courses, continuous assessment, independent study, distance learning, 'continuous flow' training (where trainees start courses when each arrives, not when the whole cohort has finally assembled), have all become much more acceptable. These moves require a more sophisticated approach to course management, for example, leading to interest in computer managed learning.

National Development Program in Computer Assisted Learning

In the United Kingdom, interest in CAL was stimulated by the Government's decision in 1972 to fund a major development program over a five year period, with a budget of around £2 millions.

The Program's main aim was to 'institutionalise' CAL, that is to say ensure that the institutions housing the development projects funded them totally at the end of outside funding.

Some 35 projects were set up, the majority in higher education, and lesser numbers in schools, military and industrial training. In Suffolk, history teachers developed the use of the computer for the manipulation of 19th Century census returns. In Glasgow, the University produced CAL materials for teaching medical students (simulations of patient conditions). In Northern Ireland, the New University of Ulster pioneered computer managed learning using the CAMOL package written by ICL. In the RAF, the computer's ability to simulate expensive radar equipment and give remedial teaching was exploited. By the end of the Program, 47 institutions in the UK were active, and another 100 here and overseas had taken on project materials. From 55 teaching staff involved in 1973/4, the total was 690 in 1977. In the final year, projects notched up 35,000 student terminal hours of teaching compared with 3000 four years earlier. Most significant of all, by the end of the Program 70% of the project institutions had decided to continue the work with internal resources.



Computer Assisted Learning at the University of Surrey

The Program commissioned two evaluations — on the educational, and financial, aspects of computer assisted learning. The key finding of the educational evaluators was, the extent to which the Program's approach to CAL moved away from the narrow types of computerised programmed instruction; and the rather interesting conclusion that the CAL work aimed at achieving high level thinking and problem-solving skills, in contrast to the American emphasis on CAI for more elementary teaching. The key finding from the financial evaluators was the high cost of developing CAL (it takes anything from 40 to 400 hours to develop one hour of good validated teaching material), and the impossibility of establishing quantifiable cost-benefits.

Micro mania and computer assisted learning

The recent rush of interest in microelectronics has begun to make CAL positively respectable after all these years!

The Little Professor from Texas Instruments, a mass consumer product, can be seen as a direct descendant of Professor Suppes' work at Stanford in the 1960s. In training circles, there is growing interest in using the computer to help with training and retraining, as a result of the computer's arrival. For example, Barclaycard and British Airways make large-scale use of computer assisted learning to train clerical staff on their computerised systems. The computer manufacturers with software products for computer assisted learning, such as CDC and IBM, see industrial training as a prime market. The personal computers coming into homes and small businesses often have computer assisted learning programs to help the new user.

The techniques of programmed instruction are being given a push from an unexpected quarter — Prestel, the new Post Office viewdata service. Prestel, the low-cost, easy-to-use mass information system linking PO computers through ordinary telephone lines to modified TV sets, routes the viewer from any one page to up to ten other pages or 'frames'. The *routing structure* of Prestel is what makes it so different from existing printed media like this magazine. Prestel looks like giving the Good Housekeeping seal of approval to educational ideas of programming instruction invented in North America twenty years ago.

There's even talk of educational revolutions . . . . . again.

Further reading

Richard Hooper, *The National Development Program in Computer Assisted Learning, Final Report of the Director,* London: Council for Educational Technology, 1977.

EVOLUTIONARY PROGRAMMING

"Computer technology is now entering a new phase, one in which it will no longer be necessary to specify exactly how the problem is to be solved. Instead it will only be necessary to provide an exact statement of the problem in terms of a goal and the 'allowable expenditure' in order to allow the evolution of a best program by the available computation facility."
Fogel *et al*

An intelligent answer?

Dr. D. J. Hand

Introduction

Early workers in Artificial Intelligence (AI) confidently predicted the quick development of intelligent machines — that is, of computer programs which could respond in an intelligent manner. Unfortunately, however, it turned out not to be as easy as they had expected. The optimism of these early researchers was replaced by a feeling of disappointment that the expected breakthroughs had not occurred. But advances — due partly to increases in computer power and partly to better understanding of the problems involved — are being made. In this article I outline one avenue which has been investigated — and which can be investigated by you, on your machine, no matter how small.

But let's not get carried away. On a relatively simple personal computer we'll not be able to emulate the performance of 10^{10} neurons, each connected to thousands of others. What we can do, however, is write a small program which will *learn* to make decisions and which will improve its problem solving ability as it gains experience. And note that I didn't say you teach it. This program can even learn to solve problems which you can't solve. That, coupled with the fact that this same program can work on a virtually unlimited type of problem, certainly means it satisfies some of the requirements for intelligence!

To see how we can write such a program let's make a slight digression to consider natural creatures possessing intelligence. More specifically, let us see how human intelligence arose.

Evolution

Evolution is one of the most fundamental and elegant concepts which man has ever discovered. The basic idea has been around for a long time although Darwin was the first to put it into an acceptable form — or perhaps the time was right. Anyway, to progress from the first simple single-celled creatures that oozed their way through the primeval sludge required a number of things.

First, the creatures had to be able to reproduce. That is, they had to be able to build or grow replicas of themselves. Note also that an essential attribute of reproduction for evolution is that it should be *imperfect* reproduction. The offspring should be similar to the parent and each other, but not identical. (In the natural world this is brought about by gene combinations and mutation). These slight differences mean that some individuals are more able than others to counter certain threats (so, for example, the slower runners get eaten). This process of natural selection produces a gradual change in the nature of the population.

The second requirement is number. A large number of organisms, each subtly different from the others, means that there is a greater chance some will survive in the face of an environmental threat.

The third requirement is time. The threats which affect the population of organisms must occur gradually relative to the reproductive period of the creatures. To take a naturalistic example, suppose that the creatures have bred in almost freezing water. Then they'll be lucky to survive if the water becomes boiling in a split second. A gradual change, lasting many generations, however, might allow some of each generation to survive. (The classic example of this is the gradual darkening of trees in the north of England due to soot deposits from factories. It was sufficiently gradual that the scalloped hazel moth was able to evolve a dark form which was well camouflaged and so could evade predators).

This basically simple mechanism allows constructs of tremendous complexity to be built up — witness ourselves. Fortunately for the readers of this magazine, however, it is not the complexity of the products which matters, it is the basic simplicity of the mechanism. It is this mechanism which we can simulate in our personal computers.

We can formalise the harsh world of nature by saying that at any moment there are criteria to be satisfied. Any creature which satisfies them (by being able to run fast, climb high, or think quickly, for example) lives, the others die. Nature's criteria are flexible, they depend on the place and the time. Not only that, but since the environment consists largely of other evolving organisms, the criteria themselves are evolving.

To simulate this process we need to be able to simulate imperfect reproduction and we need to impose selection criteria. These criteria will be chosen according to what we want to "breed" our simulated organism to do. The other two natural requirements or numbers and time are taken care of by the speed of the computer.

NEXT STATE

		Current state 1	2	3
Input	1	1	1	2
	2	1	2	3
	3	3	1	2

OUTPUT

		Current state 1	2	3
Input	1	1	3	2
	2	3	1	3
	3	3	2	1

Figure 1

Our basic approach will be the following: we begin with a number of simulated organisms and we test their performance on problems we know the solutions to. That organism which performs best will go on to be the parent of the next generation — its offspring being generated by making small random changes to the parent's structure. Each of these offspring will be tested and again the best will be selected. And so on.

This basic idea can be simplified or expanded according to the size of your computer. As I will show in the next paragraph the simulated organisms have a very simple internal structure. It is this which makes it possible to experiment with this idea on even a very small computer.

The body

Nature uses organic molecules and biological cells as the basic building blocks for man, its intelligent organism. What can play a similar role in our electronic simulation? We need something simple and yet something which can be put together to yield complex simulated organisms. Several things have been suggested and used, and the ones we shall adopt are *characters stored in rectangular arrays.* Each organism will consist of two such arrays, one labelled NEXT STATE and one labelled OUTPUT. NEXT STATE shows the organism's internal state and how this state changes in response to an input, while OUTPUT shows what output is given in response to each input.

Any alphabet of characters could be used in these arrays (one could even use words, names, letters, etc.) but for simplicity I shall stick to numbers. Figure 1 provides an example of a simple organism with three possible states.

Just as any instructions for a computer have to be translated into a language the computer can understand, so the problems for our organisms must be recoded into a suitable form. It is convenient, though not essential, if the organisms' inputs use the same alphabet as is contained in the arrays. In my case this means the digits 1 to 3. Problems to be tackled by the organisms are recoded to be streams of numbers.

Suppose that we have recoded some problem (practical examples are given below) to be the number stream 3 2 1 2 2 3 1 . . . Suppose also that the organism is initially in state 1 when the problem is fed in. This simply means that *current state* has value 1. From arrays NEXT STATE and OUTPUT in figure 1 we can see that input 3 will cause the organism to move to state 3 and to output value 3. For the number stream shown above the complete sequence of states and outputs is shown in figure 2. It is our aim that the stream of digits output by the organism shall be a solution to the problem fed in.

Input	Current state	Next state	Output
3	1 (Initial state)	3	3
2	3	3	3
1	3	2	2
2	2	2	1
2	2	2	1
3	2	1	2
1	1	1	1

Figure 2

In order for an organism to learn how to solve problems we must give it some examples to work on. Furthermore, in order to be able to improve its performance it must be able to compare the solution it gets with the correct solution. Note that this does not require us to be able to solve problems — merely that we have examples of solutions, or that we can recognise a solution when we see one. We shall begin from this point, assuming that we have a set of solved examples, but that we are unable to solve others outselves. Our aim is to develop an organism which can give solutions to the examples and which can generalise to new problems.

The way the organisms work and the outputs they yield are given by the numbers in the cells of the arrays. However, since we cannot solve the problems we cannot specify what numbers will lead to correct solutions. So let's start by putting a random selection of digits in the array; in figure 1 each cell contains a random choice from the set (1, 2, 3). We can now feed in our recoded problems one at a time and compare the outputs with the desired solutions. The difference between the output and the desired solution is a measure of the organism's performance — smaller difference means a better organism. Since it is not very reliable to assess performance on just one problem, however, we take the average of the differences over all the example problems as the measure of performance.

Now comes the evolutionary part of the approach. Starting from the organism we already have, to produce another one which is the same except for a few randomly chosen differences. We could, for example, change some of the cells in the NEXT STATE array. We could change some of the cells in the OUTPUT array. Or we could change the number of states.

When we test this new organism on the set of solved examples it will also give an average performance value. By comparing this with the parent's performance we can see which is the better organism. We then use whichever is the better to act as the parent for the next generation. And so on. We stop when we are satisfied with the performance achieved by the latest organism (or when we discover that the improvements are too slow — in which case we take steps to speed up the evolution — see "improvements".

We can test the final organism's ability to generalise to problems not in the set of solved examples by using it on a new set of problems.

Example

As a concrete illustration to make the ideas clear let's suppose we want to develop an organism which can multiply. For the sake of argument we'll suppose we don't know how to multiply, but that we have got some examples — maybe a scrap from a primary school child's exercise book. Since we are unable to teach the program we'll just have to present it with the examples and hope that it can learn by itself.

We might begin by trying a ten state organism (with states 0 to 9) so that the organism's two arrays are each ten by ten. Again, since we are unable to do better, we'll make a random choice from 0 to 9 for the numbers to fill each array's 100 cells.

This organism then tackles each problem using the following steps:

(i) Initialise — e.g. set the organism's initial state to 0.
(ii) Input the first number to be multiplied (say 8).
(iii) Ignore the output which this generates.
(iv) Feed in the second number to be multiplied (say 3).
(v) Feed in 0 to initialise the organism for the next trial.
(vii) This input will generate a single digit output (let's say 7).
(viii) Compare the outputs 79 with the desired output (8 x 3 = 24).

Not surprisingly, since the values in the arrays' cells were randomly chosen, this is not very good (79 — 24 = 55). However, as I have already said, it's unreliable to assess the organism's performance on just one example. Instead we use all the solved examples and calculate the *average* difference between the actual and the desired outputs. And note that we always take the positive value of this difference — otherwise we could get an organism which did badly but had an average difference of zero! In fact we can go one better than this. Since larger differences are worse from our point of view we can square each difference and calculate the average of these squared differences. Large differences are now even more detrimental to the overall performance score. In terms of the language of the previous section, this average of squared difference is the criterion by which we select organisms.

This is the point where we introduce evolution. We produce a new organism which is almost the same as the one above, but which has minor random differences. We then compare the performance of the new organism with that of the old one. If the new organism is better then it serves as the parent for the next generation. If not, then we make some different random changes to the original organism and try again.

At each generation we select the best organism to serve as the parent for the next generation. This means that performance can never get worse. I programmed this example — figure 3 illustrates the achieved learning curve.

The flexibility of such an approach can be demonstrated if I change my mind about what I want the organism to do. Suppose, instead of multiplication, I want it to add. It is not necessary to change the program at all — all I need to do is to replace the multiplication examples by addition examples and let the program evolve a set of arrays which minimise the difference between the examples' solutions and its attempts. Figure 4 shows the results of this.

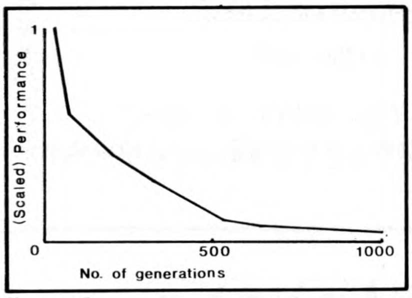


Figure 3

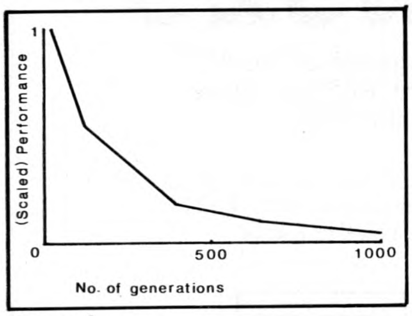


Figure 4

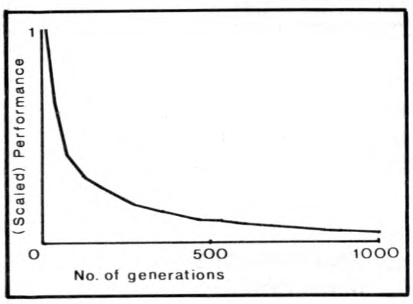


Figure 5

Then, being hard to please, I changed my mind again and decided that instead of addition I wanted the organism to take pairs of numbers and tell me which was the larger. I gave it some solved examples — coding 1 if the first number was larger and 0 if the second was larger. The learning curve is illustrated in figure 5.

Figure 6 gives the flow diagram of this basic program.

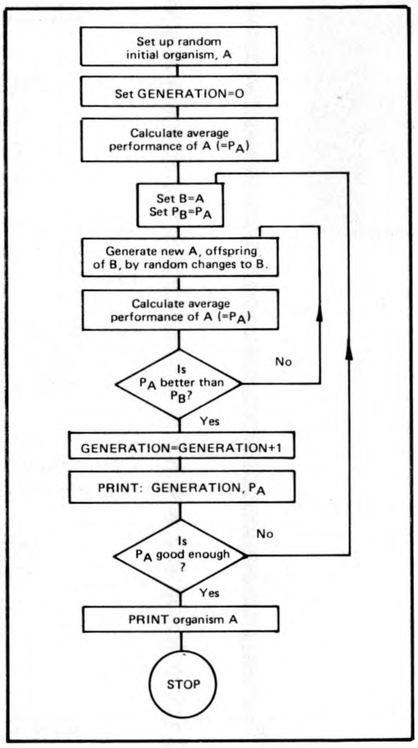


Figure 6

Improvements

In this section I suggest how the basic program can be improved.

The simplest way to increase the rate of evolutionary change is probably through multiple offspring. In the preceding examples each parent produces one offspring and we compare its performance with that of its parent. We could, instead, make five (or any number) different types of random changes to the parent and compare the six organisms (one parent and five offspring), selecting the best to father the next generation.

