2/3

PERSONAL COMPUTER WORLD

50p U.S.$2.00

BEGINNER'S GAMBIT

–

SMALL SYSTEMS ARE GOOD FOR SMALL BUSINESSES

JULY 1979

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

The Programmed Punter

Visionlink

The Soft Cursor

Getting In The Swim

Valerie Downes: Using the Sorcerer

If you buy the wrong personal computer, you can't re-program your bank account!

Buying a personal computer is not an easy task.
So many people selling them neglect the little things that enable you to get the most from your computer, such as documentation, spares, add-ons and maintenance. We believe that these 'details' are essential. We are the only British company to put all our time and energy into the personal computer market and are in the best position to advise you on your initial purchase and keep you fully informed about all the new developments relevant to your computer.
Personal Computers Limited — *the name of the game*.

Say 'hello' to a graphics **Apple II**

A business **Apple II**

Exclusive to Personal Computers Ltd. German **Apple II**. Available shortly.

Personal Computers Limited

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

Apple Software Bank
Free membership to all Apple owners
POST TO: Personal Computers Ltd. 194-200 Bishopsgate, London EC2M 4NR.
NAME ....
ADDRESS ....

PERSONAL COMPUTER WORLD

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

UK 50p
US $ 2.00

Vol. 2 No. 3
July 1979

ISSN: 0142 0232

CONTENTS

Buzzwords by Peter Reynolds, and Statpack by Colin Chatfield, will appear in the next issue.

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

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
Cover Photo: Kaveh Khan

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 with the policy outlined above.

Published monthly by Intra Press, 62A Westbourne Grove, London W2. Phone 01-229 5599. 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.
Printed by Keats Printing Limited, 12 Fouberts Place, London W1V 1HH. Sole UK Distributors Seymour Press Ltd., 334 Brixton Road, London S.W.9., England. Distribution to specialist shops by Intra Press.

Editorial

On Correcting the Menu

I had occasion once to go to a fairly "classy" restaurant. During the course of the meal a diner at the next table began cutting up a waiter with the sharp edge of his tongue. While this went on, I took shelter behind my knife and fork and (not inaptly) ruminated over the incident, thinking how much better it would have been if the customer had complained to the man who should have borne the responsibility for whatever strange animal he'd found in his food – the manager of the restaurant.

This leads me to say that, in general, authors are not to blame for errors found in PCW. Ultimately, responsibility lies with the editor.

Publisher's Letter

The NCC Show in New York, held last June, was a colossal affair. The editor went over by DC10, seated where he could watch the wing as if his life depended on it. The results of his visit will appear in the next issue of our sister publication, Computer Trader.

I would like to thank all our consultants for helping to make PCW the best magazine of its kind in Europe. In particular, it is no exaggeration to say that we would not have got off the ground had it not been for **John Coll, Charles Sweeten, Neil Harrison** and **Mike Dennis**, who were with PCW from its beginning.

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.5.

STOP PRESS! £1,500 CASH – FIRST PRIZE

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.

SPONSORSHIP INVITED FOR 2nd & 3rd PRIZES

Letters

Lost in the bewilderness

I was very interested reading the article published in your April issue by Mr. D.R. Worsley, for he has touched many points which I and some of my business colleagues feel are most vulnerable. It is however our intention to try and remedy this by filling in the gaps, but alas we had no success. Our knowledge is very limited (virtually zero), and yet our enthusiasm is great, and we aim at buying our small computer (micro) to be able to carry out all our tasks of accountancy, i.e: sales ledger and analysis, order entry, invoicing, stock control and recording, payroll, VAT. The problem is and remains our limited knowledge in the size of memory, storage, and type of system (i.e. diskette or cassette) which would be most suited to our needs.

We had thought that a dual floppy disk would be most advantageous for its speed in retrieving information, the storage and also market availability, but which one? It is difficult to say which between: TRS-80 Level II, Pet, Apple II, Exidy Sorcerer, Horizon, Cromemco System II, Challenger C24P. etc. etc. There are so many! There is also another point to consider, that is the money available for the purchase of the microcomputer is limited, at the moment to £2000 – £2500, and this, I know, will limit our potential as buyers. But we would like to look for something which will be possible to expand, when more money is available to us. And also we would like to choose the one where the manufacturer offers good software and hardware backing.

We would like some sound advice on how to go about it all, what to look for what to read and also some advice in developing our own software system (any books on analysis and design of a system for microcomputers).

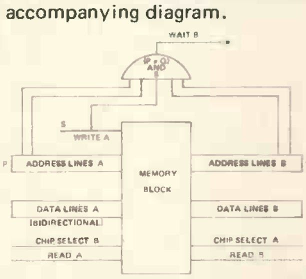
Eagerly waiting for all the advice you could send us.

G. Santini and Friends
61 Victoria Road, Cambridge CB4 3BW.

PCW *An interesting and beautifully written reply to Mr. Worsley's article, written by Mr. Braga of the* Byte Shop, *Nottingham, is published in this issue. Consultant Sheridan Williams will also go some way toward helping Mr. Santini and friends in a reply to another reader's query. The Nottingham Shop is now known as* Computerland.

Dual Access

I wonder if some manufacturer would entertain the idea of making available a memory block which is dual accessible as illustrated in the accompanying diagram.



This is certainly less interfering in the amount of time taken in holding up one system by another compared with DMA or memory mapping.

Dr. S.R. Silvapulle
1 Dale Avenue, Edgware, Middlesex.

SUNBURNT PROMS

I have a free 2708 PROM waiting for the first reader who can tell me how to cure a sunburnt PROM — and advice for readers who intend to build their own PROM laundry.

The advice: incorporate an exposure timer! I didn't and as a result I now have 6 sunburnt PROMs. You may think, as I did, that you couldn't forget to switch off after the ten minutes required to clean out a PROM. But you can fall asleep — at least I did, and as a result my PROMs got a 2 hour dose of hard U.V.

The result was that in all 1204 locations bit 1 is stuck low. All the other 7 bits program and clear O.K., but no amount of cycling has revived bit 1.

Apart from this problem, my laundry works very well, so here are a few notes for those readers contemplating building their own.

You won't find suitable U.V. tubes in stock at your local electrical shop. If they stock any kind of U.V. tube at all it will be the black tube type, which is useless for PROM cleaning because the U.V. wavelength is too long. The correct U.V. tubes, which emit "hard" — i.e. short wave U.V. are made from a special glass which is transparent to both U.V. and visible light. These are stocked by hairdresser suppliers, in the 18in 15 W size. However, they won't stock the control gear (unless you want a sterilizer instead of a PROM laundry). Back at your local electrical shop you will find that control gear for a 15 W tube is just not made! This is because the choke is the same as that made for a 3 ft. 30 W tube. You might think that this choke would overload your lamp, but this doesn't happen — the 30 W choke will run your lamp at 15 W, and, indeed, this is the lamp manufacturer's recommendation.

Use a starter switch with a translucent cover, and don't hide it inside your case. It then acts as a pilot and glows when the lamp is switched on but is not working.

Make sure your case is U.V. proof. Remember, it takes a week's exposure to the sun to erase a PROM, but only 10 mins. in the laundry — i.e. the lamp is 1000 times stronger in U.V. than the sun. Don't get sunburnt!

Grahame Coates,
4 Ventris Close, Hadleigh Road, Ipswich IP2 0DB.

Redrawpic

I enjoyed reading the program 'Drawpic' in your Feb '79 issue However I found my Level II 4K TRS-80 scarcely big enough to hold it: indeed when I entered the program with some redundant spaces to increase legibility, I ran out of memory.

I therefore recommend anyone using the program to insert between lines 3 & 4 the line

```
DEFINT B – Z
```

and renumber accordingly.

Since integers are stored in 2 bytes as against 4 bytes for reals, this will approximately halve the space required for data.

If you go further and make the line read

```
DEFINT B – Z : DEFSTR A
```

then you can omit the $ after every A, saving 19 characters. Since DEFSTR is stored as one byte in the Level II TRS-80 this saves space as well as wear on the fingers.

If you want to save even more space, since both x and y lie in the range 0 — 127, you can pack the x and y co-ordinates into one integer; e.g.

```
w (n) = 256* x + y and unpack e.g.
set (w (n)/256, w(n) – 256* (w(n)/256))
```

Thus we require only 2 arrays instead of 4, halving the space requirement again.

J.S. Linfoot
Flat 10, Pembroke Court, Rectory Road, Oxford OX4 1BY.

The Pet IPUG

In the article 'In Defense of PET' a number of items require further comment.

1. With the new improved PETs comes a new improved PET User's handbook.
2. The IEEE-488 standard has the equivalent IEC TC-66. This basically differs only in the connector standards, which PET does not use anyway.
3. The keytop wear problem on the earlier PETs has been solved and no longer occurs. Replacement keytops are available from Commodore.
4. The Newsletter of the Independent PET Users Group is another useful source of information.

R.D. Geere, Editor, IPUG Newsletter.
52 Highfield Road, Cove, Farnborough, Hants, GU14 0EB.

Creative translation

I feel that I should clarify some points that have arisen from the correspondence on my previous letter. (PCW, March '79).

The reason. I transcribed the algorithm for programmable calculators was to show how one should not automatically assume that a program written for a computer is impossible to run on a programmable calculator. The second point of my letter was, once the program works it should not be abandoned if the running time is too long; hence my modification of including the Newton-Raphson formula.

As a result of this, the initial algorithm provides an excellent starting point for N.R. when the two programs are combined. The principle of N.R. is that a starting value, which is near to the root of the equation, is modified to a more accurate answer. Any shorter program must use guess work in calculating the first value — such as taking some fraction of the initial number.

To summarise then, I did not try to create a solution to a problem, but to show how to use other solutions in a creative way.

I.W. Morrison
54 The Fairways, Leamington Spa, Warwickshire. CV32 6PS.

TRS-80 Line Renumbering Program

I found it strange that the renumbering routine listed in the May issue should give an error message when confronted with the problem of fitting a two digit line number in the space formerly occupied by a one digit number, in fact of replacing any line number by a larger one.

I have already written a program which creates spaces in the TRS-80's BASIC text. All that is necessary is to move the required block of text up as far as is desired, using the Z80's LDDR function, then to update the two bytes at the beginning of each line which indicate the start of the next line and finally to adjust the two bytes 16633 and 16634 (40F9 and 40FA Hex) which together should always contain the address of the byte just after the BASIC program text's end.

Michael Burrows (Age 16)
42 Hawthorn Avenue, Palmers Green, London N13 4JT.

Safety first

With reference to Mr. P.F.T. Tilsley's letter in PCW (May issue) on the subject of a simple solution to restricted I.F. bandwidth of a broadcast T/V. R/X.

Readers should note that it is not recommended (if you value your computer) to feed video from a UART straight to the video O/P stage of a live chassis TV. The only correct way, of course, is to fit a mains I/P isolation T/X; which is expensive, and fairly heavy! The point is that a R.F. I/P is subject to I/F noise (snow) rather than restricted I/F bandwidth; as an average TV should resolve 5.5MHz; therefore a good quality modulator should be used. Play safe, use isolated A/E skt. and don't give your first love a chance to say, "I told you that damn computer was a waste of money", as you throw it in the scrap bin.

R. Keller
"Paddock End" Polperro Drive, Freckleton, Preston, Lancs.

Miskeying the Miskeys

This morning's post brought my copy of PCW and a pleasant surprise: the publication of my letter (in PCW, May) with regard to miskeying.

However, after reading the published version of my letter, I was not so pleased. Owing to certain errors and one omission, the letter now reads like a garbled output from some delinquent computer. I realise the information I intended to impart was hardly earth-shattering, but at least someone, I would have thought, would have noticed the illogicality: 'The normal interrupt key is the division (+) key', although to be fair, on this typeface the division sign does look like the plus sign.

Could I therefore prevail upon you to perhaps publish a correction for lines 18 ('For example, if the interrupt (÷) key . . .'), on page 9, and 1 on page 10 ('The normal interrupt key is the division (÷) key.'), in addition to including the double-triangle key symbol at the end of the table; the equals sign looks a bit daft stuck there on its own!

This minor moan apart, I immensely enjoy your magazine (or should it be OUR magazine?) and I look forward to implementing Colin Chatfield's 'STATPACK': I am a medical laboratory research scientist, and statistics is my staple diet, along with system modelling and many other subjects which will benefit, eventually, from the programs the magazine outlines.

Peter G.Q. Brooks
20, Brasenose Driftway, Cowley, Oxford OX4 2QX.

PCW ***It's*** **our** ***magazine*** **PCW**

SUPERFAST SUPERBOARD

Having recently made a slight modification to our demonstration Superboard and selected the best of a batch of standard 550ns memories, our board is now running at 2MHz instead of 1MHz. Running the PCW Benchmark tests produced the following results:—

BM1	less than	1 sec
BM2		4 secs
BM3		7 secs
BM4		8 secs
BM5		9 secs
BM6		13 secs
BM7		20 secs
BM8		4 secs
BM7&8		24 secs

All measurements were rounded up to the next whole second to allow for reaction time errors and all tests were repeated several times to validate the results. The normal unmodified Superboard would of course produce double these timings.

It would appear from these results that in order to beat a 2MHz SUPERBOARD you would have to go for an IBM 370; which can't be bad for a home micro!

Peter S. Fawthrop,
Calderbrook Technical Services,
1 Higher Calderbrook, Littleborough, Lancs.

PCW *The Benchmark tests were first published by "Kilobaud" magazine, and our test results are collated or obtained by Consultant John Coll.* PCW

A tighter plot

Having been interested in plotting graphs of experimental data and in drawing pictures of molecular structures on microcomputer screens, I have examined a number of microprocessor systems closely for their graphic capabilities. I would therefore like to point out a number of omissions in Dr. Beynon's article (PCW, May) "The Sorcerer's Wand" with regard to the capabilities of the PET.

1. The Pet does have quarter square characters; we regularly plot data at a resolution of 80 by 50 using these and elementary Basic programmes.

2. The Pet also has eight x eight dot matrix pictures, so the "impossible dream" figures should be (40 x 8) by (25 x 8).

3. The Pet has 16 very useful characters which are similar to that defined in the article, i.e. a horizontal line across the character space in any of 8 positions, and similarly vertically. So by selecting the right characters, which do not need to be specially defined, it is possible to use a resolution of either 320, 25; or 40, 200. This may not be quite as good as the 64, 240 pictures shown, and of course instead of dots one gets short horizontal (or vertical) lines, but it is still very useful.

Perhaps I could point out that we have published a design for a cheap 256 x 256 point addressable graphics peripheral which would operate on any microprocessor*; the current cost of the hardware should be well under £200, and we have used it in research applications for nearly a year now. We still hope for a similar unit to be available commercially to save us building more, but so far the only British one is in the £2000 region and cheaper American units seem to be reluctant to cross the Atlantic.

*A graphics oriented data collection unit — K. Stewart and J.S. Littler, "Microprocessors" *2*, 139-145 (1978).

John S. Littler
University of Bristol
School of Chemistry, Cantock's Close, Bristol BS8 1TS.

State your art

As the indirect instigator of B.A. Martin's letter (PCW, February) correspondence, I feel compelled to say a few words. Firstly, who is to say what 'state of the art' is? Any design using 'state of the art' devices will:—

a/ be expensive since if you want to be truly 'state of the art' you must buy your devices at the early expensive stage of the cost curve, otherwise . . . .

b/ if you wait for a better price, then your device will no longer be 'state of the art'! I think that we are in danger of becoming akin to the hi-fi fanatic who updates his equipment at the merest whisper of so-called improved performance and as a result misses the whole point completely.

However, it is right to examine periodically all possibilities but, even now, I rule out 16 bit machines for most people who want to get into personal computing the DIY way. The reason is simply cost: you will need twice as much memory, more power supply capability, etc; and I presume that most people either DIY because they like designing things (in which case, they are unlikely to be interested in any design published in PCW) or because it's cheaper — ergo, cost of 16-bits rules it out.

Do we go to the other extreme a la 77-68? Well, yes, there probably will be some interest for such a system, but I wonder just how much and for how long. Where do we go? There are, it seems to me, two groups to cater for. First, those who want to design their own but don't or aren't sure how to. PCW is about the only magazine that is catering for that need. Second, are those who want to buy a kit that can be expanded in easy stages and that includes, for example, a simple VDU with alphanumeric display semi-permanently on the top line and a cheap light pen in lieu of a keyboard (which could be bought later on). However, there have been three kits produced of similar type already (NASCOM, TRITON and currently one in Wireless World) so any new design must either be innovative or be state of the art (again?) or be dirt cheap. P. Tilsley (PCW May '79) made a good suggestion that a centre should be set up to draw up a spec. for a reasonable all-round personal computer. Well, I am prepared to act as that centre but I bet I don't get more than twenty replies. What I want to know is what you want in the way of:—

a/ minimum system to start with
b/ your anticipated maximum (and if you say twin floppy discs then I will expect you to justify your extravagant desire!)
c/ preference for processor — again justify
d/ features the monitor should have
e/ resident high-level language
f/ complete kit or just pcb or just design notes.

Please send your specs to me at my address (preferably typed) but if you want a reply, then please enclose an sae. Closing date is August 1st, so come on, put pen to paper. PCW will give a prize of £30 for the most realistic and best reasoned specification and I will publish the findings.

Mike Dennis
"Blackberries" Sherriffs Lench, Evesham, Worcs.
Tel: Evesham (0386) 870841

ANADEX D.P-8000

COMMERCIAL PRINTER
Heavy Duty Printer
for
Pet ★ SWTP ★ Horizon
Apple ★ Cromemco ★ Sorcerer etc.
80 Col. Fast 112 C.P.S. Bidirectional Printing.
Upper/Lower Case, £ sign, Universal Interface.
Serial, Parallel & Current Loop
Complete with Plugs & 500 Sheets Paper,
Ready to Go. £575 + VAT

DELIVERY EX. STOCK.

Phone/Write for Further Details or Demonstration

LONDON COMPUTER STORE
43, Grafton Way, London W.1.
Tel: 01-388 5721

C CRYSTAL ELECTRONICS CC ELECTRONICS C

CRYSTAL/CC ELECTRONICS 'NASCOM' SOFTWARE

XTAL BASIC – SPECIFICATION

This is an "8K Basic" Interpreter written for the Nascom 1 system.

1. **COMMANDS:**– Call Clear CLoad Cont CSave Read.. Data.. Restore Def.. Fn Dim Edit End For.. To.. Step.. Next Gosub.. Return Goto If..Then Input List Nas Pop New On..Goto On.. Gosub Out Poke Print Rem Run Speed Stop Wait SPC< > Tab < > Print @
2. **VARIABLES:**– Names must start with a letter, but can be up to any length. First two characters used to distinguish one variable from another. Strings of up to 255 characters, also Multi-Dim. Arrays and String Arrays. Numbers range from +/– 1E+/–38, with an accuracy of six significant figures.
3. **FUNCTIONS:**– ABS ASC ATN CHR$ COS EXP INT LEFT$ LEN LOG MID$ PEEK POS RND RIGHT$ SGN SIN SIZE SIZE$ SQR STR$ TAN VAL
4. **OPERATORS:**– ARITHMETIC: + – * / ** ("To the Power of")
   RELATIONAL: < > <> >= <=
   ARITH-LOGICAL: And or Not
   STRING: + (Concatenation)
5. **CASSETTE COMMANDS:**– CSave CLoad for Saving and Loading Programs. Also CSave@ Cload@ for saving and loading of Numerical Arrays.
6. **SPECIAL COMMANDS:** EDIT – Powerful Line Editor. CALL – Machine-Code Subroutine Call, NAS – Return to 'Nasbug' Under Software Control, OUT, INP & WAIT – For Control of I/O Ports.
7. **COMPATIBILITY:**– Tape Routine Provided for Use with T2 Monitor. Fully compatible with T2, T4 & B-BUG Monitors.
8. **SIZE:**– Actually Fits in 7K of RAM (1000H – 2BFFH), but recommend >= 16k expansion Ram in your system.
9. **AVAILABILITY:**– On C12 Cassette Tape, with documentation.
10. **PRICE:**– £35 + VAT

40 MAGDALENE ROAD
TORQUAY, DEVON, ENGLAND
Tel: STD 0803 22699

Happy Memories

21L02 450ns	78p	2114 450ns	£4.95	4116 300ns	£7.45
21L02 250ns	95p	2114 250ns	£5.40	2708 450ns	£6.75

TRS-80 16K Memory Upgrade Kit: £70

S100 16K 250ns Static RAM Kit: £195 With 4K £81, 8K £119

Low Profile DIL Sockets										
	Pins:	8	14	16	18	20	22	24	28	40
	Pence:	9	10	11	15	16	18	20	25	35

Our new shop is now open at the address below. We shall be stocking a wide range of items to interest all those of you who are building or plan to build your own microcomputer, why not pay us a visit? We are open from Mon. to Sat. 10 to 6 and often much later.

We stock a range of books covering fundamentals through to advanced topics (like games)

We are NASCOM dealers for the South Coast.

Do-it-yourself with our range of wire wrapping aids and materials from the O.K. corral, or Box-it-yourself with a Vero enclosure after Soldering-it-yourself with Antex.

Our stocks are rapidly increasing; please write or call for latest lists of available products. We welcome your suggestions for stock lines. What do you find difficult to obtain? (We know about buffers)

Please add 20p p&p to all order less than £10 in value. Cheque or P.O.s payable to 'Happy Memories'. Access or Barclaycard orders may be telephoned 24hrs a day.

Prices quoted include VAT at 8% Please adjust for any change in rate.

19 Bevois Valley Road, Southampton, Hants. SO2 0JP Tel: (0703) 39267

micro software systems

SOFTWARE CONSULTANTS & SUPPLIERS FOR ALL COMPUTER SYSTEMS

DESK TOP COMPUTERS

Find it difficult to make a choice?
Utilise our experience to make the right decision.
Micro Software Systems Supply:

HORIZON
RAIR BLACK BOX
EQUINOX 300
COMPUCORP 600 SERIES
COMPACT 400
MODATA
COMMA V.03 and other LSI BASED SYSTEMS

A complete range of peripherals complements this selection.

Application Software available: Packaged or Bespoke.

Prices from £350

Quantity Discounts can be arranged.

Call or write and we will be happy to discuss your problems.

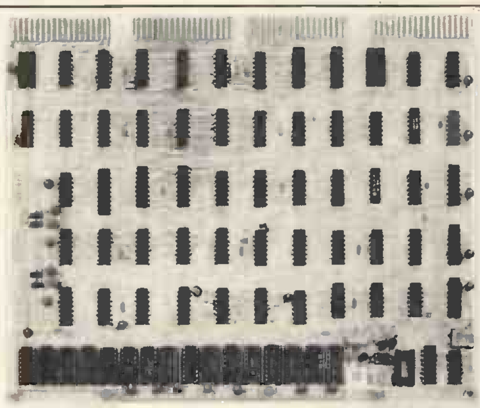
Micro Software Systems,
Stanhope House
Stanford-le-Hope
ESSEX.
Tel: STANFORD-LE-HOPE (03756) 41991/2

Tid Bits PRODUCTS . . . . COMPANY NEWS . .

Datapro has it Tabled

Datapro publishes several hundred reports annually on data processing and office system hardware, software, services and companies. There are over 35,000 subscribers to its reports, one of which, "All about Microcomputers" was sent to PCW and can be thoroughly recommended. This service is for professionals and is not cheap. Computer Information Services are agents for Datapro reports, and PCW contact with them indicates a friendly and fast service.

Information from:
Tricia Carter, Information Director,
CIS Ltd., 221 Seven Sisters Road, London N4 2DA
Tel: 01-263 4441



The "Univisor" — Gresham Lion's new single card computer graphics system. It will enable mini computer users to generate their own graphics using a standard TV monitor.
Contact: Russ Cockrell, Gresham Lion Ltd.,
Gresham House, Twickenham, Middlesex. Tel: 01-894 5511

These Self-Study packages from Sybex comprise workbooks plus two to eight audio cassettes. The courses teach all aspects of micro applications. There are ten courses ranging from an Introduction to Microprocessors to interfacing techniques.

Details from:
Sybex, 2020 Milvia Street, Berkeley, California 94704
Tel: (415) 848 8233

Forecasts on the Horizon

Competing with the large computers used by BBC and ITV to keep their pundits happy last Election Night, BBC Wales used an Horizon microcomputer supplied by Equinox Computer Systems to provide an instant check of all the local General Election results.

With programs developed by Dr. Martin Healey of University College, Cardiff, the experts were able to watch the results being analysed on individual monitors.

The cost of the system at £2,000, "compared favourably with other equipment" used on the same night and reputed to have cost £250,000.

Equinox also announce that the high level language SNOBOL 4 (for string manipulation, lists, data structures and aggregates) is available on the 16-bit Equinox-300. It calls its implementation Macro Spitbol, a compiler which can be run on a minimum 64K bytes memory.

Contact:
Michael Kusmirak,
Equinox Computer Systems Ltd., Kleeman House,
16 Anning Street, New Inn Yard, London EC2A 3HB.
Tel: 01-739 2387/9.

Plenty of Leeway

The Model 879 Serial matrix printer is a microprocessor controlled Printer suitable for use in many data processing applications with mini and micro computer systems.

Standard features: a 9 x 7 Matrix Print Head, 120 cps bi-directional printing, a 96 ASCII character set with upper and lower case and programmable extended characters, forms length and top of form control.

RS232C Serial Interface is supplied as standard with parallel and current loop interfaces being available as options. Baud rate is switchable up to 9600 baud and the Print Head is microprocessor controlled with a full 2 line buffer and 96 character overflow. (No price quoted to PCW).

Further information:
Leeway Data Products Ltd., Gresham House,
Twickenham Road, Feltham, Middx, TW13 6HA.
Tel: 01-894 5511

Newton's Lore of Computation

Photograph shows 320K configuration with 12 RS232 I/O Ports, a Floppy Disk Controller and Hard Disk Controller — PCB's from front to back are:

1 AM100, Dual Board LS11 16 Bit CPU
5 64Kb Dynamic Memory Boards
1 AM200 Floppy Disk Controller (up to 8 drives)
1 AM500 Hard Disk Controller (up to 4 drives)
2 AM300 Six RS232-port I/O Boards

All fitted in a 22 slot S100 Bus Box +PSU.

This is the Alpha Microsystem, whose UK distributor is Newtons Laboratories, 123 Wandsworth High Street, London SW18.
Tel: 01-870 4248.
The system above is priced at £8875.

Aiming to please
Portable Microsystems Ltd. of Brackley, Northants, announce the Rockwell AIM 65 microcomputer is now available in "three important versions."

AIM 65. The bare-board version which is intended for the hobbyist, student or systems manufacturer who intends to develop AIM 65 into a higher level system. Prices from £249.50.

AIM 650. A case and a power supply unit is now available for AIM and comes complete with 4K Ram, 8K Ram Basic, power supply and desk top case. Intended for the user who prefers to have a complete package and either would prefer not to, or is unable to work at the electronics level, prices from £485.

PDS 65. Called PDS 65 this product takes the basic AIM 65 and enhances it to become a portable development system. For programmers or hardware designers who need to modify and debug systems at home, or at customers' premises. "The excellent documentation of this product enables the user to 'home in' on the 'target system', to allow any necessary changes to be carried out immediately."

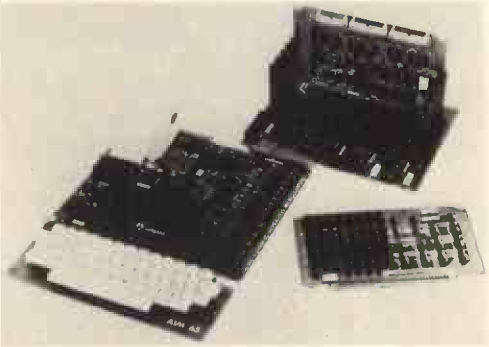
With the use of an optional acoustic coupler the user can link up with a host system to benefit from master files or the relevant data base. Prices from £950.

Dealer enquiries welcome.
Contact Mike Ayres at:
Portable Microsystems, 28 Broad Lane, Evenley, Northants.
Tel: (0280) 702017

Bubbling Over
The AIM 65 microcomputer, which sells for £249.50 can now be operated with a fully addressable add-on 1 megabit bubble domain memory system.

Key to using such external memory with the AIM 65 is a buffered expansion motherboard which Rockwell calls the "AIM 65 Expansion Motherboard" and which is now offered by Pelco (Electronics) Ltd., as an option for this R6500 based microcomputer.

Originally designed as a microprocessor educational aid and R6500 Development System, the AIM 65 has been "adapted by thousands of its initial buyers to applications such as a smart terminal, a special desk top computer and a variety of process controllers."



The AIM 65 expansion motherboard have five connector slots that can accommodate any of the Rockwell System 65 modules or Motorola Exorcisor modules, as well as add-on modules from Burr-Brown and other manufacturers.

The motherboard essentially extends the AIM 65 bus. System bus lines (address, data and control) are buffered to provide ample drive capability. Address decode logic for mapping internal and external addresses in 4K-byte increments is provided. Sixteen switches permit the user to define whether each 4K-byte portion of the R6502 address space of 65K-bytes is internal or external to the AIM 65.

Thus, under software control, the Rockwell megabit bubble memory module can be addressed providing 128K-bytes of memory expansion.

The unit price for the AIM 65 expansion motherboard is £136.50 and is scheduled for June delivery. Pricing of the various options that the motherboard allows can be obtained from Pelco (Electronics) Ltd., on Brighton (0273) 722155. Delivery of the AIM 65 products is now ex stock.

Address:
Enterprise House, 83-85 Western Road, Hove, Sussex BN3 1JB.

SYNERTEK SYSTEMS KEYBOARD: 6500 Based Keyboard Terminal
Rastra Electronics Ltd. of Hammersmith present the Synertek Systems KTM –2; "a high reliability low cost keyboard terminal module" providing a full ASCII keyboard and all the logic to display 24 lines of 40 characters each with full graphics.

Features: 54 keys, 128 ASCII characters (upper & lower case alpha, numeric, special and control); graphic and alphanumeric characters are capable of simultaneous display. With KTM–2's relative and absolute cursor addressing, graphs, game pieces etc can be placed and moved about the screen with a minimal amount of software.

Graphic capability, a major consideration, uses 24 lines of 40 characters. In addition KTM–2's 40 character line has a 3.2 MHz bandwidth which is within TV video capability.

In addition Rastra offer the full range of Synertek Systems with special kit prices for integral systems based on SYM–1.

Further details:
Rastra Electronics Limited, 275-281, King Street, Hammersmith, London W6 9NF.
Tel: 01-748 3143/2960; Telex 24443

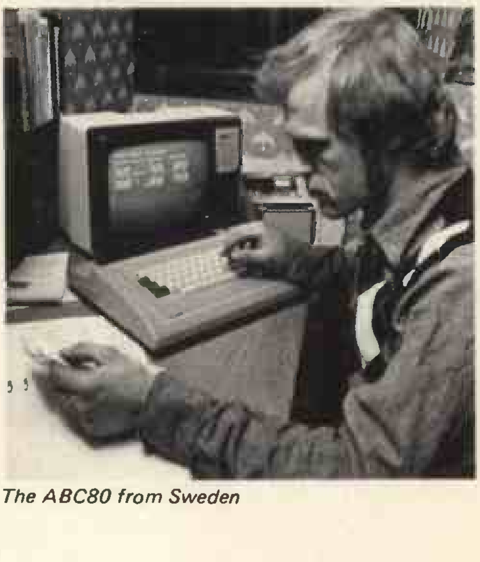
Left to Right: *Tony Plackowski, Mark Cooke, Coralie Cain* of **Tridata Micros.**

Tridata Micros Limited have recently opened an office in Birmingham and the address is: Smithfield House, Digbeth, Birmingham, B5 6BS. Telephone: 021 622 1754. Tridata are now marketing packages for micro computers and prices range between £100 and £200 approximately. All accounting functions are covered plus Stock Control and Payroll.

What's This?
It's "Whatsit": Software claimed to be guaranteed trouble-free for first time users of Apple II, North Star and CP/M™ Systems. It's a database/query system. Using simple pidgin English requests the user can store, index and retrieve information about one or more related or unrelated subjects. Systems using a minimum of 24K memory can use "Whatsit", which provides as many as 25,000 entries, each up to 200 characters in length. Utility programs are also provided, together with a comprehensive manual. (Sold by Interam in Britain).

More information from:
Information Unlimited, 146 N. Broad Street, Griffith, IN46319, USA Tel: (291) 924-3522

The Swedes Get Personal
A new Swedish personal computer system, the ABC 80, is composed of three units: a standard keyboard with built-in computer, a 12" B & W TV monitor, and a cassette unit. The processor used is the Zilog Z80A. Memory of 16K Bytes ROM, 16K Bytes RAM. There is a V24 jack enabling connection to a telephone modem, an "ABC bus" for connection to peripherals



The ABC80 from Sweden

such as floppy-disk units, printers, plotters and measuring instruments, a real-time clock, and a loud-speaker. The ABC 80 is designed for home, educational and small business use, is programmable in BASIC, and has software packages under development. No price was quoted to PCW but details are available from:
SATTCO AB, Dalvagen 10, S-17136 Solna, Sweden.
Telex: 11 588 Phone: int +0046-8 83 02 80.

NASCOM DO IT – AGAIN

The Nascom-1, which graced the first cover of PCW, has a successor, the Nascom-2, an upgraded version with more of a computing basis, and significantly more memory – a total of 20K bytes (addressable), comprising 2K monitor-in-Rom; 1K Video RAM; 1K Word Space/User RAM; 8K Microsoft Basic; 8K Static RAM/2708 EPROM. The Nascom-2 is not intended as a replacement for the Nascom-1 which, says Nascom Microcomputers, has a market for the foreseeable future.