Extending this approach it is possible to select the *several* best at each stage, running several evolutionary branches in parallel. This is, of course, what happens in nature — where fish have evolved at the same time as birds. These approaches are useful in that they decrease the chance that our system will fail because it has followed an evolutionary dead end.

Since this is obviously a desirable attribute, ways have been developed to apply it to the simple program of the preceding sections. This is done by trying more and more drastic random changes as the evolution proceeds. In the multiplication example, for instance, we might begin by making 10 random changes at each step, then increase this to 20 and so on.

Just as in nature size confers some advantages so it does here. The larger the machine the more states it has, and hence it can handle more complex problems. However, in a small personal computer size can also be a disadvantage. This property can be included in the evolutionary process by making the criterion depend not only on performance, but also on the organism's size. This would mean that as well as trying to minimise the average error, the process would also try to keep the organism's size small.

A simple criterion would be

$$C = A \times \text{(average difference between output and solution)} + B \times \text{(number of states)}.$$

A and B are constants, fixed by you, giving the relative importance of the two parts of the criterion.

In the preceding section we used the square of the difference between the output and the known solution as a measure of performance. Of course, it is not necessary to use this particular measure — anything could be used. However, some are better than others in that they lead to quicker evolution. A little experimentation could be a good idea. In theory, of course, there is no reason why one shouldn't *evolve* an effective measure of performance — but this is getting rather complicated!

An extra sophistication which can be investigated by those with larger machines is that of mating. Nature uses sex as a means of providing a good mix of genes and some experiments have been done attempting to combine attributes from two (or more!) organisms.

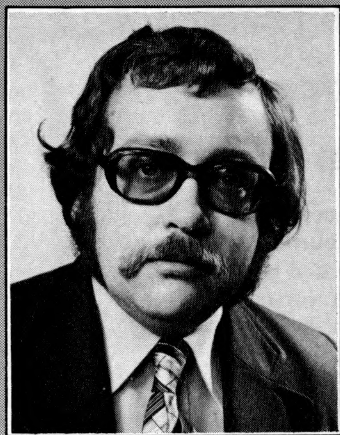
Conclusion

In this article I have tried to explain the basic ideas underlying evolutionary programming. The examples I've given have been very simple but the methods have been applied to much more complex problems (for example, they have been used to predict trends in time series and for some problems in multivariate statistics). The beauty of the method lies in the way it can be used to tackle problems that the programmer cannot solve (though a large machine would be needed for this) as well as in the way a single program can be applied to a tremendous variety of problems.

Since only two organisms at a time need be stored and since each consists of only two arrays, worthwhile experiments can be performed on even a very small computer.

I leave it to you to discover the excitement of developing your own life-form.

PCW For a fascinating comment on orthodox and "modern" ideas of evolution, see Arthur Koestler's ***The Ghost In The Machine*** (Pelican Books). **PCW.**



ON THE LINE

David Hebditch

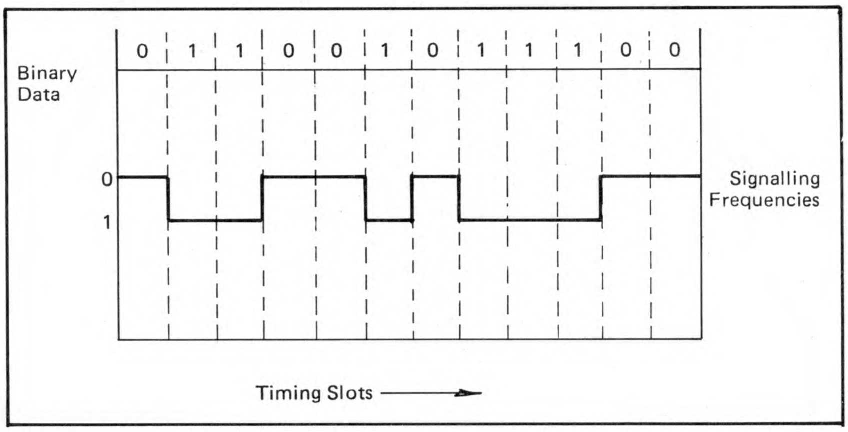


Figure 1: Time Slots in FSK Signalling

Down to Basics: Timing and Synchronism

Last month we talked about the representation of binary data on telephone lines using the Frequency Shift Keying (FSK) technique and introduced the idea of modems to convert DC signals in our computer or terminal to AC signals on the telephone line and vice-versa. You may find it useful to have that article to hand while you read this one.

This month we shall consider the problems of transmission timing and how to transmit whole characters down the line. Figure 1 shows how a time slot is allocated to each bit sent.

To keep things simple, we will just consider one channel.

The timing of the transmission is determined by an oscillator (clock) in the line interface board which is used to ensure that the DC bit signals passed over the interface to the modem are evenly spaced. The modem merely responds to this by sending the correct frequency for whatever signal level is currently on its input lead. (We will consider this interface in more detail next month).

Does this mean that the speed of transmission is a function of the rate of the clock we put on the interface board? Unfortunately it does not. The rate possible is limited by the characteristics of the circuit being used (in our case a grotty dial-up line), the sophistication of the modulation technique being used (not much hope there, either) and ultimately by the bandwidth of the channel (and if you do not believe that, read Shannon on Information Theory — it will serve you right).

The transmission rate is measured in bits per second (bit/s). This should not be confused with the baud rate which is the number of signalling intervals per second. Because some modulation techniques are able to get as many as 16 bits in each signal-slot, there is a big difference between two measures.

With FSK signalling on dialled lines, the maximum transmission rate we are likely to achieve is 300 bit/s (in each direction). I know it's not much but above that speed the cost of modems will soon become so high that they will cost as much as your whole computer system.

Obviously it will be necessary to fill in some more detail in later articles but we now have a simple mechanism for moving bits from one system to another over the public telephone network. However, in computers we usually handle data in a character format (except when doing arithmetic manipulation) and we will find it convenient to stick to this for data transmission.

This requirement raises the problem of how to indicate to the receiving device the start and end of each character. To achieve this we 'frame' each character (usually 7-bit ASCII) with start and stop bits as shown in Figure 2:

When the transmitting end starts sending the carrier frequency, the receiver begins to listen. At this time all it hears is a series of 1-bits but does not accept them as data. When the first character is to be transmitted, the sender precedes it with a 0-bit, known as the *start bit.* It has to be a zero or the receiver would not see it coming (after all, a 1-bit looks just like any other 1-bit).

The start bit has the effect of nudging the receiver and saying 'Oi, wake up! Here comes a character'. The receiver's clock is then started and the next seven (or six or eight) bits are deserialized into a buffer for passing into processor memory as a complete character. The eighth bit establishes odd parity for the character (according to the ASCII standard) and enables any errors during transmission to be detected.

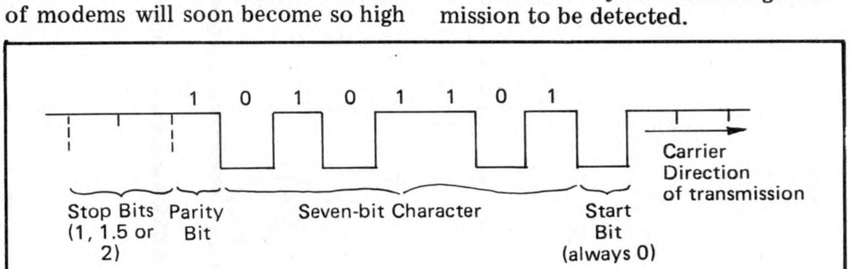


Figure 2: Character Framing for Start-Stop Transmission

After the parity bit comes 1, 1.5 (1.5?) or 2 'stop bits' before the next start bit or a further period of idle carrier. The stop bits are a hangover from telegraphy systems when it was necessary to give the receiving teleprinter time to print one character before the next one arrived. Legend has it that the 1.5 was some kind of historical mistake which is still with us (like firemen on diesel trains). As we will see in a later article, some of these variable characteristics are selectable, by the user, by strapping the board or by loading a parameter register via software.

The type of transmission is called *start-stop* or *asynchronous* and has the benefit of simplicity and low-cost and the disadvantage of inefficiency. For each seven data bits we have to send up to four additional overhead bits. This means that our effective transmission speed will be less than 30 ch/sec at a 300 bit/s data rate.

Now we have an arrangement for getting characters from one end of a dial-up line to another. Next month we will consider the interface to the modem and procedures for establishing calls and getting data from one system to another. For further reading try my own book 'Data Communications — An Introductory Guide' (Paul Elek Science Ltd. £5.50).

Second PCW Microprocessor Chess Tournament

Following the success of the first **PCW Chess Tournament** last year, we are pleased to announce that our second tournament will take place at the **PCW** show in London, November 2nd - 4th 1979.
It is hoped that some financial support may be available for private entrants from outside the U.K., to defray travelling expenses, and there will be at least one cash prize. The highest placed programs will be eligible to compete in the first **World Microprocessor Championships** which will be held at the 1980 **PCW** Show.

Detailed rules and entry forms will be available in due course.
Prospective entrants are requested to write to David Levy (c/o Personal Computer World, 62a, Westbourne Grove, London W2) who will be acting as commentator and tournament manager.

PCW OPEN PAGE

THE AMATEUR VIEW

Mike Lord

NEWS FROM A CHIP SHOP

While the headlines are being taken by bigger and better microprocessors like the Z8000, 68000, and now Western Digital's new chip set that will run Pascal p code directly, it is still worth while reading the smaller notices in technical journals. For some useful support chips are being introduced which can prove extremely useful to the computer constructor.

For example, National Semiconductor's MM74C912 and 917 6-digit BDC/HEX Display Controller/Drivers. These are designed to connect directly on to a processor data bus, appearing to the CPU as *normal* memory or IO registers into which it can write six HEX or BCD digits. The chip then performs a multiplexing and decoding function to drive a 6-digit 7-segment plus decimal point display. The 74C917 gives a HEX display (0–9,A–E). As NSC's data sheet sats; "This combination of write only memory and self-scan display makes the display controller a 'refreshing experience' for an overburdened microprocessor".

Also from NSC, the INS8208 crams 8 bi-directional transceivers into a narrow 20 pin DIL to please those who like big busses but small PCB's. One port can sink a hefty 48mA in the 'O' state, and provide 10mA in the '1' state, allowing the chip to drive a heavy 300pF bus load while still maintaining 17ns typical delay. Logically, the chip pin connections are arranged with one port on each side of the package.

AND FROM OTHER MEETING PLACES

Mr. S. C. Bird (139 The Moors, Kidlington, Oxon) reports that the *Oxfordshire AMC,* of which he is the General Secretary, is expanding all the time and is starting up branches in *Witney* and *Oxford.* The original group is based on *Kidlington* where regular fortnightly meetings have featured Tandy, Sintel and Newbear.

The *Merseyside Microprocessor Group's* latest newsletter gives every indication that this is an enthusiastic and flourishing organisation. They now have two newsletter editors; Alan Pope and Dr. Malcolm Taylor, and have organized several Special Interest Groups, organised by:

Eric Stancliffe, 4 Withensfield, Wallasey, Merseyside (051 709 6022 x 2967) for the 6800 User Group.

G. Myers, 34 Hillcrest Drive, Greasby, Wirral (051 677 9340) for NASCOM

A. G. Price, Dept. of Mathematics, Liverpool Polytechnic, Byrom Street, Liverpool L3 (051 236 0598) for PET

Mrs. M. Rouse, 7 Kingscourt Road, West Derby, Liverpool L12 (051 228 1669) for Computer Education Group.

John Stout, Dept. of Architecture, Liverpool Polytechnic, 53 Victoria Street, Liverpool L1 (051 236 0598) for PET

Dr. Martin Beer, Computer Lab, University of Liverpool (051 709 6022 x 2967) for Micro's in the Laboratory.

Recent meetings of the main MMG have included a session on Microprocessors and Medicine, and they were represented at an all day event held at Neston Comprehensive School to introduce school teachers to current activities in Science and Technology teaching. On April 19th. Alan Stirling and Eric Stancliffe will be demonstrating and discussing their 77–68 microcomputers. Further details on the MMG itself may be obtained from John Stout or Dr. Martin Beer, addresses and telephone numbers as above.

Readers living in the *Newcastle* area are invited to join a local personal computer society that holds meetings on the first Tuesday of each month; ring Dr. W. G. Allen on 0632 851528 for more details.

Finally, if you are in the process of starting up a new group, be careful how you name yourselves. The newly formed *Poole* group had a very strange response to an advert they placed in the local paper. The first reply opened expressed an ardent desire to form a genuine relationship, and several others were in the same vein. On reflection this may have been due to the wording of the advertisement; "Personal Computer Club . . .", decidedly ambiguous. But then, don't some machines have built-in error correction devices? However, those living in the *Poole/Bournemouth* area who believe that bytes are bundles of bits may wish to contact the Poole Home Computer Club at Flat 3, 43 Lindsay Road, Branksome, Poole with a view to closer encounters.

POP SHOP

The **PCW** *Open Page Service for amateurs. Buy, sell, exchange –*

One-off advertisements only accepted. Not more than 50 words.

Scarborough (Yorkshire) Computer Group monthly meetings. Next on 11th April at Talbot Hotel, Queen Street, Scarborough. at 7.30 p.m. Further details from **Des Wood** on Scarborough 63982.

FOR SALE: TRS 80 LEVEL II, 4K
Few months old only: **£450**
Contact **C. GORE,** 30 Midlane Close, BASINGSTOKE, Hants. **Tel: 21468**

NASCOM I FOR SALE
Assembled in full working order (with improved modulator and video blanking). B/W TV and cassette recorder included.
Price complete with original documentation plus Z80 programming manual, **£200.** Buyer collects.
Tel: DURHAM 69695 after 4.30 pm

FOR SALE
Data Dynamics 390, used for 480 hours (verifiable), **£480;** contact **Dave Peatroy,** 15 Norman Road, Ashford, Middlesex.

Wanted Second Hand PETS. Others considered too.
01-979 4546
for – **Richard Mortimore**

FOR SALE
As New **APPLE II**, 48K RAM, Extra Books and Programs
£1150
Also **ELF I P.C.B.,** 1802 D Processor, 1861, Manuals, Bits and Pieces **£20.** Free Creed 7 (unmodified) to purchaser of above. **Telephone: 0632-468041 Evenings**
0385-888381 Day

CHECKMATE
A computer program for playing a chess ending

PCW *Readers, look out for an article on how to approach chess programming. It's by International Chess master David Levy, and will appear in the next issue.* **PCW**

Francis Best

Introduction

I am 18 years old and started computing when I entered the sixth form at school. A thriving interest in chess caused the obvious link to some kind of chess playing program but the aim of this article is not just to explain how my program works. I also hope to indicate some of the problems encountered and how I overcame them. In short, I would like to paint a more human picture of how an idea is conceived and translated into something which is meaningful in a computer. This, I think, is important; especially since I was relatively inexperienced in programming when I started 'Checkmate' (as I decided to call it), and was not entirely sure how I was to set about writing the program.

1. Early considerations

The first thing to be considered was the basic structure of the program. One method would be to devise a set of rules for determining each move. This would have the disadvantage of only being applicable to the ending of king and rook versus king, which I had chosen. It would also be a rather inflexible approach, since it would be difficult to modify rules in the light of experience.

Another method would be to scan ahead each position a number of moves and choose the line which seems to drive the opposing king towards the edge of the board. This system is the closest to how computers are currently programmed to play a whole game of chess, but most of these programs do not play very well, so that left me with one alternative.

The final method involves considering each move and scoring it according to its various merits. Parameters could be used to adjust the comparative importance of each of the heuristics.

Although the third approach was eventually adopted, and produced a reasonable level of play, it would be much more difficult to use the same technique for a whole game, when many more factors than I used would have to be taken into account.

There were a few other minor considerations. For the input and output of moves, descriptive chess notation was used. This was possibly unwise as there was little string handling available in the version of BASIC which I used. Consequently, I stored all possible strings so that they could be compared with the input. Clearly this would be impossible on a small system. For input of the starting position, I used coordinates (file then rank).