Focus on the Brokers

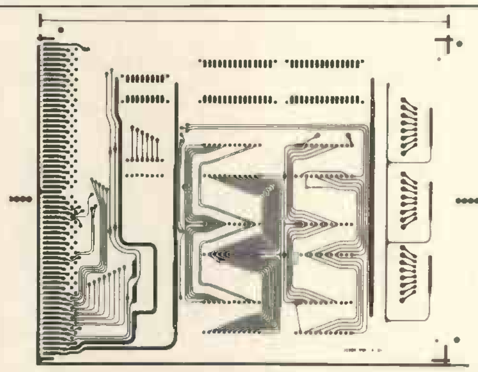
A new "low-cost" ASCII Keyboard, the Model 771, is on offer from **Electronic Brokers**. Standard features include: full ASCII alphanumeric section; cursor control; numeric pad; two-key rollover; upper and lower case plus control codes; TTY mode for uppercase only operation; timed autorepeat on all keys; all modes standard parallel interface. At £95, the Model 771 is supplied fully assembled and tested, and there are discounts for quantity. **Electronic Brokers are at 49/53 Pancras Road, London NW1 2QB. Tel: 01-837 7781.**

Keyboard Model KB771

The ATARI 800 home computer has just been released in the States at the West Coast Computer Faire, and has "knocked everybody sideways".

A young visitor to Thames Polytechnic has a go on a microcomputer provided by the "Selmic" microcomputer hobby group. The microcomputer workshop was open to the public after the Annual Grace Hopper Lecture given last May by Alex d'Agapeyeff, a past president of the British Computer Society and now Chairman of the powerful Software house CAP Microsoft (UK) Ltd.

Pupils of Longfield School in Kent demonstrate their skills on a microcomputer: Front to back: *Gary Googe, Robert Butler, Nigel Fuller, David Barnes, and Alan Farley. The occasion: A one-day conference at the Thames Polytechnic attended by over eighty teachers and staff from all over S.E. England.*



Microprocessor type read only memory printed circuit board, with provision for 3 x 8 way user programmable switches. This is a product of Shannon Circuits, *which launched its UK operation on May 15. Strong backing from the* Irish **Export** Board *for this highly reputable company points to a prospering export drive for Shannon. Its UK Operations Executive is* Colin East and information is available from: **Shannon Circuits Ltd., 47 Hawks Mill Lane, Allesley, Coventry. Tel: (0203) 333055.**

No Tears from this Town Crier

Publicity-conscious Tony White *and* Michael Monk *have formed "The White Monk" company, and decided to hire the Town Crier to ring out their wares: "an umbrella information processing, instant printing and automatic typewriting service with allied mail house facilities, from production to final mailing". They expect a heavy demand for word processing, especially letter writing. Their services are available week-through, response guaranteed on* Hastings (0424) 440204. **Address: 9 London Road, St. Leonards on Sea, East Sussex, TN37 6AE.**

Best seller gets taped

Adam Osborne and Petsoft have got together to put seventy six programs on one cassette. The programs are from Osborne's book, "Some Common Basic Programs" and range from small business applications to mathematical applications. The program cassette is at £15 (inc 8% VAT) and is available from **Petsoft, 5/6 Vicarage Road, Edgbaston, Birmingham B15 3ES.**

Keen Computers Ltd. are in the process of organising their own distributor network, to retail their Apple based computer systems. Applications for these Dealerships are being sorted out a full list of dealers will be published shortly.

In relation to this Distributor network, Keen Computers Ltd. are expanding their software development operations and "hope to provide an increased range of packaged software for the Apple in the immediate future."

New developments of products for the Apple II will be available shortly and these will ensure that the Apple "retains its position as market leader", whilst allowing flexibility at all levels of use.

ABACUS SOFTWARE announces the availability of the "PET[R] Machine Language Guide". This manual is intended to help the Pet owner who would like to progress beyond the Pet's native language, BASIC.

Included are sections on using the Pet's input and output routines, clocks and timers, floating point, fixed point and ASCII number conversion routines, built-in arithmetic function all from machine language programs.

There are suggestions for writing machine language programs. The sample programs are all written at a level which is simple enough for the newcomer to machine language to understand. More than 30 of the Pet's built-in routines are fully detailed so that the reader can immediately put them to good use.

The guide can be purchased for $6.95 from: **Abacus Software P.O. Box 7211, Grand Rapids, Mich 49510, USA.**

Exceedingly Competitive

Exidy, Inc., the makers of the Sorcerer[TM] microcomputer, will sponsor a contest for microcomputer programs this summer. Four Sorcerer computers will be awarded as grand prizes.

The purpose of the contest is to encourage people who have written good programs to share their programs with other computer owners. Exidy will publish a book of the best programs entered in the contest.

The contest is open to all Basic Language computer programs which will run on the Sorcerer. The Sorcerer's Standard Basic is compatible with 4K and 8K versions of Altair Basic which have been in use since the early days of microcomputing.

Prizes of free computers will be awarded to the program judged best in each of four categories: Business, Education, Fun & Games and Home/Personal Management. Every entrant will be rewarded with a free poster and professionally-written program in trade for the program they submit.

The contest will run from June 1 through August 31, 1979.

Further Information:
Mr. Paul Terrell, Marketing Communications,
Exidy Inc., 969 W. Maude Avenue,
Sunnyvale, California 94086, USA.

Businessman's newsletter

The "Business Computer Newsletter" is intended for the business executive interested in computerising his company. It claims it will be free of jargon and full of help and advice. On its editorial board is Malcolm Peltu, who was for some years the highly capable editor of "Computer Weekly".

Enquiries to:
Hulton Technical Press Ltd., Warwick House,
Swanley, Kent BR8 8JF. (The newsletter subscription is £32 per annum).

A stimulating initiative

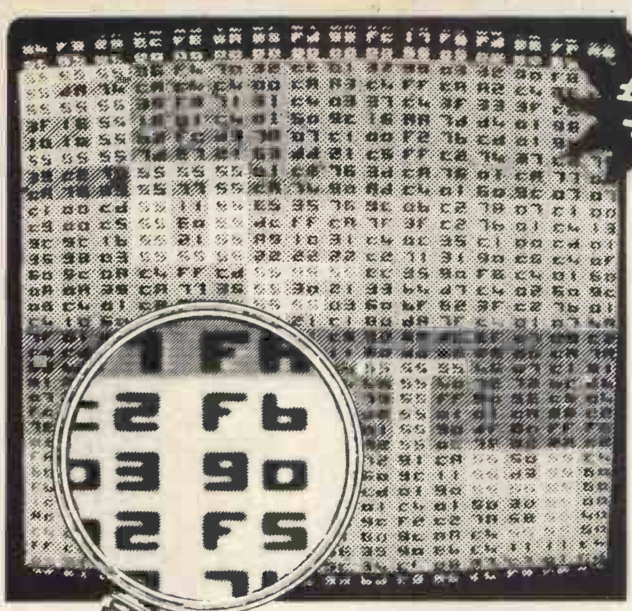
The Manpower Services Commission is to fund a £11 million three-year programme, introducing new courses, to stimulate computer training and reduce the shortage of skilled staff. An extra 2500 to 3000 training places will be created in each year of the programme's three-year period. The three new types of course will be: 1) An initial Programmer Course of 42 weeks, 24 of them practical experience. 2) A Real Time Programmers Course, lasting 24 weeks. 3) A systems analysis course of ten weeks.

Enquiries (from Companies and Colleges) to:
Manpower Services Commission, Selkirk House,
166 High Holborn, London WC1V 6PF.

A dazzling new Cromemco

The Cromemco Z-2H is about to be introduced into the UK. Its outstanding feature is an integral 11-megabyte hard disc system. The Z-2H will be marketed by an independent distributor, **Complete Microsystems Limited.** Priced at £5748, the Z-2H also has a 4MHz Zilog Z80A processor, two floppy disc drives, 64K RAM, RS232 serial interface, and a printer interface. The whole computer is housed in a compact single table-top cabinet. Software, at £65 per item, includes: ANSI Level I Cobol; Fortran IV compiler; Z-80 Macro Relocating Assembler; 16K Extended Basic; Database Management System; Word Processing; Trace System Simulator; TV Dazzler Games; Dazzler Graphics.

Enquiries to:
Micro Centre, 132 St. Stephen Street, Edinburgh EH3 5AA.
Tel: 031 225 2022.



£79.50 + VAT

A NEW TOOL FOR A NEW TECHNOLOGY

SOFTY is not just another training aid for the engineer wishing to become acquainted with microprocessors – it is a BENCH-TOOL for the system designer.

"A revolution will probably take place soon, with the appearance of complete development systems costing less than £1,000 . . ."

Until quite recently all the electronics design engineer really required was a scope and a meter. Now even the smallest of companies are investing thousands in microprocessor development laboratories — afraid of being left behind in the technology-race. But SOFTY is here to help, and a microsystem can be developed without expensive equipment right through the design and prototype stages and even into production.

WHAT SOFTY WILL DO

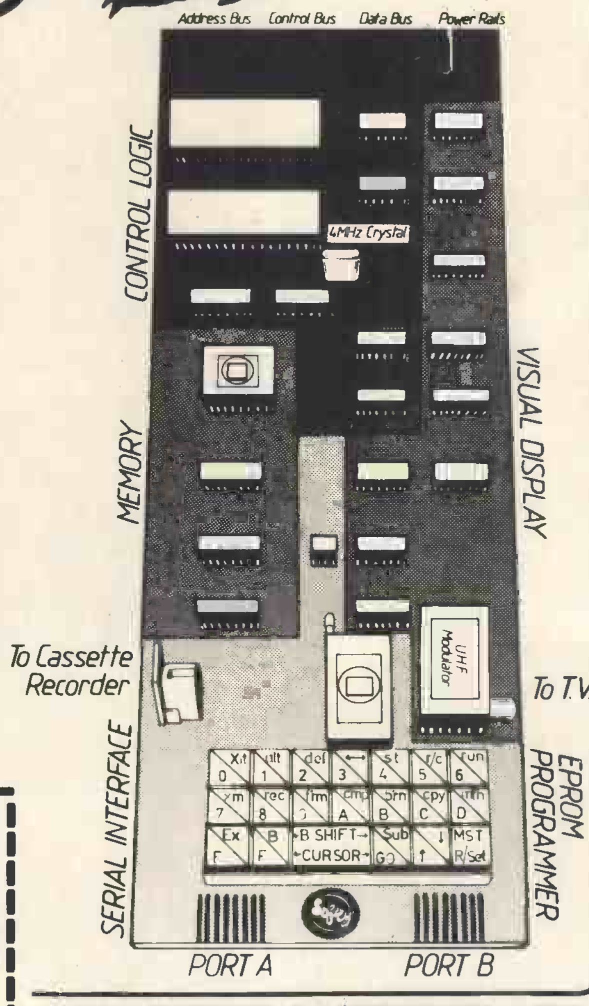
- IT COPIES MEMORY DEVICES (ROMs &c) presenting the data as an address-mapped hexadecimal display on the screen of a monitor or TV set.
- IT DEVELOPS PROGRAMS for virtually any microprocessor with facilities similar to an ASSEMBLER: you may enter, insert or delete instructions, shift blocks of data, match specific bytes, calculate displacements to labelled locations — and all with the overwhelming advantage of being able to test the program instantly and even develop it one instruction at a time!
- IT RECORDS PROGRAMS on ordinary cassette tape using an ordinary cassette recorder at ultra-high-speed — around 2000 baud equivalent!
- IT PROGRAMS EPROMS of the 2708 family at a speed which is close to the theoretical minimum (2 mins per 2708). It may therefore be used as an 'instant-copier' for software.
- IT IS A HANDY COMPUTER which may be programmed to do useful jobs in the home or workshop, and may even be included as the 'brains' of larger equipment, performing sequential or combinatorial control functions. SOFTY has a microcycle length of exactly one microsecond and there is a programmable timer. The manual lists a simple interpretive language which anyone may learn to use in ten minutes!
- IT IS A FABULOUS LEARNING AID because the trainee can actually see what is happening — SOFTY is completely transparent! The internal MPU will cease execution at a breakpoint, which may be substituted for any program step, and display contents of internal registers.
- IT FILLS THE GAP BETWEEN THEORY AND PRACTICE for the serious user who already has a computer and dedicated assembler to develop his software. The computer makes documentation — not prototypes. SOFTY places the program in addressing space to be actioned by the MPU of his choice in a real system — the proof of the pudding! Simple debugging and condensing of code may often be handled without recourse to the assembler.

SOFTY can be assembled in a couple of hours. No extras are required except for a power supply providing +5, +12 & —5 volt rails and +30 volts for the EPROM programmer. The kit includes sockets for all the 23 ICs, UHF modulator for TV use, 4MHZ crystal, DIN socket and lead for cassette interface, 21 key keyboard, a quality double-sided PCB of fibreglass with solder mask and component overlay and a comprehensive manual covering assembly and use.

A DEVELOPMENT KIT is also available which includes all of the above and a lever-operated ZERO INSERTION FORCE SOCKET for the EPROM programmer, 43 way card edge connector, ribbon cable and 24 pin header (for connection to the system under development as firmware) and a spare 2708 EPROM.

It is not possible to present a full technical specification in the space available here. We will therefore send you a SOFTY on the understanding that you may examine it and read the literature and, if you wish to do so, return the goods for a complete refund within 14 days.

VIDEOTIME PRODUCTS, 56, Queen Road, BASINGSTOKE, Hants, RG21 1RE
TEL: (0256) 56417 TELEX: 858747.
We welcome Barclay & Access orders by telephone.



Please send me: (I enclose Cheque/Company Order)

...... SOFTY Kits @ £86.36 (Incl. VAT & 50p p & p)

...... DEVELOPMENT Kits @ £106.88 (Incl. VAT & 50p p & p)

...... BUILT DEVELOPMENT Kits @ £128.48 (Incl. VAT & 50p p & p)

...... POWER SUPPLY Kits @ £16.08 (Incl. VAT & £1 p & p)

Name ....

Address ....

....

....

PCW 7

VIDEOTIME PRODUCTS, 56, Queen Road, BASINGSTOKE, Hants, RG21 1RE

SOFTY

TVJ

For Hardware, Software, Peripherals, Consultancy and Competitive Prices.

MICROCOMPUTERS ETC (0276) 62506 (0272) 422061

Registered business name

PET

commodore Authorised Dealers

Pet 2001 From £435

NEW PET 2001 with large keyboard. From £ 630.00

PET 2001-16N (16K RAM and New Large Keyboard)	£630.00
PET 2001-32N (32K RAM and New Large Keyboard)	£750.00
PET 2001-4 (Standard PET with 4K memory)	£435.00
PET 2001-8 (Standard PET with 8K memory)	£515.00
PET 2040 (Dual Drive mini-floppy 343K User Storage)	£745.00
PET 2023 (80 col. dot matrix printer with PET graphics)	£515.00
PET 2022 (as above with tractor feed)	£605.00
IEEE/RS232 Serial Interface 'A' Output only	£106.00
IEEE/RS232 Serial Interface 'B' Input/output	£186.00
IEEE-488/Centronics type parallel Interface	£45.00
PET C2N External Cassette Deck	£50.00
Interface to S100 (4 slot motherboard)	£112.00

Sorceror

Now with the S100 Bus Expansion Interface and Dual Drive min-floppy Disk

Inc. Authorised Dealers

Sorceror 16K RAM (inc.UHF Modulator)	£740.00
Sorceror 32K RAM (including UHF Modulator)	£840.00
Exidy Video Monitor (High Resolution)	£240.00
Exidy Dual Drive mini-floppy disk (630K Storage)	£1200.00
Exidy S100 Bus with Interface+Motherboard+PSU	£200.00
Exidy Mini-floppy Disk Drive (143K Storage)	£495.00

Disk Drives

Shugart Mini-floppy Disk Drive (including PSU)	£350.00
Micropolis Mini-floppy Disk Drive (incl. PSU)	£350.00
Percom FD200 Mini-floppy Disk Drive (including PSU)	£350.00
Pertec 10MB Top Load Hard Disk System (with controller+disk basic)	£6400.00

APPLE II/ITT 2020

Apple mini-floppy Disk Drive (116K Storage)	£425.00
ITT 2020 incl. PAL Modulator (16K RAM)	£895.00
RS 232C Serial Interface for ITT 2020 Motherboard	£110.00
Parallel Output Interface for ITT 2020 Bus	£95.00
Palsoft on ROM Board (extended Basic)	£110.00
RAM Upgrade (16-32K, 32-48K)	£110.00
ITT 2020 Authorised Dealers	

Advanced Systems

Altair, Equinox, Billings, Heath, Rair, Horizon. Installations to include hard disk, and multi tasking — P. O. A.

Software

Petsoft COMPUSETTES Personal Software GEMSOFT

Lifeboat Associates (Authorised Dealerships, Send for Catalogues)

PILOT (for TRS 80) text orientated language	£18.00
COMAC – Computerised Accounting for TRS 80	£50.00
STOCK CONTROL – (for TRS 80) incl. Inventory, Purchase Orders and Invoicing	£125.00
CP/M for TRS 80	£95.00
CBASIC for TRS 80 & Sorceror	£75.00

Estate/Employment Agency Systems, Fortran 80, Cobol 80, Pascal

Etc.

Diskettes 5¼ (blank) boxed (min. order 10) each	from £3.00
C60 Cassettes (Min. order 10) each	£0.40
CBM KIM 1 Microcomputer System	£94.00
Computalker Speech Synthesis for S100	£350.00

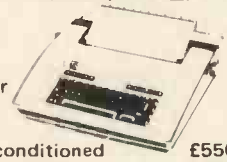
Books – Large range of Microcomputer related books & magazines.

If you don't see it – ask if we have it.

TRS 80 From £350

TRS 80, 4K Level 1 (Keyboard with 4K memory+ VDU+Cassette drive+240v PSU)	£435.00
TRS 80, 4K Level II (as above but with Level II basic)	£535.00
TRS 80, 16K Level II (as above but with 16K memory)	£645.00
TRS 80, 4K Level I – Keyboard+240v PSU only	£350.00
TRS 80, Expansion Interface with 16K RAM	£325.00
TRS 80, Expansion Interface with 32K RAM	£435.00
TVJ 232T Serial Interface for TRS 80	£45.00
TRS 80 Screen Printer (text+graphics) (110V)	£445.00
Expandor Black Box Interface for TRS 80	£48.00
Expandor Black Box Power Supply for TRS 80	£49.50
Centronics Parallel Printer Interface for TRS 80	£45.00
TRS 80 Voice Synthesizer	£345.00
TRS 80 Numeric Key Pad supplied & fitted	£89.00
New Radio Shack Micro Printer	£245.00
Radio Shack Phone Modem	£160.00
NEWDOS Super-enhanced TRSDOS	£50.00
Level III Super-enhanced BASIC	£34.00
RSM Assemble/Monitor on Disk	£25.00
MICROCHESS or SARGON CHESS Cassette/Disk	£14.00
Disk Drives for TRS 80 – see Disk Drives	
UHF Modulators (encased with leads for 625 lines)	£20.00
RAM upgrade (4-16K, 16-32K, 32-48K) Supplied and fitted at our premises	£110.00
Upgrade to increase speed 1.78 MHZ to 2.66 MHZ	£13.00
Switchable selection of Level I or Level II (ROMS required)	£25.00
Automatic volume control (AVC) for CLOAD	£25.00
'Electric Pencil' text/word processing package (on cassette)	£65.00
'Electric Pencil' text/word processing package (disk version)	£115.00
'Electric Pencil' keyboard mod. to give lower case with text/word processing package.	£28.00
S100 Interface for TRS 80	£375.00
'Library 100' – 100 progs for TRS 80 on cassette (Level II)	£39.00

Printers



TELETYPE 43 KSR £875.00
Keyboard send/receive Serial printer for PET or TRS 80
(interfaces or friction feed extra)

Teletype 33 KSR Serial (110 Baud) Reconditioned	£550.00
Expandor Black Box (80 column impact printer)	£375.00
Centronics 779 parallel (friction feed)	£790.00
Centronics 779 parallel printer (tractor feed)	£890.00
Centronics 701 parallel printer, Bi-directional+ tractor fed	£1375.00
Centronics micro printer (20, 40, 80 columns selectable)	£395.00
QUME or DIABLO daisy wheel serial printers	P. O. A.

Terminals

Soroc IQ120 VDU/Keyboard – 80 char./24 lines	£660.00
Cypher CUB VDU+separate keyboard	£380.00

Ansaback 'Phonemate' Telephone Answering Machine, voice operated twin cassette — £190.00

PRICES EXCLUDE VAT, FREIGHT & HANDLING. SEND OR 'PHONE FOR PRICE LIST & BROCHURES. (All prices correct at time of compilation)

Directors: Dr. R.V. King, BA, MIEE. S.G. Johnson, BSc. (Hons.) T.S. Johnson, ABIBA, ACMB, FBSC, MBIM A.S. Barton, ACII, ABIBA, CdipAF.

T & V JOHNSON (MICROCOMPUTERS ETC) LTD.
Member of the TV Johnson Group of Companies
165 London Road, Camberley, Surrey GU15 3JS
48 Gloucester Road, Bristol 7.
Branches at: Birmingham, Bristol, Edinburgh, Leeds, London, Louth, Newmarket, Nottingham, Oxford, Byfleet, Wokingham.

☎ (0276) 62506 (0272) 422061 Eves. + Ansaback eves and w/ends. Telex 858893 Basingstoke (0256) 24787 and Farnham (0252) 721094 (+ansaback during office hours) Hours of business 9.30–5.30 Mon-Fri. 9.30–1.00 Sat.

In which Language should I program my Microcomputer?

Martin Beer

University of Liverpool, Computer Laboratory

For some time now the advantages of using a high-level programming language, rather than Assembler, have been apparent to most microcomputer users. Programs are far easier to read and understand, making it easier to write, or modify, programs to work on any particular computer.

The language chosen by most microcomputer manufacturers has been BASIC, since it is considered by many to be the easiest computer programming language to learn, and can be implemented on current microcomputer systems very economically. BASIC statements are readily recognisable, and can be executed either as part of a program, or immediately, in calculator fashion.

This not only aids writing your program, but also debugging, and possible later alterations or additions. Unfortunately there is no generally recognised standard for BASIC, as anyone who has tried to transfer a program from one microcomputer to another will have found.

Another unfortunate feature of BASIC is the use of PEEK and POKE commands to interface with the hardware and operating system. Not only are the codes concerned totally meaningless without the relevant manuals, but they are totally different for every machine. It would be far more helpful to provide procedures which do the machine dependent operations, which would have to be rewritten for every machine, but they would be short, easy to understand, and isolated from the main part of the program.

Many Tongues

There are now a large number of computer languages. Some, like BASIC, have been written for general programming, whilst others have been designed to make the programming of particular applications easier. Two languages in common use on large computers are FORTRAN, for scientific, and COBOL, for business, applications. Both languages have been in use on a large number of computers for many years. It has been possible for standard subprogram libraries of useful routines to be built up during this time. This, together with access to large numbers of published programs and books describing the programming techniques most suited to programming in the particular language, are a great help when writing a new program. Unfortunately, neither of these languages are easy to instal on a microcomputer since there are minimum specifications for both languages such that only the largest systems could support useful compilers.

One language for which there are good compilers for microcomputers is PASCAL [1, 2]. This language was developed by Niklaus Wirth about ten years ago as an aid in teaching his students good programming methods. Later, it was implemented on a large mainframe computer. Since then compilers have been developed for most computer systems currently available. Now several compilers are being written for moderately configured microcomputer systems. There are reports in the PASCAL User Group Newsletter[3] that large programs written using a microcomputer PASCAL compiler developed at the University of California, San Diego[4] have been transferred successfully, with very little work, to one of the world's largest computers.

```
program exammarks (input, output);
const
   passmark = 50;
var
   mark, passes, failures: integer ;
 begin
   passes := 0; failures := 0;
   while not eof (input) do
   begin
      read (mark);
      if mark > = passmark then
        passes := passes +1
      else
        failures := failures + 1
   end;
   write ln;
   write ln ('number of passes   : ', passes);
   write ln ('number of failures : ', failures)
   end.
```

Fig. 1a. The Exam Marks Program written in PASCAL.

The Highly Scrutable Chinese

A useful subset of PASCAL can be implemented in as little memory as most BASIC interpreters. The compiler described by Chung and Yuen[5] does not include error recovery or allow data types other than integers, and so is not suitable for general use, but even so it is a useful guide as to what can be done when only limited machine resources are available.

```
10  PM = 50
20  P = 0 : F = 0
30  INPUT M
40  IF M < 0 THEN 90
50  IF M > = PM THEN 70
60  F = F + 1 : GOTO 80
70  M = M + 1
80  GOTO 30
90  PRINT "NUMBER OF PASSES :
    "; P
100 PRINT "NUMBER OF FAILURES :
    "; F
110 END
```

Fig. 1b. Exam Marks Program written in BASIC.

To illustrate the use of PASCAL two programs are included, written in both PASCAL and BASIC. The first is a count of the number of students who have passed or failed an examination. The marks obtained by each candidate are entered on the terminal and the number of passes and failures are displayed once all the results have been entered. In PASCAL this program can be written very simply using a loop starting with the statement

while not eof (input) *do*

This tests for end of file on the input stream. If your system does not have a simple end of file symbol, another symbol may be tested for instead. The statement following the *while* statement will be executed repeatedly until the condition *not* eof (input) is false; i.e. the end of input has been found. Since a number of steps are necessary within the loop the required program statements are enclosed between *begin* and *end* to form what is known as a compound statement. The flow of control is also made more obvious by the use of an

if . . . *then* . . . *else* . . . ;

statement to count the number of passes and failures, avoiding the conditional and unconditional jumps required in BASIC. Finally the program is made a lot more readable by careful layout and the use of meaningful variable names.

Declaring Constancy

In PASCAL it is possible to declare constants at the start of the program. This facility is used here to set the passmark at 50. If, at a later stage, a new passmark is decided upon, it is a simple matter to edit the program as only one line must be changed. No variable storage is used, as constants are looked after by the compiler, whereas in BASIC a variable had to be assigned the value 50.

```
function gcd (i,j : integer) : integer ;
(* i and j must both be greater than 0*)
var
   i1,j1 : integer;
begin
   i1 := i ; j1 := j ;
   while i1 < > j1 do
     if i1 > j1 then
       i1 := i1 – j1
     else
       j1 := j1–i1;
   gcd := i1
end
```

Fig. 2a. Procedure to find the Greatest Common Divisor of two Integers, written in PASCAL.

```
1000 REM G. C. D. SUBROUTINE.
1005 REM PARAMETERS I AND J
1010 REM RESULT RETURNED IN
     I1 and J1
1020 I1 = I
1030 J1 = J
1040 IF I1 = J1 THEN 1090
1050 IF I1 < J1 THEN 1070
1060 I1 = I1 – J1 : GOTO 1080
1070 J1 = J1 – I1
1080 GOTO 1040
1090 RETURN
```

Fig. 2b. Subroutine to find the Greatest Common Divisor of two Integers, written in BASIC.

The next example is a function to obtain the greatest common divisor of two integers (figure 2a and b). The method used is to repeatedly subtract the smaller integer from the other until both are the same. This is the result. It shows the ease with which standard functions can be written in PASCAL. In the main program this function may be called very easily within an assignment statement e.g.

```
l := gcd (m,n) ;
```

whereas in BASIC the calling statements to perform the same action would have to be

```
100  I = M
110  J = N
120  GOSUB 1000
130  L = I
```

which is far less readable. Also, care will be needed in the naming of variables in both the main program and the subroutine, since all are available to both. When a number of interrelated subroutines are involved this can be very confusing. In PASCAL variables declared within a subprogram can only be used within that sub-program, so that name clashes are unlikely to occur. If a particular variable is needed in both the main program and a procedure i it must be declared before the procedure, together with the other variables used in the main program. A library of useful procedures can be built up very rapidly and added to programs automatically, when required. This is a facility that professional programmers have made good use of for many years. Routines should be included in the library to make use of any machine dependent features you use, so that if one of your programs is to be transferred to another microcomputer only these procedures need rewriting.

Wirth It

In a single article it is impossible to do justice to any programming language, but I hope that I have given some indications of the opportunities offered by PASCAL. Byte recently ran a whole issue devoted to PASCAL[6], which is well worth reading if you can find a copy. For those wishing to investigate PASCAL's potential, Wirth has written a very good advanced programming text book[7] using it. PASCAL offers the ability to write programs in a clear and readable form, which together with the development of program libraries will help you to program your microcomputer much more easily.

I hope in a further article (with the editor's permission) to describe some of the facilities available in PASCAL for which there are no immediate counterparts in BASIC.

REFERENCES

1 K. Jensen and N. Wirth. PASCAL — User Manual and Report. Springer Study Edition (second edition), Springer-Verlag, 1975.

2 W. Findley and D.A. Watt. PASCAL — An Introduction to Methodical Programming. Pitman, 1975

3 PASCAL User Group. c/o Computer Studies Group, Mathematics Department, The University, SOUTHAMPTON, SO9 5NH. (Membership fee £4 per academic year).

5 K. Chung and H. Yuen. A 'Tiny' PASCAL Compiler. BYTE September — November 1978.

6 BYTE — August 1978.

7 N. Wirth. Algorithms + Data Structures = Programs. Prentice-Hall, 1976.

PCW *"Structured Programming and Problem-Solving with Pascal", by Richard B. Kieburtz (Prentice-Hall, ISBN 0 13 854869 2, $ 7.95), is highly recommended* PCW

THE EXIDY SORCERER Owner's Report

Valerie A. Downes

Department of Computing and Control, Imperial College, London

This is an owner's report written after two weeks' experience with the Exidy Sorcerer. It does not claim to be an exhaustive analysis of the strengths and weaknesses of this home computer, but I hope to give sufficient information to assist potential buyers in their choice.

The Criteria

I was looking for a low cost microcomputer that was not just a toy and that would give sufficient facilities for expansion to a full diskette system via a S100 bus (when finances allow!). Already having a TV and cassette recorder in the house, I did not see why I should pay for duplicating these items, preferring to use my money to buy extra computing power or memory. The Sorcerer seemed to fill my requirements, offering a Z80 processor with 32k of user RAM, floating point standard BASIC, a full ASCII character set plus graphics on a *qwerty* keyboard — all for under £1000 (£859 plus VAT). Expansion is available via an S100 bus that allows the choice of upgrading with a number of disc systems.

In fact I ordered the 16k version (£760 plus VAT) from Factor One in Penzance and it was delivered within two weeks (I was quoted six weeks delivery elsewhere). The single unit comes well packed, double boxed with polystyrene padding, measures about 19" x 12" x 5" and weighs some 12lbs. It has a wipeable light brown plastic housing with a robust typewriter style keyboard and separate numerical key-pad. There are several output points arranged along the back — phono sockets for video, TV and Cassette tape outputs and for tape input (the VHF modulator for TV input is not standard but was provided and included in the price!); one 25-pin connector socket for a serial interface to drive a motor-controlled cassette or an RS232 interface; another 25-pin connector socket for a parallel data port, and an edge connector for the S100 bus in the Exidy Expansion Unit. There are also three leads, two manuals and a keyboard template — but more of these later.

To say that there is 16k of RAM is deceptive — the Z80 can address 64k bytes and the Sorcerer design allows a maximum of 48k to be provided as user workspace. The remaining 16k is used by the system and provides a 4k power-on monitor, 4k of control areas concerned with the video display, and 8k for a control program on a ROM PAC that can be plugged into a slot in the side of the machine.

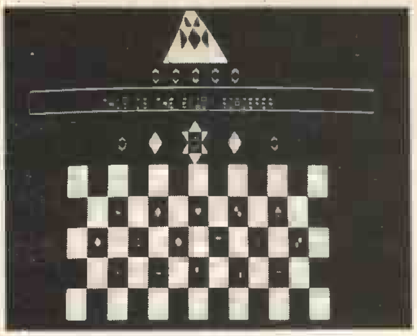
This is one of the most attractive features of the Sorcerer and a BASIC PAC is supplied as standard.

Installation

I had only minor problems in connecting the Sorcerer to my own TV and cassette recorder. The TV, a Ferguson 12" Black and White portable, needed a phono to co-axial lead which had to be made up, since only a phone to phono lead was provided. The *mic* and *ear* leads were supplied for the cassette recorder but my first tests of saving and reloading programs all ended in the response:—

```
ERROR — TAPE CRC ERROR
>
```

which left control in the monitor and needed a PP command to return to BASIC. A subsequent list would then



give assorted gobbledygook, ending in endless repetition of a single character, which could only be stopped by RESET. This error was not documented in the manuals and could be disconcerting. However, I decided that the problem probably lay with my rather elderly Sony TC-100A portable cassette recorder, so I gave it a good clean with cotton-wool buds soaked in methylated spirits. After this I had no further trouble except when accidentally attempting to read human speech.

With everything connected I was left with an unsightly collection of leads, particularly because each component needed connecting to the mains electricity. This could be improved by the use of a single distributor board that would also make the siting of the equipment more flexible.

Documentation

The Sorcerer comes complete with two introductory manuals — 'A Guided Tour of Personal Computing', and 'A Short Tour of BASIC'. The first of these gives a chatty introduction to the equipment, simple BASIC programs and the commands available under the monitor.

Most of this book is presented in a palatable fashion designed for the beginner, but I feel that the sections on the monitor and interfacing fall between two stools: they will either frighten the novice or prove not detailed enough for serious use. The information on connecting the cassette tape recorder did not warn of the error that I encountered, or suggest trying a tape rate of 300 instead of 1200 baud, and there is no description of monitor error messages.

The manual on Basic is well presented but I found it misleading on the subject of strings. There is an unqualified statement that, "You can have as many as 255 characters in a string", which is not true under the default limit set on the string space. Only in an appendix are we told that the string space is normally set to 50 bytes. There are some puzzling omissions like the fact that Print CHR$ (12) will clear the screen and return the cursor to the top left-hand corner. Those not familiar with BASIC will need a more detailed text book if they intend to do any serious programming.

Anyone wishing to do much outside the standard BASIC, like driving peripherals, will require further documentation such as the Sorcerer Technical Manual which is advertised in the back of the BASIC manual but is, apparently, not yet generally available.