2. Move generation

Given that we have stored the starting position in the computer, it is a relatively simple matter to generate a list of all legal moves by performing simple additions on the coordinates of each piece. For the rook, we start in the square on the same file, on rank 1, then we add one to the second (rank) coordinate. If the original position of the rook is encountered, that move is eliminated and if the position of the white king is encountered, move generation along that file is stopped. The process is then repeated for the moves along the rank on which the rook is standing.

This system assumes that the white king is not between the square of the rook and the square, on which move generation is started. If this situation does arise, the method is similar, except that generation occurs *in front of* the (white) king to the last rank on that file. This is made clearer by a diagram:

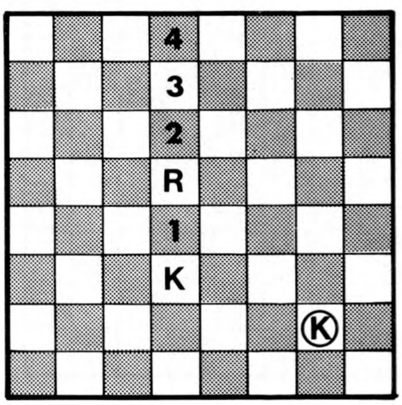


Fig. 1 Ⓚ = **black king**
Numbers indicate order in which rook moves are generated along file.

The king moves present little problem. Here, 1, 9, 10 and 11 are added and subtracted to the original position and a subroutine is called into play to check the legality of the move. The move is stored, if legal; and rejected otherwise.

2. The legality subroutine

This subroutine has a number of uses. It checks that the coordinates of the pieces are neither less than 1 nor greater than 8, and that no two pieces are on the same square. Finally, if the rook is attacking the black king, then the position is counted as illegal. The variable L is set to 1 or 0 according to whether legality is verified or not.

The immediate usefulness of the subroutine is in checking the original position input. It is also useful in generating the moves of the white king as indicated above. Another use is to see if black is in check. If the king is actually attacked, it depends on whose move it is whether a position is really legal or not.

4. The heuristics

(i) This heuristic encourages reducing the number of squares that the black king can move to. The score given to each move is kx(8 - s), where s is the number of moves available to the black king, and k is a weighting factor which will be discussed later. The only problem with this heuristic is that it would encourage stalemate. Therefore, the position is examined to see if black is in check (this is after s has been found to be 0). If he is, then that move will be made (since it is checkmate), otherwise the score for that move is taken to be -1000, thus assuring that that move will not be made. Incidentally, a score of -1000 is also given to a position where the white rook can be taken.

An example of the usefulness of this heuristic can be seen from the following diagram:

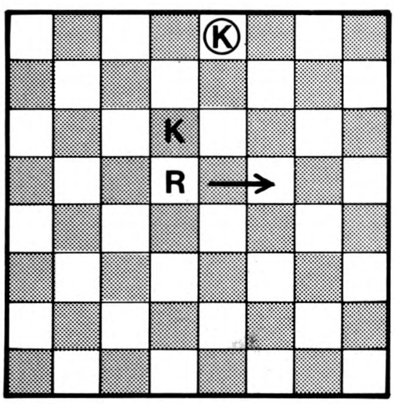


Fig. 2

Here, the move which allows black only one move would be encouraged by heuristic 1. Black would then be forced to play K—Q1 and R—KB8 is checkmate.

(ii), (iii) & (iv) These heuristics are all linked, so it is better to treat them as one unit. They are governed by the following expressions:

Heuristic	*Expression*
2	k_1 (14-distance of white and black kings)
3	k_2 (14-distance of w. rook and b. king)
4	k_3 (14-distance of white rook and king)

The distance being simply defined as the sum of the differences of the corresponding coordinates. The general idea is for the program to keep the pieces close together, the weightings (k_1) providing some way of introducing priorities. The following diagram illustrates how the parameters work.

The three moves indicated would all be quite possible: heuristic (i) would encourage move 1. But if R—Q7 is played, black can reply . . . K—K3, and white will have to move his rook away from danger. With suitable weightings for heuristics (ii) — (iv), though move 2 or 3 can be encouraged, either being better than move 1.

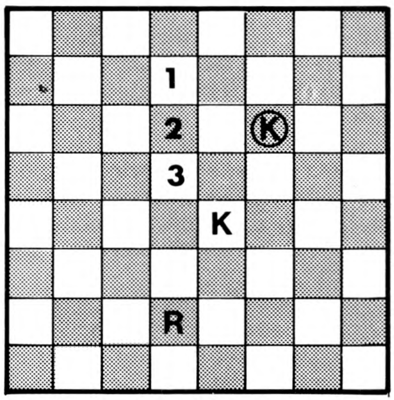


Fig. 3

(v) This heuristic is relatively unimportant. The idea is that the zone marked on the diagram is deemed less preferable for the rook to be on than the rest of the board. Similarly for the white king.

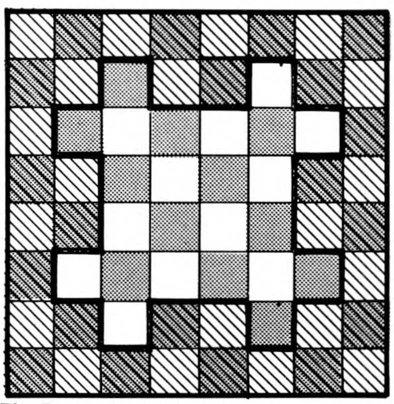


Fig. 5
Zone marked is less preferable for white king

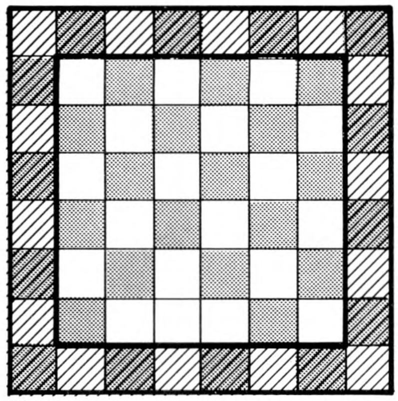


Fig. 4
Zone marked is less preferable for rook

The underlying idea of this heuristic is that the white pieces should avoid moving to squares on the edge of the board.

The preceding heuristics were those I arrived at somewhat informally at the outset. They were based on a simplified analysis of my own thoughts on how the ending should be played. After a little experimentation I decided to add another heuristic:—

(vi) This heuristic turned out to be useful in several unexpected situations. Basically, its aim is to encourage white to place the rook so that it is more or less the same distance from both of the kings. This is accomplished by dividing the number of moves that the black king would require to reach the rook by the corresponding number for the white king. This is then fed into a quadratic whose maximum is 1¼. The reason for having the maximum just greater than 1 is that, generally, it is better for the rook to be slightly closer to the white king than the black.

For calculating the shortest number of moves that a king requires to reach a certain square, it is only necessary to take the maximum of the two differences in corresponding coordinates (see diagram).

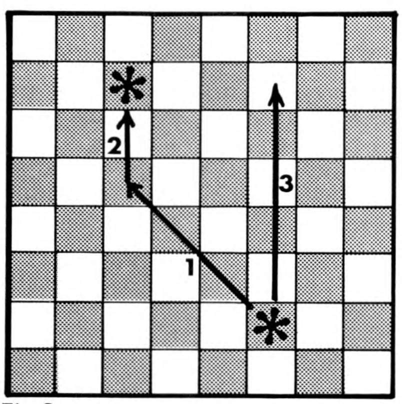


Fig. 6
Shortest route between the 2 squares marked with an asterisk is indicated by arrows 1 and 2. Route is same length as arrow 3, which is largest difference between co-ords.

The following position illustrates the usefulness of the heuristic (see Fig. 7). Here, white does not really want to do anything for a move. The moves indicated are examples of reasonable ways in which this can be done. Move 3 has the disadvantage that the black king can escape towards the centre of the board. If, however, we weight heuristic (iv) so that it is more important than (ii), that move will be eliminated. Move 2 is slightly inferior to 1 because the movement of the rook is more limited. This is where heuristic (vi) comes in. The actual quadratic used was: $2\frac{1}{2}X-X^2$, where X is the fraction of the two distances. Now the value of the expression is ½ more for move 1 than 2, so if we weight the heuristics appropriately, our aim will have been achieved.

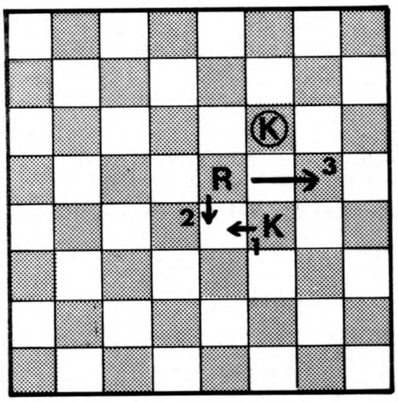


Fig. 7

5. Problems encountered

The weightings for the heuristics was the most difficult problem with this program. I first tried likely looking values and played various positions out to see the kind of move which it was found difficult to deal with. Having done this, it was then possible to set up a number of inequalities by comparing two high-scoring moves. This is best illustrated by an example.

Consider the following position:

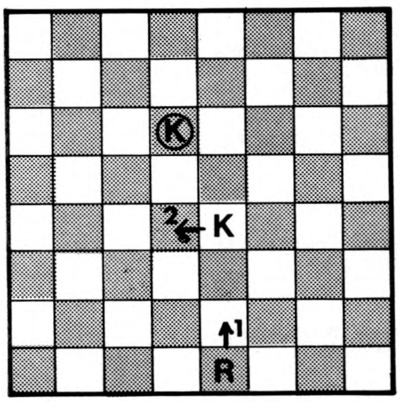


Fig. 8

Surprisingly, it was found that move 1 scored fairly highly. Move 2 is easily the best here, though. Taking each parameter in turn, we can compare them for both positions.

(i) After move 2, black has three moves. After 1, he has six.

(ii) Here the white king is one square closer after move 2 than 1.

(iii) This is better for move 1; the rook would be one square closer than for move 2.

(iv) This also favours 1; after move one, the rook and white king are 2 squares closer together than after 2 (remembering that it is not the actual distance but the difference in co-ordinates).

(v) This favours move 1 also. The king stays out of the low-scoring zone for both moves but the rook is on the edge of the board for 2 and not for 1.

(vi) This is in favour of move 2.

After move 2, we have:

$(5/3).2\frac{1}{2}-(5/3)^2$

After move 1, we have:

$(4/2) . 2\frac{1}{2}-(4/2)^2$

Therefore, the overall advantage for 2 is (25/18)−1 or (7/18).

Taking all these factors into account, we require

$$k_5' + 3k_1 + k_2 + (7/18)k_6 > k_3 + 2k_4 + k_5''$$

where k_1 is the weighting parameter for heuristic number i, k_5' is the score added on when the rook is on the edge of the board and k_5'' the score added on when it is not on the edge.

Thus, it is possible to go about obtaining the best parameters in a methodical way. It also revealed rather a serious problem. In calculating the inequalities, I reached several requirements which could not be fulfilled together. So the heuristics were not sufficient on their own.

There were two ways round this problem. Either more heuristics could have been introduced, or a provision could have been made for changing the parameters according to the position. The first method, I felt, would have quite a high chance of failing for reasons similar to those for the failure of the original heuristics. Also, the second method seemed more true to life; a human chess-player adjusts the priorities of a position as the game progresses.

There was one type of position which seemed to contradict the tendencies of the heuristics (see Fig. 9). If the rook moves away as indicated, black has to play . . . K–K1 and is then mated with R–QB8. The heuristics, though would almost certainly give the highest score to K–Q6, which is a much slower ein. In fact, a similar move is probably best even when the king is not confined to the back rank. Such a move is therefore played automatically in positions of this type (as is the follow up of the rook checking the black king). This may seem a bit of a cheat but I maintain that this, too, is how a human chess player works: he knows a few general principles together with a limited number of exceptions.

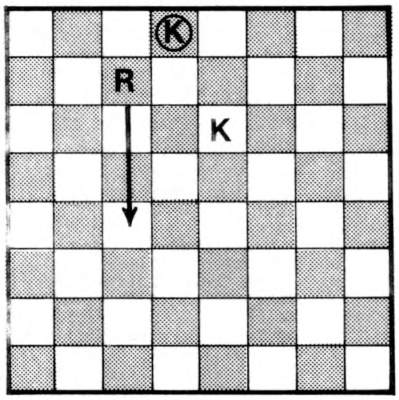


Fig. 9

Besides these special moves, a modification was introduced to recognize three classes of position and use a different set of parameters for each class. First, a check is made to see if the corresponding coordinates of the two kings differ by more than 2. If either set does, then parameters are chosen to encourage moving the white king closer to the black king. The other sets of parameters correspond to positions where the black king can be forced to an unfavourable square by a rook check and, finally to positions which most commonly occur i.e., those where white has no immediate win but must improve his position according to the general principles built into the program.

6. Success

At this point, it was possible to adjust the three sets of parameters so that a fairly good game could be played. I hasten to add that the program does not always checkmate in the most efficient manner, but the worst I have had so far is taking 23 moves to win. The average seems to be around 14 moves.

I enclose a listing of the program with a few sample runs. I leave the reader to form his own conclusions, but whatever one thinks of 'Checkmate', it was fun, and that's the main thing!

Final comments

The program was run on an ICL 1903 machine, where there was no shortage of storage. On a small system, though, an alternative approach would be to utilize a certain amount of look-ahead, thus reducing the need for so many heuristics. With care, look-ahead need not involve storing a large number of positions if moves are analysed and assessed fully before going on to consider another one. The actual process of look-ahead, being an essentially repetitive procedure, would not take up too much storage.

CHECKMATE

```
10 REM  ROOK & KING V. KING CHESS ENDING (MK II)
30 FILES MOVES
40 DIM T$(440),U$(440),V$(100),W$(100)
50 DIM R(14),D(14),F(14)
60 DIM Y(30)
70 REM  READ IN DATA FOR CHESS NOTATION.
80 RESTORE #1
90 FOR  A = 1 TO 8
100    FOR  A1 = 1 TO 8
110      READ #1,T$(10*A1+A)
120    NEXT A1
130 NEXT A
140 FOR  A2 = 1  TO 8
150    FOR  A3 = 1 TO 8
160      READ #1,U$(10*A3+A2)
170    NEXT A3
180 NEXT A2
190 READ Z(0),Z(1),Z(2),Z(3),Z(4),Z(5),Z(6),Z(7)
200 REM  READ PARAMETERS FOR SCORING POSITIONS.
210 READ A(0),A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8)
220 READ B(0),B(1),B(2),B(3),B(4),B(5),B(6),B(7),B(8)
230 READ C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8)
240 PRINT TAB(20);"?";
250 INPUT A$
260 IF  A$ = "END"  THEN 9999
270 IF  A$ = "NEW"  THEN 770
280 IF  A$ = "INFO"  THEN 520
290 IF  A$ = "OFF"  THEN 540
300 IF  A$ = "C1"  THEN 560
310 IF  A$ = "C2"  THEN 590
320 IF  A$ = "C3"  THEN 610
330 IF  A$ = "SHOW"  THEN 630
340 FOR  A4 = 1  TO 8
350    FOR  A5 = 1  TO 8
360      A = 10*A5+A4
370      IF  A$ = U$(A)  THEN 420
380    NEXT A5
390 NEXT A4
400 PRINT "INVALID INPUT";
410 GO TO 240
420 H = 9-A+20*INT(A/10)
430 A = ABS(H-H1)
440 IF  (A-1)*(A-10)*(A-11)*(A-9) = 0  THEN 470
450 PRINT "ILLEGAL MOVE";
453 H = H1
454 H2 = INT(H/10)
455 H3 = H-10*H2
460 GO TO 240
470 H2 = INT(H/10)
475 H3 = H-10*H2
476 GO SUB 3290
480 IF  L = 0  THEN 450
485 H1 = H
490 W$(I) = A$
500 I = I+1
510 GO TO 950
520 N3 = 1
530 GO TO 240
540 N3 = 0
550 GOTO 240
560 PRINT "?";
570 INPUT A(0),A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8)
580 GO TO 240
590 PRINT "?";
595 INPUT B(0),B(1),B(2),B(3),B(4),B(5),B(6),B(7),B(8)
600 GO TO 240
610 PRINT "?";
615 INPUT C(0),C(1),C(2),C(3),C(4),C(5),C(6),C(7),C(8)
620 GO TO 240
630 PRINT
640 PRINT
650 PRINT
660 PRINT TAB(6);"WHITE","BLACK"
670 FOR  A6 = 1  TO I
680    IF  Y(A6) = 1  THEN 710
690    PRINT A6;TAB(6);V$(A6),W$(A6)
700    GO TO 720
710    PRINT A6;TAB(6);V$(A6);"+",W$(A6)
720 NEXT A6
730 PRINT
740 PRINT
750 PRINT
760 GO TO 240
770 PRINT "TYPE STARTING POSITION";
780 INPUT F,G,H
790 PRINT
800 I = 1
810 F1 = F
820 G1 = G
830 H1 = H
840 F2 = INT(F/10)
850 F3 = F-10*F2
860 G2 = INT(G/10)
870 G3 = G-10*G2
880 H2 = INT(H/10)
890 H3 = H-10*H2
900 REM  CHECK THAT POSITION IS LEGAL.
910 GO SUB 3240
920 IF  L = 1  THEN 950
930 PRINT "ILLEGAL POSITION";
940 GO TO 240
950 REM  DECIDE ON SET OF PARAMETERS
960 P = ABS(G2-H2)
970 P1 = ABS(G3-H3)
980 IF  (1-SGN(P-2))*(1-SGN(P1-2)) = 0  THEN 1250
990 P2 = ABS(F2-H2)
1000 P3 = ABS(F3-H3)
1010 P4 = ABS(G2-H2)
1020 P5 = ABS(H3-G3)
1030 P6 = ABS(F2-G2)
1040 P7 = ABS(F3-G3)
1050 IF  ABS(P2-1)+ABS(P4-1)+ABS(P6-2) = 0  THEN 1140
1060 IF  ABS(P3-1)+ABS(P5-1)+ABS(P7-2) = 0  THEN 1180
1070 IF  ABS(G2-H2)+ABS(1-SGN(P2-2)) = 0  THEN 3620
1080 IF  ABS(G3-H5)+ABS(1-SGN(P3-2)) = 0  THEN 3650
1090 IF  (H2-1)*(H2-8)*(H3-1)*(H3-8) <> 0  THEN 1290
1100 IF  (H-11)*(H-18)*(H-81)*(H-88) = 0  THEN 1330
1110 IF  (H-12)*(H-21)*(H-71)*(H-82) = 0  THEN 1330
1120 IF  (H-17)*(H-28)*(H-87)*(H-78) = 0  THEN 1330
1130 GO TO 1290
1140 F = 10*F2+2-SGN(F3-1)
1150 IF  H3 > G3  THEN 1210
1160 F = 10*F2+7-SGN(F3-8)
1170 GO TO 1210
1180 F = F3+10*(2+SGN(1-F2))
1190 IF  G2 < H2  THEN 1210
1200 F = F3+10*(7-SGN(F2-8))
1210 F2 = INT(F/10)
1215 F1 = F
1220 F3 = F-10*F2
1230 GO TO 2580
1250 FOR  B5 = 0 TO 8
1260    X(B5) = A(B5)
1270 NEXT B5
1280 GO TO 1360
1290 FOR  B6 = 0 TO 8
1300 X(B6) = B(B6)
1310 NEXT B6
1320 GO TO 1360
1330 IF  ABS(P-2)+ABS(P1-2) = 0  THEN 1333
1331 IF  (G2-1)*(G2-8)*(G3-1)*(G3-8) = 0  THEN 1250
1332 GO TO 1339
1333 IF  (H-11)*(H-18)*(H-81)*(H-88) = 0  THEN 1339
1335 GO TO 1290
1339 FOR  B7 = 0 TO 8
1340    X(B7) = C(B7)
1350 NEXT B7
1360 REM  GENERATE ALL LEGAL MOVES
1365 M1 = M2 = 1
1370 IF  F2 = G2  THEN 1500
1380 IF  F3 = G3  THEN 1680
1390 FOR  A7 = (10*F2+1)  TO (10*F2+8)
1400    IF  A7 = F  THEN 1430
1410    R(M1) = A7
1420    M1 = M1+1
1430 NEXT A7
1440 FOR  A8 = (F3+10)  TO (F3+80)  STEP 10
1450    IF A8 = F  THEN 1480
1460    R(M1) = A8
1470    M1 = M1+1
1480 NEXT A8
1490 GO TO 1860
1500 IF  F > G  THEN 1620
1510 FOR  A9 = (10*F2+1)  TO (G-1)
1520    IF  F = A9  THEN 1550
1530    R(M1) = A9
1540    M1 = M1+1
1550 NEXT A9
1560 FOR  A0 = (F3+10)  TO (F3+80)  STEP 10
1570    IF  A0 = F  THEN 1600
1580    R(M1) = A0
1590    M1 = M1+1
1600 NEXT A0
1610 GO TO 1860
1620 FOR  B = (G+1)  TO (10*F2+8)
1630    IF  B = F  THEN 1660
1640    R(M1) = B
1650    M1 = M1+1
1660 NEXT B
1670 GO TO 1560
1680 IF  F > G  THEN 1800
1690 FOR  B1 = (F3+10)  TO (G-10)  STEP 10
1700    IF  B1 = F  THEN 1730
1710    R(M1) = B1
1720    M1 = M1+1
1730 NEXT B1
1740 FOR  B2 = (10*F2+1)  TO (10*F2+8)
1750    IF  B2 = F  THEN 1780
1760    R(M1) = B2
1770    M1 = M1+1
1780 NEXT B2
1790 GO TO 1860
1800 FOR  B3 = (G+10)  TO (F3+80)  STEP 10
1810    IF  B3 = F  THEN 1840
1820    R(M1) = B3
1830    M1 = M1+1
1840 NEXT B3
1850 FOR  N9 = (10*F2+1)  TO (10*F2+8)
1851    IF  N9 = F  THEN 1858
1852    R(M1) = N9
1855    M1 = M1+1
1858 NEXT N9
1860 M2 = 1
1870 FOR  B4 = 0 TO 7
1880    G = G1+Z(B4)
1890    G2 = INT(G/10)
1900    G3 = G-10*G2
1910    GO SUB 3530
```

```
1920   IF  L = 0  THEN 1950
1930   K(M2) = G
1940   M2 = M2+1
1950 NEXT B4
1960 REM  EVERY LEGAL MOVE HAS NOT BEEN GENERATED
1970 G = G1
1980 G2 = INT(G/10)
1990 G3 = G-10*G2
2000 M1 = M1-1
2010 M2 = M2-1
2020 REM  NOW SCORING OF POSITIONS CAN START
2025 I5 = 0
2030 IF  N3 =,0  THEN 2050
2040 PRINT "ROOK"
2050 FOR  B8 = 1 TO M1
2060   F = R(B8)
2070   F2 = INT(F/10)
2080   F3 = F-10*F2
2090   GO SUB 3340
2100   IF  L = 0  THEN 2170
2110   F(B8) = 0
2120   GO SUB 2750
2125   IF  E0 = 1  THEN 2742
2130   IF  N3 = 0  THEN 2150
2140   PRINT F;S
2150   D(B8) = S
2160   GO TO 2190
2170   F(B8) = 10
2180   GO TO 2120
2190 NEXT B8
2200 F = F1
2210 F2 = INT(F/10)
2220 F3 = F-10*F2
2230 IF  N3 = 0  THEN 2245
2240 PRINT "KING"
2245 I5 = 1
2250 FOR  B9 = 1 TO M2
2260   G = K(B9)
2270   G2 = INT(G/10)
2280   G3 = G-10*G2
2290   GO SUB 2750
2300   IF  N3 = 0  THEN 2320
2310   PRINT G;S
2320   E(B9) = S
2330 NEXT B9
2340 G = G1
2350 G2 = INT(G/10)
2360 G3 = G-10*G2
2370 REM  FIND MOVE WITH BEST SCORE
2380 J9 = D(1)
2390 R0 = 1
2400 FOR  B0 = 2 TO M1
2410   IF  D(B0) <= J9  THEN 2440
2420   J9 = D(B0)
2430   R0 = B0
2440 NEXT B0
2450 K9 = E(1)
2460 S0 = 1
2470 FOR  C = 2 TO M2
2480 IF  E(C) <= K9  THEN 2510
2490   K9 = E(C)
2500   S0 = C
2510 NEXT C
2515 REM  PRINT COMPUTER MOVE
2520 Y(I) = 0
2530 IF  K9 > J9  THEN 2640
2540 F = R(R0)
2545 F1 = F
2550 F2 = INT(F/10)
2560 F3 = F-10*F2
2570 IF  F(R0) = 10  THEN 2610
2580 PRINT T$(F);
2590 V$(I) = T$(F)
2600 GO TO 240
2610 PRINTT$(F);"CHECK";
2620 Y(I) = 1
2630 GO TO 2590
2640 G = K(S0)
2645 G1 = G
2650 G2 = INT(G/10)
2660 G3 = G-10*G2
2670 GO SUB 3340
2680 IF  L = 0  THEN 2720
2690 PRINT U$(G)
2700 V$(I) = U$(G)
2710 GO TO 240
2720 Y(I) = 1
2730 PRINT U$(G);"CHECK";
2740 GO TO 2700
2742 PRINT T$(F);"CHECKMATE";
2743 V$(I) = T$(F)
2744 W$(I) = "MATE"
2745 GO TO 240
2750 REM  SUBROUTINE FOR SCORING MOVES
2755 E0 = 0
2760 Q = ABS(F-H)
2770 R = ABS(F-G)
2780 IF  (Q-1)*(Q-9)*(Q-10)*(Q-11) <> 0  THEN 2820
2790 IF (R-1)*(R-9)*(R-10)*(R-11) = 0  THEN 2820
2800 S = -1000
2810 GO TO 3230
2820 S = 0
2830 FOR  C1 = 0 TO 7
2840   H = H1+Z(C1)
2850   H2 = INT(H/10)
2860   H3 = H-10*H2
2870   GO SUB 3270
2880   IF  L = 0  THEN 2900
2890   S = S+1
2900 NEXT C1
2910 H = H1
2920 H2 = INT(H/10)
2930 H3 = H-10*H2
2940 IF  S > 0  THEN 3000
2950 IF  I5 = 1  THEN 2800
2960 IF  F(B8) = 10  THEN 2980
2970 GO TO 2900
2980 E0 = 1
2990 GO TO 3230
3000 S  = X(0)*(8-S)
3010 S = S+X(1)*(14-ABS(G3-H3)-ABS(G2-H2))
3020 S = S+X(2)*(14-ABS(F2-H2)-ABS(F3-H3))
3030 S = S+X(3)*(14-ABS(F2-G2)-ABS(F3-G3))
3040 IF  (F2-1)*(F2-8)*(F3-1)*(F3-8) = 0  THEN 3070
3050 S = S+X(4)
3060 GO TO 3075
3070 S = S+X(5)
3075 IF (G-32)*(G-23)*(G-26)*(G-37) = 0  THEN 3100
3076 IF (G-67)*(G-76)*(G-73)*(G-62) = 0  THEN 3100
3080 IF (G2-1)*(G2-2)*(G2-8)*(G2-7) = 0  THEN 3120
3090 IF (G3-1)*(G3-2)*(G3-8)*(G3-7) = 0  THEN 3120
3100 S = S+X(6)
3110 GO TO 3130
3120 S = S+X(7)
3130 Y0 = ABS(F3-H3)
3140 W = ABS(F2-H2)
3150 IF  Y0 >= W  THEN 3170
3160 Y0 = W
3170 X0 = ABS(F2-G2)
3180 W1 = ABS(F3-G3)
3190 IF  X0 >= W1  THEN 3210
3200 X0 = W1
3210 X0 = Y0/X0
3220 S = S+X(8)*X0*(2.5-X0)
3230 RETURN
3240 REM  SUBROUTINE FOR CHECKING LEGALITY
3250 IF  (1+SGN(F2-1))*(1-SGN(F2-8)) = 0  THEN 3470
3260 IF  (1+SGN(F3-1))*(1-SGN(F3-8)) = 0  THEN 3470
3270 IF  (1+SGN(G2-1))*(1-SGN(G2-8)) = 0  THEN 3470
3280 IF  (1+SGN(G3-1))*(1-SGN(G3-8)) = 0  THEN 3470
3290 IF  (1+SGN(H2-1))*(1-SGN(H2-8)) = 0  THEN 3470
3300 IF  (1+SGN(H3-1))*(1-SGN(H3-8)) = 0  THEN 3470
3310 IF (F-G)*(G-H)*(F-H) = 0  THEN 3470
3320 M = ABS(G-H)
3330 IF  (M-1)*(M-9)*(M-10)*(M-11) = 0  THEN 3470
3340 IF  F2 = H2  THEN 3400
3350 IF  F3 <> H3  THEN 3380
3360 IF  F3 = G3  THEN 3400
3370 GO TO 3470
3380 L = 1
3390 GO TO 3480
3400 IF  F > H  THEN 3440
3410 IF  G < F  THEN 3470
3420 IF  G > H  THEN 3470
3430 GO TO 3380
3440 IF  G > F  THEN 3470
3450 IF G < H  THEN 3470
3460 GO TO 3380
3470 L = 0
3480 RETURN
3490 DATA 1,-1,10,-10,9,-9,11,-11
3500 DATA 5,20.1,1.7,7,2,1,14,1,10
3510 DATA 9,11.5,3,11,1.1,1,6.6,1,102
3520 DATA 5,5,5.9,-2,1,16.1,2,1,50
3530 REM  SUBROUTINE FOR LEGALITY OF KING MOVES
3535 IF  F = G  THEN 3600
3540 IF  (1+SGN(G2-1))*(1-SGN(G2-8)) = 0  THEN 3600
3550 IF  (1+SGN(G3-1))*(1-SGN(G3-8)) = 0  THEN 3600
3560 M = ABS(G-H)
3570 IF  (M-1)*(M-10)*(M-9)*(M-11) = 0  THEN 3600
3580 L = 1
3590 GO TO 3610
3600 L = 0
3610 RETURN
3620 F = 10*F2+H3
3621 F2 = INT(F/10)
3622 F3 = F-10*F2
3623 F1 = F
3630 IF  (H3-8)*(H3-1) = 0  THEN 2742
3640 GO TO 2610
3650 F = 10*H2+F3
3651 F2 = INT(F/10)
3652 F3 = F-10*F2
3653 F1 = F
3660 IF  (H2-1)*(H2-8) = 0  THEN 2742
3670 GO TO 2610
9999 END
```

ERROR MESSAGE

In my article "Drawpic" in the February issue of PCW, there are some errors:

Here they are:—

Line 5000 (C61) should be (C—1)

Line 8000 a greater than sign is missing i.e.

:IF (F(N)+1) > 127 THEN 8020

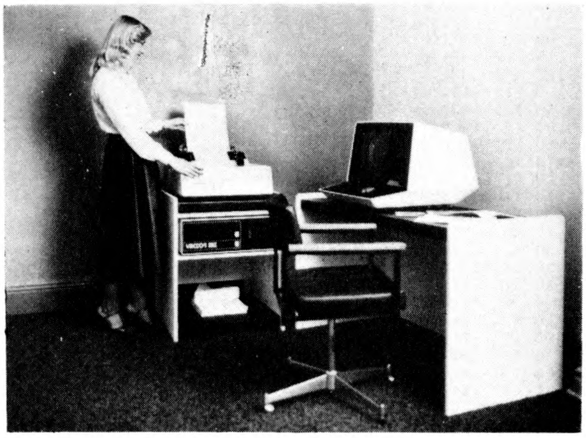
Line 1030 IF A$="L" THEN X=X—1: SET(X,Y):
RESET (X+1,Y):
IF Q=2 THEN SET (X+1,Y)

A. O. Ellefsen,
121 The Furlongs, Ingatestone, Essex CM4 0AL

Need A Business Computer?

then you need a VECTOR GRAPHIC MZ based system from ALMARC

why?