Graphics

The video display screen is made up of 30 lines of 64 characters, each character being formed by an 8 by 8 dot matrix. This gives a resolution of 240 by 512 dots over the whole screen although not every point is independently addressable. Even using a standard TV through the UHF Modulator the resolution is acceptably clear and perfectly steady, only giving slight interference when the cassette motor is on.

The display is index mapped by 1920 bytes of system memory, each of which contains the numerical coding of one of the 256 available characters given by the ASC function. Each character is stored in further 8 bytes of system memory. Values can be POKEd directly into the screen area from a BASIC program.

The Sorcerer is supplied with the 128 standard ASCII characters plus 64 predefined graphics symbols — in line with those found on the PET. By using a combination of shift keys the keyboard can also be used to access another 64 characters which can be defined by the user. By sketching the shapes required on graph paper one can work out the pattern of dots required, convert them to hexadecimal and load them into memory, either by using the monitor or by POKEing from BASIC. The process is well described in both manuals and can also be used to *redefine* the 64 supplied graphics characters.

Once defined, symbols for different applications — greek letters, mathematical symbols, or whatever — can be saved on tape and reloaded when required. The template of the keyboard can be used to note these special symbols — it has a wipe-clean surface, from which flow-pen marks can easily be removed, and an adhesive back for fixing to the Sorcerer's casing. These facilities are quite adequate for games, simple graphs and charts, circuit diagrams etc. although they do not compare with the high resolution graphics of the APPLE II.

Plug-In Memory

One of the nicest features is the plug-in ROM PAC which contains the standard 8k BASIC interpreter. This is housed in the type of plastic container usually used for 8-track cartridge tapes but with an edge connector at the open end. It slots into the side of the Sorcerer and is given control by the monitor. This design allows one to change control programs with the minimum of time and effort, and Exidy claim that several PACs are now under development — including an assembler, word processing facilities and an APL interpreter. There will also be blank PACs available into which users can load their own programs.

However, it remains to be seen how robust this system will turn out to be under constant use which could lead to wear on the gold flashings of the connectors. The manual warns against moving the PAC in or out with the power on and I have not experimented to see what damage an accident here might cause.

The Monitor

Control remains with the monitor program if a ROM PAC is missing

BENCHMARK PROGRAMS[1]

	SORCERER	APPLE II[2]	TANDY TRS-80[3]		PET[2]
BM1	1.8	1.3	2.5	0.79	1.7
BM2	10.0	8.5	18.0	5.7	9.9
BM3	20.7	16.0	34.5	10.9	18.4
BM4	22.2	17.8	39.0	12.3	20.4
BM5	24.3	19.1	45.0	14.2	21.7
BM6	37.6	28.6	67.0	21.2	32.5
BM7	53.7	44.8	109.0	34.6	50.9
BM8	9.6	10.7	–		12.3

1. PCW Volume 1, Number 1 page 57/58; 2. PCW November 1978. P.52; 3. PCW September 1978 P.28.

and it can also be entered by issuing a BYE command. It has a useful range of commands — sufficient to allow Z80 machine code routines to be loaded and executed, memory locations to be inspected to assist debugging, user defined graphical characters to be set up, and a tape of monitor commands to be set up and executed. It is through the monitor that various modes of operation can be selected — 1200 or 300 baud tape transfer rate, use of different I/O ports etc.

Using Basic

The standard BASIC interpreter, supplied in a ROM PAC, is compatible with floating point BASIC now offered with a number of systems of similar size. All numerical values are stored to an accuracy of six significant figures and cover the range 1.70141E38 to 2.93874E-39.

However, mathematical operations give rounding errors and it would be foolish to rely on more than five significant figures. Although there is a range of mathematical and trigonometrical intrinsic functions, serious mathematical computations could, and should, be performed far more accurately on a scientific pocket calculator than on a computer of this type.

I ran the benchmark programs listed in PCW Vol 1, No. 1 and the results are given in the table below together with those for the APPLE II, Tandy TRS-80 and PET quoted in earlier issues of PCW.

There is a random number generator which is initiated by a supplied seed, useful for simulation programs. I did a rough test by printing ten-point histograms of a run of numbers produced by a given seed and those tried were all reasonably rectangular after a run of 1000 numbers.

There are PEEK and POKE facilities, USR which allows entry to a machine code subroutine and FRE which tells one how much memory is left un-used. All these allow clever programming and are particularly useful for building special displays or driving peripherals.

There are string variables and various operations that can be performed upon them. But here a word of caution — in default mode the Sorcerer does not like strings, it limits each string to 50 characters and total string space to 50 bytes. This is to maximise space available for programs. However, if, like me, your first test is to type in a published program to see if it works, and that program starts with a large number of statements printing strings, then you are in for a nasty shock — the string space limit includes PRINT statements as well as variables and the first run of the program will give an "out of string" error. The default setting can be overruled by a CLEAR n statement, where n is the number of string bytes required, either as a direct command or as a program statement.

Programs can be transferred to and from cassette tape by using the CSAVE and CLOAD commands, with the tape motor being controlled by hand. An extra connection can be bought which enables the serial interface to be used to control the motor as well as transfers. However, I have not found manipulating the tape controls to be a problem; and programs can be written to issue prompts at the points where tape control actions are required. The ability to issue a CLOAD command from within a program allows over-writing by a replacement program — and this means that suites of programs can be designed with an initial program that interfaces with the user and allows him to choose the next function.

Arrays of any size can be transferred to and from tape with the CSAVE* and CLOAD* commands but, annoyingly, this feature does not extend to strings or arrays of strings. I consider this to be a serious omission since many of the applications to which a personal computer will be put will envolve handling character data-names, addresses, stock component descriptions etc. — which is to be stored on tape.

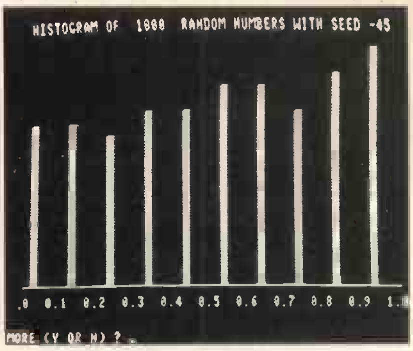
Conversion

However, this problem can quite easily be overcome, without resorting to machine code, by converting strings into their ASCII equivalents (using the function ASC), storing these equivalents in an array and then transferring the array to tape. The reverse process can be effected using the function CHR$ which converts the ASCII code back into character form. In fact these functions do not only work on the 128 ASCII characters but also on the graphics symbols — including user defined graphics. The following short program illustrates the use of this technique.

```
10  REM EXAMPLE OF TRANSFER
    OF STRINGS TO TAPE
20  REM STRING SPACE SET AT 500
    BYTES
30  CLEAR 500
40  REM ARRAY S TO HOLD MAX 50
    CHARACTERS
50  DIM S (51)
60  INPUT "STRING"; S$
70  REM STORE LENGTH OF STRING
    IN S (o)
80  L = LEN (S$)
90  S (o) = L
100 REM STORE STRING IN S
110 FOR K = I to L
120 S (K) = ASC (Mid$ (S$, K))
130 NEXT K
140 REM TRANSFER STRING
150 INPUT "RECORD"; X$
160 CSAVE* S
170 INPUT "OFF"; X$
180 REM REPEAT IF MORE DATA
190 INPUT "MORE DATA (Y or N)"; X$
200 IF X$ = "Y" THEN 60
210 INPUT "REWIND"; X$
220 END
```

The string can later be recalled from tape by using CLOAD S followed by:—

```
L = S (o)
S$ = " "
FOR K = 1 to L
S$ = S$ + CHR$ (S (K) )
NEXT K
```



I have used this technique for a program that handles a cassette tape file of names and address. Each line of the data is input into the same string variable which is then converted into one row of two dimensional array, ready for transfer to tape. The conversion process is so fast that there is no noticeable delay in the prompts for each line of input.

To speed up the process of typing programs the Sorcerer provides almost all of the BASIC key words as single key inputs, using the GRAPHIC shift in the context of the BASIC instruction. The program looks rather strange with — for PRINT π for GOSUB and the like, but a LIST restores the full words. The key words are not printed on the keys but I have copied them from the manual to my keyboard template (more than one of these would be useful). At first I found using them slower than typing the full word since I had to search for the correct key, but the positions of the commonly used words soon became familiar.

There is no line editing, hardly to be expected in an 8k interpreter, but single characters can be erased on a SHIFT and RUB and a whole line aborted with @. The control shift and C breaks into execution of a program and allows one to input directly executable statements before CONT continues execution. This facility is very useful for debugging programs, since it allows the contents of variables to be inspected or changed. If all else fails, simultaneous pressing of the *two* RESET keys, a good precaution, will take one back to the start of BASIC — losing any program but not user-defined graphics. Although reset is rarely necessary. I would advise frequent saving of program versions during development.

One very annoying aspect is the inability to list a program by the screen-full: the LIST command lists the whole program or *from* a given line, but not *to* a given line. LISTings can be caught as they scroll past by pressing the RUNSTOP key but this only holds while the key is depressed. It would be an improvement if this key was fitted with a lock, like the SHIFT LOCK Key, so that both hands can be free for making notes or photographing the screen.

Conclusion

The Sorcerer represents excellent value for money either for the home enthusiast, teacher, or small business man who wants something with expansion potential. The minimum configuration allows serial processing (if you do not mind driving the tapes by hand), the graphics potential is fine for both game playing and diagrams, the standard BASIC gives access to a large printed repertoire of established software and the ROM PAC promises easy expansion into new languages. My only words of caution would be to wait and see how durable the PACs prove and how soon the projected software actually becomes available.

Since this article was written, two months ago, Valerie Downes has attained greater familiarity with, and expertise on, the Sorcerer.


WE'LL SHOW YOU EVERYTHING
Because we've got the biggest and widest range of micro-computers, there's more for you to bite on at a Byte shop.
So you're not only sure of finding exactly what you want: you can take the opportunity to experiment before you buy.
And because we're backed by the huge financial resources of a major investment group, we'll be here tomorrow as we are today. All over the UK.
So whether you want a micro-computer for your home, your business, for industry, for education — or if you'd just like to find out which model you get on best with — you'll find a visit to a Byte Shop a new and invaluable experience.
You can call at a Byte Shop any time from Monday to Saturday.
Z-2D
Computer System
the BYTE SHOP Ltd
The Byte Shop
426/428 Cranbrook Road,
Gants Hill, Ilford, Essex
Telephone: (01) 554 2177
Telex 897311
Branches at:
LONDON (WEST END):
48 Tottenham Court Road —
tel (01) 636 0647
NOTTINGHAM: 92a Upper
Parliament Street —
tel (0602) 40576
MANCHESTER: 7-8 Corn
Exchange Building,
Fennel Street —
tel (061) 834 0220

SMALL SYSTEMS ARE GOOD FOR SMALL BUSINESSES

John Braga
Computerland, *Nottingham*

PCW *This reasoned response to a previous* PCW *article is a good indication of the quality of the people engaged in the business of selling microcomputers* PCW

Mr. Worsley's article in the June issue of PCW, dealing with whether or not small businesses were ready to adopt microcomputers in large numbers, raised a number of interesting points.

The fact that I disagree with most of his conclusions has prompted me to reply!

One point on which we do agree: Small businesses will turn to computers for one reason only — if it makes economic sense for them to do so. But anyone who is involved with small companies and who has had the benefit of a systems analysis training must agree that there are countless opportunities in almost every business to increase efficiency by the *judicious* use of computers. Mr. Worsley makes two important points with which I agree, namely:—

- most businessmen are not prone to seek innovation for its own sake; they will wait until the product is presented to them in an easily digestible form.
- However well a system is designed, installing a computer is going to be much more of an upheaval than installing dictating equipment or an electric typewriter. The smallest system may well involve a fundamental change in the way a company operates.

Many businessmen have not thought deeply about computers, believing that they are an excellent invention — for someone else! If pressed to say why they have not installed a computer they have sheltered behind three main defences; every computer salesman is only too familiar with them!

1. Computers are too expensive.
2. It is quicker by hand.
3. My business is too complicated to program on a machine.

The revolution in technology of the past two years has meant that the first argument above has been rendered totally invalid. There is now no reason why any company turning over £500,000 or so should not afford a computer system which now costs little more than a photocopier and less than a Chairman's company car! The other two arguments will persist but are generally due to an ignorance about how a computer works.

This last point is not meant to be a criticism of businessmen in general; it is an unfortunate fact that computers *are* complicated to understand and to install. It is a challenge to the software house, systems analyst, micro manufacturer and retailer to make systems that are flexible and easy to use; but, since each company operates in a different way, each company will ideally want a different computer system. Some bending will therefore need to take place, and either the company or the computer system will have to adapt!

I am certain, however, that businesses will install large numbers of micros. Any system needs three ingredients: good hardware, operating system, and application software.

We already have the first two, and application software is becoming available in increasing quantities. As the professional software houses become aware of the falling price of machines we will see a large increase in the amount of *good* packages becoming available. No company producing systems can afford to ignore the micro market (although some have shown every sign of wishing it would go away, and leave them in peace making large margins on 'expensive' minis!) A company will be left stranded if its software will only work on a £20,000 system where the competitor's will run on a £5,000 machine.

Remember too that COBOL compilers have now become available for the micro and this means that a large amount of existing software can speedily be converted to run on micros. My own company is already doing this and there are hundreds of software houses that will be doing likewise over the next few years. The market is too large to ignore.

In summary, there is no earthly reason why good software should not be available on low-priced systems. The price is likely to vary widely, depending on the size (and overheads) of the company producing it, but we are already seeing packages *now* available that can be safely used by the business entirely new to computers. Nor is there any reason to think, as Mr. Worsley does, that small computer companies will be driven out of the market by big companies. Small computers have several advantages when producing software for small businesses:—

1. They are small themselves and can relate to the small business's problems and systems. A rapport can easily be established between buyer and seller, which is not so common when the seller is the sales office of a large multinational.
2. They have small overheads so can frequently sell at lower margins. This also means that they can look at business which larger companies do not find profitable — remember, a large company works best by selling a *standard* product in large numbers, and is frequently less willing to tailor to individual requirements.
3. They can be very flexible, changing their products at short notice to meet a new market place.

For this reason I believe a professionally run small company will survive in the micro world, though it may need agile footwork at times to adjust! I also believe, though, that there are too many companies, large and small, which do not deserve to survive the next few years because they are not offering a competitive service or product.

So where does this leave the owner of a small business who is

wondering whether or not to buy a computer? Here are a few guidelines which may help him to decide:

1. Read as much about computers as you can. There are several good books and magazine articles which cover the application of small computers to business problems without introducing too much jargon.

2. Attend several different demonstrations. This will educate you in the way of different machines and — just as important — enable you to weigh up the capabilities of potential suppliers.

3. When considering what part of your business could be 'computerised', concentrate on one simple application rather than on an integrated system affecting every facet of the operation. If you devote your energies and time to getting one application up and working, your computer will be earning its keep while you are developing subsequent phases. Also, you will learn a lot from the initial stage and may well modify your subsequent development in the light of initial experience.

4. Do not try to dream up 'Your Perfect System' before the machine is even installed. You are dealing with too many variables to look ahead with precision. You must of course look ahead far enough to be sure that what you may want to do is feasible on the equipment you are considering, but what you *think* you need is probably very different from what you will *actually* want, with experience of a computer system under your belt.

5. Learn to distinguish between a micro suitable for playing Startrek and one suitable for Sales Ledger. There is a world of difference between the hobby and business system. Many manufacturers and dealers do not distinguish, perhaps preferring to try to catch both markets with the same machine (it makes life simpler for them that way!). There are some vital differences, however, so make sure you understand them before purchase.

6. While on the subject of dealers, most people are likely to need advice and help from their supplier after they have bought. So choose him with care — if he seems more interested in the electronics of his box-of-tricks than in applying the machine to solve your particular business application problem, go elsewhere. You need the advice of a knowledgeable businessman, not an engineer, at this stage.

7. Finally and perhaps most important, never forget that in the world of micros, as in other fields, you get what you pay for! If a machine costs 50% less than another system there is usually a good reason. It always surprises me, for instance, that people expect a £500 printer to perform as well as a £1200 model and last just as long. In a fiercely competitive market prices are generally what they are for a good reason . . . All of which does not mean a cheap system need not be excellent value, but do make sure you understand what has been left out. All systems have their limitations and you should be suspicious if the salesman is too glib on this point.

The responsible dealer or manufacturer realises that by educating the computer buying business world to become more knowledgeable and critical he stands a better long-term chance of selling his products. Satisfied businesses are his best reference. So please let us work jointly to keep expectations and standards high!

ON THE LINE

David Hebditch

Communications I/O

I have earlier in this series, considered:
* techniques for representing data bits on analogue telephone lines
* the use of modems and acoustic couplers for converting bits from digital form to analogue form (and vice versa)
* timing and character formats
* the V21 interface between the computer and the modem.

We have now reached the stage where I can describe what needs to be done to get a seven-bit ASCII character from the main memory of our micro-computer across the V-series interface so that the modem can modulate it on to the telephone line. The reverse operation to get bits assembled into sensible characters is also relevant.

To save you having to revise the previous articles in the series, this is what needs to be done:

TRANSMIT
1. Generate a parity bit and append it to the character
2. One start bit and at least one stop bit will also have to be appended.
3. The resultant framed character then needs to be *serialized,* i.e. passed bit-by-bit to the 'transmit data' lead of the interface.

The regular reader will remember that the last operation is necessary because all data transmission takes place in a *bit serial* fashion, but data is held in the micro in *bit-parallel* (or character-serial) form.

It should come as no surprise that the sequence for incoming data is:

RECEIVE
1. *Deserialize* the bits into a 'framed character' form.
2. Strip off the start and stop bits.
3. Check the parity bit, set status accordingly (good or bad) and dispense with the bit (unless it is going to be taken all the way into RAM).

There is another important stage:

4. Tell the processor that a character has arrived.

Exactly how this is done will be dependent on the architecture of the micro being used. At a simple level, a status bit may be set. Increasingly, an interrupt may be raised.

Other things need to be done (e.g. assemble the incoming characters into a meaningful message) but the steps I have described so far are

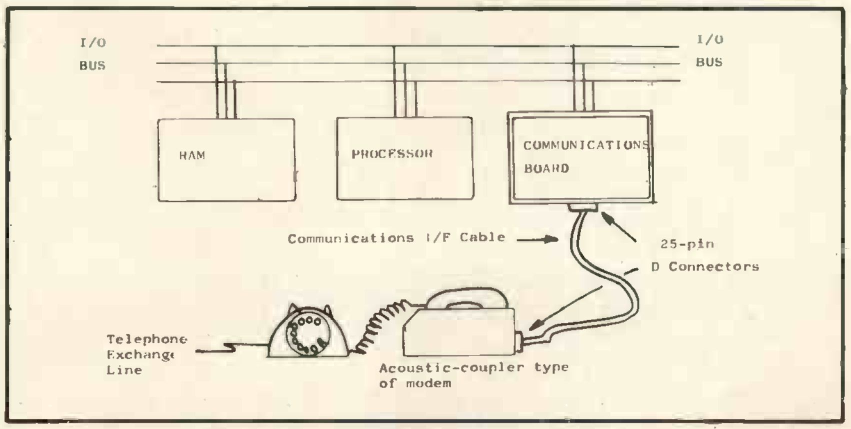


Fig. 1. Single-Line Communications Board

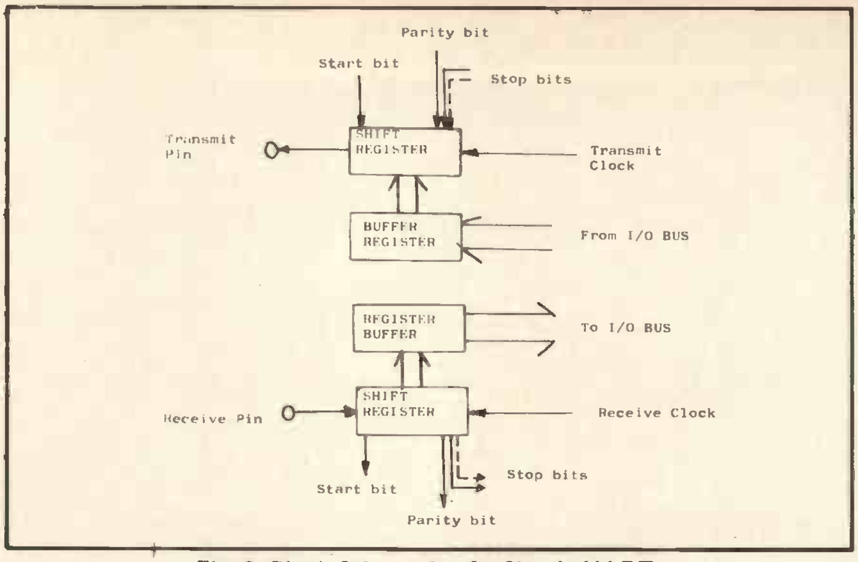


Fig. 2. Block Schematic of a Simple UART

usually implemented in hardware and the rest in software. This month I shall restrict myself to the hardware which usually takes the form of an 'add-on' PC Board which interfaces to the micro's I/O bus on one side and to one or more communications interfaces on the other. A single-line communications board is illustrated in Figure 1. Note that some Single-Board computers incorporate serial I/O capability as standard.

I admit to being old enough to remember the (pre-LSI) days when all the above functions were performed in software. Yes, that's right, you had to write programs which deserialized characters and stuffed the bits down the telephone line. Fortunately, that did not last too long and boards based upon TTL components soon became available. The big breakthrough came with the development of the Universal Asynchronous Receiver/Transmitter (UART). The UART was invented by any semiconductor company you care to ask.

Figure 2 shows how a UART chip does what we need.

Many of you will have already come across UARTs being used in cassette tape and printer interface boards. For the sake of completeness, I should tell you that there are now a whole family of UART-like devices from a variety of suppliers.

On the simpler ones, the facilities required (e.g. bit rate, odd/even parity, number of stop bits) are selected by the use of jumpers. More recent models are sometimes called (confusingly, in my opinion) 'programmable'! What it means is that the option can be specified via a parameter register, the values of which can be loaded from a program. The arrangement is easier to understand if it is described as having 'software selectable options'.

As the density of chips has increased the functions performed in the UART have increased accordingly. 'Super UARTs' are known as Universal Synchronous/Asynchronous Receiver/Transmitters (USARTs or just USRTs) and are capable of performing advanced checking functions (Cyclic Redundancy Checking), data link controls (more of which next time) and perform timing functions at both block and character levels. As few of these functions will be needed for our planned Personal Computer Network, we only need to be concerned about the more simple UARTs.

For the homebrew enthusiast and those who like more detail: Figure 3 shows a more complete picture of a typical communications board. Not being famous for my hardware capabilities, I had my good friend Dr. Martin Healey, of University College Cardiff, draw up this schematic. The additional logic needed to support the UART is as follows:

A: Raises 'Request-to-Send' on the V-Series interface and checks for 'Clear-to-Send' which indicates that the modem is sending a stable carrier and is ready for the data bits.

B: The baud-rate generator receives a clocking signal from a crystal oscillator or from the I/O bus (e.g. 2 MHz on the S100). This is then subdivided to produce the required bit-rate (e.g. 110, 300, 600 or 1200 bit/s). Note that, with many UARTs, the transmit and receive rates may be different.

C: The optional interrupt logic is used to tell the processor that the UART can receive the next character for transmission or that an input character has been received satisfactorily.

D: The address select and interface logic looks after the inter-working with the processor.

So far, so good. Next time we shall look at data link control procedures.

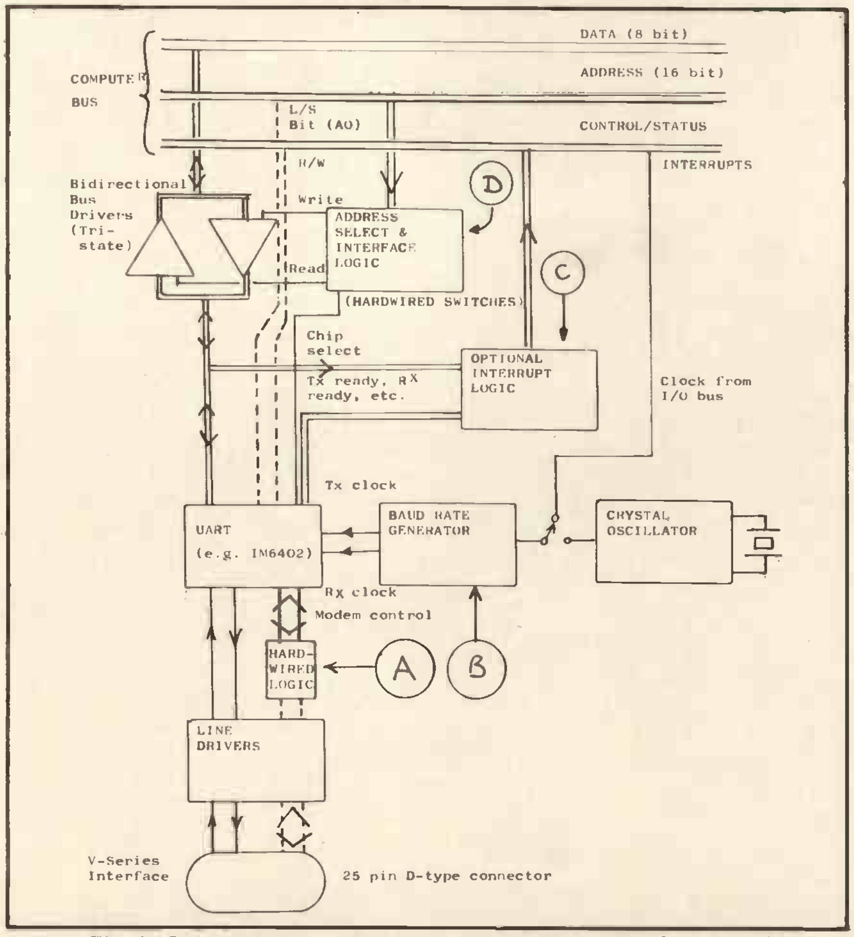


Fig. 3. General Schematic of a Communications Interface Board

D to A Converter

John Coll

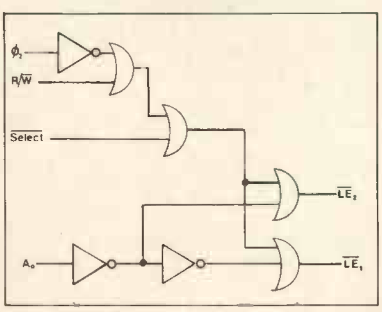
Oundle School, Peterborough

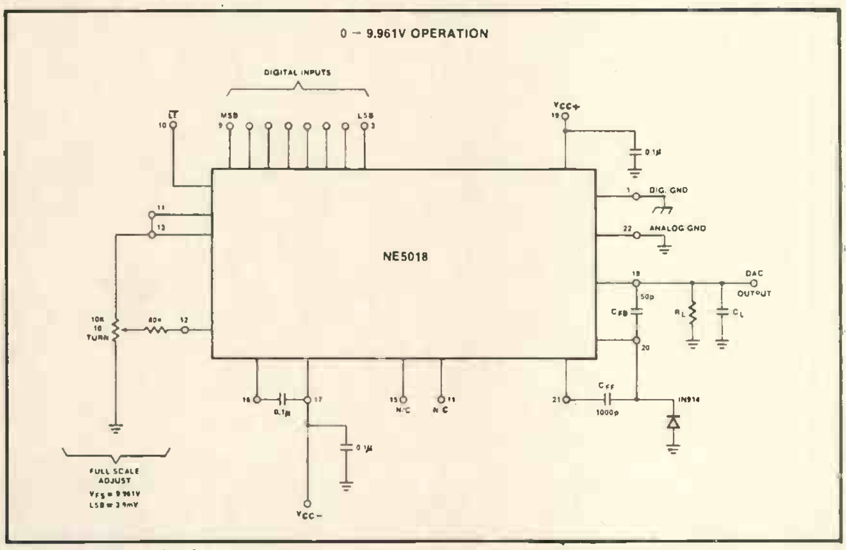
A Digital to Analogue converter is used to produce signals to drive oscilloscopes or amplifiers and loudspeakers to draw pictures or play tunes. It can also be used in servo systems.

Signetics have produced a single I.C. micro-processor D to A converter which makes the whole process simplicity itself. It is quite a lot nicer than the Ferranti ZN425 since it includes a Latch enable input which greatly simplifies connection to the data bus. The part number is NE5018 and it is suitable for most micros.

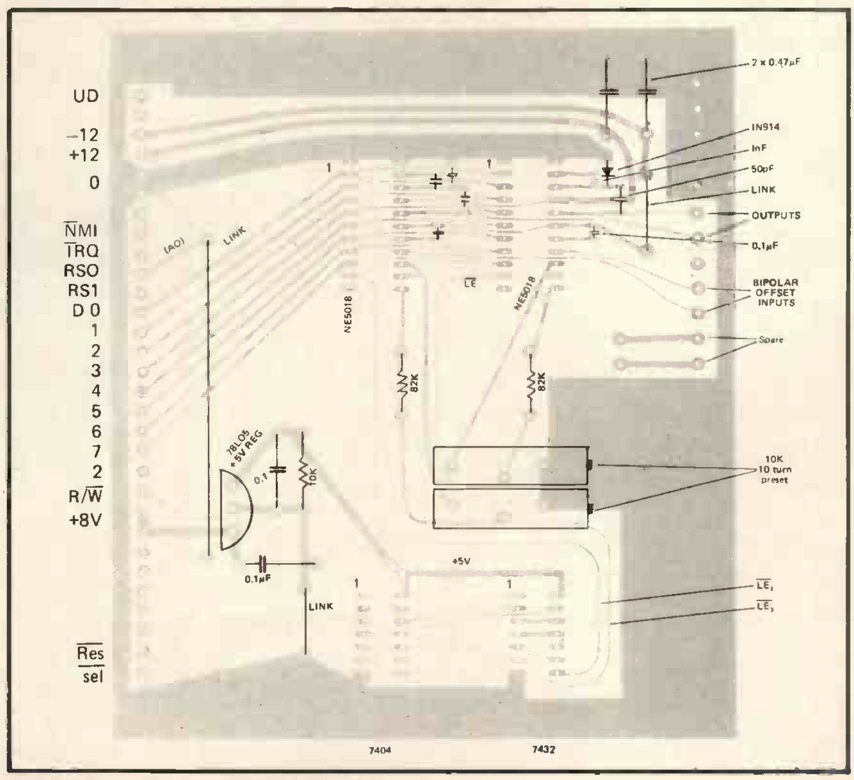
I have designed a printed circuit board for the SWTPC system (since that machine was available) with 2 D to A converters on one board. This enables X Y plotting to be done on a 'scope screen — and music to be played in stereo I suppose! The circuit is shown below together with a reduced copy of the artwork used.

The final design gives out 0 to 10 volts with a settling time of about 2 micro-seconds.





The Signetics D/A Converter



Top View of PCB for SWTPC System

BRISTOL COMPUTER CENTRE

At the Bristol Computer Centre we have a complete range of professional micro-computer systems. Not kits or toys but fast, reliable systems that will get the job done . . . . . without fuss.
We are the main agents in the South West for Modata, SWTP and Commodore. So we can supply just about any system, for any task – from a 4 k-byte PET to a 28 m-byte advanced Modata hard-disc communicating system.
And all our systems have a choice of visual display units, printers and other peripherals plus full maintenance support.
Software packages including stock control, order-processing, word-processing, accounts, etc can give you instant control of your business.
Telephone now for a demonstration and see for yourself.

THE BRISTOL COMPUTER CENTRE
MICRO HOUSE,
ST. MICHAEL'S HILL,
BRISTOL
TEL: 0272 23430

XITAN SYSTEMS

Southern England stockists for **CROMEMCO**

Selected nationwide by Government departments, Research Establishments, Universities, Professional Consultants, Assurance Companies, Industry and the serious hobbyist. Features include a range of software equalled by no other manufacturer for price and performance. Fortran, Basic, Cobol, Z-80 Macro Assembler, Trace, and Word Processing, all at only £85 each. Xitan Systems can supply CP/M for use on Cromemco equipment with 8 in. floppy discs.

THE machine for professional use

XITAN SYSTEMS,. 23 CUMBERLAND PLACE, SOUTHAMPTON SO1 2BB. TEL: (0703) 38740

Also suppliers of: PET · HORIZON · PT SOL · DYNABYTE MEMORY · BOOKS · VDUs · PRINTERS

Practising a little Micro-control

Mike Dennis

Much has been said on busses and the basic schematic layout of personal computers. However, little has been published on what the various *control signals* that come out of the CPU chip do, and how you can use them. To list them all would fill this magazine and so I intend to fill the gap with reference to the Z-80.

The basic design

Figure 1 shows the familiar block diagram of a typical computer. *In order to determine what control signals may be necessary, we must list those operations that are required from the computer.* Table I shows them.

1/ CPU sends data to memory
2/ CPU receives data from memory
3/ CPU sends data to I/O (Input/Output)
4/ CPU receives data from I/O.

TABLE I

Larger computers like the PDP-11 have the added facility to move data to and from memory and I/O but in the micro world this is generally not the case and so transferring data from I/O into memory would require a combination of operation (4) then (1). Table I can be re-arranged thus:—

a/ The CPU communicates with either memory or I/O
b/ The CPU either sends or receives data.

TABLE II

The system needs to know *which one of these is to take place* and the control signals do precisely that. There are four control signals from the Z-80 and they are:—

1/ $\overline{\text{MREQ}}$ – indicates CPU wishes to communicate with memory and that the address bus is stable and valid. ('Memory REQuest.')

2/ $\overline{\text{IORQ}}$ – indicates CPU wishes to communicate with I/O and that the address bus is stable and valid. ('I/O ReQuest'.)

3/ $\overline{\text{WR}}$ – WRite – CPU wishes to send data – the data bus is now stable and valid.

4/ $\overline{\text{RD}}$ – ReaD – CPU wishes to receive data

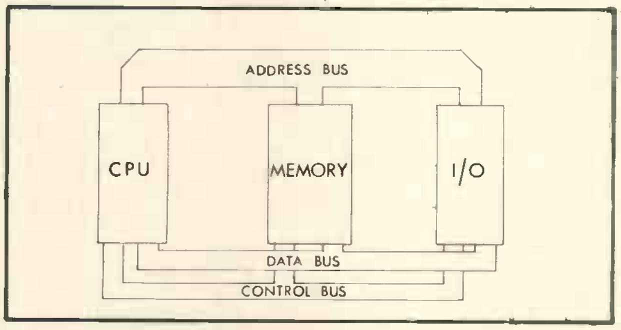


Figure 1.

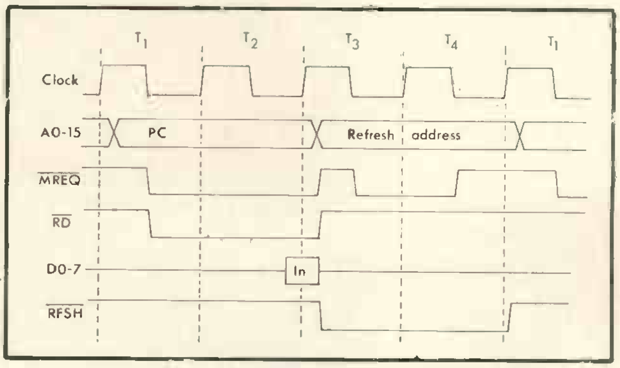


Figure 2.

These signals are all automatically generated by the CPU. Moreover, as far as the CPU is concerned, they are generated at the right time *with respect to each other.* It is up to the user to do something useful with them!

For example, when operation (2) is in progress, the CPU would make both $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$ active. The bar over the top means that they are 'active low', ie when they are active, they are at logic 0 or 0v or whatever you like to call it. Table III shows the logic state of the four control signals for each of the operations.

(Table I operations)

	MREQ	IORQ	RD	WR
(1)	0	1	1	0
(2)	0	1	0	1
(3)	1	0	1	0
(4)	1	0	0	1

TABLE III

Some computer manufacturers, eg Tandy, gate these controls together and derive four different ones:— $\overline{\text{MEMRD}}$ and $\overline{\text{MEMWR}}$ (memory read and write) and $\overline{\text{IORD}}$ and $\overline{\text{IOWR}}$. Personally, I think this is a waste of time as one can lose the flexibility of the four original signals.

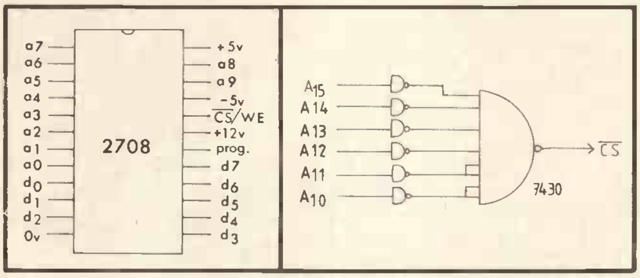


Figure 4.

How are the control signals used?

The basic concept of the tri-state bus is that there is only *one* device on the bus at any one time that is supplying any data — the talker. All the other devices are sitting across the bus and listening. The control signals help discriminate between the talker and the listeners and allow them to talk or listen at the correct time.

After reset is hit, all the internal registers are cleared and the program counter is forced to a specific value or address, depending on which CPU chip is used. The contents of the program counter are then transferred to the address bus as an address and, in the case of the Z-80, this address is 0000. The Z-80, therefore, expects to find the first program instruction at this address.* This instruction will reside in memory, the CPU wants to read it and so $\overline{\text{MREQ}}$ and $\overline{\text{RD}}$ *will be made active* (in this case, active low) by the CPU — see Figure 2, which has been taken from the Z-80 Technical manual. Notice that the whole operation takes *four* clock cycles.

The next step is to apply the controls to memory in such a way that the data at address 0000 finds its way onto the data bus so that the CPU can read it.

How Memory Uses Control Signals

Figure 3 shows the pin connections for a 2708 EPROM. This device has a capacity of 1k bytes (2^{10} bytes) and so 10 pins are needed to access all the memory locations within the chip. The data is output onto the data bus via pins 9-11, 13-17 inclusive, AND only when pin 20 ($\overline{\text{CS}}$; CS = Chip Select) has been taken low otherwise the o/ps (outputs) of the 2708 are tri-stated. The CPU has put out 0000_H or 0000 0000 0000 0000 onto the address bus but the 2708 is *only* interested in the logic state of the ten lower bits of the bus (A_0 to A_9). The remaining six bits can be used to uniquely define the 2708's position in the possible 64k bytes of memory that the micro *could* address. There are two methods of decoding and they are:—

**There are hardware tricks that fool the rest of the computer into thinking that a different address is present on the bus. I do not intend to go into them now.*

1/ Full decoding

Here the remaining six bits are combined via a logic circuit that will only give an output when a certain bit pattern is present. The 2708 must respond to a base address of 0000 and so the circuit of Figure 4 will decode this. The output from the decoder is 0 which, conveniently, is needed by $\overline{\text{CS}}$ in order to enable the 2708.

2/ Partial decoding

This is only suitable for a system of limited size and one that is not going to expand. Assuming that it consists of, say, 4k bytes of RAM and 2k bytes of ROM, then each of the spare six address lines could be used *directly* as the Chip Select ($\overline{\text{CS}}$) for each k byte of memory. Figure 5 shows how they could all be used and the various base addresses that they will each respond to. Notice that address 0000 *still* needs to be completely decoded and this has not been shown. Care must be taken when programming to prevent any attempt at reading or writing to addresses that would select two or more blocks of memory. For example, an instruction to read data at address F000 would be disastrous as all four blocks (A, B, C & D) would be selected simultaneously!

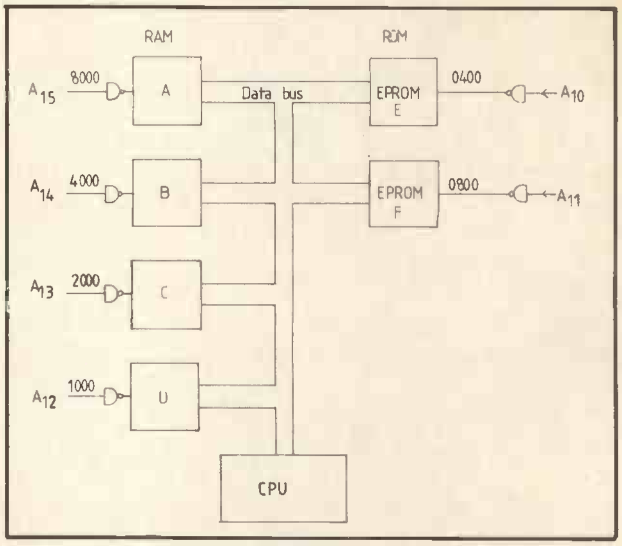


Figure 5.

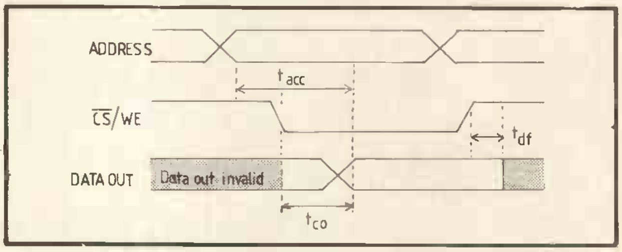


Figure 6.

Full decoding is certainly well worth while and always pays dividends in the end as bus conflicts are avoided. You perhaps are wondering about $\overline{\text{RD}}$ and $\overline{\text{WR}}$? Well . . . .

All memory has an inherent delay between the address being presented at the memory chip's pins and then *valid* data appearing on the output pins. There will amost certainly be rubbish on the pins until a certain time has passed — the access time. Access time is often quoted as part of the chip number, eg —2102-4 (450nS) access time. It can also be found from data sheets, and Figure 6 shows that for the 2708 while Table IV gives the timings.

Symbol	Parameter	Min	Typical	Max	Unit
tacc	Address to o/p valid		280	450	nS
tco	Chip select to o/p valid		60	120	nS
tdf	Chip select to o/p tri-stated			120	nS

TABLE IV

We are told that the data will be valid at a maximum time of 450 nSecs after the address has been presented *and* $\overline{CS}$ has been low for at least 120 nSecs. Relate Figure 6 to Figure 2. If we are using a clock frequency of 2MHz then one clock cycle will take 0.5uS or 500nSecs. The CPU *samples* data on the rising edge of T3 and so our data must be valid and stable on the data bus by this time.

Have we enought time? The address is present on the 2708 for nearly all of T2 and most of T1 before the critical edge of T3 occurs. That is nearly 1000 nSecs and more than enough for our 2708. We can use Figure 4 to chip select the 2708 and further combine this signal with $\overline{MREQ}$ and $\overline{RD}$ to enable a separate chip that buffers the outputs of the 2708 from the data bus. This is shown in Figure 7.

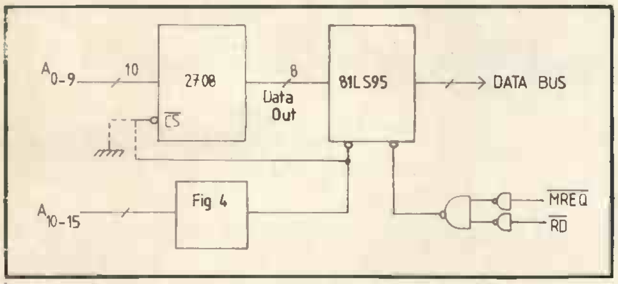


Figure 7.

In fact, with Figure 7, there is no reason why CS of the 2708 shouldn't be permanently tied low provided that the tri-state buffer is only enabled when A_{10} — A_{15} are low and $\overline{MREQ}$ and $\overline{RD}$ are active — shown dotted. A similar process can be applied to RAM and this is discussed next.

Read and write

RAM really is an awful word! We don't actually random access the memory; if we did, there's no telling what data would come out. What is done is Read/Write but no-one seems to call it that. Figure 8 shows the timing diagram for a Memory Write by the Z80; Figure 9 — the timing diagram for a 2102 RAM; and table V gives the timings.

Symbol	Parameter	Min. Time (nSec)
t_{wc}	Write cycle	450
t_{aw}	Address to write set-up	20
t_{wp}	Write pulse width	300
t_{dw}	Data setup	300
t_{cw}	Chip enable (select) to write setup	300
t_{dh}	Data hold time	

TABLE V

A brief explanation of some of the above follows now:—

t_{aw} — the address must be valid for at least 20nSec before write goes low.

t_{dw} — the data must be valid 300nSec before the write pulse goes high and moreover:—

t_{dh} — data must remain valid for 0nSec after write goes high.

Comparison between Figure 8 and 9 shows that the timings of the control signals from the Z80 are capable of providing the times shown in Table V. There is a fair margin as well. For example, data remains valid for some time (almost 200nSecs) after write goes high and so there is plenty of room for a chip that has a longer t_{dh}. Figure 10 shows how a 2102 could be connected up to a buffer and address decoder. Work out what base address it will respond to — answer given at the end*.

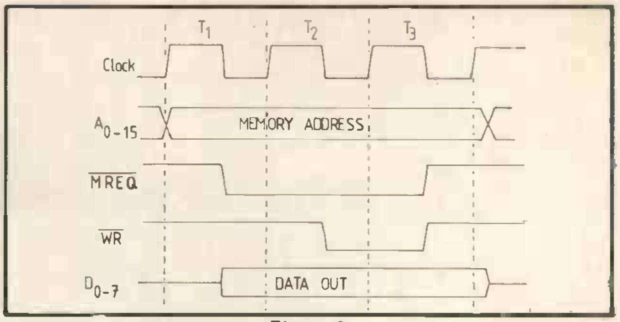


Figure 8.

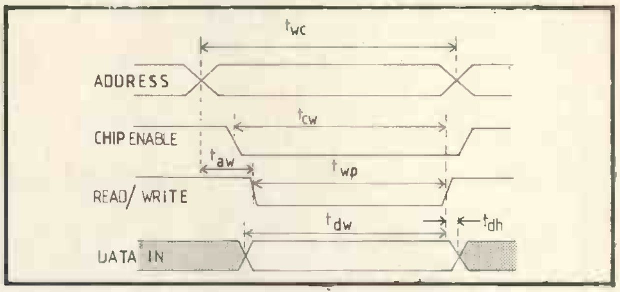


Figure 9.

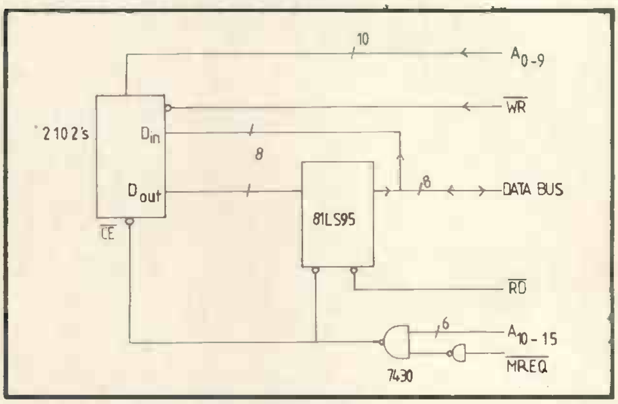


Figure 10.

What about I/O?

So far, we have only considered memory. I/O is very similar but this time, as mentioned previously, $\overline{IORQ}$ is used and not $\overline{MREQ}$. $\overline{RD}$ and $\overline{WR}$ still perform the same functions. However, only the bottom eight address lines are used to decode which port is being accessed. Eight lines allow 256 different ports to be decoded and $\overline{RD}$ or $\overline{WR}$ will tell them which direction the data is going. Sometimes, several ports exist within the one chip (eg PIO's which have two data ports, two command or status ports at least) and the chip performs the decoding for itself for these ports. Timing is not a problem if you use peripheral chips from the same stable like the Zilog PIO or CTC. With other I/O devices, timing may or may not be a problem. It all depends on the speed with which the CPU transfers data to and from the device. Those devices that transfer data slowly, like UARTS, are easy, whereas those that transfer data quickly, like CRT or floppy disc controllers, are more difficult. You have to get the data sheets, burn the midnight oil and try it for yourself.

That completes this quick look on control signals but one further point remains regarding convention. When the CPU chip is reading data from either memory or I/O, then that device is *writing to* the CPU, so should the $\overline{RD}$ or $\overline{WR}$ line go active? Convention states that it is the action of the CPU that dictates which line goes active and, in this case, it will be the $\overline{RD}$ line — common sense really, since it's the CPU that generates the control signals and it should know what it's doing!

**(Ans: FC00)*

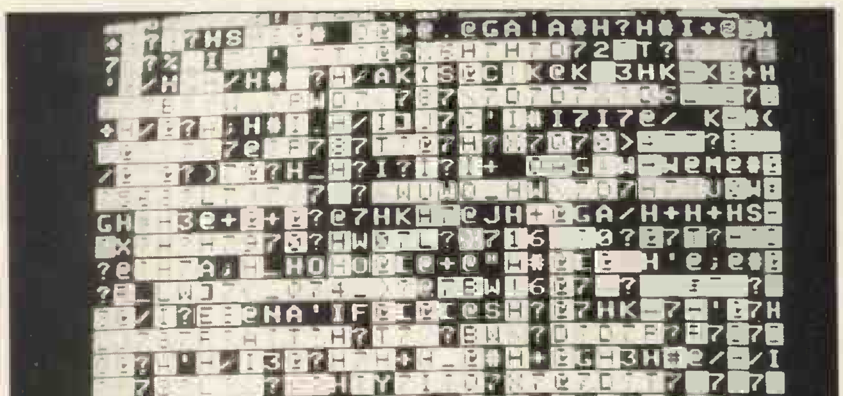
VISIONLINK

MORE VDU BITS

Interfacing & Software for Superscamp's VDU

W.G. Marshall

The first article (Jan '79) on Superscamp's VDU outlined the basic hardware module, without showing in detail how the computer (in my case SC/MP based) *controls* the appearance of characters on the screen. The suggested hardware and software interfacing will now be described, together with the optional 'extras' of hardware screen clear and keyboard repeat. The advantages of using memory-mapped RAM should become apparent, as any amount of cursor control is possible – with a large enough program! The subroutine listed in this article however, is limited to four main functions: *display character, carriage return, line feed with scroll* and *backspace cursor.* As a result the storage requirement for keyboard and VDU handler is less than 200 bytes. The backspace cursor function is included so that the subroutine may be used in place of the teletype routines on the end of the NIBL BASIC.



Random Display when power first applied.

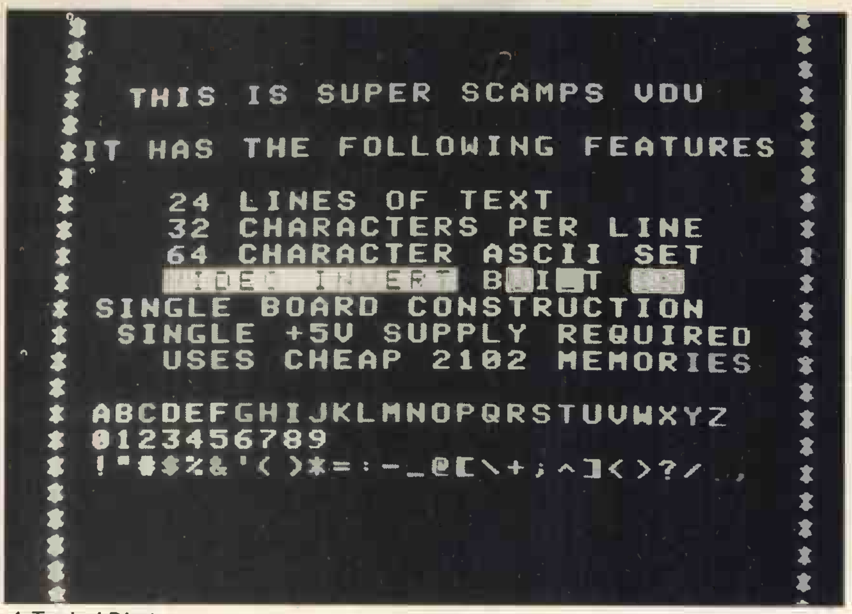
VDU Interface & Screen Clear

The first control signal that we need to generate is $\overline{\text{VSEL}}$, the $\overline{\text{VDU}}$ select. The simplest way of doing this is to use one of the upper address lines (A10 – A15) suitably inverted, but this results in inefficient use of storage space as it means that a number of addresses will access the VDU. If we decode the top four address lines (A12 – A15) with a 4–16 line decoder, sixteen 4k page select signals are produced. Now decode the next two lines (A10 – A11) with a 2-4 line decoder to yield four 1k block select signals. If a page select and a block select are 'anded' together, every 1k block within the 65k range can be addressed uniquely. One of these 'anded' signals become our $\overline{\text{VSEL}}$. (Fig. 1.)

The 1k VDU RAM can now be placed anywhere in 65k.

The next signal to consider is RW, which is easily derived from the MPU write data strobe (NWDS in the case of SC/MP). Simply route NWDS through one of the spare multiplexers of X17, in such a way that it is only allowed to influence the RAMs while $\overline{\text{VSEL}}$ is low. i.e. during computer access.

The tri-state enable $\overline{\text{TSE}}$ is obtained by inverting NWDS and send-



A Typical Display

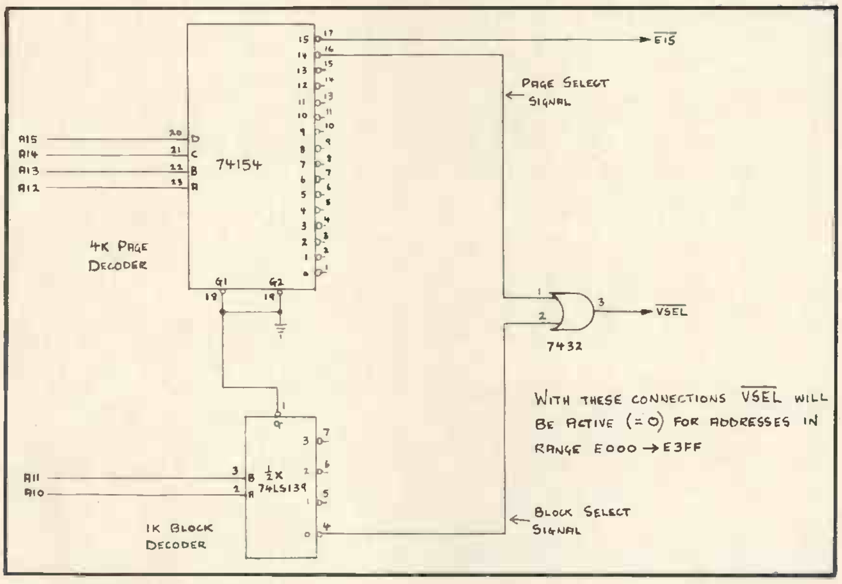


Fig. 1. Address Bus Decode

ing it through a second spare multiplexer in X17 (Fig. 2.) In a similar way to $\overline{RW}$, $\overline{TSE}$ can only switch on the output buffers during a read computer access when $\overline{VSEL}$ is low.

With the data bus $\overline{SDO\text{-}SD7}$, the address bus A0-A9, $\overline{VSEL}$ and NWDS, the computer now has complete control of the VDU RAM, so now let's move on to the manual control, SCREEN CLEAR. Why is this required? The answer lies in the second paragraph of the first article showing the display when the VDU is switched on. All we see is the random data set up in the RAMs. For a blank screen, every RAM location must contain the ASCII code for a space, 20 in hex.

The rest of the hardware in fig. 2 is designed to force this code into every location when the front panel push button SCREEN CLEAR is pressed momentarily. When the clear button is pressed, CLR goes low taking $\overline{RW}$ low all the while $\overline{VSEL}$ is high. In this way we can force data ito the RAMs without computer access. This data will be the space code as $\overline{CLR}$ also switches over the data-in multiplexers X30 and X32 from the system data bus SDO-SD7, to a fixed code 20.

Note the two inverters X16a,b. These merely form a delay so that when the button is released, the write strobe is switched off the RAMs before the data-in multiplexers switch back. The button is de-bounced by latch X16c,d. While the button is pressed, ASCII space is written into every RAM location as the video address counters cycle through all possible addresses, thus 'clearing' the screen.

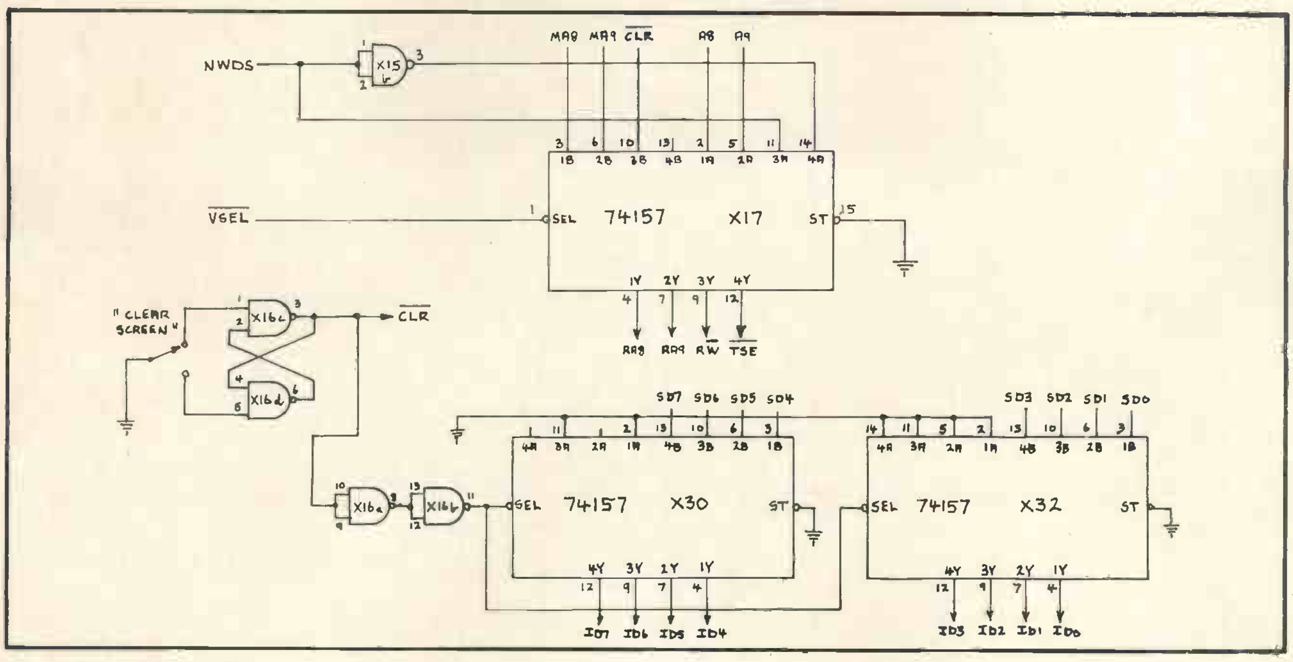


Fig. 2. VDU Control

Keyboard Interface

The keyboard is treated as a peripheral independent of the VDU, and the MPU addresses it as a single read-only memory location. The KB6, in common with most keyboards available, has seven data output lines (eight if parity is used), and a data ready strobe, making interfacing simple.

Two approaches are possible: the first is to connect the strobe to a non-maskable interrupt on the MPU, the second to latch the strobe and have the processor examine the latch periodically to see if a key has been pressed. The interrupt method is not very convenient on a SC/MP system, as the MPU only has a single interrupt arrangement requiring a pointer register to be reserved for the handler.

I chose to implement a latched set-up using one of SC/MP's sense inputs — SENSE B. The latter is tested by program when required, to 'sense' a key application. Refer to the hardware diagram Fig. 3. Having sensed the keystrobe, the program addresses the keyboard data buffers X33, X34 and switches them on by taking $\overline{KSEL}$ low, at the same time clearing the strobe latch X35a. The REPEAT button merely prevents the latter event from happening, so that the program keeps reading the same data over and over.

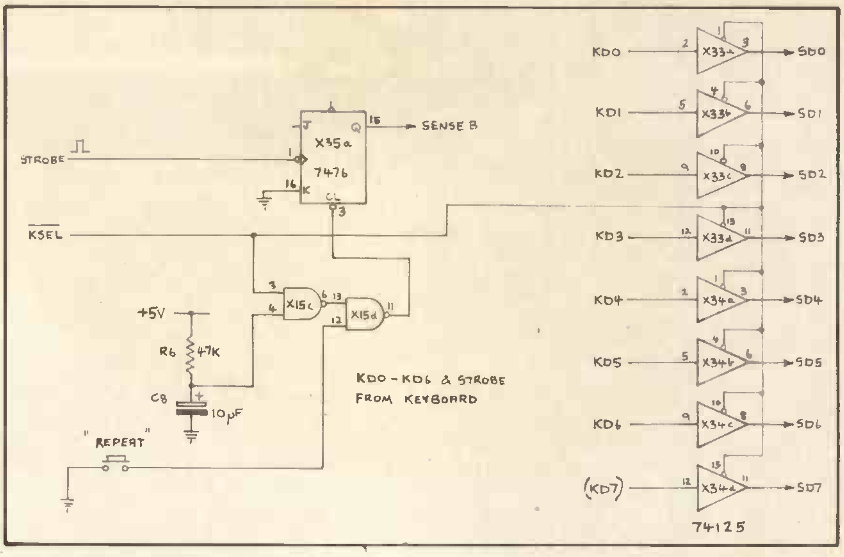


Fig. 3. Keyboard Control

Peripheral Decode

How do we produce $\overline{KSEL}$? The same method used for $\overline{VSEL}$ is possible, but why waste a whole 1k select for one unique address? This would be reasonable for just one peripheral, but what if we need a dozen or more? In order to overcome these problems, let's do some more decoding of the address bus. I propose that the top 4k page of store, addresses FXXX, be reserved for peripherals. The top output, E15, of the page decoder (Fig. 1.) enables the G inputs of the peripheral decoder (Fig. 4.) which in turn decodes the bottom four address bits A0-A3 to one of sixteen peripheral select signals. Any one of the latter can be our $\overline{KSEL}$, and in my own case the allocation is as follows:—

PS0 — Panel display & data switches	— Address FXX0
PS1 — Paper tape reader	— Address FXX1
PS2 — Keyboard (KSEL)	— Address FXX2
PS3 — UART status (cassette & teletype)	— Address FXX3
PS4 — UART data (cassette & teletype)	— Address FXX4
PS5 — Paper tape punch	— Address FXX5
N.B. X = I don't care	

Some storage space capability is still lost, but a big advantage lies in simplified programming for the SC/MP. If a pointer register in SC/MP, say P1, is set to F000 in hex, then instructions accessing the peripherals (VDU apart) can select between devices using the displacement byte of the instruction thus:—

LD 0(P1)	Loads accumulator from panel switches
XOR 2(P1)	Exclusive-ORs keyboard data with accumulator.
ST 4(P1)	Dump accumulator to UART.

Software — The VDU & Keyboard Subroutine

Included with this article is a keyboard and VDU control subroutine for a SC/MP based system, using the addressing techniques discussed. It is relocatable apart from the CRSR variable, but the latter can be placed in any convenient location. The following features should be noted:—

1. The cursor position is indicated by inverted video, and never moves off the bottom line of the 24 line display.
2. Carriage return code (OD) moves the cursor to the leftmost character position.
3. Line feed code (OA) causes the display to scroll up one line leaving the bottom line blank except for the cursor.
4. Backspace code (CTRL H = 08) moves the cursor back one character position.
5. All other control codes are ignored by VDU control section.
6. A normal character appears at the current cursor position, and then the cursor is advanced one character to the right.
7. Entry into the subroutine at KBD causes the MPU to wait for a keyboard entry, which is then echoed by the VDU section. The subroutine returns with the input character in the extension register.
8. Entry into the subroutine at VDU displays the character in the accumulator.
9. The VDU RAM in my case lies in the address range D000-D3FF, although only addresses D100-D3FF (= 24 lines) are visible on my monitor.
10. On entry at either point, P2 is assumed to be push-down stack pointer, pointing to the last stack entry. P1 contents are not preserved.
11. The main program must include setting CRSR to E0 in hex, amongst its initialisation routines.

Conclusion

The addressing techniques shown here are also applicable to other types of MPU systems, and should help the owner of an evaluation kit to expand the usually limited store provided. When used as main store decode, the 'anded' 4k, 1k select signals drive the $\overline{CS}$ inputs of each 1k block of 2102s.