ADS — 100 Business system with Centronics 779 printer: £3855.00 + VAT. Delivered & Installed.

Leasing terms available — under £30.00 p.w.

Software prices on application.

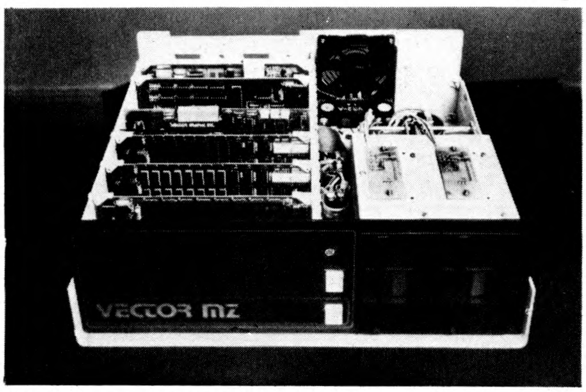
- **BECAUSE –**
  The MZ gives enough storage to make your system workable – at an unequalled price.
- **BECAUSE –**
  The MZ is easily expanded.
- **BECAUSE –**
  Almarc understand YOUR needs. Notice how our systems are packaged to fit your office. The processor is in the printer stand to save space; and these are separated from the V.D.U. to fit into awkward spaces. The V.D.U. desk is at the right height for the typist. *Good ergonomics* – Comfort and Safety.
- **BECAUSE –**
  Almarc understand your problems. If you want to write your own packages then we can give you advice. If you want us to write your system we can do that as well, and at Almarc we realise that your operator will probably be a typist and not a specially trained member of staff. We won't try to sell you packages we haven't got, or botched up U.S.A. packages.
- **BECAUSE–**
  Almarc can offer hardware backup. If you live in the Midlands or London area then Almarc can give you on-site service. Maintenance contracts are available at surprisingly low cost.

Write or ring now! Our address is at the bottom of the page.

Need A Scientific Computer?

then you need a VECTOR GRAPHIC MZ from ALMARC

why?



Vector Graphics MZ & 32K of RAM + two 315K BYTE Disk Drives.

CP/M* Software on the MZ
CBASIC Compiled basic
COBOL Subset of ANSI
FORTRAN Runs in 24K; also has an assembler and a linker.
Plus Macro assemblers, linkers, text editors etc.

- **BECAUSE**
  The MZ uses a fast 4 MHz Z80A processor.
- **BECAUSE**
  The MZ uses the S-100 bus.
- **BECAUSE**
  At £2300.00 the MZ gives you more bytes for your money. Discounts are available too.
- **BECAUSE**
  Almarc give you complete systems. For example "OPT", our own graphics package, has been designed to reside at the end of Basic and works with the HIGH RESOLUTION GRAPHICS BOARD. OPT provides for point, line and shape plotting, and also has an alpha-numeric character generator. All mathematical functions are completely independent of the DOS.
- **BECAUSE**
  Almarc offers a complete range of system software. As well as MDOS, MBASIC, and a Z80 Assembler supplied with the system, CP/M* is also available. Check the list opposite for details.

*Trademark of Digital Research.

the Almarc of quality

ALMARC DATA SYSTEMS LTD.
29 Chesterfield Drive, Burton Joyce, Nottingham. Tel: 0602 248565

STATPACK

Continued

Colin Chatfield

This part produces only one result, that being the Chi-square value of any number of columns of data required and the degrees of freedom.

The module starts at line 5000 and asks for the number of columns Chi-square is required for. Two arrays are then set up at line 5020 followed by a request for the column numbers that are to be looked at in the small loop from 5030 to 5047. The calculation starts at 5050 and the results are printed out at 5170.

The usual opening is between lines 5 and 1240 with sub-routines between lines 9000 and 9690.

It has been suggested that in module 2, the CHAINing can be done by concatenating the word 'STAT' and the STRing value of the numeric variable A. This neater idea unfortunately does not work in MSI BASIC V1.2, BASIC2C or V1.3 but if anybody knows of a BASIC version where it will I would be glad to hear of it. The suggested line alterations are:—

```
450  IF A=0 THEN END
460  CHAIN ("STAT" + STR$(A))
```

The next part of this package, module 7, produces Bar Charts.

```
0005 REM STAT6 - CHI-SQUARE
0020 INPUT "     ENTER PORT # ",Z9
0080 LINE= 132: GOSUB9380
0100 ? TAB(25);"CHI-SQUARE":?
1000 GOSUB 9600
1010 ? "YOUR ARRAY IS ";A;"X ";B;:?CHR$(8);:?". ";A*B;"ITEMS."
1110 INPUT "     CARRIAGE RETURN WHEN READY",A$
1120 GOSUB 5000
1200 GOSUB 9360
1210 IF LEFT$(A$,1)="N"THEN1240
1220 IF LEFT$(A$,1)<>"Y"THEN1200
1230 INPUT "     IF CHI-SQUARE INPUT `Y´ ",A$
1232 IF A$="Y"THEN1120
1234 CHAIN STAT1
1240 ? TAB(20);"STATPACK END": END
5000 GOSUB 9380:?#(Z9):?#(Z9),TAB(25);"CHI-SQUARE":?#(Z9)
5005 X=X+1:IFX>1 THEN5030
5010 INPUT "     ENTER NUMBER OF COLUMNS CHI-SQUARE REQUIRED FOR",M
5015 IF M>B THEN5010
5020 DIM P(M),R(M)
5030 ? "ENTER COLUMN NUMBERS EACH ONE FOLLOWED BY RETURN"
5040 FOR I=1TOM
5045 INPUT "     # ",P1:IFP1<1 THEN?" RE-ENTER":GOTO5045
5046 IF P1>B THEN?"  RE-ENTER":GOTO5045
5047 P(I)=P1:NEXTI
5050 FOR I=1TOM: S=0
5055 I1=I:I=P(I)
5060 FOR J=1TOA:S=S+C(J,I):NEXTJ
5070 I=I1:R(I)=S:NEXTI
5080 S1=0
5090 FOR J=1TOA:S=0
5100 FOR I=1TOM:I=P(I):S=S+C(J,I):NEXTI
5110 C(J)=S:S1=S1+S:NEXTJ
5120 ? :C=0
5130 FOR I=1TOM:I1=I:I=P(I):FORJ=1TOA
5140 I=I1:E=R(I)*C(J)/S1
5145 C=C+((C(J,I)-E)*(C(J,I)-E))/E
5150 NEXT J
5160 L=0: NEXTI:?#(Z9)
5170 ? #(Z9),"CHI-SQUARE = ";C;"ON ";(M-1)*(A-1);"DEGREES OF FREEDOM"
5390 ? #(Z9):RETURN
9000 REM SUB PROGRAMS
9300 FOR K=1TO5:?#(Z9):NEXTK:RETURN
9360 INPUT "     ENTER `Y´ FOR MORE, `N´ FOR NONE ",A$:RETURN
9380 ? CHR$(25);:?CHR$(25);:?CHR$(22);
9385 ? CHR$(12);:RETURN
9400 IF B=1 THENB2=1:GOTO9430
9410 ? :INPUT"     COLUMN # STATISTICS REQUIRED FOR ",B2
9420 IF B2>B THEN?"TOO HIGH";:GOTO9410
9430 ? #(Z9):RETURN
9600 OPEN #10, STATFL1 FOR INPUT
9610 OPEN #20,STATFL2 FOR INPUT
9620 FIELD #10,F=6
9630 FIELD #20,A=6,B=6
9640 SET #10=1:SET#20=1:GET#20
9650 DIM C(A,B),B3(A)
9660 FOR I=1TOA:FORJ=1TOB:GET#10:C(I,J)=F:NEXTJ:NEXTI
9683 INPUT "     ENTER `Y´ FOR VISUAL OF DATA",A$:IFA$<>"Y"THEN9690
9685 ? :FORI=1TOA:FORJ=1TOB:?C(I,J);:NEXTJ:?:NEXTI:?
9690 CLOSE #10:CLOSE#20:RETURN

MSI READY
#
```

```
CHAINSTAT6
     ENTER PORT # ? 1

                         CHI-SQUARE

     ENTER `Y´ FOR VISUAL OF DATA? Y

3 2 0 0 2 3 1 1 50 5 0 0 0 1 0 0 0 0 0
3 1 0 0 2 4 1 1 100 2 0 2 0 0 0 0 0 1 0
3 1 0 1 0 1 1 1 100 3 3 2 0 1 0 1 0 1 1
3 1 0 1 0 2 1 1 2 1 0 2 0 0 0 1 0 1 0
3 1 0 0 2 3 1 3 2 1 0 0 0 0 1 1 0 0 0
3 1 1 0 0 2 1 1 50 2 0 0 0 0 1 1 0 1 0

YOUR ARRAY IS 6 X 19. 114 ITEMS.
     CARRIAGE RETURN WHEN READY?

                         CHI-SQUARE

     ENTER NUMBER OF COLUMNS CHI-SQUARE REQUIRED FOR? 3
ENTER COLUMN NUMBERS EACH ONE FOLLOWED BY RETURN
     # ? 2
     # ? 6
     # ? 9


CHI-SQUARE = 580.27353 ON 10 DEGREES OF FREEDOM

     ENTER `Y´ FOR MORE, `N´ FOR NONE ? N
                    STATPACK END

THE GHOST
#
```

"Sorry — Just Calculated I Can't Afford It!"

ORDER BY POST FROM THE WIDEST RANGE OF MICRO-COMPUTERS IN BRITAIN... OR VISIT OUR STORES

the BYTE SHOP Ltd

A WIDE VARIETY OF SYSTEMS AND BOOKS AVAILABLE 'OFF THE SHELF'

SOUTHWEST TECHNICAL PRODUCTS — FOR 6800 USERS

MP-68CPU, 4K RAM	£355
8K additional RAM	£140
560KB twin disk system	£1,400
160KB twin mini floppy system	£860
CT-64 V.D.U. upper/lower case, monitor	£455
PR-40 40-column printer	£250
Cassette interface	£100
GT-61 Graphics Terminal	£105

A favourite with schools and colleges — low priced system that can grow.

Attractive low-priced software!

FOR THE PROFESSIONAL - CROMEMCO

Z-2 system: fully assembled	£575
Z-2 system: Kit	£395
Disk system from	£1,425
CS-3 System 3, 32K, dual disk, expandable	£4,175
4 MHZ Single card computer	£345
Bytesaver board with PROM Programmer	£135
16K PROM card	£135
Analogue Interface	£135
Disk — BASIC, FORTRAN, ASSEMBLER, COBOL, etc. etc...	each — £85

A rugged well-engineered system designed for hard use. Expandable to 512K,21 Boards.
Excellent software support CP/M AVAILABLE

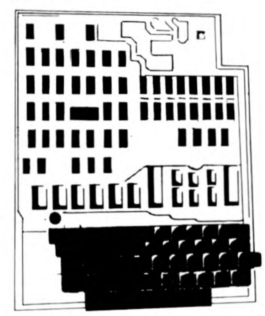
PROCESSOR TECHNOLOGY SOL 20/16

Computer system, 16K, V.D.U. etc.	£1,785
Additional memory, 16K dynamic	£275
Mini-floppy disk (first)	£635
High quality word processing printer from	£1,800

A quality 8080 based system.
Over 7,000 sold! Excellent word processing software available.

Ohio Scientifics

Superboard II



4K user RAM
Full 8K basic
Built and tested

£263.84

COMMODORE SYSTEMS

PET 2001	£550
PET MKII — Full Keyboard Plus Assmblr.	£675
PET Disk Drives	£740
PET Printer	£550
Software from	£3.00

NORTH STAR COMPUTER PRODUCTS

NEW HORIZON-I dual density	£1330
Additional mini floppy	£325
Additional 16K static memory 4MHZ	£315
Printer interface	£70
Extended BASIC and DOS included in above	

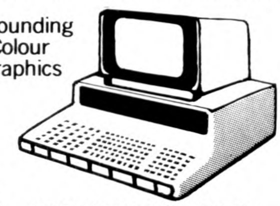
Double Density Disks now available separately CP/M Horizon	£130

NEW SYSTEMS

ITT 2020 EX STOCK

16K RAM integer basic	£950
32K RAM integer basic	£1150
48K RAM integer basic	£1278
DISK II floppy subsystem	£450
EXIDY SOCERER 32K	£850
S100 expansion unit	£250
Disk Drive & Control Card	£650
NASCOM-I kit	£197.50
Buffer Board	£25
32K RAM	£200
Compucolor II	from £1,390

Astounding Colour Graphics



AN AMAZING SYSTEM BUT A COMPUTER IN ITS OWN RIGHT! DISK DRIVE BUILT IN!

MAIL ORDER

Please send me ....
....
....
....

(if preferred, attach order on separate sheet to coupon)

Name ....

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

☐ I enclose cheque for £ ....
☐ Please debit my Diners Club/ American Express/Access/Barclaycard Account no. ....

Please add 8% VAT to all items except books

The Byte Shop Ltd
426-428 Cranbrook Road, Gants Hill, Ilford Essex. Tel: (01) 554 2177
All prices correct at time of going to Press

PCW 5

VISIT US AT: LONDON — 45 TOTTENHAM COURT ROAD. (01) 636 0647
NOTTINGHAM — 92A UPPER PARLIAMENT ST. 0602-40567
OPENING SHORTLY AT: BIRMINGHAM, LEEDS, MANCHESTER, GLASGOW.

MORTGAGES

Can You Afford One?

John D. Lee *(Dept. of Chemistry, University of Technology, Loughborough)* and Timothy D. Lee

The most expensive single item that most people purchase is a house. Almost inevitably this necessitates obtaining a mortgage. It is prudent to analyse the financial details such as the purchase price and the monthly repayments required before embarking on expenditure on this scale.

Most people can remember the formula for simple interest:

$$\text{simple interest} = \frac{\text{principal x rate x time}}{100}$$

Many hire purchase companies calculate interest charges based on this formula, and it should be noted that interest is paid for the total life of the loan on the full amount borrowed rather than on the outstanding debt. This is a very expensive way to borrow money. However, building societies use compound interest in which the interest paid is calculated at monthly intervals based on the outstanding debt at that time. Normally repayments to a building society are made monthly, and the interest charges are also calculated monthly. The first repayment is normally one month after the date the loan commenced.

Compound interest may be calculated using the simple interest formula for one month, then adjusting the principal (now equal to the outstanding debt), and repeating in a similar manner for the appropriate number of months. This approach is extremely tedious if a time period of 20 or 25 years is involved since this would involve 240 or 300 cycles. If one assumes monthly repayments P, the equation for repayment of a mortgage using compound interest is ....

$$P = \frac{Ar\,(1 \times r)^n}{(1 + r)^n - 1}$$

where A is the total amount borrowed, n is the number of repayments and r is the interest rate per month as a fraction rather than a percentage.

Rearrangement of this equation easily yields A, the amount borrowed. Rearrangement to obtain n, the number of repayments, is more difficult, since n occurs twice in the equation, but this can be achieved with the use of logarithms. It is not possible to rearrange the equation to give r. The program described is written in a simple subset of BASIC, which should be easy to implement on most computers. It calculates P, A and n as outlined above, and will perform an iterative approximation to evaluate r should this be required. The computer will perform these calculations speedily, painlessly and reliably, and it prints the results for future reference.

The program MORTGAGE ANALYSIS first asks the user whether he wishes to calculate (i) the monthly repayment, (ii) the life i.e. number of years for which the mortgage is to run, (iii) the annual rate of interest or (iv) the maximum amount which may be borrowed. Having selected which one of these four is to be calculated, the computer then asks for the numerical values of the other three to be typed in one at a time. Helpful messages are printed at each stage, and the input data are checked to ensure that they are physically possible and reasonably likely. The annual rate of interest must be greater than 0 and less than 50%. The life of the mortgage must be at least 1 year and less than 50 years. The amount borrowed and the monthly repayment must be greater than zero!

Should unacceptable data be typed, it is rejected, and a message is typed instructing the user to re-input an acceptable value. From the three values typed in, the value of the fourth term is calculated, and a table showing all four mortgage terms is printed.