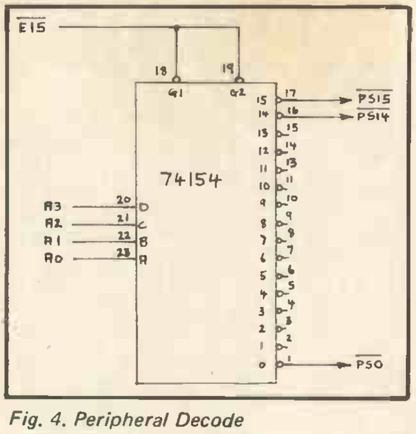


Fig. 4. Peripheral Decode

Sources of Information

1. THE TTL DATA BOOK (Texas Instruments)
2. SC/MP PROGRAMMING AND ASSEMBLER MANUAL National Semiconductor

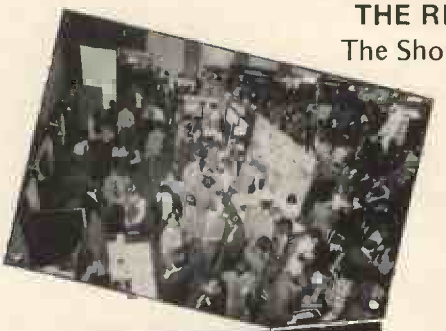
VISIONLINK

```
          TITLE, VDU  'SUPERSCAMP VDU SUBROUTINES'

          ; P2 = PUSH-DOWN STACK POINTER.

          ; KEYBOARD SECTION.
          ; MSB OF KEYBOARD DATA = 1

EØØØ  Ø8              NOP
EØØ1  C4 FØ  KBD:     LDI   H(PER)
EØØ3  35              XPAH  1
EØØ4  C4 ØØ           LDI   L(PER)
EØØ6  31              XPAL  1
EØØ7  Ø6     KEY:     CSA
EØØ8  D4 2Ø           ANI   X'2Ø
EØØA  98 FB           JZ    KEY          ; KEY PRESSED?
EØØC  C1 Ø2           LD    2(1)         ; LOAD KEYBOARD DATA

          ; VDU SECTION

EØØE  Ø1     VDU:     XAE
EØØF  C4 2Ø           LDI   L(CRSR)
EØ11  31              XPAL  1
EØ12  C4 DØ           LDI   H(CRSR)
EØ14  35              XPAH  1
EØ15  C1 ØØ           LD    Ø(1)         ; LOAD CURSOR POINTER
EØ17  33              XPAL  3
EØ18  CE FF           ST  @-1(2)         ; PUSH LP3
EØ1A  C4 D3           LDI   H(CURPOS)
EØ1C  37              XPAH  3
EØ1D  CE FF           ST  @-1(2)         ; PUSH HP3
EØ1F  4Ø              LDE
EØ2Ø  D4 7F           ANI   X'7F         ; REMOVE PARITY
EØ22  E4 Ø8           XRI   X'Ø8
EØ24  98 34           JZ    BS           ; BACKSPACE CODE?
EØ26  E4 Ø2           XRI   X'Ø2
EØ28  98 55           JZ    LF           ; LINE FEED CODE?
EØ2A  E4 Ø7           XRI   X'Ø7
EØ2C  98 1A           JZ    CR           ; CARRIAGE RETURN CODE?
EØ2E  4Ø              LDE
EØ2F  D4 6Ø           ANI    X'6Ø
EØ31  98 39           JZ    RET          ; IGNORE OTHER CONTROL CODES
EØ33  4Ø     VDU1:    LDE
EØ34  D4 3F           ANI   X'3F
EØ36  CF Ø1           ST  @+1(3)         ; DISPLAY CHARACTER
EØ38  33              XPAL  3
EØ39  9C Ø4           JNZ   NEXT         ; CURSOR AT END OF LINE?
EØ3B  33              XPAL  3
EØ3C  C7 FF           LD  @-1(3)         ; BACK UP CURSOR
EØ3E  33              XPAL  3
EØ3F  33     NEXT:    XPAL  3
EØ4Ø  C3 ØØ           LD    Ø(3)
EØ42  DC 4Ø           ORI   X'4Ø
EØ44  CB ØØ           ST    Ø(3)         ; RESET CURSOR
EØ46  9Ø 24           JMP   RET

          ; CARRIAGE RETURN ROUTINE

EØ48  C3 ØØ  CR:      LD    Ø(3)
EØ4A  D4 3F           ANI   X'3F
EØ4C  CB ØØ           ST    Ø(3)
EØ4E  33              XPAL  3
EØ4F  D4 EØ           ANI   X'EØ
EØ51  33              XPAL  3
EØ52  C3 ØØ           LD    Ø(3)
EØ54  DC 4Ø           ORI   X'4Ø
EØ56  CB ØØ           ST    Ø(3)         ; RESET CURSOR
EØ58  9Ø 12           JMP   RET

          ; BACKSPACE ROUTINE

EØ5A  C1 ØØ  BS:      LD    Ø(1)
EØ5C  E4 EØ           XRI   X'EO
EØ5E  98 ØC           JZ    RET          ; CURSOR AT FRONT OF LINE?
EØ6Ø  C3 ØØ           LD    Ø(3)
EØ62  D4 3F           ANI   X'3F
EØ64  CB ØØ           ST    Ø(3)
EØ66  C7 FF           LD  @-1(3)
EØ68  DC 4Ø           ORI   X'4Ø
EØ6A  CB ØØ           ST    Ø(3)

          ; RETURN TO PROGRAM ROUTINE

EØ6C  C6 Ø1  RET:     LD  @+1(2)         ; POP HP3
EØ6E  37              XPAH  3
EØ6F  C6 Ø1           LD  @+1(2)         ; POP LP3
EØ71  33              XPAL  3
EØ72  C9 ØØ           ST    Ø(1)         ; STORE CURSOR POINTER
EØ74  4Ø              LDE
EØ75  D4 8Ø           ANI   X'8Ø
EØ77  98 Ø3           JZ    VDUA         ; ENTRY AT KBD OR VDU?
EØ79  3F              XPPC  3
EØ7A  9Ø 85           JMP   KBD
EØ7C  3F     VDUA:    XPPC  3
EØ7D  9Ø 8F           JMP   VDU

          ; LINE FEED AND SCROLL ROUTINE

EØ7F  C3 ØØ  LF:      LD    Ø(3)
EØ81  D4 3F           ANI   X'3F
EØ83  CB ØØ           ST    Ø(3)
EØ85  C4 D1           LDI   X'D1
EØ87  37              XPAH  3
EØ88  C4 2Ø           LDI   X'2Ø
EØ8A  33              XPAL  3
EØ8B  CE FF           ST  @-1(2)         ; PUSH LP3
EØ8D  C7 Ø1  ROLL:    LD  @+1(3)
EØ8F  CB DF           ST  -33(3)
EØ91  33              XPAL  3
EØ92  9C Ø8           JNZ   LOOP
EØ94  37              XPAH  3
EØ95  E4 D4           XRI   X'D4
EØ97  98 Ø6           JZ    FIN          ; FINISHED SCROLLING?
EØ99  E4 D4           XRI   X'D4
EØ9B  37              XPAH  3
EØ9C  33     LOOP:    XPAL  3
EØ9D  9Ø EE           JMP   ROLL
EØ9F  C4 D3  FIN:     LDI   X'D3
EØA1  37              XPAH  3
EØA2  C4 EØ           LDI   X'EØ
EØA4  33     LINE:    XPAL  3
EØA5  C4 2Ø           LDI   X'2Ø
EØA7  CF Ø1           ST  @+1(3)         ; CLEAR BOTTOM LINE
EØA9  33              XPAL  3
EØAA  9C F8           JNZ   LINE
EØAC  33              XPAL  3
EØAD  C7 FF           LD  @-1(3)         ; BACK UP CURSOR
EØAF  C6 Ø1           LD  @+1(2)         ; POP LP3
EØB1  33              XPAL  3            ; RESET CURSOR
EØB2  C4 6Ø           LDI   X'6Ø
EØB4  CB ØØ           ST    Ø(3)
EØB6  9Ø B4           JMP   RET

DØ2Ø  CRSR = X'DØ2Ø

      .END
```

Scenes from last year's stunningly great event....

THE REAL SHOW FOR REAL PEOPLE!

The Show that makes other Shows look like Sunday School outings.






Now.... the 2nd Great Personal Computer World Show

will be held on

November 1-3rd 1979

at the West Centre Hotel, Lillie Road, London.

Excitement . . . hard-headed practicality . . . discovery . . . business . . . the Show that's serious but not solemn.

- **PCW** readers are truly the shapers of the future of computing in Europe, exhibitors will be taking the opportunity of presenting their latest products.
- The **Features Area** will have: The **PCW** Chess Championship . . . a robot or two . . . individual machines . . . innovations . . . Computer art . . . anything interesting that comes up between now and November . . . the Amateur Computer Club . . . Special School's Section. The Show will be vital, cheerful — and very profitable for visitors and exhibitors alike.

And for the first time in Europe!
David Hebditch, PCW Consultant, demonstrating practical communications with personal computers

- **PCW** gives and will be giving consistent support to exhibitors. A comprehensive, detailed catalogue of products of every size and description will appear in the **October** and **November** issues of **PCW.**
- A three day **Conference** (1st – 3rd November) will be held during the Show.
  Day One (Nov. 1st) The Small Computer and the Businessman.
  Day Two (Nov. 2nd) Computer Applications.
  Day Three (Nov. 3rd) Computing for the home and the amateur.

Over thirty stands have been booked so far . . . with the publicity campaign only just beginning.

Over five hundred enquiries from readers already . . . with information only just being released.

The Second Great Personal Computer World Show

☐ Please book me . . . . . . Admission tickets @ £1.00 (15% discount for groups of 10 and over).

☐ Please send me Conference details.

Name ____________________ Address ____________________

Post to:- Personal Computer World
62a Westbourne Grove
London W2

Hello Z8000

Mike Lord

Now that Zilog are releasing samples of their Z8000 chip, and Advanced Micro Devices have announced that they will second-source it, it is an appropriate time to look at this microprocessor which can outperform many mini-computers; and see how it is likely to influence future generations of personal computers.

Hardware

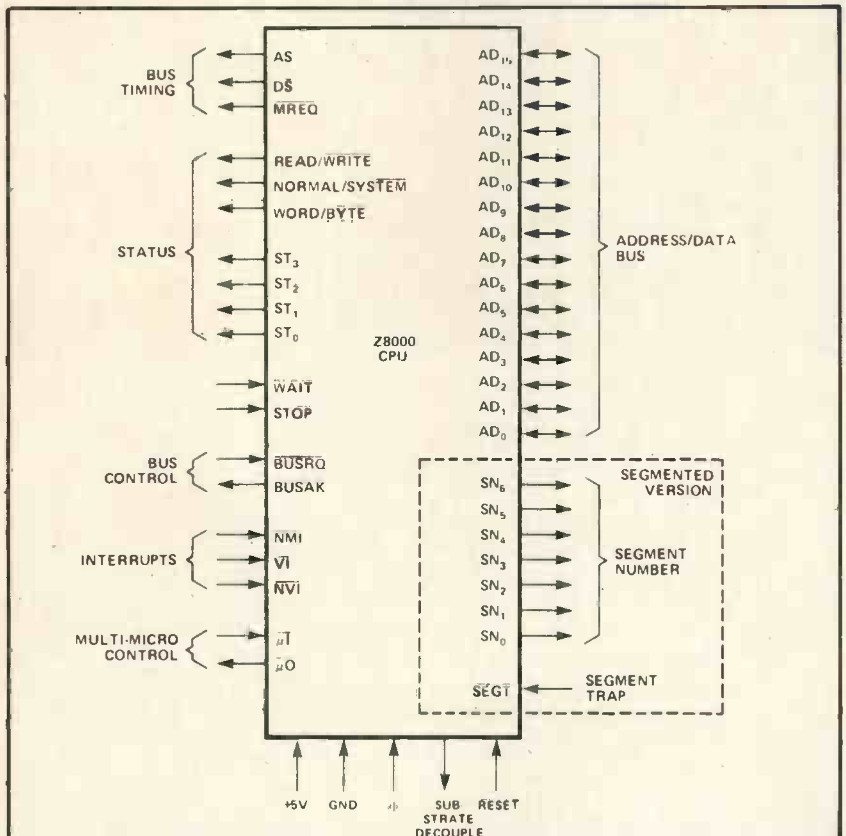
The Z8000 comes in two versions which differ only in their addressing capabilities. One, in a 40 pin package, provides 16 address lines. The more complex chip uses a 48 pin package and has a total of 23 address lines.

The data bus is 16 bits wide, and is multiplexed with the low order 16 address lines. Although this means that address latches will be needed to separate data and address signals, address buffers would be needed anyway in any reasonable size system, and a combined latch/buffer such as Intel's 8282 can be used.

As one would expect in a modern CPU chip, the Z8000 requires only a single (5V) supply and a single phase (4MHz) TTL level clock. To allow it to work efficiently in a multi processor system, CPU flag lines uO and uI control access to critical shared resources. (A multi Z8000 system would make a nice basis for serious chess programmers!) Bus Request and Bus Acknowledge lines allow other parts of the system to take control of the Bus; as for example during DMA transfers.

Of the three interrupt inputs provided, NMI (Non Maskable Interrupt) would generally be reserved for an emergency condition such as Power Fail. The other two inputs; Vectored and Non Vectored Interrupt Requests, are separately maskable; otherwise interrupt priority must be determined by external hardware.

The four Status output lines STO – ST3 can be decoded to provide information about the processor activity, such as Data, Stack, Instruction or I/O request, VI, NVI or NMI acknowledge, Bootstrap set or reset, Refresh cycle, Internal Operation or Halt. Using these signals, I/O addresses can be separated from memory addresses, and different memories used for Data, Stack and Program storage.



The CPU

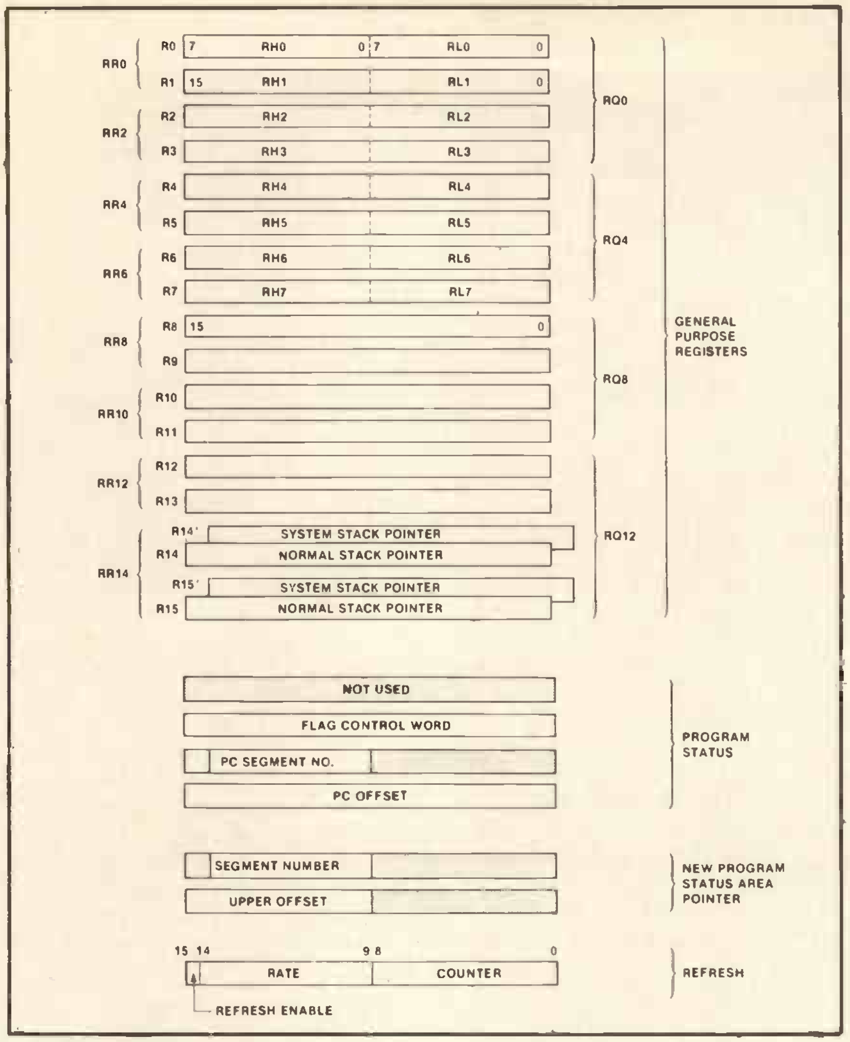
Addressing & Memory Management

Since the Z8000 is designed to handle 8-bit data bytes as easily as 16-bit words, it has to be able to read or write individual bytes to memory as well as full words. To allow this the Z8000 addresses memory in bytes, and uses the 'word/byte' signal to indicate whether 8 or 16 bit data is to be transferred. When referencing 16-bit words the least significant address bit (AO) is ignored.

The 23 address bits from the 48 pin ('segmented') version consist of a seven bit segment number plus a 16-bit 'offset', allowing 8 Mbytes of address space. In theory, by using the CPU status signals to differentiate between Stack, Data and Program references in both Normal and System modes, one could have 6 times 8 Mbytes of memory on a system, although this would be a very specialised application.

A 48 pin Memory Management chip is being produced to complement the Z8000. This will provide for segment re-location by a logical (software) to physical (hardware) address translation, and will also give a measure of memory protection. In larger systems, address re-location is an important feature as it allows:

– Programmers to write for a 'virtual' memory which is larger than the actual amount of RAM available on the system. The extra data or program is



CPU Registers (Segmented Version).

stored on disc and swapped into RAM when called for; address re-location making it appear to be in the right address area.

- A shared program (e.g. a BASIC interpreter on a time shared system) to operate on different data areas by simply re-locating the data addresses.
- Programs and data in a multi-programming/multi-tasking system to be dynamically re-located to make the best use of the available memory space.

Normal & System Modes

In terms of raw processing power and address space, the Z8000 is well suited to be used in a multi-tasking or multi-user system. This type of application depends on an Operating System to allocate resources (CPU time, peripherals, memory space) to the various users; and, to ensure that an individual user or task cannot corrupt the rest of the system, it is necessary that:

- The Operating System cannot be 'locked out'; i.e. some aspect of the Operating System must have the highest system priority.
- Memory and I/O requests from an individual task should only be allowed if legal (within the bounds allocated by the Operating System). Any attempt at an illegal access must not be allowed to succeed.

To meet these requirements the Z8000 can run in either 'System' or 'Normal' modes. Separate stack pointers are provided for each mode, and the System stack is used to save the old program status when interrupts or traps occur. Certain 'privileged' instructions are only valid when the processor is in the System mode; particularly those which affect the Memory Management chip. I/O would normally be handled via the Operating System, which can resolve conflicts and attempts at illegal accesses of peripheral devices. The Memory Management chip itself checks the validity of memory addresses against programmed bounds; causing a 'segment trap' if an illegal reference is attempted.

Registers

The Z8000 contains sixteen 16-bit registers RO-R15, all of which may be used as accumulators. R1-R15 may also be used as Index Registers, and R14, R15 double as the Stack Pointer.

RO-R7 may also be used as sixteen 8-bit registers (RHO, RLO – RH7, RL7), all of which may be used as accumulators. For long word (32-bit) data, eight register-pairs are used (RRO-RR14), while a few instructions such as Multiply and Divide use the register-quadruples RQO-RQ12.

The Program Counter is a separate register, as are the program Status Word register and the New Program Status Area Pointer (used by Traps and Interrupts).

Like its predecessor, the Z80, the Z8000 can provide automatic refresh for dynamic RAMs. The Refresh Register allows the programmer to control the refresh rate and count.

Instruction Set

The Z8000 can handle seven types of data; bits, BCD digits, 8-bit bytes, 16-bit words, 32-bit long words, byte strings and word strings.

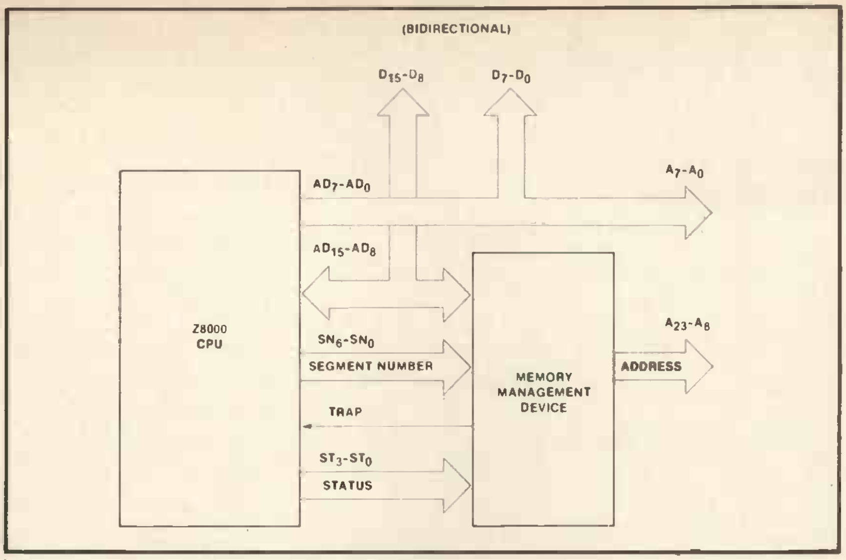
The basic instruction set provides the usual set of arithmetic and logical instructions, most of which can be specified to operate on bytes, words or long words, plus some less usual types such as Move Multiple Registers to/from memory and Load Address (the address of the 'operand' is loaded into the specified register). In most cases a register must be either the source or the destination of a data manipulation instruction, although the contents of memory locations can be directly transferred to or from the stack (PUSH & POP).

Signed Multiply and Divide Instructions can handle 16 or 32 bit operands. INC and DEC, increment or decrement a byte or word in a register or memory location by any number from 1 to 16. Similarly, bytes, words or long words can be shifted left or right by any number of bits by a single instruction.

For fast loops, a Decrement (register) and Jump on Non-zero instruction is provided and has a relative jump range of 0 to –256 bytes.

The System Call (Trap) instruction allows user programs running in Normal mode to communicate with the Operating System; as, for example, to request I/O routines. The instruction carries a programmer specified value (0-255) to specify which system routine is being called.

The most notable feature of the Z8000 instruction set must be the comprehensive range of Block Transfer and String Manipulation instructions. These operate on byte or word strings in RAM, string length being held in a register. Working in either autoincrement or autodecrement mode, a string can be searched for a particular byte or word, strings can be searched for a particular byte or word, strings can be compared, or copied into another area of memory,



Memory Management Device with Z8000 CPU.

or translated via a look-up table. Considering how much time Compilers and Interpreters spend performing this type of function, these special instructions should give a significant improvement in program run times. The I/O instructions include similar block move commands, useful for high speed data links.

Performance

While it is impossible to make a precise comparison of processors, as so much depends on the particular application and the skills of the programmer, it is possible to get a feel for their relative performances by looking at the instruction times for common instruction types. On this basis, the Z8000 is slightly faster than a PDP 11/45, and almost twice as fast as a 4MHz Z80; e.g. a 16 bit LOAD register from memory instruction takes 2.25, 2.86 and 4uS respectively. But the enhanced instruction set will also have an effect on the execution time of real programs. So, bearing in mind the power of the Z8000's Block & String instructions, and the 16 and 32 bit arithmetic capability including multiply and divide, Zilog's claim that the Z8000 will perform some 5 — 10 times faster than existing 8-bit microcomputers is believable.

And what shall we do with it?

So, we now have a micro which, in terms of speed and address range, is more than a hundred times faster than the 8008 which launched the whole personal computing scene. And the cost of memory is falling every day. And hopefully someone will produce the software to match the Z8000's power. So, the question must be; do we want it — what can we do with it?

One area will be applications where running time is critical, or at least annoyingly long on present microcomputers. Chess, for example, and computer animation. Also time-shared systems such as multi-user BASIC. Coupled with increasing memory size, and perhaps using virtual memory techniques, micro based real-time multi-tasking systems become more realistic. Surely the microcomputers developed around the Z8000 will have at least a rudimentary provision for connecting multiple terminals; if the price is low enough this makes sense even in the home or small business environment.

And surely another use for increased computer power will be to take some of the load from the user; by higher level languages that are easier to program in; by more 'intelligent' checking of input data and program structure; by providing more helpful error messages; and by implementing automatic hardware and software error detecting or diagnostic routines to give immediate warning of system failure.

And finally, bearing in mind that memory may be getting cheaper but 8 Mbytes represents an awful lot of programming effort, we will see a tendency towards 'sloppy programming'; that is, the use of techniques which may be extremely inefficient in terms of computer utilisation, but which get the job done at the minimum cost. After all, if the machine is powerful enough to cover up for long winded algorithms, who cares? If the computer responds to a query in a quarter of a second instead of a theoretically achievable tenth of a second, who is to tell the difference? And if a 16k byte ROM containing a collection of utility routines is cheaper than writing the hundred or so bytes of code that are needed for a particular job, then why not use it?

"Another Export Job — For Pisa!"

are . . . software . . . software . . . software . . . sof

Ready to use, Commercial and other software packages are now available for your Apple II

Mailing and Addressing

Build up a mailing list with this program! Addresses of customers and enquirers can be kept on file for use with all direct mailing applications. Can be sorted by up to six parameters and printed onto envelopes or labels.

Dual Disk system £50.00

Word Processor

Standard letters can be typed into your Apple and saved onto disk for future use. Features include: line insertion/deletion, justification, line centering and replacement of strings. After editing, your letter can then be printed in both lower and upper case.

Single Disk System £50.00

Estate Agents Management

Allows information of house purchaser and vendor to be held on file, referenced uniquely by a house code. Produces house adverts and keeps a record of advertising accounts, the word processor facility allows letters to solicitors, building societies etc. to be produced. A house enquiry routine is also included.

Dual Disk System from £250.00

Other programs include: Incomplete Records Accounting, Information Retrieval, Shape-Create, Sales Ledger, Order Entry Invoicing, Management Information, Roots of Polynomials, Matrix Inversion, Microade Assembler, Regression, Variance, Standard Deviation, Correlation, Datatesting, Co variance of a Time Series, plus a variety of Games. Dealer discounts available.

Name ....

Address ....

....

....

....

Please send me ....

....

....

....

....

I enclose cheque/PO No. ....

Please debit my Access/Barclaycard

Account No. ....

....

Amount ....

All prices include VAT and carriage.

Cut out and send to:-

Keen Computers Ltd., 5 The Poultry, Nottingham Tele: 583254/56

are . . . software . . . software . . . software . . . sof

Expand your Pet to its full capability with these professional peripherals

Dual floppy disk unit

A COMMERCIAL GRADE DISK SYSTEM FOR THE PET

- Up to 800K Bytes on Line Mass Storage
- High Speed D.O.S. in ROM
- Effectively doubles PET operating system commands
- Complete documentation allows use of full resources
- Handles up to four disk drives
- Includes Commercial Printer Support
- Adds 16 commands to PET basic
- Automatic self reorganisation of free space on disk after each save or erase command
- Free space is never fragmented

PRICE £840·00 + VAT

Expandem memory expansion board

General purpose expansion system for PET and other 6502 Computer Systems (e.g. KIM, SIM, AIM).

- 24K to 32K Low Dissipation RAM
- Mounting slots for 4 option cards
- All necessary cables and brackets
- Disk Controller
- Parallel I/O
- Sockets for Eprom
- Flexible Serial I/O
- S100 I/O Driver

24K Board £320·00 + VAT

Plessey Petite add-on memory From Europe's largest memory board supplier

Stand-alone, plug-in memory expansion. Available in 8K, 16K, 24K, and 32K versions. Complete in case with built-in P.S.U.

24K £449·00 + VAT

PET 2001-4 4K Bytes of Memory £460·00 + VAT

PET 2001-8 8K Bytes of Memory £550·00 + VAT

PET 2ND Cassette Unit £55·00 + VAT

PET Sound Box Plugs direct into 2nd Cassette Port complete with Demo Software £12·98 + VAT

PET Dust Covers To protect your PET—(Four Colours) Rust, Blue, Beige, Green £8·33 + VAT

PET TV Interface Plugs direct into user port Superb Specification £45·29 + VAT

Software

Estate Agency: Very cheap, very effective method of showing properties to customers. Automatic deletion. Printed copies for prospective purchasers (Now available on disk). £25·00

Mail List: Printout all your mailshots, names and addresses; printout on self-adhesive labels; disk only, £15.

Payroll: Complete suite, abundant storage on disk. End-of-year summary; coin analysis; two payslips; excellent value; disk only, £50.

We supply all Commodore, Petsoft and Compusette programs. We should be glad to quote for dedicated software applications.

R.H. Mastermind: Surely the best Mastermind ever. 3 Games levels. £5·00

R.H. Shape Matching: Match up two different shapes to score. You against the other guy. You will have to be sharp for this one. £5·00

Access and Barclaycard accepted for Telephone and Postal Orders.

All prices include post and packing (insured post extra).

We should be glad to discuss your problems or requirements relating to technical matters, software, etc—in-house software experts available.

Dealer enquiries invited for all our stock items.

U.K.-wide service by our specialist technical staff. Maintenance contracts available after our one-year guarantee at 7½% of capital purchase price.

All new equipment sold by us carries a 12 months' guarantee

HB COMPUTERS LTD

Computers for Business & Home

22 Newland Street
Kettering Northants
Telephone (0536) 83922/
520910

David Goadby *Strumech Engineering Electronics Developments*

The 6800 microprocessor is probably the most popular amateur "chip" in use (certainly in the Midlands) and I have seen many machines based upon it. A lot of the designs whilst being very commendable are often unsuited to running commercial software packages. The reason is that many such products assume either Mikbug or one of the many derivatives (lookalikes) that are available. The traditional Mikbug machine has its I/O at 8000 and various other oddball memory addresses like A000 making the memory map quite a headache. The system is quite constrained and particularly so if you want to put 60k of RAM in it! Another hazard is the pocket, if you can't afford a nice VDU or TTY but you *can* afford a Creed 7B then you have to do ASCII to Baudot conversion, as well as tackle the problems described. (See PCW April '79 for article on Creed 7B).

Universal Solution

Somehow a universal solution was required to overcome the melee of homebrew monitors and "thin wallet" I/O enabling everyone to use 8K Basics, Startrek and the like.

The idea of a machine-independent (SWTPC, MEK, MSI etc), monitor-position independent (put it anywhere you like), and I/O-independent monitor seems a bit like Utopia but it is certainly possible with the monitor I use.

I used to work for IBM and I remembered how the Jolly Giant used to call supervisory functions, namely the SVC (supervisor call). The call was followed by a byte which was the call number (to select the routine you wanted) and the SVC worked by causing an interrupt, the SWI instruction also caused an interrupt so there was the answer . . . . (Eureka?).

My particular machine was a SWTPC with 32k RAM and memory mapped VDU with parallel ASCII keyboard, and later on a MSI processor with minifloppy disks. The software, once patched, was easily transported to my MSI processor and disk operating system. In a SWTPC system I had to use a promvertor to enable replacement of the Mikbug prom by a 2708.

The actual call mechanism I used is illustrated in fig. 1. The SWI is followed by a single byte which is the call value. When the SWI occurs then the 6800 processor will load an address from location FFFA (Hex) and this is the address of the monitor (within reason, anywhere you like).

The monitor then has to get the call value ready to work out the routine required, this is done by using the program counter value on the stack. Since, when the SWI occurs the program counter +1 (for the return) is stored on the stack, then this value points to the call value.

Since we do not want to return to the next location (as this is the call value) we increment the program counter value on the stack. The call value is doubled and is used as an index pointer into the call jump table. The routine is jumped to (using JSR) and after execution we can either return directly using the RTI instruction or, if return register values are required, then these must be placed on the stack before the RTI instruction is executed.

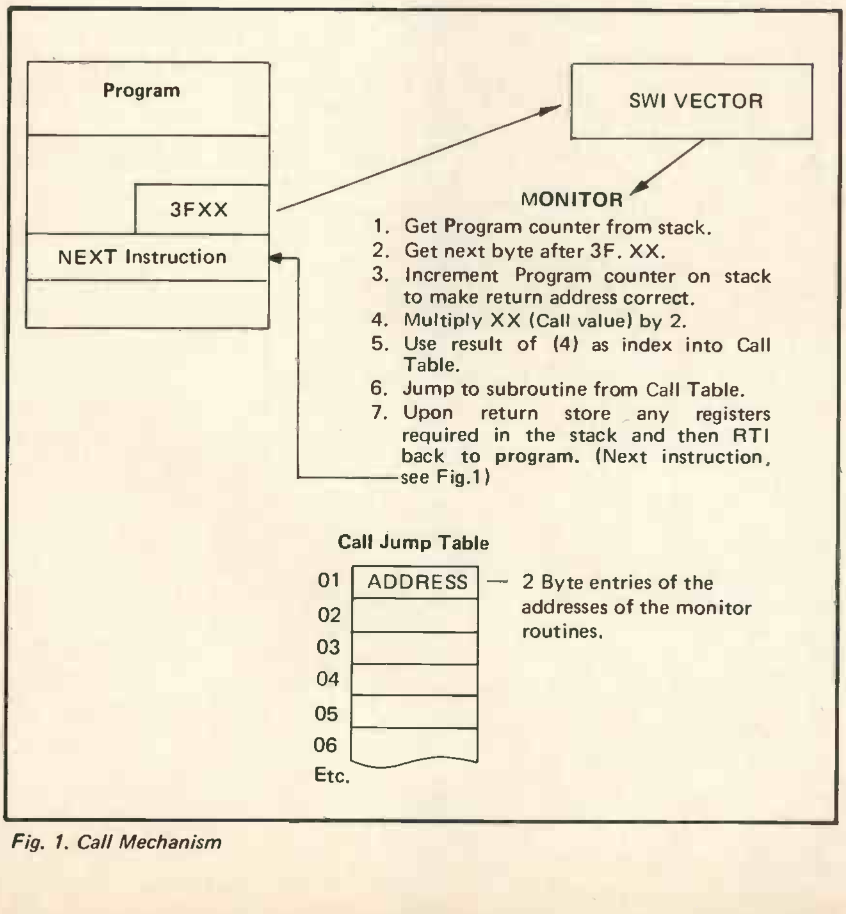


Fig. 1. Call Mechanism

The mechanism I used appears to be the best compromise bearing in mind the compatability that is required with Mikbug machines.

The Stages

To implement such an operating system requires two stages. First: identify where existing monitor calls are, using a search routine to assist you. Second: replace existing routines with SWI calls. There are two types of monitor calls that you are likely to encounter

1. JSR XXXX
where XXXX is the monitor routine address.

Replace JSR with 3FYY01 where YY is the routine call value, the NOP pads the instruction back to three bytes.

2. JMP XXXX
Here the JMP is replaced by 3FYY39. The 39 (RTS) is there because some programs, to save space, have a single jump point and monitor calls are made using BSR's (Branch Subroutine) to the jump instruction and the RTS (Return Subroutine) instruction will be the last one in the monitor subroutine. In our system all returns are made to the byte after the call value so we put a return subroutine instruction in, to complete the sequence.

It would be only fair to list the advantages and disadvantages of my system, so here goes:

PROs
1. Machine type independent (so long as it's a 6800!)
2. I/O independent (VDU, TTY, LED's etc; your mix!)
3. Software is fully compatible once standards established.
4. All monitor calls are 2 bytes in length.
5. Existing software easy to modify once you've found the places to change.

CONs
1. All software *has* to be patched to enable to be used with the monitor unless you run two monitors.
2. Slightly greater overheads, but only noticeable when using very fast terminals (19200 Baud); but who needs data transfer that fast anyway?
3. No currently available standards, so you're on your own!

Figure 2 shows the calls I used and it is by no means complete or definitive.

When writing in assembly language you can define the calls thus: INPUT EQU S3F01; then, when an input is required, code the following: FDB INPUT.

If you have written your own assembler then you can write your own for the calls eg: OUT, IN, DOS, etc.

CALL TABLE

Call Value	Function or Mikbug equivalent
01	OUTEEE (To your output device)
02	Register dump and enter monitor
03	OUT HXL
04	OUT HXR
05	Output Carriage return line feed
06	Register dump and return to program (Debug)
07	OUTS
08	BYTEOP Output A Reg. as a byte
09	Input 8 bits from input device (for binary loaders)
0A	INEEE
0B	INHEX
0C	BYTE
0D	Get address and return it in X Register
0E	OUT 2HS
0F	OUT 4HS
10	PDATA 1
11	Output 8 bits to output device (Binary dump)
12	Print error Msg. and return to monitor
13	Monitor entry via initialize routines
14	ACIA Input routine, port number in B
15	ACIA output routine, port number in B
*16	PIA input routine, port number in B
*17	PIA output routine, port number in B
18	Delay: Time depends on value in X Register
19	Task swap
1A	Get time and date
1B	ACIA poll to see if input pending (Port in B)
1C	PIA poll to see if input pending (Port in B)
1D	Message print, Msg number in B register
1E	Call multi-tasking executive
1F	DISK FILE MONITOR CALL (DOS)
20	Memory Bank Switch
21	Multiply Routines
22	Divide Routines
23	User defined Call (Set by user program).