A message is then printed asking the user whether he would like a table and graph showing how his debt decreases with time. If this information is required the user may choose between a table showing the changes year by year, or a more detailed printout showing the changes monthly within each calendar year, and the total changes for that year. The number of years for which a table is required is specified by the user, together with the date on which the loan commenced. (Remember that the first repayment is one month after this date).

The table shows the amount of interest charged, the amount of principal repaid and the outstanding debt, and the graph is printed alongside, giving a pictorial representation of the debt. Examination of the table reveals that in the early stages most of the monthly repayments are used to cover the interest charges and so the outstanding debt decreases only slowly. In the later stages because the debt is much smaller, the interest charges are also much smaller, and thus the outstanding debt decreases much more rapidly.

A few technical points are worthy of mention to facilitate implementation of the program on other computers.

(i) The output uses 72 characters across one line.

(ii) The symbol ** is used for raising to the power, rather than the upward arrow ↑ or circumflex Λ

(iii) The largest integer which could be stored in the computer used to develop this program was 32K.

To prevent an overflow when using the function INT, checks are built into the program (lines 630, 670, 690). On other computers the maximum permissible integer may differ from this, in which case the value in these lines should be changed accordingly.

```
10 DIM Q$(10)
20 PRINT TAB(7); "MORTGAGE ANALYSIS"
30 PRINT TAB(7); "======== ========"
40 PRINT
50 LET M = 0
60 PRINT "TYPE IN ONE OF THE FOLLOWING:"
70 PRINT "PAYMENT", "LIFE", "RATE", "LOAN"
80 PRINT "TO CALCULATE THE MONTHLY REPAYMENT,"
90 PRINT TAB(13); "THE LIFE OF THE LOAN,"
100 PRINT TAB(13); "THE ANNUAL RATE OF INTEREST"
110 PRINT TAB(10); "OR THE AMOUNT BORROWED."
120 PRINT "THEN PRESS RETURN."
130 INPUT Q$
140 PRINT
150 IF Q$ = "PAYMENT" THEN 210
160 IF Q$ = "LIFE" THEN 210
170 IF Q$ = "RATE" THEN 290
180 IF Q$ = "LOAN" THEN 210
190 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD.  RETYPE CORRECTLY."
200 GOTO  130
210 PRINT "TYPE THE ANNUAL RATE OF INTEREST IN PERCENT";
220 INPUT R
230 IF R > 50 THEN 250
240 IF R > 0 THEN 270
250 PRINT "INTEREST RATE UNLIKELY - RETYPE CORRECTLY"
260 GOTO  220
270 LET R = R / (12 * 100)
280 IF Q$ = "LIFE" THEN 420
290 PRINT "TYPE THE LIFE OF MORTGAGE IN YEARS";
300 INPUT Y
310 IF Y < 0 THEN 380
320 IF Y < 1 THEN 360
330 IF Y < 50 THEN 400
340 PRINT "MORTGAGE LIFE TOO LONG - YOU WILL NOT LIVE TO PAY IT OFF !"
350 GOTO  290
360 PRINT "MORTGAGE LIFE TOO SHORT - GET A LOAN FROM THE BANK & RETYPE"
370 GOTO  300
380 PRINT "IMPOSSIBLE MORTGAGE LIFE - RETYPE"
390 GOTO  300
400 LET N = 12 * Y
410 IF Q$ = "LOAN" THEN 480
420 PRINT "TYPE IN THE AMOUNT TO BE BORROWED IN POUNDS";
430 INPUT A
440 IF A > 0 THEN 470
450 PRINT "DONT BE SILLY - RETYPE CORRECTLY"
460 GOTO  430
470 IF Q$ = "PAYMENT" THEN 530
480 PRINT "TYPE IN THE AMOUNT OF ONE MONTHLY PAYMENT IN POUNDS";
490 INPUT P
500 IF P > 0 THEN 530
510 PRINT "VERY CLEVER!  RETYPE CORRECTLY."
520 GOTO  490
530 PRINT
540 IF Q$ = "RATE" THEN 720
550 IF Q$ = "LOAN" THEN 620
560 IF Q$ = "PAYMENT" THEN 660
570 IF A * R > P THEN 840
580 LET N = INT(-LOG(1 - (A * R) / P) / LOG(1 + R)) + 1
590 LET Y = INT(N / 12)
600 LET M = N - 12 * Y
610 GOTO  820
620 LET A = P * (1 - 1 / ((1 + R) ** N)) / R
630 IF A > 32000 THEN 820
640 LET A = INT(A + .5)
650 GOTO  820
660 LET P = A * R * (1 + R) ** N / ((1 + R) ** N - 1)
670 IF P > 3200 THEN 820
680 LET P = INT(10 * P + .5) / 10
690 IF P > 320 THEN 820
700 LET P = INT(P * 100 + .5) / 100
710 GOTO  820
720 LET R = (P * N / A - 1) / 12
730 LET R1 = R
740 LET C = P * (1 / (R / ((1 + R) ** N - 1) + R))
750 IF ABS(C - A) < .01 THEN 820
760 LET R1 = R1 / 2
770 IF C < A THEN 800
780 LET R = R + R1
790 GOTO  740
800 LET R = R - R1
810 GOTO  740
820 PRINT
830 IF P > (R * A) THEN 880
840 PRINT "YOUR FIRST YEARS'S PAYMENTS ARE"; 12 * P
850 PRINT "THE FIRST YEARS'S INTEREST IS"; R * A * 12
860 PRINT "THEREFORE, THE MORTGAGE WILL NEVER BE PAID OFF!"
870 GOTO  1750
880 PRINT TAB(10); "MORTGAGE TERMS"
890 PRINT TAB(10); "-------- -----"
900 PRINT
910 PRINT "LIFE OF MORTGAGE   ="; Y; " YEARS,"; M; " MONTH(S)"
920 PRINT
930 PRINT "MONTHLY REPAYMENT  ="; P; "POUNDS"
940 PRINT
950 PRINT "RATE OF INTEREST   ="; R * (12 * 100); " % PER YEAR"
960 PRINT
970 PRINT "AMOUNT BORROWED    ="; A; " POUNDS"
980 PRINT
990 IF Q$ <> "LIFE" THEN 1010
1000 PRINT "LIFE OF MORTGAGE HAS BEEN ROUNDED UP TO THE NEAREST MONTH"
1010 PRINT
1020 PRINT "WOULD YOU LIKE TO SEE HOW YOUR DEBT DECREASES (YES/NO)"
1030 INPUT Q$
1040 IF Q$ = "YES" THEN 1080
1050 IF Q$ = "NO" THEN 1760
1060 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD. PLEASE ANSWER YES OR NO"
1070 GOTO  1030
1080 PRINT "TYPE DATE LOAN COMMENCED (MONTH,YEAR) EG. 1,1979";
1090 INPUT T1, T2
1100 PRINT "TYPE THE TABLE LENGTH REQUIRED IN YEARS";
1110 INPUT T3
1120 IF T3 = INT(T3) THEN 1150
1130 PRINT "RETYPE A WHOLE NUMBER OF YEARS."
1140 GOTO  1110
1150 PRINT "TYPE YEAR OR MONTH TO GIVE INTERVAL REQUIRED IN SUMMARY";
1160 INPUT Q$
1170 IF Q$ = "YEAR" THEN 1210
1180 IF Q$ = "MONTH" THEN 1210
1190 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD. TYPE YEAR OR MONTH"
1200 GOTO  1160
1210 PRINT
1220 PRINT TAB(10); "MORTGAGE TABLE"
1230 PRINT TAB(10); "-------- -----"
1240 LET S1 = 0
1250 LET S2 = 0
1260 LET A1 = A
1270 LET T2 = T2 + INT(T1 / 12)
```

```
1280 LET M2 = T1 - INT(T1 / 12) * 12
1290 LET M3 = M2 + 1
1300 PRINT TAB(22); "PRINCIPAL"; TAB(35); "OUTSTANDING"; TAB(57);"GRAPH"
1310 PRINT Q$;TAB(10);"INTEREST    REPAYMENT    DEBT";TAB(56);"OF DEBT"
1320 PRINT
1330 PRINT TAB(47); "0"; TAB(69); "MAX"
1340 IF Q$ = "YEAR" THEN 1360
1350 PRINT "FOR THE YEAR"; T2;
1360 PRINT TAB(46); "I----+----+----+----+----+"
1370 FOR M1 = M3 TO 12 * T3
1380 LET I1 = INT(A * R * 100 + .5) / 100
1390 IF P < (A + I1) THEN 1420
1400 LET P1 = A
1410 GOTO  1430
1420 LET P1 = P - I1
1430 LET A = A - P1
1440 LET S1 = S1 + I1
1450 LET S2 = S2 + P1
1460 LET M2 = M2 + 1
1470 IF Q$ = "YEAR" THEN 1640
1480 PRINT M2; TAB(10); I1; TAB(22); P1; TAB(35); A; TAB(46); "I";
1490 PRINT TAB(46.5 + 25 * A / A1); "*"
1500 IF M2 = 12 THEN 1520
1510 IF A > 0 THEN 1740
1520 PRINT
1530 PRINT "PRINCIPAL REPAID DURING"; T2; S2
1540 PRINT "INTEREST PAID DURING    "; T2, S1
1550 PRINT "PRINCIPAL OUTSTANDING AT YEAR END ="; A
1560 IF A <= 0 THEN 1750
1570 LET T2 = T2 + 1
1580 PRINT
1590 PRINT
1600 IF M1 = 12 * T3 THEN 1740
1610 PRINT TAB(47); "0"; TAB(69); "MAX"
1620 PRINT "FOR THE YEAR"; T2; TAB(46); "I----+----+----+----+----+"
1630 GOTO  1710
1640 IF M2 = 12 THEN 1660
1650 IF A > 0 THEN 1740
1660 PRINT T2; TAB(10); S1; TAB(22); S2; TAB(35); A; TAB(46); "I";
1670 PRINT TAB(46.5 + 25 * A / A1); "*"
1680 LET T2 = T2 + 1
1690 IF M1 = 12 * T3 THEN 1750
1700 IF A <= 0 THEN 1750
1710 LET S1 = 0
1720 LET S2 = 0
1730 LET M2 = 0
1740 NEXT M1
1750 PRINT
1760 PRINT
1770 PRINT "WOULD YOU LIKE ANOTHER GO? (YES/NO)"
1780 INPUT Q$
1790 IF Q$ = "YES" THEN 40
1800 IF Q$ = "NO" THEN 1830
1810 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD.  RETYPE CORRECTLY."
1820 GOTO  1780
1830 PRINT "END OF JOB"
1840 END
```

```
          MORTGAGE ANALYSIS
          ======== ========

TYPE IN ONE OF THE FOLLOWING:
PAYMENT        LIFE           RATE           LOAN
TO CALCULATE THE MONTHLY REPAYMENT,
             THE LIFE OF THE LOAN,
             THE ANNUAL RATE OF INTEREST
          OR THE AMOUNT BORROWED.
THEN PRESS RETURN.
? LOAN
TYPE THE ANNUAL RATE OF INTEREST IN PERCENT ? 8
TYPE THE LIFE OF MORTGAGE IN YEARS ? 25
TYPE IN THE AMOUNT OF ONE MONTHLY PAYMENT IN POUNDS ? 50


          MORTGAGE TERMS
          -------- -----

LIFE OF MORTGAGE   = 25  YEARS, 0  MONTH(S)

MONTHLY REPAYMENT  = 50 POUNDS

RATE OF INTEREST   = 8  % PER YEAR

AMOUNT BORROWED    = 6478  POUNDS


WOULD YOU LIKE TO SEE HOW YOUR DEBT DECREASES (YES/NO)
? YES
TYPE DATE LOAN COMMENCED (MONTH,YEAR) EG. 1,1979 ? 1,1978
TYPE THE TABLE LENGTH REQUIRED IN YEARS ? 25
TYPE YEAR OR MONTH TO GIVE INTERVAL REQUIRED IN SUMMARY ? YEAR

          MORTGAGE TABLE
          -------- -----
                      PRINCIPAL    OUTSTANDING           GRAPH
YEAR      INTEREST    REPAYMENT    DEBT                 OF DEBT

                                               0                     MAX
                                              I----+----+----+----+----+
 1978      472.51     77.49         6400.51   I                        *
 1979      508.73     91.27         6309.24   I                        *
 1980      501.17     98.83         6210.41   I                       *
 1981      492.95     107.05        6103.36   I                       *
 1982      484.08     115.92        5987.44   I                       *
 1983      474.46     125.54        5861.9    I                      *
 1984      464.05     135.95        5725.95   I                      *
 1985      452.73     147.27        5578.68   I                     *
 1986      440.53     159.47        5419.21   I                    *
 1987      427.29     172.71        5246.5    I                    *
 1988      412.97     187.03        5059.47   I                   *
 1989      397.44     202.56        4856.91   I                  *
 1990      380.6      219.4         4637.51   I                 *
 1991      362.41     237.59        4399.92   I                *
 1992      342.69     257.31        4142.61   I               *
 1993      321.35     278.65        3863.96   I              *
 1994      298.22     301.78        3562.18   I             *
 1995      273.16     326.84        3235.34   I            *
 1996      246.05     353.95        2881.39   I           *
 1997      216.65     383.35        2498.03   I         *
 1998      184.85     415.15        2082.88   I        *
 1999      150.39     449.61        1633.27   I      *
 2000      113.08     486.92        1146.35   I    *
 2001      72.65      527.35        619.003   I  *
 2002      28.88      571.12        47.8829   I*


WOULD YOU LIKE ANOTHER GO? (YES/NO)
? NO
END OF JOB
```

Situations Vacant

PCW is read by a vast number of the most intelligent and upwardly mobile people in Britain. Advertise your job and staff vacancies here.

CAREER IN MICRO COMPUTERS.

Experienced micro-computer (hardware/software) engineers wanted.

Please write with full details to:-

THE BRISTOL COMPUTER CENTRE
MICRO HOUSE,
ST. MICHAEL'S HILL, BRISTOL.

MICRODIGITAL are expanding their engineering capabilities. Your chance to join the **premier** microcomputer firm. Ring Bruce on 051 236 0707

PROGRAMMERS

Part-time Z80 programmers, with particular reference to the TRS-80 for business applications.
Contact: ROSTRONICS COMPUTER CENTRE, 118 WANDSWORTH HIGH STREET, LONDON SW18.
Tel: 01-870 4805

Secretary required for a busy computer centre in Wandsworth. Accurate typing, a knowledge of book-keeping, the ability to organize and an interest in microcomputers are the key requirements for this interesting and challenging position. We are offering an excellent salary to a suitable applicant.

CONTACT:
Mrs Battersby
Rostronics
118 Wandsworth High Street,
London SW18 Phone: 01-870 4805

THE MICRO ENVIRONMENT

If you are interested in getting involved with micro's and have relevant background experience in electronics and/or computing then get in touch with us.

We are looking for a dynamic, go-ahead person who is adaptable and learns easily. If you possess these necessary qualities we will in return offer generous remuneration, rapid advancement and total involvement in an expanding company.

Audition for an active part in the new revolution. Reply to BOX 39.

Keen Computers Ltd.,

We require:

2 PROGRAMMERS with experience in BASIC	**£3,500 – £4,000**
1 SYSTEMS ANALYST with experience in BASIC and Commercial Systems	**£5,500 – £6,000**
1 ENGINEER to service and maintain: Micro-computers, printers and general	**£3,500 – £4,000**

ARE THESE JOBS FOR YOU?

If so contact:
KEEN COMPUTERS LIMITED,
5 THE POULTRY,
NOTTINGHAM NG1 2HW
(Telephone: 0602 583254)

SINTROM MICROSHOP

COMPUTER SERVICE ENGINEER

An experienced service engineer is urgently required. After initial training, duties will include testing and servicing of Z-80 and 6800 based microcomputers and their peripherals.

Applicants should preferably have a degree or similar qualification and previous experience of digital electronics. They should be adaptable and capable of working with minimum supervision.

Benefits include four weeks holiday and profit sharing.