*with handshaking

A refreshing swim

The article has, I hope, stimulated some thoughts and possibly some standards could be set up around the idea. The universal program seems an almost impossible dream, but with this system I believe that we can get very close to it. As a sweetener the 6809 (the 6800's successor, to Z80 freaks) has three software interrupts and SW2 has been deliberately kept "nondedicated" by Motorola so the system can be carried over into the next generation of machines; which means that any work done now is not wasted at all.

Finally, you might ask, where does SWIM come into it? — obvious really: SoftWare Interrupt Monitor.

ADVERTISERS' INDEX

PCW OPEN PAGE

THE AMATEUR VIEW

Mike Lord

MORE FROM THE CLUBS

Mr. M.J. Patrick of 28 West Drive, Cheam, Surrey, is interested in forming a microcomputer group serving *North/North-East Surrey* (centered around Sutton) and invites prospective members to write or 'phone him on 01-642 8362. He is also trying to set up a *Sorcerer User Group* – on a national basis – and would welcome enquiries from owners or users of this machine.

Des Wood writes to say that there is now a group going in *Scarborough*, meeting once a month at the Talbot Hotel in Queen St. Anyone wishing to go can ring 63982 for details of the next meeting.

The *Nottingham Micro Computer Club* started last year by Mr. Braga has now established regular meetings; at 7.30 on the first Monday of each month in the Trent Polytechnic, Newton Building, Burton St. Mr. P.C. McQuoney is the club secretary and enquiries should be addressed to him at 28 Seaford Ave., Wollaton, Nottingham. He would also be interested to hear of any co-ordination between the various user groups (PET, Tandy, Apple, etc.) as the exchange of BASIC software, with appropriate modifications, should be possible and could benefit everybody.

POP SHOP

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

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

SWTP 6800 MK II Computer. SWATBUG Monitor, 16K fast RAM, Serial Interface, SWTP tape interface for two tapes on full auto. Terminal with Cherry pro keyboard, works up to 19,600 Baud. Over £100 Software free, includes co-res, two versions 8K BASIC, 3K BASIC, disassembler, etc.
Cost over £1000 – **open to offers**
ring (0244) 547575

FOR SALE: 7768 4K RAM BOARD Intel 450ns Low Power RAM Chips, All I.C's socketed. Offers write – **N.K. Wright, 25 Penny Park Lane, Coventry, West Midlands CU6 2GU.**

EXTENDED NASCOM 1 COMPUTER
With T4 Monitor, 2.5A Power Supply, 3-Slot Nasbus, Buffer Board, 8K RAM + Tiny Basic, All mounted with keyboard.
£360.00 o.n.o.
Also Elektor SC/MP RAM I/O + CPU, 5v 1A Supply, ideal for experiments. **£20.**
John Oliver. London 01-886 5046

FOR SALE
KSR 33 + 10 Rolls of Paper £225
Tel: 025 125 4217

TRS80 LEVEL II 16k RAM + Expansion Interface As new + boxed. Will accept **£700**
Tel: Low Fell (0632) 872991 after 6pm

NASCOM 1 MICROCOMPUTER
For Sale: Nascom 1, fully working & virtually unused. Consists of CPU board, keyboard, 3A PSU, high resolution modulator + CPU controlled cassette deck. All housed in Custom Verocase. All Manuals & assorted programs.
£295 o.n.o.
Tel: 021-382 5815 evenings or weekends.

FOR SALE: ELF II COMPUTER with Manual & Expansion Power Supply. Fully working and housed in a temporary case
£85.00 o.n.o.
Ring 0602 266748 (Weekends Only)

Some 4096 Bit Dynamic RAM Chips @ **£1.99 each.**
Some DRAM Refresh Control Chips @ **£1.99 each**
One ASCII Touch-Type Keyboard Kit @ **£25 unused**
One 8080 CPU System with Power Unit @ **£199 working**
St. Albans 64077 after 7 pm

FOR SALE
OSI SUPERBOARD II £230
(New – Must Sell)
M. Lawrence 01-253 4321 (Daytime)

FOR SALE
SILENT 700 TERMINAL 30 cps thermal printer with manual
£380
Tel: 0582 603027 evenings.

FOR SALE
HEWLETT-PACKARD HP-25
PROGRAMMABLE CALCULATOR
49 Steps (147 Keystrokes), 8 register, 4 entry memory stack. Complete with Users Guide, Applications Manual and mains unit
£40 o.n.o.
Phone: Peter Lusmore Guildford (0483) 36311

FOR SALE
Thompson 'superchip' VDU and GRI 756 Keyboard in separate boxes with connectors and power supplies with lead to TV set. In good working order.
VDU £130 Keyboard £70 together £195
DEC memory board for CSI 11 MSVII-DC 16k by 16 bits hardly used **£700 or best offer**
S. Eisenbach 01-794 8419

NASCOM MICROCOMPUTER In Nice Cabinet with heavy duty fan cooled and protected power supply. Cuts Interface, B-Bug, Cassette Deck and all Data Books. Keyboard on Wooden Plinth.
£250
8k of 4027 Dynamic Ram £40
C.R. Bonner (G3TGF)
29 Hillcrest Road, Orpington, Kent.
Tel: Orpington 26802

STAR CLUSTERS V BYTES
For exchange. 900mm professionally made, Newtonian astro-telescopy outfit. Excellent (refigured mirror) eye pieces, equatorial mount on tripod, barlow etc. Cost **£230.** Exchange for sound Nascom or similar, Texas 159 etc. Cash adjustment if appropriate.
Sam Weller, St. Just Cornwall 788217 Day or **788286 Evenings.**

SORCERER OWNERS. Programs available including Tiny Pilot, Auto Line No. Etc. Etc. Also similar progs for PET, on Tape.
Contact: **Pete Scargill & Jim Cocallis** at
24 Percy Park, Tynemouth, Tyne & Wear

PET 4K FOR SALE. Only Three Months Old, including Pet Owners Club Manual. Price **£410.00**
Tel: Maidstone (0622) 43337

TRS-80 LEVEL 1 4k perfect working order, hardly ever used.
£350 o.n.o.
Contact: **P. Bryant** 2 Broom Road, Sittingbourne, Kent. 22753

SWTP 6800 MK II Computer SWATBUG Monitor, 16k fast RAM, Serial Interface, SWTP tape interface for two tapes on full auto. Terminal with Cherry pro keyboard, works up to 19,600 Baud. Over £100 Software free, includes co-res, two versions 8k BASIC, 3k BASIC, disassembler, etc.
Cost over £1000 – open to offers
Ring (0244) 547575

FOR SALE 7768 4K RAM BOARD
Intel 450ns low power Ram Chips, all I.C's socketed. **Offers.**
Write – **N.K. Wright, 25 Penny Park Lane, Coventry, West Midlands, CU6 2GU.**

EXTENDED NASCOM 1 COMPUTER
With T4 Monitor, 2.5A Power Supply, 3-Slot Nasbus, Buffer Board, 8K Ram + Tiny Basic, All mounted with keyboard.
£360.00 ono
Also Elektor SC/MP RAM I/O + CPU 5V 1A Supply, ideal for experiments. **£20.**
John Oliver. London 01-886 5046

NASCOM 1 FOR SALE
Plus Ithaca 8k Static RAM Board, COMP S100 expansion board and 3A PSU. Various game programs, 2K BASIC ontape. Fully working + all manuals.
£350
Tel: Wakefield 79163

MK14 SCMP MICRO For Sale. Built and Working, with Extra RAM, Improved Monitor, updated documentation.
£45. Cuts interface extra.
Tel: J. Wragg, Chesham 3744

FOR SALE – NASCOM 1
Fully built & tested, beautifully housed in wooden cabinet. TV modulator + all the manuals and taped programs.
£225 ono
T D Botterill, 48 Yardey Drive, Northampton
Tel: Northants 844338

TRITON COMPUTER, No Case, Hence **£195 o.n.o.** or exchange Hi Fi Cassette Deck, or Portable Colour TV.
Ring **Mike Gill** on Huddersfield (0484) 44176
6 – 8.30 pm.

TRITON 8K EPROM EXPANSION CARD ready built (without eproms) **£35.00**
7 Bonsor Road, Folkestone Kent.

MK14 SCMP MICRO FOR SALE. Built and working, with extra Ram, Improved Monitor, Updated Documentation.
£45. Cuts I/face extra.
Tel. J. Wragg, Chesham 3744

NASCOM 1 FOR SALE
Plus Ithaca 8K static Ram Board, COMP S100 expansion board and 3A PSU. Various games programs, 2K Basic ontape Fully Working + All Manuals **£350**
Tel: Wakefield 39163

SCIENCE OF CAMBRIDGE MK14 Microcomputer. Many extras and options: extra RAM, cassette interface, revised monitor PROM, new keyboard, PSU, full documentation **£60**
Tel: 061-224 0717 (evenings)

OHIO SCIENTIFIC CHALLENGER 1P with full 8K Basic/M-Code, u/l case, graphics, 4K RAM, cased with 4½ amp supply. Free UK supply adaptor – bought March. Sale
£400 ono + p+p in carton.
Contact: **Myb Shahid**
44 York Street, Walton, Liverpool L9 1BW

PET FOR SALE
8K Pet 2001-8 Six months old hardly used. With set of work books, microchess, ten spare tapes. **£500,** offers welcome.
Phone Headley Down (0248) 712206 (evenings)

FOR SALE
TRS-80 Level II Basic £395
5 Months Old – As New
VDU, Keyboard and Cassette
Tel. Holmes Chapel (0477) 37080

FOR SALE
TRS-80 Level II, 4K, keyboard unit only, bought late April, hardly used £275
Andy Crofts, 141 Greville Road, Warwick
Tel: 0926 43868

FOR SALE – NASCOM 1
Fully built & tested, beautifully housed in wooden cabinet. TV modulator + all the manuals and taped programs.
£225 ono
T.D. Botterill, 48 Yardey Drive, Northampton
Tel: Northants 844338.

TRITON COMPUTER, No Case, Hence **£195.ono** or exchange Hi Fi Cassette Deck, or Portable Colour TV.
Ring Mike Gill on Huddersfield (0484) 44176
6 – 8.30 pm

CASSETTES
I have a large quantity of brand new, High Quality Cassettes which are ideally suited to programme storage because they are only 10 mins per side.
Price: £1.00 each (inclusive of library case, postage etc.)
S.G. Smith, 81, Cholmeley Road, Reading Berks.

For Sale: Dual 8cm Shugart disc drives: Includes power supply and I/O cable. Both enclosed in aluminium case. Nearly new. **Offers.**
Write Flat 5, 46 Blackwater Road, Eastbourne BN20 7DH.

TRS-80 LEVEL II 16k with numeric keypad, as new – only two months old. Complete with video and cassette deck, lower case, KBFIX some games programs. Owner has genuine reason for selling. **Price £600**
Phone Romford (70) 48786

PCW Book Review

Michael James

BASIC COMPUTER GAMES

Ed. by D.H. Ahl, 1978
183 pages & 101 programs,
(Workman Publishing, 8" x 1" £4.50)

This is a collection of 101 games written in Microsoft 8080 BASIC (MITS Altair Rev 4.0). Only the minimum of advice about conversion to other BASIC's is given. An important omission is that some do not allow logical connectives in IF statements and while

IF "condition true" OR "condition true" THEN . . . is easy to convert into two IF's,

IF "condition true" AND "condition true" THEN . . . is more difficult. However, don't be too put off by this, for it's all a matter of knowing your own version of BASIC well enough. If you do, then conversion will be easy. If you don't, then certainly you will after converting a few of these games!

Each program is provided as a reasonably clear listing along with a sample run — invaluable for debugging — and usually a small amount of comment. A pleasant extra feature of this book is the drawings by George Beker scattered throughout.

As I can hardly comment on each of the 101 games, I will list some of the more noteworthy: Super Star Trek (the Klingons even move about), Poker, Mastermind, Hammurabi (a "run a country" game), Civil War (a rerun of civil war battles with you in charge), Checkers (draughts, I think!) and Blackjack. I tend to prefer games that involve an element of skill or that produce interesting graphics, and the bulk of these 101 programs rely on luck too much and become boring quickly. This is not so much a criticism of this collection of games, but more of the general level of game playing that is possible without extra input devices such as joysticks, slider controls, etc. If you're limited to a standard VDU or printer, then you run out of simple games very soon and the more interesting ones such as Chess need a book each.

A second boring aspect of any collection of programs in book form, is the need to type them in. Surely we have reached the stage when we can exchange programs in machine readable form? Obviously not!

In conclusion, this is a good collection of a wide variety of games. The ones I've tried work and have kept me amused. If you're looking for some computer games, then £4.50 is well spent on this book.

COMPUTER MODELS IN THE SOCIAL SCIENCES

R.B. Coats and A. Parkin, 1977
184 pages, (Edward Arnold, 7" x 8¾" £3.75)

At first sight this might not look like a book that would interest the computer hobbyist, but before you pass on to another review or article, let me say that you need not be put off by the words "model" and "Social Sciences". This is a practical book about programs and programming. It contains much that would be valuable to anyone interested in using computers to solve problems — for fun or profit! Slowly the reader is introduced to the idea of computer modelling (or simulation) as a method of obtaining information that would be too difficult to calculate. We begin with an amusing description of how the treasurer of the Ruritanian Sports and Social Club decides — using a simulation of the weather and hence income from the various outdoor functions to be held in the following year — how much the club can afford to spend. A very readable and painless introduction to computer simulation indeed.

After being convinced that computer modelling is a good idea, we are taken through the problems of Erewhon Airport, and answer the question of how many runways it needs. This serves to introduce the simple building blocks of models: events, queues, entities, etc.

Chapter four, "Experiments with Computer Models", deals with what should be done with a computer model after it has been constructed — how one should obtain results and test assumptions. This area of the subject always threatens to become very theoretical, but this is avoided and the chapter never loses sight of why we are building (and hence testing) a computer model.

The second half of the book is a collection of examples. The first is a program "TIM", supposed to be a first approximation to the workings of human memory. Whether it is or not is difficult to judge, but it is very interesting and should be a great deal of fun to play with. How it fits into the general picture of a computer model is not clear; it's a rare example of a non-stochastic (stochastic random) model that's not based on a set of equations.

The second example is of bed usage in a surgical suite and as such is a more standard type of simulation problem. It is well explained but not as exciting as the first model.

The third and last example is an economic model. Parameters that define the economy can be adjusted and the outcome judged by the GNP, % unemployed, etc. The purpose of this simulation is to teach students the various economic concepts such as the Philips curve, multipliers, etc. Unlike other "run a country" games, eg. Sumer, King or Hammurabi, this model is supposed to be a reasonably accurate representation of economic theory — and hence is not strictly speaking a game!

No complete computer program listings are given and the ones that are, are in FORTRAN. This need not worry anybody however, because FORTRAN is very close to BASIC and the descriptions and flow charts are enough, in themselves, to allow anyone to write their own versions of the simulations.

In conclusion, the book is well written in a chatty style. It seems to have been reproduced directly from a typed manuscript but is, even so, very clear and easy to read. If you are interested in any of the examples, then this book is for you. If you are interested in learning about simulation techniques, then don't be put off by the "Social Sciences" part of the title.

NewBear

NEW BASF 5¼ ins disk drives

BASF 6106 - slimline
Double density £220.00 8% VAT

BASF 6108 - slimline
Double density
Double sided £279.50 8% VAT

MEMORIES DOWN IN PRICE

4116 only £8.50

2102-1	£0.85	2708	£6.99
2102L-1	£0.99	SWATBUG	£16.00
2112	£2.25	MC 6803L7 (MIKBUG)	£13.65
2114	£6.99		

plus 8% VAT

WE STOCK

VERO. OK TOOLS. JIM PAK. NASCOM. SYM-1. ITHICA AUDIO. 77-68 BEAR BAGS. SORCEROR. HORIZON AND MUCH MORE...

INTEGRATED CIRCUITS

Microcomputing I.C.'s

MC 6800	£7.15
MC 6821	£4.63
MC 6850	£6.74
MC 6810AP	£3.61
MC6840	£12.72
MC8602P	£2.88
MC 14536P	£3.69
MC3459	£2.43
Z80 CPU 2.5MHz	£8.99
Z80 PIO 2.5MHz	£7.99
Z80 CTC 2.5MHz	£7.99
Z80A CPU 4MHz	£13.99
Z80A PIO 4MHz	£10.00
Z80A CTC 4MHz	£10.00
SC/MP11 (INS 8060N)	£8.88
INS 8154N	£8.18
8080A	£6.00
6502	£12.00
6522	£6.58
6532	£12.56

NEW

100.0	KHz	£3.25
200.0	KHz	£4.20
204.8	KHz	£3.25
1.0	MHz	£3.50
1.008	MHz	£2.80
1.8432	MHz	£2.80
2.0	MHz	£2.60
4.0	MHz	£2.60
5.0	MHz	£2.60
10.0	MHz	£2.60
13.478	MHz	£3.25
36.0	MHz	£3.25

plus 8% VAT

INTEGRATED CIRCUITS

Character Generators

74S 262	£13.70
2513 (U)	£6.50
2513 (L)	£6.50
MCM 6571P	£6.71
9368 (7 segment display decoder)	£2.45

UARTS

6402	£4.50
AY-5-1013	£4.50

plus 8% VAT

Send for New Hardware Catalogue & booklist

DEVLIN MODULAR 60 KEY KEYBOARD

Pet replacement
Mounted on PCB
X-Y matrix £38.50
Matching Hex keypad £7.20
Send for details

GAMES BOOKS

101 Basic Computer Games	D.AHL	£6.50
Star Ship Simulation		£5.10
Game Playing with Computers		£10.20
Game Playing with Basic	D. Spencer	£4.20
SARGON-A Chess Computer Program in Z80 ASSEMBLY LANGUAGE		£9.50

PASCAL BOOKS

Pascal: User Manual & Report	Springer-Verlag	£5.52
Problem Solving Using Pascal	Springer-Verlag	£7.84
Programming in Pascal	P. Grogono	£7.50
A Practical Introduction to Pascal	A. Addyman	£3.50

PRACTICAL APPLICATION BOOKS

Programming the 6502 (SYBEX)		£8.00
The 6800 Microprocessor		£3.60
8080A BUGBOOK		£6.95
Data Aquisition Handbook		£2.00
77-68 Construction Manual		£7.50
Fundamentals of Digital Computers	D. Spencer	£7.50
National Semiconductors Interface Circuits Data Book		£2.75
Programming a Micro 6502	C.C.Foster	£7.95
Semiconductor Circuit Design Vol 1	T.I	£6.50
Semiconductor Circuit Design Vol 2	T.I	£6.50
Semiconductor Circuit Design Vol 3	T.I	£6.50
Semiconductor Circuit Design Vol 4	T.I	£6.50
Semiconductor Circuit Design Vol 5	T.I	£7.95

TERMS AND CONDITIONS

Minimum official order £10.00
Postage & packing 30p on Hardware
Hardware price plus 8% VAT.
Callers welcome Mon-Sat 9.00-5.30
but please phone us first on 0635 30505.

Barclaycard & Access Welcome

ALL MAIL ORDER TO NEWBURY

INTRODUCTORY OFFER

MODEL 700 VDU

- Assembled and tested
- 64 characters x 16 lines
- Upper case only
- 110-1200 baud

£299 with video monitor

Newbear Computing Ltd.,
40 Bartholomew St, Newbury, Berks.
0635 30505
Visit our Northern office at 2A Gatley Rd, Cheadle, Cheshire.
Tel 061-491-2290

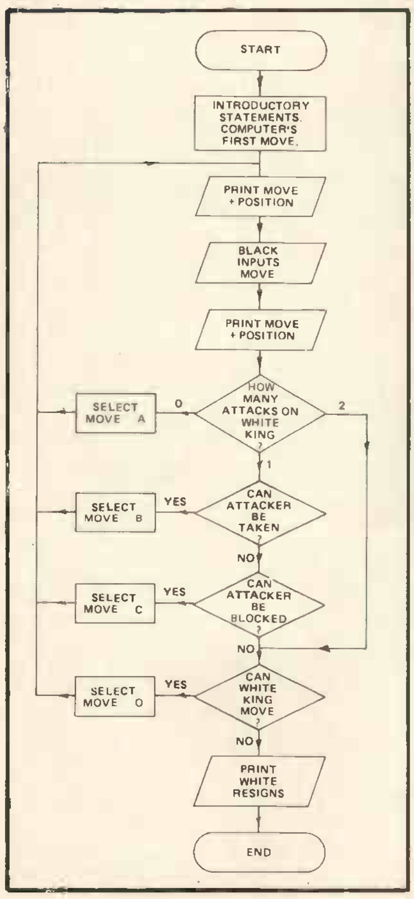
24 HOUR DELIVERY IN MOST CASES

Beginner's Gambit

NOTES ON "32"~ A Version of Chess

R. A. Diamond, *Tottenham College of Technology, London.*

Many amateur programmers will have attempted to write a program that plays chess but will have been defeated by the complexity of the rules. Those who pass this stage may well be daunted by the task of programming a strategy that plays useful chess. These notes may encourage people to deal with the former problem whilst avoiding the latter. The suggestion is to write a program that plays a version of *chess* called "32" in which any piece taken is replaced on the board *by its captor.*



The Game Strategy

The strategy for playing a good game of 32 involves restricting the scope of opposition pieces, and this usually involves tucking away taken major pieces on the back rank or taken pawns on the next rank. Generally it is a relatively simple matter to program this. Equally well, a human opponent unless experienced in this version of chess, will find it somewhat difficult to adjust to the fact that material loss cannot arise and a more even game ensues.

Suppose we decide to play 32 according to the simplified rules

1. Any piece taken must be replaced on the board.
2. Check cannot be blocked by using an opponent's piece.
3. No castling or en passant are allowed.
4. Pawns remain pawns and cannot be put on the back rank.
5. Otherwise according to normal chess rules.

Suppose further we label the board as shown and allocate values to pieces as shown.

WHITE : COMPUTER

11	12	13	14	15	16	17	18
21	22	23	24	25	26	27	28
31	32	33	34	35	36	37	38
41	42	43	44	45	46	47	48
51	52	53	54	55	56	57	58
61	62	63	64	65	66	67	68
71	72	73	74	75	76	77	78
81	82	83	84	85	86	87	88

BLACK : PLAYER

	PAWN	KNIGHT	BISHOP	ROOK	QUEEN	KING
WHITE	1	2	3	4	5	6
BLACK	−1	−2	−3	−4	−5	−6

Piece Values

Let the current position be contained in array P and the current position of the white king be K. Thus initially P(83) = −3, P(17) = 2, K = 14 etc. Sections of the program involving printing instructions, the current position and move are straightforward as are the initial move (or opening lines if required) and the arrangements for black to input his response. A 6 figure move, e.g. 614317, could be used for: the piece on square 61 takes the piece on square 43 and puts it on square 17; whereas 614300 would mean: the piece on square 61 moves to square 43.

The flowchart

Bearing in mind that if white is in check then he has to do something about it, an outline flowchart (disregarding offers/acceptances of draws and black's resignation — which are easily catered for) might be designed as shown.

To implement this flowchart perhaps the major subroutine specification will be : given a chess position defined by array Z and a target square S, find the number of attacks on square S and record the positions and types of attack. If we use subscripts of array J to record the positions of attacking pieces and corresponding contents to record types of attacking pieces then part of the subroutine written in BASIC might appear as:

```
100  REM; 110 – 140 DEALS WITH
       PAWN ATTACKS ON SQUARE S
110  IF Z(S + 9) = –1 THEN J(S + 9)
       = –1
120  IF Z(S + 11) = –1 THEN J(S +11)
       = –1
130  IF Z(S – 9) = 1 THEN J(S – 9) = 1
140  IF Z(S – 11) = 1 THEN J(S – 11)
       = 1
 . . . . . . . . . . . . . . . . . . . . .
300  REM; 310 – 350 DEALS WITH
       QUEEN/ROOK ATTACKS
       FROM THE LEFT OFS
310  FOR P = S –1 TO 10* INT(S/10)
       + 1 STEP – 1
320  IF ABS (Z (P)) = 4 THEN J(P) =
       Z(P) \ GOTO 360
330  IF ABS (Z (P)) = 5 THEN J(P) =
       Z(P) \ GOTO 360
340  IF ABS (Z (P)) > O THEN 360
350  NEXT P
360  . . . . . . . . . . . . . . . . . .
```

Note that 340 deals with the line of action being blocked. In 300 — 350 we are searching for attacks on square S by queens and rooks from the left of S on the diagrammed board. Starting from square S we go across the board to the left. If we meet queen or rook we record it and jump out of this section. If we meet any other piece we jump directly out of this section. Towards the end of the subroutine we can split array J to count and find black and white attacks on square S separately.

This subroutine can be used for example to find out if white is in check by

```
50    MAT Z = P \S = K GOSUB 100
```

Punishing the offender

Suppose now that white is in single check. Can the offending black piece be taken? Again we use the same subroutine to see if the square from which check is given is attacked by a white piece or pieces. But we must be careful to make sure that white can indeed take the attacking piece without revealing a discovered check on himself. So once again we use the same subroutine for the position that would arise if white takes the black checking piece to see whether the white king is still under attack.

Suppose we have now gathered all the moves that can take the checking piece; it remains to decide which one to choose. This could be done on a random basis or in order of priority pawn, knight, bishop, queen and king. The taken piece could be placed on the highest numbered free square (provided it doesn't check white) or if a pawn is being taken, replace it on the highest numbered free square less than 79 (again provided it doesn't check white).

Similarly, to block a check, a subroutine can be used to store in an array all squares intermediate between U and V. Whether a square X can be reached by a white pawn depends on whether P(X − 10) = 1 or in some cases whether P(X − 20) = 1 and P(X − 10) = 0; and in all cases that the new position is free from white being in check. Whether a square X can be reached by a white piece other than a pawn can be established using the major subroutine referred to earlier. The decision of which block to choose could be random or pawns first etc.

The fugitive

Suppose on the other hand we cannot take the checking piece or interpose a piece (as occurs in all examples of double check) then we will be looking for escape squares for the white king. Free squares adjacent to the white king are easily found and the main subroutine can check that they are not also under attack.

Alternatively when under attack all possible moves corresponding to selection boxes B, C, D on the flowchart can be gathered and some other criterion for choice can be used.

A Moving Opportunity

We are now left with the case of the white king not being under attack. That is the selection box A on the flowchart. How do we decide on our move? In normal chess a program that's going to play even at club standard has to come up with a very good answer to that question but in 32 *this is not so.* One can frustrate the opposition by quite simple strategies occupying up to 20% of the total program. Even a fairly transparent policy such as take and replace a piece with one of the knights; advance a piece on a random (or weighted random) basis etc., can be difficult to combat and heavy use of the major subroutine facilitates programming.

It is easy to incorporate an illegal move finder after black inputs his move. The framework described is amenable to further development to include a more positive strategy for white rather than the spoiling tactics referred to.

Even in a simple form the program leads to bizarre and brightly entertaining positions and it is a routine matter to print out the final position and a record of the game. If anyone knows an infallible strategy for winning 32 I shall be delighted to hear from him or her . . . meantime good luck.

Pet Preening

PLAY SNAP WITH YOUR PET

Rupert Steele, *St. John's College, Oxford.*

This is essentially a reaction tester: the SNAP format is simply to make the game more interesting, particularly for younger users. There are ten levels of difficulty from 1 (trivial) to 10 (impossible); I draw with the machine on 5.

The program monitors the keyboard with a GET instruction and times the response. If the response is faster than the loop in lines 320–330 a point is awarded to the human, and if slower the computer takes a point. If the user presses a key when there is no 'snap' he is penalised two points.

Notes:

At present, there are six possible digits shown, but line 300 may be altered to give more or less.

Line 290: the string is " < clear screens > 14 x cursor down > < 15 x Cursor Right > ". The printer does not print these characters.

Line 400: the string is "< 6 x Cursor left > ".

The program will run with only minor changes (lines 290, 400 and 260–270) on any micro running 8k Microsoft BASIC or similar, if GET is supported. TI$ is the time string on the PET.

Sometimes, the computer will appear not to print the next digit; this is when the new one is identical with the old; this is a small trick as the player is put off and unsure which digit to watch.

```
100 REM  ** REAL TIME SNAP **
110 REM  RUPERT STEELE 28-NOV-78
115 DIM C(2)
118 PRINT"DO YOU KNOW THE RULES";
120 INPUT N$:N$=LEFT$(N$,1)
130 IF N$="Y" THEN 200
140 PRINT"THE OBJECT OF THE GAME IS TO BEAT THE"
150 PRINT"COMPUTER AT SNAP. THE COMPUTER PRINTS"
160 PRINT"TWO DIGITS NEXT TO EACH OTHER ON THE"
170 PRINT"SCREEN; IF THEY MATCH THERE IS A SNAP"
180 PRINT"IF YOU PRESS ANY KEY FAST ENOUGH, YOU"
190 PRINT"WIN THE SNAP, OTHERWISE I WIN"
192 PRINT"TYPING ! WHEN THERE IS NO SNAP TERM-"
194 PRINT"INATES THE GAME":PRINT
195 REM ** RANDOMISE **
200 PRINT"PLEASE GIVE A NUMBER BETWEEN 20 AND 100"
210 INPUT A:IF A>100 OR A<20 THEN 200
220 FOR N=1 TO A: B=RND(1): NEXT N
230 PRINT"LEVEL OF DIFFICULTY (1 EASY 10 HARD)"
240 INPUT D:IF D<1 OR D>10 THEN 230
243 REM ** INITIALISATION **
245 Y=0:M=0
250 PRINT"YOUR SCORE:"Y"  MY SCORE:"M
251 C(2)=0:P=1
252 FOR A=1 TO 2
260 Z$=TI$
270 IF Z$=TI$ THEN 270
280 NEXT A
285 GET L$
286 IF L$<>"" THEN 285
290 PRINT"";
300 C(P)=INT(6*RND(1)+1)
310 PRINTC(P);
320 FOR L=1 TO 130+450/D
330 NEXT L
340 IF C(1)=C(2) THEN 420
345 N=150+INT(RND(1)*400)-300/D
350 FOR L=1 TO N:NEXT L
360 GET L$:IF L$="" THEN 390
365 IF L$="!"THEN 470
370 PRINT:PRINT"CHEAT! 2 POINTS TO ME"
380 M=M+2:GOTO 250
390 P=P+1:IF P=2 THEN 300
400 P=1:PRINT"";
410 GOTO 300
420 GET L$:IF L$="" THEN 450
430 PRINT:PRINT"SNAP! WELL DONE, 1 POINT TO YOU"
440 Y=Y+1:GOTO 250
450 PRINT:PRINT"SNAP! BAD LUCK, I WAS FIRST, 1 TO ME"
460 M=M+1:GOTO 250
470 PRINT:PRINT"YOUR SCORE:"Y"  MY SCORE:"M
475 IF M=0 AND N=0 THEN 490
480 PRINT"YOUR PERCENTAGE SCORE:"100*Y/(M+Y)
490 PRINT:PRINT"REPLAY";:INPUT N$:N$=LEFT$(N$,1)
500 IF N$="Y" THEN 230
510 PRINT"THANKS FOR THE GAME"
```

TANDY TRS-80
The Soft Cursor

M.O. Benjamin

The January issue of PCW contained a very interesting article by P.J. Turner, on the Tandy TRS-80 Level 2.

At the end of his article, Mr. Turner gave the listing for a Cursor Graphics program which I tried out and found to add considerably to the graphics capability of the TRS-80. However, having spent many happy hours drawing fancy pictures, I realised the need for a method of storing the pictures on tape, so I set to, to extend Mr. Turner's program, and in the process added several other facilities as well.

The added facilities are as follows: Save/retrieve picture on tape, change from continuous line drawing/cursor movement to single block drawing & vice versa, mix alphanumeric characters on screen with the graphics.

One or two notes on the program itself: to save a picture on tape, simply press the 'M' key, but ensure the tape recorder is set to record first. The picture is stored on tape in blocks of ten screen memory locations, blank areas of the screen being ignored, so the tape will start and stop as the computer reads past blank screen areas. When all the screen has been read, the screen will clear and "DATA DUMP COMPLETE" will appear. This is simply to let you know when it has finished recording. To retrieve the picture from tape, just follow the screen instructions when the program is run.

With regard to the TRS-80 Level two, I purchased my system last July, opting for the Level 2 from the outset. I had no previous programming experience (apart from a Casio calculator), but from all the articles I had read, I realised that the Level 2 was far better then the Level 1. Tandy's instruction manual is so good that I was able to learn very easily, and I cannot recommend the Tandy highly enough. The instruction manual is really excellent, and very clear, although the Level 2 manual assumes one already has some knowledge of Basic, so I also purchased the Level 1 Manual, which gives a very simple and easily understood introduction to the Basic language.

There has been some criticism concerning difficulty in setting the cassette volume levels with the TRS-80, and I think this should be clarified. The difficulty only arises on the initial setting up of the recorder, and once the correct level has been found, no further trouble occurs. I control three recorders with my system (2 cassette, and 1 Reel recorder), and have had absolutely no problems whatsoever. I feel that this has been exaggerated in many reviews, as, once set up, the tape system is extremely reliable.

The only other major criticism, (with which I agree) is the modular design. I was not too happy with the separate items all connected with a mass of cables. I overcame this problem by constructing a new cabinet to hold the entire system as one unit (photo above), with sufficient room for further expansion.

To sum up, I think the Tandy is excellent value for money, extremely well thought out, and with a very powerful Basic, and apart from criticism concerning the modular construction, is to my mind the best buy within the £500—£1000 price range.