CONTACT:
Personnel
Sintrom Microshop
14 Arkwright Road, Reading, Berks.
Phone Reading 85464

PCW Vol 1.
BACK NUMBERS

Sold Out – No. 5
Sold Out – No. 4
Just a few left of – Nos. 3, 6, 7, 9.
If you think No. 10 will be available in 3 months' time – don't count on it. Nos. 1, 2, and 8 are still in good supply.

Think of the Future – Look Backwards!

THE RESEARCH MACHINES 380Z COMPUTER SYSTEM

THE RESEARCH MACHINES 380Z
A UNIQUE TOOL FOR RESEARCH AND EDUCATION

Microcomputers are extremely good value. The outright purchase price of a 380Z installation with dual mini floppy disk drives, digital I/O and a real-time clock, is about the same as the annual maintenance cost of a typical laboratory minicomputer. It is worth thinking about!

The **RESEARCH MACHINES 380Z** is an excellent microcomputer for on-line data logging and control. In university departments in general, it is also a very attractive alternative to a central mainframe. Having your own 380Z means an end to fighting the central operating system, immediate feedback of program bugs, no more queuing and a virtually unlimited computing budget. You can program in interactive BASIC or, using our unique Text Editor, run very large programs with a 380Z FORTRAN Compiler. If you already have a minicomputer, you can use your 380Z with a floppy disk system for data capture.

What about Schools and Colleges? You can purchase a 380Z for your Computer Science or Computer Studies department at about the same cost as a terminal. A 380Z has a performance equal to many minicomputers and is ideal for teaching BASIC and Cesil. For A Level machine language instruction, the 380Z has the best software front panel of any computer. This enables a teacher to single-step through programs and observe the effects on registers and memory, using a single keystroke.

WHAT OTHER FEATURES SET THE 380Z APART?

The 380Z with its professional keyboard is a robust, hardwearing piece of equipment that will endure continual handling for years. It has an integral VDU interface – you only have to plug a black and white television into the system in order to provide a display unit – you do not need to buy a separate terminal. The integral VDU interface gives you upper and *lower* case characters and low resolution graphics. Text and graphics can be mixed *anywhere* on the screen. The 380Z has an integral cassette interface, software and hardware, which uses *named* cassette files for both program and data storage. This means that it is easy to store more than one program per cassette.

Owners of a 380Z microcomputer can upgrade their system to include floppy (standard or mini) disk storage and take full advantage of a unique occurence in the history of computing – the CP/M™* industry standard disk operating system. The 380Z uses an 8080 family microprocessor – the Z80 – and this has enabled us to use CP/M. This means that the 380Z user has access to a growing body of CP/M based software, supplied from many independent sources.

380Z mini floppy disk systems are available with the drives mounted in the computer case itself, presenting a compact and tidy installation. The FDS-2 standard floppy disk system uses double-sided disk drives, providing 1 Megabyte of on-line storage.

**Trademark, Digital Research.*

Versions of BASIC are available with the 380Z which automatically provide controlled cassette data files, allow programs to be loaded from paper tape, mark sense card readers or from a mainframe. A disk BASIC is also available with serial and random access to disk files. Most BASICs are available in erasable ROM which will allow for periodic updating.

If you already have a teletype, the 380Z can use this for hard copy or for paper tape input. Alternatively, you can purchase a low cost 380Z compatible printer for under £300, or choose from a range of higher performance printers.

380Z/32K complete with SINGLE MINI FLOPPY DISK SYSTEM MDS-1

£1787.00

380Z/16K System with Keyboard

£965.00

RESEARCH MACHINES Computer Systems are distributed by RESEARCH MACHINES LTD., P.O. Box 75, Chapel Street, Oxford. Telephone: OXFORD (0865) 49792. Please send for the 380Z Information Leaflet. Prices do not include Carriage or VAT @ 8%.

PCW Book Review

Michael James

MICROPROCESSOR PROGRAMMING FOR COMPUTER HOBBYISTS

By Neil Graham
Published by TAB Books, No. 952. First Edition 1977
Distributed by W. Foulsham & Co. Ltd., Yeovil Road, Slough SL1 4JH ISBN 0-8306-6952-3

This book is a remarkable book in several ways. Firstly, there is no mention of BASIC at all (not even in the useful index). Secondly it is the first book I've seen on computer science for amateurs. When Donald Knuth set out to write a comprehensive treatise covering a similar area he produced a nine volume encyclopaedia. Neil Graham has done well to edit and compress so much into 400 pages with many diagrams, though inevitably the proofs of many performance expressions have had to go.

The algorithms are presented in a pseudo-language which is rather like PL/I. This is rather more useful in general than any particular assembly language; it translates easily into an assembly language (even that for the TMS9900, a 16 bit micro) and can be readily understood. System software (e.g. compilers, editors and operating systems) is not discussed though anyone who sets out to write any without at least the wisdom in this book is likely to learn a lot the hard way!

Subjects that are dealt with are dealt with in sufficient detail to enable an efficient implementation of the basic algorithms and data structures, as well as to tempt the reader to refer to one of the excellent references. The author of this review had little difficulty in implementing an efficient floating point package for the TMS9900 in less than 256 words in under a week. The contents of the book are given in the form below. The book can be recommended to anyone wanting to write better programs.

Bill Davy

BUZZWORDS

Peter Reynolds

Fail-Safe. An arrangement in computer hardware or software designed to avoid catastrophe but possible at the expense of convenience. For example, when a fault was detected in a computer controlling traffic lights a fail-safe arrangement might set all the lights to red rather than turn them off. Similarly, in a commercial system, overheating might simply disconnect the power supply.

Fail-Soft. A more convenient variation of the *fail-safe* protection described above. To take the same two examples, the traffic lights might turn to flashing amber rather than red and the overheated system might maintain battery power for volatile memories while the main source of power was unexpectedly turned off. Roughly synonymous with *'graceful degradation'*.

Fall-Back. Arrangements brought into use in emergency, especially the reserve data files that would be switched in quickly, or even automatically, in the event of a detected fault in a real-time system.

Father-Son. A system of updating records which retains a copy of the original record as well as providing an amended version. The technique is particularly applicable to files held on magnetic media such as tape or disc. For example, payroll details with gross pay up to week 39 might be held on Tape A (the 'Father' tape) and this could be run with details of pay for the following week to provide a 'Son' Tape B, holding details of cumulative gross pay to week 40. The purpose of the Father-Son arrangement is that it provides opportunity both to test the accuracy of the new record (for example, the hash total of pay data on Tape B should equal the total of the hash totals of Tape A and the data for week 40) and to reconstitute the new record if any error should come to light. On the next updating occasion (week 41 in the payroll instance) the 'Son' tape would become a 'Father' tape and the original Father tape might be cleaned or over-written.

Fax. A system for photo-electronically scanning a document, transmitting it as data by telephone line and reproducing an image of the document on a distant instrument.

Feed. The mechanical process whereby lengthy materials, for example, paper or magnetic tape, printing paper, printer ribbon, are moved along the required operating position.

Feedback. An arrangement whereby information is brought back to an earlier stage in a computer program or operating system. A physical example is found in the feedback from thermostat to heat source. In data processing information arising from a particular stage of processing could provide a feedback to affect the processing of subsequent data — for example, the fact that an area of storage was nearly full might either delay the acceptance of more data or divert it to some other storage area.

Female (connector). The socket half of a plug and socket pair or coupling.

Ferrite Core. Once the standard medium for computer memory, ferrite core has been displaced by less expensive, mass produced, transistor memory chips. The *core* is a form of *bistable* magnetic data storage comprising matrices of small "doughnuts" made of a magnetisable material called ferrite. These ring-like elements are threaded on fine wires which are strung as it were from north to south and from east to west, so that an element is held at every intersection of the wires with both wires cutting its axis. Ferrite core memory has the advantage of being non-volatile: the data pattern remaining when power is removed.

FET. Field Effect Transistor.

Fetch (instruction). A processor instruction to bring the next word of data from, say, an input buffer store to the accumulator where it can be processed.

FF. Form Feed. This may refer to either the physical transport of continuous stationery to the beginning of a new page or to the standard ASCII character to cause a form feed to occur, just as other ASCII characters will generate a line feed or sound a 'bell' (in practice, often a bleeper).

Field. A specified area of a record used for a particular purpose. For example, the first three columns (the first "field") in a punched card may be used to identify the customer. Similarly a particular set of *bit* locations in a computer word may be specified to identify the address of the operand.

FIFO. First In, First Out. Descriptive of a storage protocol (for goods in a warehouse and for computer data alike) where that which came into the store first is the next to be removed. Contrast with LIFO.

Figure Shift. A keyboard key (or the code it generates) which signifies that the following characters are to be read as figures (or punctuation) until a letter shift appears in the message. Typically found on a teleprinter whose 5-level code restricts the directly encodable repertoire to 32 characters.

File. In computer context a file is an ordered set of data (or a set capable of being put into order), held for example on punched cards or magnetic tape. A computer file is very similar to a conventional one of papers except that it is designed to be handled automatically and, when a magnetic medium is used, the updating of a record may expunge what was recorded previously.

File Librarian. A computer specialist, typically an *operator* or junior *programmer,* who has responsibility for the safe-keeping of all computer files, for example, program and data files on cards, tape or exchangeable discs. Where duplicates are kept he will ensure that amendments get into all copies; where copies show signs of wear or deterioration he will replace them before they fail.

File Maintenance. The periodic revision of a *file* of data to incorporate changes other than basic *updating* — for example, changes of address or deletion of dead records.

Filter (1). An electronic device to separate certain signals, for example, unwanted interference or possibly the second channel of a data transmission stream of information.

(2). A fibrous or similar mesh through which air is drawn to rid it of dust particles. Magnetic peripheral devices are easily damaged by the dirt particles prevalent in most cities, and it is generally necessary for the atmosphere in a computer room to be filtered down to one or two *microns.* Dust filters quickly become clogged with dirt, and must be renewed or removed for cleaning from time to time. It is important that this should be achieved without letting in dirt to the clean area.

Filler Characters. Characters in data processing whose function is to occupy storage or to fill time, thereby keeping out other characters. For example, a sequential printer on line to a computer and lacking its own buffered storage might require two or three 'rubouts' to be transmitted after 'carriage return' to fill the time while the carriage is moving back and nothing can be printed.

Firmware. Wired-in software routines in a computer, for example a BASIC interpreter in a read-only memory chip.

First Generation. A term applied retrospectively to the early computers which used radio valves before transistors were available.

Fixed Point (Arithmetic). A now-rare system of organising arithmetical registers, for example in a computer or a calculator, which requires the operator to keep track of the position of the decimal point. This was necessary with mechanical calculators and with some of the earlier electronic ones. Contrast *floating point.*

Flag. A coded indicator, typically a single bit at one end of a register, which is set from 0 to 1 by the occurrence of some anticipated event, such as a carry digit being generated, a number going negative or an interrupt signal being received. *Flag* has also been used in the sense of *label,* to identify a particular line in a program as an address to be jumped to.

Flip Flop. A *bistable* device. A flip flop switch may be on or off, but cannot be in any intermediate position. Computer circuitry is generally based on the grouping of numerous flip flop devices; these may be inter-connected, and a pulse of information may set a device into a given position, the next pulse restoring it to the first position but at the same time also sending a pulse to a later device in the chain. Binary addition and subtraction are effected in this way.

Floating Point. Electronic calculations in which the user need not worry about the position of the decimal point, the solution being expressed as five or six significant digits and a multiplier in some power of 10.

Flow Chart. A graphic representation of the more important steps of work in a process, using symbols to represent machines or human action. The emphasis is on where and by whom work is done rather on how it is done. A flow chart normally forms part of any systems analysis preparatory to the design of a computer program.

Format. The way in which data is presented for input or writing, taking account of the numbers of digits to be printed, including punctuation marks, etc., and of means of expression — for example, format may require a date to be expressed in the form 09 03 79 rather than 9 III 79. It is generally important to follow the format of significant examples in program writing and in data input.

Floppy (Disc). Generally refers to the larger (8 inch diameter) size of non-rigid disc used for data and program storage. The 5 inch disc is distinguished by the name *mini-floppy* but the term *diskette* may be applied to either size.

FORTRAN. A long-established computer programming language, particularly designed for scientific applications. The name derives from FORmula TRANslation. As with other programming languages, various dialects become established after a few years: thus both FORTRAN IV and FORTRAN 77 may be found.

Form Feed. — see explanation under FF.

Frequency. Speed, conventionally expressed in complete cycles per second (usually termed *Hertz* or *Hz*). Frequencies are designated in distinct bands as follows:

VLF (very low frequency) . . . . . Below 30 KHz (0.03 MHz)
LF (low frequency) . . . 30 to 300 KHz (0.03 to 0.3 MHz)
MF (medium frequency) . . . . . 300 to 3 000 KHz (0.3 to 3 MHz)
HF (high frequency) . . . . 3 000 to 30 000 KHz (3 to 30 MHz)
VHF (very high frequency) . . . . . 30 to 300 MHz
UHF (ultrahigh frequency) . . . . 300 to 3 000 MHz
SHF (superhigh frequency) . . . 3 000 to 30 000 MHz
FHF (extremely high frequency) . 30 000 to 300 000 MHz

Front Feed. Arrangement in a typewriter or similar machine to allow stationery for typing upon to be inserted from the front. In this way successive entries can each appear on separate ledger cards as appropriate while a continuous proof record is built up on continuous stationery fed normally from the rear.

Front Panel. The collection of switches and indicators (or the place where they are located) whereby the operator may control a computer. In many modern machines the functions formerly served by separate rows of lights and switches have now been incorporated in the normal keyboard and display.

Full Adder. A binary adder which includes provision for recording a carry digit, when one arises.

Full Duplex. Same as *duplex* but contrasting with *half-duplex.*

Function. A specific purpose of an entity or its characteristic action, for example, to add two quantities or to test the contents of a store against some pre-determined value.

A Program for the Z80 for producing hard copy on a Creed 7B page printer

R. J. Chance

This program and its predecessor have been used by me for about a year with a Creed 7B and RML 380Z microcomputer to produce hard copy from machine code, Basic and Text editing programs. It may be of interest especially as the decoding from ASCII is done with soft rather than hardware as described previously in PCW.

It is in the form of a subroutine which expects to find an ASCII character for printing in Register A. One bit of an 8 bit port is used as the serial output which is amplified to provide the drive for the Creed directly.

Notable features are as follows:

1) It will run anywhere in memory and thus can be incorporated into, for instance, a Basic interpreter.

2) It is promable and uses no dedicated RAM except that which should be provided by the system.

3) Recoding of selected non Creed characters to suitable printable ones for the particular application is easy.

Introduction:

The routine uses the following features of the 380Z which could be altered to suit most Z80 based machines.

A jump to the execution start known as the line printer vector (LPV) at address 401C. This must be initialised to point to CREOUT: and in this case will be

C3 D4 6C

Bit 0 of the lower address byte (401D) is used to store upper/lower case condition of the 7B. So the NOP at line 40 must be at an even address. It is then immaterial whether the LPV points to this instruction or the following one. This feature is useful in a PROM version when it avoids allocating RAM for this.

A memory mapped port known as UPORT at address 0FFF.

An address in RAM known as UMASK which is used to store the current state of UPORT. This is vital if less than all the bits of a port are to be changed.

How it Works

The conversion from ASCII to Creed is accomplished by counting down a list of 64 characters looking for a match. If found, the Count = the equivalent Creed code. This list (lines 22-39) has lower case characters first, then upper, so that upper case count 32 higher than lower case i.e. bit 5 indicates case. Unwanted positions are filled with NULL; see lines 45-49.

If no match then a recode list of unprintable ASCII characters is searched. This time the desired Creed character is found 'RECNO' lower in memory. See lines 50-55.

Creed List = Line 16; ASCII List = Line 19. Bit 5 is tested for UP/LO case and a start bit (0) and 2 stop bits (1) inserted to complete the 8 bits to be transmitted. (Lines 55-59).

Bit 0 of LPV+1 is updated to the case of the character and register E loaded with the correct case code. (Lines 60-66). If the desired case is the same as the one indicated in LPV+1, the character is loaded to E and output. If not the case character is output first. Bits are output 0 first to bit 7 of UPORT and UMASK.