EXTENDED CURSOR GRAPHICS

```
1    CLS: CLEAR100
2    ?: "THIS PROGRAM ALLOWS YOU TO DRAW ANY
     PICTURE ON THE SCREEN AND SAVE THE PICTURE
     ON TAPE IF DESIRED": ? STRING $ (64, "#")
3    ? "THE CONTROL KEYS ARE . . . : V = SINGLE BLOCK
     DRAWING ↓ "TAB (28) " B = CONTINUOUS LINE
     DRAWING ↓ "TAB (28) "M = SAVE PICTURE ON
     TAPE ↓"TAB (28) "C = CLEAR SCREEN"
4    ? "THE DRAWING KEYS ARE . . . : ↓↓U = UP : D =
     DOWN : R = RIGHT : L = LEFT ↓↓CURSOR MOVE-
     MENT (NON DRAWING) ↓ W= UP : Z = DOWN : F =
     RIGHT : A = LEFT : P = PRINT (CHARACTERS): @ =
     RETURN TO GRAPHICS":?
6    INPUT"ENTER TO START":W:CLS:?:INPUT"DRAW
     PICTURE. . 1 : RETRIEVE FROM TAPE . . 2";N:CLS:
     ON N GOTO 8,3000
8    ?:?:? "BEFORE STARTING YOUR DRAWING, ↓- - -
     IT SHOULD BE NOTED THAT WHEN SAVING THE
     DRAWING ON TAPE, THE TAPE WILL STOP FOR
     SHORT PERIODS WHILST THE COMPUTER COUNTS
     PAST BLANK PORTIONS OF THE SCREEN, THIS IS
     TO AVOID TAPE";
9    ?" WASTAGE ANDTIME. WHEN THE ENTIRE SCREEN
     HAS BEEN CHECKED, THE COMPUTER WILLPRINT
     . . . ** DATA DUMP COMPLETE **": STRING$ (64,
     "-")
10   X=1:Y-1:INPUT"SELECT TYPE OF CURSOR MOVE-
     MENT REQUIRED AT START ↓ "TAB(7) "SINGLE
     BLOCK. . 1 : LINE. . 2"; V:CLS
20   B$ = INKEY $ ; IFB$= "M" THEN2000
21   IFV = 1THEN23ELSE25
23   IFB$ < > " "THENA$ =B$
24   GOTO30
25   IFB$= < > " "THENA$ =B$
30   IFA$ = "L"THENX=X-1:GOSUB1000
31   IFA$ ="V"THENV=1:GOTO20
32   IFA$ ="B"THENV=2:GOTO20
35   IFA$ = "A" THENRESET (X,Y):X=X-1
36   IFA$ = "P" THEN3090
37   IFX <0THEN X = 0
40   IFA$= "R" THENX = X+1:GOSUB1000
45   IFA$= "F" THENRESET (X,Y):X=X+1
47   IFX > 127 THEN X=127
50   IFA$ ="D"THENY=Y+1:GOSUB1000
55   IFA$ ="Z"THENRESET (X,Y):Y=Y+1
57   IFY >47 THEN Y=47
58   IFA$ ="C"THENCLS
60   IFA$ ="U"THENY=Y-1:GOSUB1000
65   IFA$ ="W"THENRESET (X,Y):Y=Y-1
67   IFY <0 THEN Y=0
68   P=POINT (X,Y)
69   SET (X,Y):FORF=1TO30:NEXT:RESET (X,Y):IFP=1
     THENSET (X,Y)
70   GOTO20
1000 IFX <0THENX=0:RETURN
1001 IFY <0THENY=0:RETURN
1002 IFX >127THENX=127:RETURN
1003 IFY >47THENY=47:RETURN
1010 SET (X,Y): RETURN
```

▶

Note: the following section saves the picture on tape

```
2000 POKE16382, 43:FORI=15360TO16383:REM prints '+'
     at lower r.h corner to indicate dump in progress
2006 A=PEEK(I)
2007 IF A > 32 AND A < 97 OR A > 128 AND A < 192 THEN
     L=L+1:C(L)=I:D(L)=A:REM counts past blank screen
     locations and records only picture/character blocks
2008 IFL < 10 AND I < 16383 THEN2020
2010 PRINT # -1, C(1), D(1), C(2), D(2), C(3), D(3), C(4),
     D(4), C(5), D(5), C(6), D(6), C(7), D(7), C(8), D(8),
     C(9), D(9), C(10), D(10): L=0:REM dumps in blocks
     of 10 to save tape and time
2020 NEXTI
2040 CLS:PRINT @ 462, "*** DATA DUMP COMPLETE
     ***":END
```

Note: following section retrieves picture from tape.

```
3000 CLS:?:?:?:?"PRESS ENTER WHEN READY ** NOTE
     THAT WHEN PICTURE IS COMPLETE, THERE WILL
     BE A SHORT DELAY BEFORE THE TAPE SWITCHES
     OFF, DUE TO THE COMPUTER CHECKING TO THE
     END OF THE SCREEN. A FLASHING CURSOR WILL
     BE DISPLAYED AT THE LOWER";
3001 ?"RIGHT HAND CORNER WHEN RETRIEVAL IS
     COMPLETE. ADDITIONS OR ALTERATIONS TO THE
     PICTURE MAY THEN BE MADE":?:INPUTW:CLS
3005 FORI=1TO1023STEP10
3010 INPUT# -1, C(1), D(1), C(2), D(2), C(3), D(3), C(4),
     D(4), C(5), D(5), C(6), D(6), C(7), D(7), C(8), D(8),
     C(9), D(9), C(10), D(10):FORJ=1TO10:POKEC(J), D(J):
     NEXTJ:X=PEEK (16382):IFX < > 43THENNEXTI
3030 X=124:Y=46:GOTO20:REM sets cursor to lower r.h.
     corner and returns to drawing mode.
```

Note: the following section allows characters to be mixed with graphics, commencing from the current graphics cursor position + 1

```
3090 U=INT (Y/3) *64+INT (X/2) +1
3095 PRINT @ U, ".";
4000 B$=INKEY$ :IFB$="  "THEN N4020E LSEPRINT B$;
4010 GOTO4000
4020 A$=" ":GOTO20
```

Summary of control keys

KEY	*FUNCTION*
C	Clear Screen
V	Convert to single block drawing
B	Convert to continuous (rpt) line drawing
P	Convert to character printing (commences from current graphics cursor position +1
@	Revert to graphics
M	Save picture on tape. (ensure recorder is ready & in record mode before pressing 'M' key)
DRAWING KEYS	
R	Draw RIGHT
L	Draw LEFT
U	Draw UPwards
D	Draw DOWNwards
CURSOR CONTROL KEYS	
W	Move cursor UP
Z	Move cursor DOWN
A	Move cursor LEFT
F	Move cursor RIGHT

N.B. Moving the cursor over a lit block, will delete that block.

In the Character printing mode, the key works as normal. To print characters on more than one line, press key at end of the first line, cursor will then be displayed, then move cursor to start of new line and press P key, the next line can then be typed.

When using the continuous line drawing mode, pressing the space bar (or any non-control key) will stop the cursor at its present position until the next control key is pressed.

In the Character printing mode, the SHIFT ↓ will move the print position down one line without returning to the start of line, and the SHIFT ↑ will move the print position UP one line.

Appledem

G.F. Filbey, *Department of Physics, Polytechnic of the South Bank, London*

continued from June 79 **PCW**

```
1000-  20 00 0C   JSR  $0C00
1003-  20 46 10   JSR  $1046
1006-  A9 FF      LDA  #$FF
1008-  8D 2C 03   STA  $032C
100B-  EA         NOP
100C-  A9 00      LDA  #$00
100E-  A8         TAY
100F-  85 50      STA  $50
1011-  85 51      STA  $51
1013-  85 69      STA  $69
1015-  EA         NOP
1016-  8D 12 03   STA  $0312
1019-  20 ED 11   JSR  $11ED
101C-  C9 25      CMP  #$25
101E-  D0 03      BNE  $1023
1020-  20 DB 10   JSR  $10DB
1023-  18         CLC
1024-  A9 01      LDA  #$01
1026-  65 51      ADC  $51
1028-  85 51      STA  $51
102A-  A9 00      LDA  #$00
102C-  65 50      ADC  $50
102E-  85 50      STA  $50
1030-  C9 03      CMP  #$03
1032-  D0 E5      BNE  $1019
1034-  A6 69      LDX  $69
1036-  EA         NOP
1037-  DE 00 0B   DEC  $0B00,X
103A-  BD 00 0B   LDA  $0B00,X
103D-  AC 12 03   LDY  $0312
1040-  20 7C 0C   JSR  $0C7C
1043-  4C 51 10   JMP  $1051
1046-  A9 C0      LDA  #$C0
1048-  A2 00      LDX  #$00
104A-  9D 00 0B   STA  $0B00,X
104D-  E8         INX
104E-  D0 FA      BNE  $104A
1050-  60         RTS
1051-  A6 69      LDX  $69
1053-  BD 00 0B   LDA  $0B00,X
1056-  D0 B4      BNE  $100C
1058-  60         RTS
1059-  00         BRK
105A-  20 00 0C   JSR  $0C00
105D-  20 46 10   JSR  $1046
1060-  A9 FF      LDA  #$FF
1062-  8D 2C 03   STA  $032C
1065-  20 ED 11   JSR  $11ED
1068-  EA         NOP
1069-  EA         NOP
106A-  85 69      STA  $69
106C-  A6 69      LDX  $69
106E-  DE 00 0B   DEC  $0B00,X
1071-  EA         NOP
1072-  BD 00 0B   LDA  $0B00,X
1075-  A0 00      LDY  #$00
1077-  20 7C 0C   JSR  $0C7C
107A-  A6 69      LDX  $69
107C-  BD 00 0B   LDA  $0B00,X
107F-  D0 DF      BNE  $1060
1081-  60         RTS
1082-  20 00 0C   JSR  $0C00
1085-  20 46 10   JSR  $1046
1088-  A9 FF      LDA  #$FF
108A-  EA         NOP
108B-  EA         NOP
108C-  85 80      STA  $80
108E-  20 ED 11   JSR  $11ED
1091-  E6 80      INC  $80
1093-  A5 67      LDA  $67
1095-  29 03      AND  #$03
1097-  EA         NOP
1098-  EA         NOP
1099-  D0 F3      BNE  $108E
109B-  A5 80      LDA  $80
109D-  0A         ASL
109E-  0A         ASL
109F-  0A         ASL
10A0-  85 69      STA  $69
10A2-  AA         TAX
10A3-  DE 00 0B   DEC  $0B00,X
10A6-  D0 E0      BNE  $1088
10A8-  E8         INX
10A9-  DE 00 0B   DEC  $0B00,X
10AC-  BD 00 0B   LDA  $0B00,X
10AF-  85 6B      STA  $6B
10B1-  F0 0C      BEQ  $10BF
10B3-  EA         NOP
10B4-  EA         NOP
10B5-  EA         NOP
10B6-  EA         NOP
10B7-  A9 FF      LDA  #$FF
10B9-  20 BB 11   JSR  $11BB
10BC-  4C 88 10   JMP  $1088
10BF-  60         RTS
10DB-  18         CLC
10DC-  A9 09      LDA  #$09
10DE-  65 69      ADC  $69
10E0-  85 69      STA  $69
10E2-  60         RTS
10E3-  00         BRK
112B-  20 00 0C   JSR  $0C00
112E-  A9 00      LDA  #$00
1130-  85 69      STA  $69
1132-  85 6A      STA  $6A
1134-  A9 9E      LDA  #$9E
1136-  85 6B      STA  $6B
1138-  A9 FF      LDA  #$FF
113A-  20 BB 11   JSR  $11BB
113D-  C6 6B      DEC  $6B
113F-  C6 6B      DEC  $6B
1141-  A5 6B      LDA  $6B
1143-  C9 8E      CMP  #$8E
1145-  D0 F1      BNE  $1138
1147-  E6 69      INC  $69
1149-  E6 69      INC  $69
114B-  D0 E7      BNE  $1134
114D-  20 ED 11   JSR  $11ED
1150-  29 FE      AND  #$FE
1152-  85 69      STA  $69
1154-  20 ED 11   JSR  $11ED
1157-  30 1B      BMI  $1174
1159-  29 0E      AND  #$0E
115B-  18         CLC
115C-  69 90      ADC  #$90
115E-  85 6B      STA  $6B
1160-  A9 00      LDA  #$00
1162-  20 BB 11   JSR  $11BB
1165-  38         SEC
1166-  A5 6B      LDA  $6B
1168-  E9 40      SBC  #$40
116A-  85 6B      STA  $6B
116C-  A9 FF      LDA  #$FF
116E-  20 BB 11   JSR  $11BB
1171-  4C 8C 11   JMP  $118C
1174-  29 0E      AND  #$0E
1176-  18         CLC
1177-  69 90      ADC  #$90
1179-  85 6B      STA  $6B
117B-  A9 FF      LDA  #$FF
117D-  20 BB 11   JSR  $11BB
1180-  38         SEC
1181-  A5 6B      LDA  $6B
1183-  E9 40      SBC  #$40
1185-  85 6B      STA  $6B
1187-  A9 00      LDA  #$00
1189-  20 BB 11   JSR  $11BB
118C-  20 ED 11   JSR  $11ED
118F-  C5 50      CMP  $50
1191-  B0 BA      BCS  $114D
1193-  20 ED 11   JSR  $11ED
1196-  29 FE      AND  #$FE
1198-  85 69      STA  $69
119A-  20 ED 11   JSR  $11ED
119D-  29 0E      AND  #$0E
119F-  18         CLC
11A0-  69 90      ADC  #$90
11A2-  85 6B      STA  $6B
11A4-  A9 FF      LDA  #$FF
11A6-  20 BB 11   JSR  $11BB
11A9-  38         SEC
11AA-  A5 6B      LDA  $6B
11AC-  E9 40      SBC  #$40
11AE-  85 6B      STA  $6B
11B0-  A9 00      LDA  #$00
11B2-  20 BB 11   JSR  $11BB
11B5-  4C C8 11   JMP  $11C8
11B8-  60         RTS
11B9-  00         BRK
11BA-  00         BRK
11BB-  8D 2C 03   STA  $032C
11BE-  A6 69      LDX  $69
11C0-  A4 6A      LDY  $6A
11C2-  A5 6B      LDA  $6B
11C4-  20 7C 0C   JSR  $0C7C
11C7-  60         RTS
11C8-  C6 81      DEC  $81
11CA-  D0 C0      BNE  $118C
11CC-  C6 82      DEC  $82
11CE-  EA         NOP
11CF-  EA         NOP
11D0-  D0 BA      BNE  $118C
11D2-  EA         NOP
11D3-  EA         NOP
11D4-  EA         NOP
11D5-  EA         NOP
11D6-  38         SEC
11D7-  A5 84      LDA  $84
11D9-  E9 2A      SBC  #$2A
11DB-  85 82      STA  $82
11DD-  85 84      STA  $84
11DF-  A5 50      LDA  $50
11E1-  38         SEC
11E2-  E9 7E      SBC  #$7E
11E4-  85 50      STA  $50
11E6-  C6 83      DEC  $83
11E8-  D0 A2      BNE  $118C
11EA-  4C B8 11   JMP  $11B8
11ED-  A9 00      LDA  #$00
11EF-  85 08      STA  $08
11F1-  85 09      STA  $09
11F3-  A2 07      LDX  #$07
11F5-  18         CLC
11F6-  B5 00      LDA  $00,X
11F8-  65 08      ADC  $08
11FA-  85 08      STA  $08
11FC-  A9 00      LDA  #$00
11FE-  65 09      ADC  $09
1200-  85 09      STA  $09
1202-  CA         DEX
1203-  D0 F0      BNE  $11F5
1205-  18         CLC
1206-  A5 09      LDA  $09
1208-  F0 08      BEQ  $1212
120A-  65 08      ADC  $08
120C-  85 08      STA  $08
120E-  90 02      BCC  $1212
1210-  E6 08      INC  $08
1212-  A5 08      LDA  $08
1214-  C9 FF      CMP  #$FF
1216-  D0 02      BNE  $121A
1218-  E6 08      INC  $08
121A-  E8         INX
121B-  B5 01      LDA  $01,X
121D-  CA         DEX
121E-  95 01      STA  $01,X
1220-  E8         INX
1221-  E0 08      CPX  #$08
1223-  D0 F5      BNE  $121A
1225-  A5 07      LDA  $07
1227-  85 67      STA  $67
1229-  60         RTS
```

concluded

Error Messages

With reference to my renumbering routine for TRS-80 published in the MAY 1979 issue, an error crept into the listing and line 224 should be ADD HL, DE not BC.

I have made several improvements since writing the article last year, which although not necessary to its functioning, I would recommend adding to tidy things up.

1. Delete lines 210 to 212 inclusive and replace them by

```
        LDIR
        LD A, STAR          ; check if any trailing
        EX DE, HL           ; stars left
CHKSTR  CP (HL)
        JR NZ, ERRRET–$
        LD (HL), SPACE      ; clear them out
        INC HL
        JR CHKSTR–$
ERRRET  POP IX
```

2. Delete lines 230 to 247 inclusive and replace them by

```
        LD HL, ERRMES
        CALL OUTSTR         ; output 'omitted'
        POP HL
        CALL OUTSTR         ; output number
                              omitted
        LD HL, (CURSOR)
        LD A, L
        OR OFH              ; tab across screen
        LD L, A
        INC HL
        LD A, H
        LD (CURSOR), HL
        CP 40H              ; end of screen?
        JR NZ, ERRRET–$
        CALL KBD            ; wait for key
        CALL CLS
```

3. Also declare OUTSTR EQU 28A7H; this prints a string pointed to by HL and terminated by O.
   To provide the zero for the above a DEFB O must be put after the DEFM 'OMITTED:' and also after the TEMP DEFS 5.

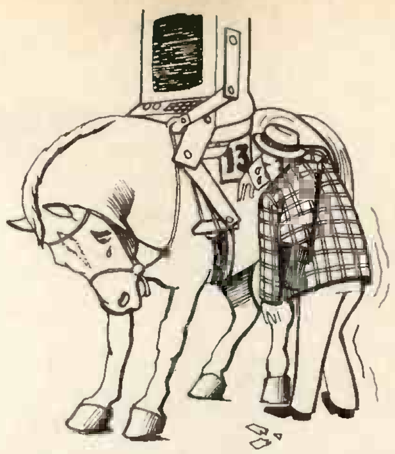
This increases the size of the program so the ORG must be changed to 4E3DH for a 4k and 7E3DH for a 16k machine and the entry points will now be 20029 for 4k and 32317 for 16k machines. MEMORY SIZE must be adjusted similarly to one less.

The entry points for the published version should of course be 20040 and 32328.

I. Addinsell
Author's address: Top of the Hill, Windmill Lane, Preston on the Hill, Nr. Warrington, Cheshire.

In Clifford Clark's 'Diagnostic Program', (PCW, May) p23, col 2, para 1, lines 14 & 15 should read: "These locations are in the 'Display Memory Section' as illustrated in the March 79 Issue, p25, fig. 1.'

The Programmed Punter



PERMS

A BASIC COMPUTER PROGRAM FOR THE BETTING MAN

Dr. M.R.J. Morgan

Department of Biological Sciences, University of Lancaster

Many of us enjoy the occasional flutter at the races, and filling in the weekly football pools coupon is something of a national pastime. Napoleon Bonaparte saw us as a "nation of shopkeepers" and the betting shop has become a familiar feature of every High Street, a symptom of recent explosive growth in the gambling industry. The professional gambler studies the odds as a way of life because for him betting is a serious business. We amateurs can increase the fun of betting if we make the effort to understand something of the mathematical basis of gambling and a good place to start is with permutations. This BASIC program offers the punter the means of determining the number of possible permutations over a wide enough range to cover all likely requirements. The calculation of permutations also has more "serious" applications, notably in the fields of probability and statistics. Closer familiarity with such concepts and a better appreciation of the odds may give the punter food for thought.

Background

The number of possible permutations of N from M is given by:

$$\frac{M!}{N!\ (M-N)!}$$

where the shriek notation! denotes the factorial of the adjacent number, e.g. the factorial of 5 is 5! or 5 x 4 x 3 x 2 x 1 = 120. Suppose we want to determine the number of possible treble bets for 8 horses (in different races), then the calculation uses N = 3, M = 8 and (M-N) = 5 so that the expression becomes:

$$\frac{8!}{3! \times 5!} = \frac{8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1}{(3 \times 2 \times 1) \times (5 \times 4 \times 3 \times 2 \times 1)} = \frac{40320}{6 \times 120} = 56$$

If funds are short we might decide to eliminate one or two of our selections to reduce the number of treble bets to 35 for 7 horses, or 20 for 6 horses. Alternatively we might prefer to retain the original 8 selections but to bet all possible doubles instead, giving 28 bets. Similarly, when filling in our football pools coupon, we can calculate that for a full perm of any 8 from 10 we will need to subscribe to 45 lines. Increasing the number of selections to 11 or 12 increases the number of lines required to 165 or 495 respectively, so we may well want to reduce our stake per line.

For fairly small numbers these calculations are not terribly difficult but they tend to become tedious if more than one or two have to be executed. With larger numbers the calculations can become quite time-consuming and are very prone to error. Why not let the personal computer do the donkey work? (PCW *Here we have the not implausible implication that only donkeys bet on horses* PCW)

Running the Program

The program makes full use of the conversational facilities of BASIC and is largely self-explanatory. Two modes of operation are offered. Mode 1 provides a complete tabulation of permutations of N from M for all values of N and M between 1 and 20. The table is best suited to hard-copy output and is printed in four sections so as to fit into standard teletype format; the complete table is then obtained by cutting the teletype paper and pinning up the four sections in horizontal alignment. The complete table is suitable for display on the wall of your local betting shop, at the club, for your pools syndicate, or your den at home. A permanent copy of the table will meet many regular needs and stand you in good stead with the lads of the betting fraternity. It will come in handy for settling arguments, or even bets!

The second, alternative, mode of the program permits the calculation of particular permutations, one at a time, over a much wider range of values of N and M. This mode of operation is suitable for any output device, including VDUs; the program accepts your chosen values of N, M and recycles indefinitely to assist your studies of probability theory. When you have finally perfected your understanding and conjured up the ideal full perm or master bet, your exit from the program is accompanied by some sound advice from the computer.

How the Program Works

For the calculation of individual permutations the necessary factorials M!, N! and (M-N)! are evaluated in turn in a simple subroutine and then the ratio M!/N! x (M-N)! is determined. When using larger numbers for N or M the factorial calculation involves many terms and it is easy to cause an overflow at statement 2040 unless steps are taken to avoid doing so. For the BASIC system under which the program was written the upper limit of the numerical range is approximately 10^{76} which means that the largest factorial that can be calculated is $57! = 4.05 \times 10^{76}$.

Statements 1200 – 1260 check that N and M are within the range 1 to 57 and that M is greater than N. It is essential that N, M and (M-N) should not assume the value 0 or become negative. By definition, the factorial of zero is unity, 0! = 1, but this special case would not be correctly handled by the factorial subroutine as it stands. Although it would be quite easy to incorporate a special treatment for 0! in the subroutine, it is not worth doing so in this application because zero values cannot arise in real-life input. Similarly, it is not possible to calculate factorials for negative values, so the program checks that these have not been inadvertently introduced. There is one real condition not catered for by the program, namely where N = M so that (M-N) = 0, but this is trivial because it should be fairly obvious that there is only 1 possible permutation of, say, 37 from 37 (or 22 from 22, etc.). Clearly, it is only possible to calculate factorials for integral values, so as a final safeguard, if any non-integral values are input inadvertently for N or M, the program takes the corresponding integer numbers.

In the alternative mode of operation leading to the output of a table of permutations, the program builds up the numbers required for the table in a matrix. The value taken by the numbers occupying the cells of the matrix is derived from adjacent elements of the array, without employing the factorial subroutine.

	x	y	
		z	

This is possible because the number in each cell is equal to the sum of the number in the cell immediately above it in the same column plus the number in the cell one column to the left and one row above, z = x + y.

Technical Points

Strings

In the teletype listing of the program, the dollar sign $ is represented as a back-slash \; thus A\ should be read as A$.

For the ICL BASIC Interpreter X98B under which the program was written there is a standard string length of 15 characters. For other BASIC Interpreters or Compilers it may be essential or desirable to dimension all strings. The program only uses a single string and this could be catered for by inserting the statement:

```
1075 D1M A$ (3)
```

Matrix Statements

For versions of BASIC without matrix-handling facilities, the statement:

```
1680 MAT PRINT Q
```

can be replaced by FOR loops enclosing a PRINT statement referring to the required element of array Q.

FOR Loops

The statement:

```
2030 FOR J = 2 TO X
```

will only rarely give rise to problems but, if your version of BASIC will only step from 1, replacing this statement with:

```
2030 FOR J = 1 TO X
```

will not affect the calculations.

Numerical Range

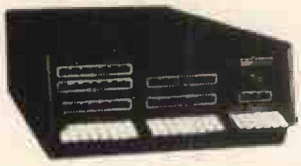
The ICL BASIC Interpreter can handle numbers up to approximately 10^{76}. If your version of BASIC has a lower limit you should decrease the value of 57 appropriately at statements 1240 and 1250 to avoid overflow problems at statement 2040. Conversely, if your BASIC can cope with numbers up to, say 10^{99} you will be able to calculate factorials up to 69! and can modify statements 1240 and 1250 accordingly.

```
1000 PRINT
1010 PRINT
1020 PRINT
1030 PRINT "PROGRAM FOR THE DETERMINATION OF PERMUTATIONS"
1040 PRINT "**********************************************"
1050 PRINT
1060 PRINT
1070 PRINT
1080 PRINT "DO YOU WANT A FULL TABLE OF PERMUTATIONS OF N FROM M"
1090 PRINT "FOR ALL VALUES OF N AND M BETWEEN 1 AND 20 ?"
1100 PRINT "PLEASE ANSWER YES OR NO";
1110 INPUT A\
1120 IF A\ = "YES" THEN 1500
1130 IF A\ = "NO" THEN 1170
1140 GOTO 1100
1150 PRINT "SORRY, N AND M MUST BOTH BE IN THE RANGE 1 TO 57"
1160 PRINT "AND, OF COURSE, M MUST BE GREATER THAN N"
1170 PRINT "INPUT YOUR CHOSEN VALUES OF N,M AND I WILL TELL YOU"
1180 PRINT "THE NUMBER OF POSSIBLE PERMUTATIONS OF N FROM M";
1190 INPUT N,M
1200 LET N = INT(N)
1210 LET M = INT(M)
1220 IF N < 1 THEN 1150
1230 IF M < 1 THEN 1150
1240 IF N > 57 THEN 1150
1250 IF M > 57 THEN 1150
1260 IF M <= N THEN 1150
1270 LET X = M
1280 GOSUB 2020
1290 LET M1 = F
1300 LET X = N
1310 GOSUB 2020
1320 LET N1 = F
1330 LET X = M-N
1340 GOSUB 2020
1350 LET O1 = F
1360 PRINT
1370 PRINT
1380 PRINT
1390 PRINT"THERE ARE";M1/(N1*O1);"POSSIBLE PERMUTATIONS OF";N;"FROM";M
1400 PRINT
1410 PRINT
1420 PRINT
1430 PRINT "DO YOU WANT TO CALCULATE ANOTHER PERMUTATION ?"
1440 PRINT "PLEASE ANSWER YES OR NO";
1450 INPUT A\
1460 IF A\ = "YES" THEN 1170
1470 IF A\ = "NO" THEN 2070
1480 GOTO1440
1490 REM PERM ANY N FROM M
1500 DIM P(20,20), Q(20,5)
1510 FOR N = 1 TO 20
1520 FOR M = 1 TO 20
1530 LET L = 1
1540 FOR J = 1 TO N
1550 LET K = M+1-J
1560 LET L = L*K/J
1570 NEXT J
1580 LET P(M,N) = L
1590 NEXT M
1600 NEXT N
1610 FOR J = 0 TO 3
1620 FOR N = 1 TO 5
1630 FOR M = 1 TO 20
1640 LET Q(M,N) = P(M,N+J*5)
1650 NEXT M
1660 NEXT N
1670 GOSUB 1770
1680 MAT PRINT Q
1690 PRINT
1700 PRINT
1710 PRINT
1720 PRINT
1730 PRINT
1740 NEXT J
1750 GOTO 2070
1760 REM TABULATION SUBROUTINE
1770 PRINT
1780 PRINT
1790 PRINT
1800 PRINT TAB(20);
1810 PRINT "TABLE OF PERMUTATIONS"
1820 PRINT TAB(20);
1830 PRINT "*********************"
1840 PRINT
1850 PRINT
1860 PRINT
1870 PRINT
1880 PRINT "ANY --->"
1890 FOR K = 1 TO 5
1900 PRINT TAB(15*(K-1));
1910 PRINT K+J*5;
1920 NEXT K
1930 PRINT
1940 PRINT
1950 PRINT
1960 IF J > 0 THEN 1990
1970 PRINT "FROM"
1980 GOTO 2000
1990 PRINT
2000 RETURN
2010 REM FACTORIAL SUBROUTINE
2020 LET F = 1
2030 FOR J = 2 TO X
2040 LET F = F*J
2050 NEXT J
2060 RETURN
2070 PRINT "OK   BEST OF LUCK WITH YOUR PUNTING"
2080 PRINT "BUT REMEMBER THE ODDS YOU ARE UP AGAINST !"
2090 END
```

DPS-1

Introducing the DPS-1 the full IEEE S100 bus computer system from Ithaca Intersystems — the S100 experts.

FOR EDUCATION, INDUSTRY, RESEARCH and all professional uses, including hardware and software development, low cost OEM systems, teaching applications etc.



A MINI COMPUTER using MICRO technology at a ridiculous MICRO price!! The front panel with a backplane and power supply accepts S100 bus boards from many manufacturers.

Just look at these professional features!

- ★ FRONT PANEL (we won't ask you to debug our hardware, but we will give you the tools to debug yours!). Includes breakpoints, write or jump to any location, single or slow step (0.1 to 1000 IPS), stop on any data byte or address, scope trigger on breakpoint, repeat instruction (e.g. NOP's) and many other hardware diagnostic facilities.
- ★ 25 Amp power supplies (all rails seperately fused)
- ★ 20 slot IEEE S100 motherboard
- ★ Full 16 data bit, 24 bit address lines as per IEEE (1979) S100 specification
- ★ Will run 16 bit micro S100 boards when available (e.g. Z8000 and 8086)
- ★ Suitable for 6800, 6502, 8080, 8085, Z80, Z80A etc.

The DPS-1 comes as a mainframe with front panel, motherboard and power supply. The system is truly modular allowing the user to build up the system he requires in his own time.

S100 boards from a number of manufacturers will plug into the DPSI IEEE S100 bus.

Just add S100 Memory Boards — S100 cpu board — S100 disk controller boards — S100 I/O boards — S100 video and/or graphics boards — S100 EPROM boards — S100 disk boards etc.

All Ithaca Intersystems OEM products including K2 disk operating system and PASCAL/Z on 8" floppy drives will run in the DPS-1.

DPS-1 with S100 Z80 cpu board £695

DPS-1 less S100 Z80 cpu board £645

OEM S100 boards from the experts!

	bare board	Assembled and tested
8K Static RAM board	£18.75	£123.75
Z80 cpu board (2MHZ)	£26.25	£131.25
Z80 cpu board (4MHZ)	£26.25	£153.75
2708/2716 EPROM board	£18.75	£63.75
Prototype board	£18.75	—
Video display board (64x16, 128U/L Ascii)	£18.75	£108.75
New products from Ithaca audio!		
High density graphics (1024 x 512 points)	T.B.A.	T.B.A.
Disk controller (up to 4 single or double sided drives)	£26.25	£131.25
I/O board (serial and parallel outputs)	T.B.A.	T.B.A.
S100 front panel (as used in DPS1)	—	£245.00

Over 15,000 boards delivered worldwide

InterSystems

Software for your S100 system

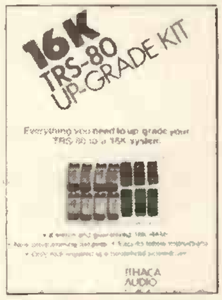
K2 operating system
8" disk based operating system — distributed on Shugat compatible 8" floppy disk ★ TED—52 command text editor with Macros. ★ PIP—File and directory handler. ★ ASMBLE—full Z80 2 pass assembler. ★ HDT—Hex debug tool. ★ QCI—Utility overlay/command decoder. ★ SYSGEN—System builder. ★ COPY—disk to disk file copier. ★ DUP—disk duplicator. **£56.25**

PASCAL/Z The new language for Micros £131.25
Runs under K2 operating system.
★ Compiler that produces direct assembler code — NO NEED for slow run time P-code interpreter. ★ Comes complete with Macro assembler. ★ Produces binary object modules — small and fast. ★ Modules are re-entrant and can be put into ROM. ★ IMBED, TRACE and ERROR debug facilities.

ASMBLE/Z Z80 Macro assembler £37.50
★ Full 2 pass Macro Assembler. ★ IF and ELSE — 255 nesting levels. ★ Produces symbol table. ★ Relative jumps.

UP-GRADE KITS

Trying to add computer memory is not much fun if you dont get everything you need.



Receiving unprogrammed jumpers and having to program them yourself is not much better. Most Important, that's the place where the problems are introduced.
So Ithaca Audio's better Idea is the Simple Up-Grade. Each Simple Up-Grade is specially designed to make adding memory foolproof. We include all the parts you'll need; 8 prime, tested 16K RAMs, along with concise step by step directions and diagrams. And if a personality jumper is required, *it's premade*.

The TRS-80* memory expansion was our first Simple Up-Grade. Now there are two more — for owners of Apple II** and Exidy Sorcerer*** computers. Each kit is 100% guaranteed — if a part ever fails, we replace it FREE. Your Ithaca Audio dealer has them in stock, only £99. Now you can afford to add high quality, high density memory to your system for remarkably little — far less than you would expect to pay from Radio Shack, Apple, or Exidy directly.
These Simple Up-Grades are Ithaca Audio's first step in adding more capability and reliability to your computer at lower cost. Other Up-Grades are on the way to your dealer now.

*TRS-80 is a registered trademark of Tandy Corp.
**Apple II is a registered trademark of Apple Computer, Inc.
***Sorcerer is a registered trademark of Exidy, Inc.