At 'TIMLP:' HL is loaded with 'TIMCO' which is counted down (lines 80-83) to delay to the correct 50 baud rate. The value given is for 4 MHz with 1 wait state and should be adjusted if other speed Z80 systems are used.

N.B.
Before using, LPV must be initialised to point to CREOUT: (Listing lines 93-101) and UPORT and UMASK set to the resting state (bit 7=1). The listing shown is used with BASIC and a jump to BASIC start comes next. The system may still start off in the wrong 'case' as the state of the printer itself is unknown. Linefeed and carriage return have been put on different cases so that issuing a CR, LF corrects this.

This initialisation is not position independent as the address of 'CREOUT:' must be known although in the 380Z it can be made so.

CREED 7B

```
0000            0001.01 :CREOUT
0000            0002.01 :------
0000            0003.01 :OUTPUTS ASCII CHARACTER IN A TO CREED
0000            0004.01 :7B PAGE PRINTER.
0000            0005.01 :
6C87            0006.01 ORG 6C87H
6C87            0007.01 LPV EQU 401CH
6C87            0008.01 UMASK EQU 4054H
6C87            0009.01 UPORT EQU OFFFH
6C87            0010.01 TIMCO EQU 2580.
6C87            0011.01 RECNO EQU 7
6C87            0012.01 BSTART EQU 4200H
6C87            0013.01 :
6C87            0014.01 :RECODE CHARACTERS FOLLOW
6C87            0015.01 :
6C87            0016.01 :CREED CHARS ARE
6C87 002D012E3A1A12
                0017.01 DEFB 00H,2DH,01H,2EH,3AH,1AH,12H
6C8E            0018.01 :EQUIVALENT ASCII ARE
6C8E 00235E3B2A3E3C
                0019.01 DEFB 00H,23H,5EH,3BH,2AH,3EH,3CH
6C95            0020.01 :
6C95            0021.01 :CHARACTER LIST
6C95 00         0022.01 DEFB 0
6C96 45         0023.01 DEFM 'E'
6C97 0A         0024.01 DEFB 0AH :LF
6C98 4120534955
                0025.01 DEFM 'A SIU'
6C9D 00         0026.01 DEFB 0
6C9E 44524A4E46434B545A4C574859505 14F4247
                0027.01 DEFM 'DRJNFCKTZLWHYPQOBG'
6CB0 00         0028.01 DEFB 0
6CB1 4D5856     0029.01 DEFM 'MXV'
6CB4 0C00       0030.01 DEFB 0,C
6CB6 33         0031.01 DEFM '3'
6CB7 00         0032.01 DEFB 0
6CB8 2D20223837
                0033.01 DEFM '- '87'
6CBD 0D00       0034.01 DEFB 0DH,0
6CBF 34         0035.01 DEFM '4'
6CC0 C7         0036.01 DEFB 07H
6CC1 2C253A28352B29322436303139 3F40
                0037.01 DEFM ',%:(5+)2£6019?@'
6CD0 00         0038.01 DEFB 0
6CD1 2E2F3D     0039.01 DEFM './='
6CD4 00         0040.01 CREOUT: NOP :START HERE
6CD5 E5         0041.01         PUSH HL :OR HERE
6CD6 D5         0042.01         PUSH DE
6CD7 C5         0043.01         PUSH BC
6CD8 F5         0044.01         PUSH AF :SAVE REGS
6CD9 2A1D40     0045.01         LD HL,(LPV+1) :HL=PROG START
6CDC CB85       0046.01         RES 0,L :HL=CREOUT:
6CDE 014000     0047.01         LD BC,64.
6CE1 EDB9       0048.01         CPDR :FIND CHAR
6CE3 2809       0049.01         JR Z,STBI :IF MATCH
6CE5 0E07       0050.01         LD C,RECNO
6CE7 EDB9       0051.01         CPDR :RECODE CHAR?
6CE9 11FAFF     0052.01         LD DE,1-RECNO
6CEC 19         0053.01         ADD HL,DE :(HL)=NEW CHAR
6CED 4E         0054.01         LD C,(HL) :C=NEW CHAR
6CEE 79         0055.01 STBI:   LD A,C
6CEF 87         0056.01         ADD A,A :START BIT AT 0
6CF0 F6C0       0057.01         OR A,0C0H :STOP BITS AT 6,7
6CF2 CB69       0058.01         BIT 5,C :UP/LO CASE?
6CF4 4F         0059.01         LD C,A :COMPLETE CHAR TO C
6CF5 211D40     0060.01         LD HL,LPV+1
6CF8 7E         0061.01         LD A,(HL) :BIT0 (HL) A=CASE BIT
6CF9 1EFE       0062.01         LD E,0FEH :LO CASE CODE
6CFB CBC6       0063.01         SET 0,(HL) :LO CASE BIT
6CFD 2804       0064.01         JR Z,XCASE :IF LO CASE
6CFF 1EF6       0065.01         LD E,0F6H :UP CASE CODE
6D01 CB86       0066.01         RES 0,(HL) :UP CASE BIT
6D03 AE         0067.01 XCASE:  XOR A,(HL)
6D04 2001       0068.01         JR NZ,CHOUT1 :IF CASE CHANGE
6D06 59         0069.01 CHOUT0: LD E,C
6D07 0608       0070.01 CHOUT1: LD B,08 :B=BIT COUNTER
6D09 215440     0071.01 CHOUT2: LD HL,UMASK
6D0C CBBE       0072.01         RES 7,(HL) :CLEAR BIT 7
6D0E CBCB       0073.01         RRC E :BIT 0 TO CY
6D10 9F         0074.01         SBC A,A :CY TO BIT 7,A
6D11 E680       0075.01         AND A,80H :Z BITS 0-6
6D13 B6         0076.01         OR A,(HL) :ADD OTHER UMASK BITS
6D14 77         0077.01         LD (HL),A :UPDATE UMASK
6D15 32FF0F     0078.01         LD (UPORT),A :UPDATE PORT
6D18 21140A     0079.01         LD HL,TIMCO :TIME DELAY
6D1B 2B         0080.01 TIMLP:  DEC HL
6D1C 7C         0081.01         LD A,H
6D1D B5         0082.01         OR A,L
6D1E 20FB       0083.01         JR NZ,TIMLP :BIT TIMED OUT?
6D20 10E7       0084.01         DJNZ CHOUT2 :BYTE FINISHED?
6D22 7B         0085.01         LD A,E
6D23 B9         0086.01         CP A,C
6D24 20E0       0087.01         JR NZ,CHOUT0 :A=C=CHAR FINISHED
6D26 F1         0088.01         POP AF
6D27 C1         0089.01         POP BC
6D28 D1         0090.01         POP DE
6D29 E1         0091.01         POP HL :RESTORE REGS
6D2A C9         0092.01         RET
6D2B 3EC3       0093.01 INIT:   LD A,0C3H :INITIALISE HERE
6D2D 321C40     0094.01         LD (LPV),A
6D30 21D46C     0095.01         LD HL,CREOUT
6D33 221D40     0096.01         LD (LPV+1),HL :LOAD LPV
6D36 215440     0097.01         LD HL,UMASK
6D39 CBFE       0098.01         SET 7,(HL)
6D3B 7E         0099.01         LD A,(HL)
6D3C 32FF0F     0100.01         LD (UPORT),A
6D3F C30042     0101.01         JP BSTART :TO BASIC START
0003            0102.01 END 3

00 ERRORS
401C LPV      4054 UMASK    0FFF UPORT
09C4 TIMCO    0007 RECNO    4200 BSTART
6CD4 CREOUT   6CEE STBI     6D03 XCASE
6D06 CHOUT0   6D07 CHOUT1   6D09 CHOUT2
6D1B TIMLP    6D2B INIT
```

INTERESTED IN HOME COMPUTING?

FREE B BUG valued at £23.00
plus 10 x C12 cassettes valued at £4.00
plus Standard Modulator valued at £2.25
WITH EVERY NASCOM

Start now and don't get left behind THE NASCOM 1 is here Ex-stock with full technical services

Plus the opportunity to join the fastest moving club of personal computer users enabling you to get the most our of your computer. You can OBTAIN and EXCHANGE programs and other software — many now available.

The Powerful Z80 Microprocessor
Professional Keyboard
1 Kbyte Monitor in EPROM
2 Kbyte RAM (expandable)
Audio Cassette interface
Plugs into your domestic TV
Easy construction from straightforward instructions
— no drilling or special tools
— Just neat soldering required.

SAVE £60

Only ~~£197.50~~ + 8% VAT (includes p & p + insurance)

NEW LOW PRICE £165

Manuals seperately	2.95
Z80 programming Manual	6.90
Z80 Technical Manual	2.95
PIO Technical Manual	2.95

(All prices add 8% VAT)

Power supply suitable for NASCOM 19.90

NASCOM AD ONS — Nascom improved monitor B Bug (2K) featuring — *Four times tape speed *Direct text entry without ASCII *Extended keyboard facility *Additional useful subroutines **£23.00**

Nascom Vero Case **£22.50**

Nascom Joy Stick Kit **£14.90**

Nascom Music Box Kit **£9.90**
(write your own tunes and play them on your Nascom. Complete with full documentation).

GRAPHICS ADD ON BOARD £9.90
Complete kit to upgrade your NASCOM for graphics capability includes full documentation and demonstration program.

NASCOM IMMEDIATE EXPANSION S100 from COMP —strongly recommended

The only available S100 motherboard kit (fully buffered) that plugs directly into your Nascom. Designed for the insertion of S100 boards (e.g. Static RAM, EPROM and discs etc.).

S100 Motherboard/Buffer (Complete kit + documentation) **£47.50**
Suitable 8K Static RAM Memory ~~£125~~ £110 (fully assembled tested and guaranteed)

Motherboard comes complete with 2K Tiny Basic On cassette.

ATTENTION! TRS 80 & APPLE II USERS

SAVE £90

Simple to fit — only a screw-driver is required.

16K UP GRADE KIT — HALF NORMAL PRICE **£99** + VAT — LIFETIME GUARANTEE

SHORT C12 CASSETTES 10 for £4.00
FOR COMPUTER PROGRAMMES

TV GAME BARGAINS
TELEPLAY PROGRAMAGAME
Fully Assembled Reduced to **£24.90** + VAT
SAVE £10

ATARI VIDEO SYSTEM
REDUCED TO **£139** plus VAT
SAVE £30

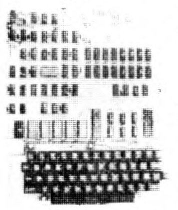
PET COSTS LESS AT COMP and it's a pedigree

RRP £550

The No. 1 Personal Computer in the U.K. Affordable £499 + VAT **SAVE £50**
for the first time user and the professional check out the PET, the world's most popular personal computer. **8K**

OHIO SUPERBOARD II *NEW* FIRST IN UK—ON DEMO

For electronic buffs. Fully assembled and tested. Requires +5V at 3 Amps and a video monitor or TV with RF converter to be up and running.



STANDARD FEATURES
Uses the ultra powerful 6502 microprocessor
8K Microsoft BASIC-in-ROM
Full feature BASIC runs faster than currently available personal computers and all 8080-based business computers.
4K static RAM on board expandable to 8K
Full 53-keyboard with upper/lower case and user programmability
Kansas City standard audio cassette interface for high reliability
Full machine code monitor and I/O utilities in ROM

Direct access video display has 1K of dedicated memory (besides 4K user memory), features upper case, lower case, graphics and gaming characters for an effective screen resolution of up to 256 by 256 points. Normal TV's with overscan display about 24 rows of 24 characters; without overscan up to 30 x 30 characters.

EXTRAS
Available expander board features 24K static RAM (additional), dual mini-floppy interface, port adapter for printer and modem and an OSI 48 line expansion interface.
Assembler/editor and extended machine code monitor available.

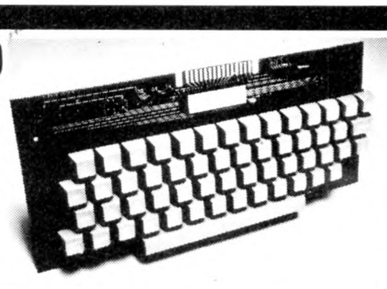
~~£289.00~~ **£249.00** + VAT — *Send £10 to reserve one - pay balance on delivery.*

MODULATORS UHF Channel 36
Standard 6 meg band width **£2.25**
High Quality 8 meg band width £4.90

BULK PURCHASE VDU MONITORS
CASED AND GUARANTEED *12"*
~~£99~~ **£69** + VAT

TRS 80 SOFTWARE *NEW*
100 MIXED PROGRAMMES on cassette **£49.00**

KEY BOARD
756 GEORGE RISK
Brand new professional ASCII keyboards (USA)
Full technical details included. **RRP £60.00**
Only **£49.90** + 8% VAT.
Ready built, tested and guaranteed.



All prices include VAT except where shown. Orders over £5 post and packing free otherwise add 20p. Please make cheques and postal orders payable to COMP, or phone your order quoting BARCLAYCARD or ACCESS nur

OPEN — 10am to 7pm — Monday to Saturday
CONTINUOUS DEMONSTRATIONS

SEND FOR OUR SPRING '79 CATALOGUE - Group

SEND FOR OUR SPRING 79 CATALOGUE

Exidy SORCERER™ COMPUTER SPEAKS YOUR LANGUAGE

THIS COMPUTER HAS ALL OF THE FACILITIES NEEDED TO FORM THE NUCLEUS OF A SMALL BUSINESS SYSTEM FOR AN EXCEPTIONALLY SMALL OUTLAY.

EX STOCK

Are you looking for a PERSONAL or SMALL BUSINESS Computer? Look at these features for only £850 + VAT

- 256 Character Graphics Set includes 128 that you can define.
- Z80 Based computer.
- 32K of RAM on board.
- Power-on 4K Monitor
- Excellent bit-mapped graphics (512 x 240 point resolution)
- Full 79 key dp keyboard **including** 16 key numeric pad (Full ASCII)
- Plug-in ROM facility (e.g. 8K BASIC, (supplied), Assembler/editor, word processing etc.) so that you can change your language or utility by simply plugging in the one you want.
- Serial I/O ports for cassette/RS232 etc. (300 or 1200 Baud)
- Parallel I/O port for printer (Centronics Compatable)
- Z80 bus available at the rear for connection to the optional 6 slot S100 expansion box (e.g. S100 Floppy disk drives, graphics boards, extra memory etc.)

EXIDY SORCERER 32K RAM computer	£850
(Add £9 for UHF output) 16K RAM computer	£751
12" Video monitor	£69
S100 expansion box (includes 6 Slot Motherboard and power supply)	£210

DISK SYSTEM as shown
MICROPOLIS S100 disks — 315 K bytes per drive
Drive 1 (includes S100 controller, Micropolis M DOS and BASIC software) **£649**
Additional 315 K bytes drives (up to 4 per controller) **£349**
Cables and regulators for 2 disk system £41

PRINTER as shown
ANADEX DP8000
80 columns, 112 char/sec 5 x 7 dot matric head
Connects to parallel or serial ports £575

ADDITIONAL SOFTWARE FOR SORCERER PLUS DISK SYSTEM

CP/M operating system	**£145**
CP/M with C BASIC (Compiler/Interpreter)	**£235**
FORTRAN 80	**£350**
COBOL 80	**£490**

MICROPOLIS BASIC PACKAGES FROM COMPUMAX: STOCK CONTROL (INVENTORY), PAYROLL, ACCOUNTS RECEIVABLE, LEDGER, PERSONNEL — **£99 each**

COMP COMPUTER COMPONENTS
(PART OF THE COMPSHOP LTD. GROUP)

For technical and sales literature send SAE to:
COMP COMPUTER COMPONENTS (Exidy UK Sales),
Freepost, 14 Station Road, New Barnet, Herts.
Tel: 01-441 2922 (Sales) 01-449 6596 Telex: 298755

TAKE ADVANTAGE OF OUR SHORT DELIVERY TIME.

All prices exclusive of VAT