All prices quoted are exclusive of VAT.

CONTACT THESE DEALERS

NEWBEAR COMPUTING STORE — Telephone: Newbury (0635) 30505 **AIRAMCO** — Telephone: 0294 57755 Telex: 779808
SIRTON PRODUCTS — Telephone: 01-660 5617 **COMPSHOP LTD** — Telephone: 01-441 2922 Telex: 298755

ITHACA InterSystems
(formerly ITHACA AUDIO of New York)

UK & EUROPEAN DEALER ENQUIRIES INVITED — CONTACT
EUROPEAN SALES HEADQUARTERS
58 Crouch Hall Road, London N8 8HG. UK. Telephone: 01-341 2447 Telex: 299568

NORTH STAR COMPATIBLE SOFTWARE

ALLEN ASHLEY ENTERPRISES

PDS Program Development System for 8080 or Z80 computers . . £75.00
PDS is an exceptionally powerful assembly language development package. PDS includes a unified assembler/editor, a macro assembler with a relocating linking leader, a string-oriented text editor and trace debugger/disassembler. PDS supports full Z80 code favouring the Intel Instruction mnemonics, treating the Z80 superset as a logical and syntactical extension.

The following is a list of source modules compatible with PDS:–

MODULE	FUNCTION	REQUIREMENTS	
ALPHSORT	High speed alphabetic sort	None	£10.00
NUMRSORT	High speed numeric sort	None	£10.00
FPPACK	BCD floating point arithmetic	None	£10.00
FOURIER	Fast Fourier transform	FPPACK	£10.00
MINV	Matrix inversion	FPPACK	£10.00
MATPED	Matrix product	FPPACK	£ 7.50
RATPOL	Rational function and utilities	FPPACK	£ 7.50
SQRT	Square root	FPPACK	£ 5.00
TRIGS	Sine, cosine, TAN, ATAN	FPPACK, RATPOL	£10.00
LOGEXP	Exponential, logarithm, y^x	FPPACK, RATPOL	£10.00
FPIOP	Floating point I/O	None	£10.00
FORMAT	Formatted floating point output	None	£ 7.50
NFILES	North Star disk handler	None	£10.00
INOPS	Integer multiply/divide	None	£ 5.00
The complete set of modules listed above			£59.00

BYTE SHOP OF WESTMINSTER

NORTHSHARE North Star BASIC time sharing package £37.00
The NORTHSHARE system allows you to make much better use of your 8080 or Z80 CPU by allowing you to connect two to four terminals to your system and running North Star BASIC in a timesharing mode. Each terminal shares the same copy of BASIC simultaneously; but independent of the other user(s). All book-keeping is done by a separate task supervisor.

DIGITAL RESEARCH

CP/M* with Horizon customisation £90.00
CP/M is a proprietary general-purpose operating system for Intel 8080 or Zilog Z-80 microcomputer systems. CP/M's basic facilities include dynamic file management, a fast assembler, a general purpose text editor and an advanced debugger.

MAC Macro Assembler, features include assembly-time expressions, conditional assembly, page formatting and a powerful macro assembler £65.00

SID Symbolic Instruction Debugger £60.00

TEX Text formatter £60.00

DESPOOL permits simultaneous printing of data from disk while user executes another program from the console £45.00

NORTH STAR SOFTWARE EXCHANGE

NSSE1 Six games plus four utility programs . . . . £ 6.00
Games: Chomp, Lunar, Slots, Match, Startrek and Wumpus. Utilities: Memory 16, Memory 24, Memory 32 and Defunct.

NSSE2 Assortment of utility programs . . . . £ 6.00
Utilities: File, Heaps, TST16K Compact, Hexdec, Memset, Filedit, Memrymap, Sort and Alphsort.

NSSE3 This disk is entirely devoted to the Guess game . . . . £ 6.00

NSSE4 Mathematical routines mostly for matrix manipulations £ 6.00
Routines: Fast Fourier Transform, Gaussian Reduction, Matrix Multiplication, Matrix Inversion and Gaussian Elimination.

NSSE5 Complete assembler source code for the PILOT system . . £ 6.00

NSSE6 Assortment of applications programs . . . . £ 6.00
Programs: Library, Mailer, Intrate, Blorthm, Chekbook.

NSSE7 Games, mainly for use with SOL video board £ 6.00

NSSE8 Games programs, mostly in the form of simulations . . . . . £ 6.00
Games: Othello, Zoo, Hamurabi, Gunner, CIBM, Trek 16K.

The above programs have been certified by their contributors as being in the public domain. Neither North Star Computers Inc. or Interam Computer Systems Ltd. make any claim or offer any warranty that the programs are correct or suitable for any purpose. The set of eight diskettes containing software is priced at £35.00

NORTH STAR

PASCAL development system includes an editor, compiler, debugger and file handler £37.00

PAS-AUX Auxillary package for above, includes an assembler and utilities £27.00

The above software is supplied on 5.25" diskettes. Prices are exclusive of VAT @ 8% and postage & packing (add 75p). Please send large s.a.e. for full details.

Coming soon – Applications software, including General Ledger package for Incomplete records and an Inventory package.

Software products are being continually added to our range, so please enquire about software products not listed above.

01-834 0261/2733

INTERAM Computer Systems Ltd.
59 Moreton St. Victoria, London SW1

INTERAM

NASCOM FLOATING POINT

BASIC LEVEL C Runs on an 8K system
Full floating point arithmetic
Software compatible with Levels A and B
Contains all features found on A and B plus more
Line editor, 52 variables
Commands include:– FOR; NEXT; READ; DATA; PEEK; POKE; IN; OUT; USR; INKEY

BASIC LEVEL A Runs on the minimum Nascom
No extra memory needed
Fitted in 2 minutes in place of monitor
Integer arithmetic < , > , < > , < =, > =, =
Fast execution time
Keyboard pause and interrupt
Abbreviated commands allow economical use of your memory, and consist of:– LET; PRINT; IF; GOSUB; RETURN; INPUT; REM; STOP; GOTO; LIST; NEW; RUN; SAVE; ABS; RND

BASIC LEVEL B As previously advertised

LEVEL A or B in 2x2708 PROMS	**£21.50**
LEVEL B Cassette	**£ 7.50**
LEVEL C in 4x2708 PROMS	**£42.00**
LEVEL C Cassette	**£12.50**
Partial disassembler/debug	**£ 5.00**
Superstartrek for Level B or C	**£ 4.50**
Relocator package	**£ 4.00**

All products fully documented

C C SOFT (Southfields)
83 Longfield St. London SW18
Tel: 01-870 4891 (Anytime)

Callers by appointment please

EQUINOX 300

A powerful multi-user
multi-tasking
multi-language

16-bit microcomputer time-sharing system

supporting
* BASIC
* LISP
* PASCAL
* Floppy discs
* Hard discs

including a powerful Text Formatter, Assembly Language Development System and disc-based Sort utilities.

Priced from under £5,000

Write or phone for further information

EQUINOX COMPUTER SYSTEMS LTD
16 Anning Street,
New Inn Yard
London E.C.2.
(Tel: 01-739 2387/8/9)

PARKINSON'S REVAS

Continued from Vol. 2. No. 2.

D. W. Parkinson

A Z80 Reverse Assembler – Addendum

Murphy's Law (variation n):– No matter how many copies of a program are in use, any errors will only surface once the listing is set in type.

There is one small error in the published listing which results in RETI not being decoded. This is easily corrected by changing the first line in the RETN/RETI section from CP SA9 to CP SAA.

The REGPR routine produces a wrong set of register-pairs for the 16-bit arithmetic mnemonics when operating with IX or IY in place of HL. (eg ADD IX, BC appears as ADD IX, DE; ADD IY, BC as ADD IY, HL). This can be corrected by recording the REGPR routine as below. (The final jump is necessary to keep the change within the bounds of the old routine.)

When I wrote the original version of REVAS I did not bother to bar four invalid opcodes. They and their associated mnemonics are:–

```
ED 63  n  n  LD    (nn), HL
ED 6B  n  n  LD    HL, (nn)
ED 70        IN    (HL), (C)
ED 71        OUT   (C), (HL)
```

The first two the Z80 actually executes (look at the 'ED' instruction map to see why) but I assume are not documented because of the shorter 8080 equivalent. (Also the DD/FD prefix for IX/IY does not work). The latter two don't exist and I apologise for their appearance in the published version of REVAS. They can be prevented by adding the following code at the start of the IN R, (C) and OUT (C), R sections:–

```
    CP $E   ($2E for OUT section)
    JP Z, NOTVAL
```

Purists who want to remove the first two as well can insert:–

```
    AND $E
    CP   $C
    JP    Z,NOTVAL
```

at the start of the LD PP, ($NNNN) . . . . section. (Line 848)

Another point mentioned in the text of the article, but not explained as fully as it might have been, concerns DD and FD, the index operators. These only serve to set a flag to indicate use of the appropriate index register in place of HL, and if the following opcode does not reference HL the flag is not checked. A code sequence such as FD 3E 4F is decoded as LD A,$4F or DD C3 . . . . as JP $ . . . . rather than as invalid code.

```
                 0293 ;
                 0294 ;REGISTER PAIR DECODE/PRINT
                 0295 ;
      47         0296 REGPR:  LD    B,A          ;SAVE IN B
      0F         0297         RRCA               ;SHIFT PAIR ID BITS DOWN
      0F         0298         RRCA
      0F         0299         RRCA
      C6 02      0300         ADD   2            ;ALTER REG. CODING
      0E 06      0301         LD    C,6          ;PUT 6 IN C
      A1         0302         AND   C            ;ISOLATE REG. ID
      21 A1 F9   0303         LD    HL,RPRTAB    ;LOAD BASE ADDRESS
      20 07      0304         JR    NZ,NOTSP     ;SKIP IF NOT 'SP'
      2B         0305         DEC   HL           ;POINT TO "SP"
      CB 78      0306         BIT   7,B          ;IS IT PUSH/POP?
      28 94      0307         JR    Z,CCPY2      ;NO DONE
      2B         0308         DEC   HL           ;YES,POINT...
      2B         0309         DEC   HL           ;...TO "AF"
      B9         0310 NOTSP:  CP    C            ;IS IT "HL"?
      20 04      0311         JR    NZ,*+4       ;NO,SKIP
      3A 0B 10   0312         LD    A,(HXYFLG)   ;LOAD FLAG
      81         0313         ADD   C            ;ADJUST IF IX/IY
      C3 B2 FA   0314         JP    CCODES+7     ;SKIP TO CALL FTADR,JP COPY2
      00         0315         NOP                ;FILLER
```

altered code

```
F91F  ED A0      0237 COPY5:  LDI
F921  ED A0      0238 COPY4:  LDI
F923  ED A0      0239 COPY3:  LDI
F925  ED A0      0240 COPY2:  LDI
F927  ED A0      0241         LDI
F929  C9         0242         RET
F92A             0243 ;
F92A             0244 ;FORM ADDRESS FOR TABLE LOOK UP
F92A             0245 ;
F92A  4F         0246 FTADR:  LD    C,A          ;PUT OFFSET IN C
F92B  06 00      0247         LD    B,0          ;CLEAR B
F92D  09         0248         ADD   HL,BC        ;ADD IN TO ADDRESS BASE
F92E  C9         0249         RET
F92F             0250 ;
F92F             0251 ;INC/DEC REGISTER/REGISTER PAIR
F92F             0252 ;
F92F  21 47 F9   0253 INC:    LD    HL,INCM      ;POINT TO "INC"
F932  18 03      0254         JR    *+3          ;SKIP
```

(Listing continued)

```
F934  21 4A F9        0255 DEC:     LD    HL,DECM     ;POINT TO "DEC"
F937  CD 23 F9        0256          CALL  COPY3       ;COPY OVER
F93A  13              0257          INC   DE          ;SPACE
F93B  13              0258          INC   DE          ;SPACE
F93C  F1              0259          POP   AF          ;RECOVER OPCODE
F93D  CB 57           0260          BIT   2,A         ;TEST 8/16 BIT
F93F  28 3D           0261          JR    Z,REGPR     ;JUMP IF 16 BIT
F941  0F              0262          RRCA              ;SHIFT REG. ID....
F942  0F              0263          RRCA              ;...DOWN TO LOW A
F943  0F              0264          RRCA
F944  C3 C4 F9        0265          JP    SREG        ;RETURN AFTER PRINTING REGISTER
F947                  0266 ;
F947  49 4E 43        0267 INCM:    DB    'INC'
F94A  44 45 43        0268 DECM:    DB    'DEC'
F94D                  0269 ;
F94D                  0270 ;16-BIT DIRECT LOAD
F94D                  0271 ;
F94D  CD 06 F9        0272 LD16:    CALL  WRLD        ;WRITE "LD"
F950  F1              0273          POP   AF          ;RECOVER OPCODE
F951  CD 7E F9        0274          CALL  REGPR       ;PRINT REGISTER PAIR
F954  CD 11 F9        0275          CALL  COMMA       ;","
F957  CD DB F8        0276 LD16A:   CALL  BYTE        ;GET LO BYTE
F95A  4F              0277          LD    C,A         ;SAVE IN C
F95B  CD DB F8        0278          CALL  BYTE        ;GET HI BYTE
F95E  67              0279          LD    H,A         ;PUT IN H
F95F  69              0280          LD    L,C         ;RECOVER FIRST BYTE
F96D  ED 53 10 10     0281 LD16B:   LD    (LABELP),DE;SAVE POINTER
F964  22 0E 10        0282          LD    (LABELA),HL;SAVE ADDRESS (?)
F967  CD 17 F9        0283          CALL  DOLLAR      ;"$"
F96A  C3 D0 F8        0284          JP    HEX4        ;RETURN,PRINTING HL.
F96D                  0285 ;
F96D                  0286 ;ADD HL
F96D                  0287 ;
F96D  21 34 FA        0288 ADDHL:   LD    HL,ARTAB    ;POINT TO "ADD"
F970  CD 23 F9        0289          CALL  COPY3       ;WRITE IT
F973  13              0290          INC   DE          ;SPACE
F974  13              0291          INC   DE          ;SPACE
F975  3E 20           0292          LD    A,$20       ;MAKE LIKE "HL"
F977  CD 7E F9        0293          CALL  REGPR       ;WRITE HL,IX,OR IY
F97A  CD 11 F9        0294          CALL  COMMA       ;","
F97D  F1              0295          POP   AF          ;RECOVER OPCODE
F97E                  0296 ;
F97E                  0297 ;REGISTER PAIR DECODE/PRINT
F97E                  0298 ;
F97E  47              0299 REGPR:   LD    B,A         ;SAVE IN B
F97F  0F              0300          RRCA              ;SHIFT PAIR ID BITS DOWN
F980  0F              0301          RRCA
F981  0F              0302.         RRCA
F982  C6 02           0303          ADD   2           ;ALTER REG. CODING
F984  E6 06           0304          AND   6           ;ISOLATE REG. ID
F986  21 A1 F9        0305          LD    HL,RPRTAB   ;LOAD BASE ADDRESS
F989  20 09           0306          JR    NZ,NOTSP    ;SKIP IF NOT 'SP'
F98B  2B              0307          DEC   HL          ;CORRECT POINTER
F98C  CB 78           0308          BIT   7,B         ;IS IT PUSH/POP?
F98E  28 95           0309          JR    Z,COPY2     ;NO DONE
F990  2B              0310          DEC   HL          ;YES,POINT...
F991  2B              0311          DEC   HL          ;...TO 'AF'
F992  18 91           0312          JR    COPY2       ;COPY OVER.
F994  47              0313 NOTSP:   LD    B,A         ;SAVE REG.
F995  3A 0B 10        0314          LD    A,(HXYFLG);LOAD FLAG
F998  80              0315          ADD   B           ;OFFSET IF IX/IY
F999  CD 2A F9        0316          CALL  FTADR       ;ADJUST ADDRESS
F99C  18 87           0317          JR    COPY2       ;COPY OVER
F99E                  0318 ;
F99E  41 46 53        0319          DB    'AFS'
F9A1  50 41 42 43     0320 RPRTAB:  DB    'PABCDE'
      44 45
F9A7  48 4C 49 58     0321 HXYTAB:  DB    'HLIXIY'
      49 59
F9AD                  0322 ;
F9AD                  0323 ;8-BIT LOAD INSTRUCTIONS
F9AD                  0324 ;
F9AD  F1              0325 LOAD8:   POP   AF          ;RECOVER OPCODE
F9AE  FE 76           0326          CP    $76         ;HALT?
F9B0  28 60           0327          JR    Z,HALT      ;YES,JUMP
F9B2  F5              0328          PUSH  AF          ;SAVE OPCODE
F9B3  CD 06 F9        0329          CALL  WRLD        ;WRITE "LD"
F9B6  0F              0330          RRCA              ;SHIFT REG ID DOWN
F9B7  0F              0331          RRCA
F9B8  0F              0332          RRCA
F9B9  CD C4 F9        0333          CALL  SREG        ;PRINT IT
F9BC  CD 11 F9        0334          CALL  COMMA       ;","
F9BF  F1              0335 L8B:     POP   AF          ;GET OPCODE BACK
F9C0  CB 77           0336          BIT   6,A         ;TEST FOR IMMED. DATA
F9C2  28 45           0337          JR    Z,IMM       ;JUMP IF SO
F9C4                  0339 ;
F9C4                  0340 ;SINGLE REGISTER DECODE
F9C4                  0341 ;
F9C4  3C              0342 SREG:    INC   A           ;ADJUST CODING
F9C5  E6 07           0343          AND   7           ;ISOLATE IT
F9C7  FE 07           0344          CP    7           ;IS IT MEMORY?
F9C9  28 09           0345          JR    Z,MEM       ;YES,JUMP
F9CB  21 A2 F9        0346          LD    HL,RPRTAB+1;LOAD BASE ADDRESS
F9CE  CD 2A F9        0347          CALL  FTADR       ;FORM ADDRESS
F9D1  ED A0           0348          LDI               ;COPY OVER REG
F9D3  C9              0349          RET
F9D4  3E 28           0350 MEM:     LD    A,'('       ;MEM INDIRECT
F9D6  12              0351          LD    (DE),A      ;WRITE "("
F9D7  13              0352          INC   DE          ;BUMP ADDRESS
F9D8  21 A7 F9        0353          LD    HL,HXYTAB
F9DB  3A 0B 10        0354          LD    A,(HXYFLG);OFFSET IF INDEX
F9DE  CD 2A F9        0355          CALL  FTADR       ;FORM ADDRESS
F9E1  CD 25 F9        0356          CALL  COPY2       ;COPY OVER
F9E4  3A 0B 10        0357          LD    A,(HXYFLG);IS IT INDEX?
F9E7  A7              0358          AND   A           ;TEST
F9E8  28 1A           0359          JR    Z,NOTIXY    ;NO,JUMP
F9EA                  0360 ;
F9EA  CD DB F8        0361          CALL  BYTE        ;YES,GET OFFSET
F9ED  A7              0362          AND   A           ;TEST IT
F9EE  0E 2B           0363          LD    C,'+'       ;SET POSITIVE
F9F0  28 12           0364          JR    Z,NOTIXY    ;SKIP IF ZERO
F9F2  F2 F9 F9        0365          JP    P,PLUS      ;JUMP IF IT WAS +VE
F9F5  ED 44           0366          NEG               ;2'S COMP
F9F7  0E 2D           0367          LD    C,'-'       ;CHANGE SIGN
F9F9  47              0368 PLUS:    LD    B,A         ;SAVE MAGNITUDE
F9FA  79              0369          LD    A,C         ;GET SIGN
F9FB  12              0370          LD    (DE),A      ;PRINT IT
F9FC  13              0371          INC   DE          ;BUMP ADDRESS
F9FD  78              0372          LD    A,B         ;GET MAGNITUDE
F9FE  CD 17 F9        0373          CALL  DOLLAR      ;"$"
FA01  CD B8 F8        0374          CALL  HEX2        ;PRINT MAGNITUDE
FA04  3E 29           0375 NOTIXY:  LD    A,')'       ;CLOSE BRACKETS
FA06  12              0376          LD    (DE),A
FA07  13              0377          INC   DE
FA08  C9              0378          RET
FA09                  0379 ;
FA09                  0380 ; GET SINGLE BYTE IMMEDIATE
FA09                  0381 ;
FA09  CD DB F8        0382 IMM:     CALL  BYTE        ;GET DATA
FA0C  CD 17 F9        0383          CALL  DOLLAR      ;PRINT "$"
FA0F  C3 B8 F8        0384          JP    HEX2        ;PRINT DATA
FA12                  0385 ;
FA12                  0386 ; HALT INSTRUCTION
FA12                  0387 ;
FA12  21 18 FA        0388 HALT:    LD    HL,HALTM
FA15  C3 21 F9        0389          JP    COPY4       ;EXIT VIA COPY4
FA18  48 41 4C 54     0390 HALTM:   DB    'HALT'
FA1C                  0391 ;
FA1C                  0392 ;8-BIT ARITHMETIC
FA1C                  0393 ;
FA1C  F1              0394 ARITH8:  POP   AF          ;RECOVER OPCODE
FA1D  EE 40           0395          XOR   $40         ;TOGGLE BIT
FA1F  F5              0396          PUSH  AF          ;PUT IT BACK
```

```
FA20 0F           0397          RRCA             ;GET OP ID
FA21 0F           0398          RRCA
FA22 E6 0E        0399          AND   $E         ;ISOLATE IT
FA24 47           0400          LD    B,A        ;*3 IT
FA25 0F           0401          RRCA
FA26 80           0402          ADD   B
FA27 21 34 FA     0403          LD    HL,ARTAB   ;LOAD BASE ADDRESS
FA2A C4 2A F9     0404          CALL  NZ,FTADR   ;FORM ADDRESS
FA2D CD 23 F9     0405          CALL  COPY3      ;COPY MNEMONIC
FA30 13           0406          INC   DE         ;BUMP ADDRESS
FA31 13           0407          INC   DE
FA32 18 8B        0408          JR    L8B        ;PRINT OPERAND
FA34              0409 ;
FA34 41 44 44 41  0410 ARTAB:   DB    'ADDADCSUBSBC'
     44 43 53 55
     42 53 42 43
FA40 41 4E 44 58  0411          DB    'ANDXOROR CP '
     4F 52 4F 52
     20 43 50 20
FA4C              0412 ;
FA4C              0413 ;POP
FA4C              0414 ;
FA4C 21 5F FA     0415 POP:     LD    HL,POPM    ;LOAD POINTER
FA4F 18 03        0416          JR    *+3        ;SKIP
FA51              0417 ;
FA51              0418 ;PUSH
FA51              0419 ;
FA51 21 5B FA     0420 PUSH:    LD    HL,PUSHM   ;LOAD POINTER
FA54 CD 21 F9     0421 PUSH1:   CALL  COPY4      ;PRINT MNEMONIC
FA57 13           0422          INC   DE         ;SPACE
FA58 C3 7D F9     0423          JP    REGPR-1    ;GET OP & PR. REG PAIR
FA5B 50 55 53 48  0424 PUSHM:   DB    'PUSH'
FA5F 50 4F 50 20  0425 POPM:    DB    'POP '
FA63              0426 ;
FA63              0427 ; CALL/IX/EXTENDED/IY
FA63              0428 ;
FA63 F1           0429 CALETC:  POP   AF         ;RECOVER OPCODE
FA64 FE ED        0430          CP    $ED        ;EXTENDED?
FA66 CA 08 FD     0431          JP    Z,EXTND    ;YES,JUMP
FA69 FE CD        0432          CP    $CD        ;CALL?
FA6B 28 0E        0433          JR    Z,CJR+1    ;YES,JUMP
FA6D              0434 ; SET IX/IY FLAG
FA6D D6 F9        0435          SUB   $F9        ;LEAVES 4 IF IY
FA6F 30 02        0436          JR    NC,*+2     ;SKIP IF 4
FA71 3E 02        0437          LD    A,2        ;ELSE LOAD 2
FA73 32 0B 10     0438          LD    (HXYFLG),A;SET FLAG
FA76 F1           0439          POP   AF         ;THROW AWAY RETURN
FA77 C3 19 F8     0440          JP    MORE       ;GO FOR MORE
FA7A              0441 ;
FA7A              0442 ; CALL/JUMP/RETURN
FA7A              0443 ;
FA7A F1           0444 CJR:     POP   AF         ;RECOVER OPCODE
FA7B F5           0445          PUSH  AF         ;SAVE IT
FA7C E6 06        0446          AND   6          ;ISOLATE ID
FA7E 07           0447          RLCA
FA7F 21 9F FA     0448          LD    HL,CJRTAB ;LOAD BASE ADDRESS
FA82 C4 2A F9     0449          CALL  NZ,FTADR   ;FORM ADDRESS
FA85 CD 21 F9     0450          CALL  COPY4      ;WRITE MNEMONIC
FA88 13           0451          INC   DE         ;SPACE
FA89 F1           0452          POP   AF         ;GET OPCODE BACK
FA8A FE C9        0453          CP    $C9        ;UNCOND RETURN?
FA8C C8           0454          RET   Z          ;YES,RETURN
FA8D CB 47        0455          BIT   0,A        ;UNCOND. CALL/JUMP?
FA8F 20 0B        0456          JR    NZ,UNCND   ;YES,SKIP
FA91 F5           0457          PUSH  AF         ;SAVE AGAIN
FA92 CD AB FA     0458          CALL  CCODES     ;ADD CONDITION
FA95 F1           0459          POP   AF         ;RECOVER AGAIN
FA96 E6 07        0460          AND   7          ;CHECK FOR RETURN?
FA98 C8           0461          RET   Z          ;YES,RETURN
FA99 CD 11 F9     0462          CALL  COMMA      ;","
FA9C C3 57 F9     0463 UNCND:   JP    LD16A      ;EXIT WRITING "$NNNN"
FA9F              0464 ;
```

```
FA9F 52 45 54 20  0465 CJRTAB:  DB    'RET JP  CALL'
     4A 50 20 20
     43 41 4C 4C
FAAB              0466 ;
FAAB              0467 ; DECODE CONDITION CODES
FAAB              0468 ;
FAAB 0F           0469 CCODES:  RRCA             ;SHIFT DOWN
FAAC 0F           0470          RRCA
FAAD E6 0E        0471          AND   $E         ;ISOLATE
FAAF 21 B8 FA     0472          LD    HL,CCTAB   ;LOAD BASE ADDRESS
FAB2 C4 2A F9     0473          CALL  NZ,FTADR   ;FORM ADDRESS
FAB5 C3 25 F9     0474          JP    COPY2      ;RETURN VIA COPY2
FAB8              0475 ;
FAB8 4E 5A 20 5A  0476 CCTAB:   DB    'NZ ZNC CPOPE P M'
     4E 43 20 43
     50 4F 50 45
     20 50 20 4D
FAC8              0477 ;
FAC8              0478 ; SINGLE BYTE ROTATE & MISC.
FAC8              0479 ;
FAC8 F1           0480 ROTMIS:  POP   AF         ;RECOVER OPCODE
FAC9 0F           0481          RRCA
FACA E6 1C        0482          AND   $1C        ;ISOLATE ID
FACC 21 D5 FA     0483          LD    HL,RMTAB   ;LOAD BASE ADDRESS
FACF C4 2A F9     0484          CALL  NZ,FTADR   ;FORM ADDRESS
FAD2 C3 21 F9     0485          JP    COPY4      ;WRITE TO BUFFER
FAD5              0486 ;
FAD5 52 4C 43 41  0487 RMTAB:   DB    'RLCARRCARLA RRA '
     52 52 43 41
     52 4C 41 20
     52 52 41 20
FAE5 44 41 41 20  0488          DB    'DAA CPL SCF CCF '
     43 50 4C 20
     53 43 46 20
     43 43 46 20
FAF5              0489 ;
FAF5              0490 ; RESTART
FAF5              0491 ;
FAF5 21 06 FB     0492 RST:     LD    HL,RSTM    ;WRITE "RST"
FAF8 CD 23 F9     0493          CALL  COPY3
FAFB 13           0494          INC   DE         ;SPACE
FAFC 13           0495          INC   DE         ;SPACE
FAFD F1           0496          POP   AF         ;RECOVER OPCODE
FAFE E6 38        0497          AND   $38        ;ISOLATE ADDRESS
FB00 CD 17 F9     0498          CALL  DOLLAR     ;WRITE "$"
FB03 C3 B8 F8     0499          JP    HEX2       ;PRINT RST ADDRESS
FB06              0500 ;
FB06 52 53 54     0501 RSTM:    DB    'RST'
FB09              0502 ;
FB09              0503 ; WRITE "NOP"
FB09              0504 ;
FB09 21 0F FB     0505 NOP:     LD    HL,NOPM
FB0C C3 23 F9     0506          JP    COPY3
FB0F 4E 4F 50     0507 NOPM:    DB    'NOP'
FB12              0508 ;
FB12              0509 ; DJNZ
FB12              0510 ;
FB12 21 1B FB     0511 DJNZ:    LD    HL,DJNZM   ;WRITE "DJNZ"
FB15 CD 21 F9     0512          CALL  COPY4
FB18 13           0513          INC   DE         ;SPACE
FB19 18 20        0514          JR    UCD        ;REST IN "JR" ROUTINE
FB1B 44 4A 4E 5A  0515 DJNZM:   DB    'DJNZ'
FB1F              0516 ;
FB1F              0517 ; NOP/DJNZ/JR NZ/JR NC
FB1F              0518 ;
FB1F F1           0519 NOPETC:  POP   AF         ;RECOVER OPCODE
FB20 FE 10        0520          CP    $10        ;DJNZ?
FB22 28 EE        0521          JR    Z,DJNZ     ;YES,JUMP
```

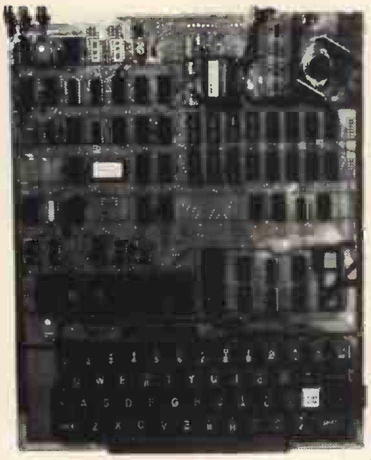
To Be Continued

EUROPES FASTEST SELLING ONE BOARD COMPUTER — JUST CHECK THE SPEC'S.

COMPUKIT UK101

SAMPLE TAPE included FREE

LOW COST SUPERBOARD IN KIT FORM



The Compukit UK101 Character Set

The Compukit UK101 has everything a one board 'superboard' should have.

- ★ Uses ultra-powerful 6502 microprocessor.
- ★ 50Hz Frame refresh for steady clear picture (U.S.A. products with 60Hz frame refresh always results in jittery displays)
- ★ 48 chars by 16 lines — 1K memory mapped video system providing high speed access to screen display enabling animated games and graphs.
- ★ Extensive 256 character set which includes full upper and lower case alphanumerics, Greek symbols for mathematical constants and numerous graphic characters enabling you to form almost any shape you desire anywhere on the screen.
- ★ 8K full Microsoft Basic in ROM compatible with PET, APPLE SORCERER hence taking the headache out of programming by using simple English statements. Much faster than currently available personal computers.
- ★ Professional 52 Key keyboard in 3 colours — software polled meaning that all debouncing and key decoding done in software.

FULL CONSTRUCTION DETAILS IN P.E. AUG 1979 EDITION

Delivery date June 1979 at the 1979 MicroComputer Show Customer orders in strict rotation only.

COMMANDS

CONT LIST	NEW	NULL	RUN

STATEMENTS

CLEAR DATA	DEF	DIM	END	FOR
GOTO GOSUB	IF..GOTO	IF..THEN	INPUT	LET
NEXT ON..GOTO	ON..GOSUB	POKE	PRINT	READ
REM RESTORE	RETURN	STOP		

EXPRESSIONS

OPERATORS

+ .*./.↑ NOT.AND.OR. >.<.<>.>=<= RANGE 10^{-32} to 10^{+32}

VARIABLES

A.B.C. ....Z and two letter variables
The above can all be subscripted when used in an array. String variables use above names plus $.e.g.A$.

FUNCTIONS

ABS(X)	ATN(X)	COS(X)	EXP(X)	FRE(X)	INT(X)
LOG(X)	PEEK(I)	POS(I)	RND(X)	SGN(X)	SIN(X)
SPC(I)	SQR(X)	TAB(I)	TAN(X)	USR(I)	

STRING FUNCTIONS

ASC(X$)	CHR$(I)	FRE(X$)	LEFT$(X$.I)	LEN(X$)	MID$(X$.I.J)
RIGHT$(X$.I)		STR$(X)		VAL(X$)	

- ★ Video output and UHF Highgrade modulator (8Mz Bandwidth) which connects direct to the aerial socket of your T.V. Channel 36 UHF.
- ★ Fully stabilised 5V power supply including transformer on board.
- ★ Standard KANSAS city tape interface providing high reliability program storage — use on any standard domestic tape or cassette recorder.
- ★ 4K user RAM expandable to 8K on board £49 extra.
- ★ 40 line expansion interface socket on board for attachment of extender card containing 24K RAM and disk controller. (Ohio Scientific compatible).
- ★ 6502 machine code accessible through powerful 2K machine code monitor on board.
- ★ High quality thru plated P.C.B. with all I.C.'s mounted on sockets.

A tape of 10 programs on cassette — educational games, etc. will be supplied free of charge with each kit.

Simple Soldering due to clear and consise instructions compiled by Dr. A.A. Berk, BSc.PhD

NO EXTRAS NEEDED JUST HIT 'RETURN' AND GO.

Build, understand, and program your own computer for only a small outlay.

ONLY **£219** + VAT
Including RF Modulator & Power supply.
Absolutely no extras.

Due to the new prices of TTL this price will be increased shortly. So order now to beat the price increases and the rush.

SPECIAL CHARACTERS

@ Erases line being typed, then provides carriage return, line feed.
◄ Erases last character typed.
CR Carriage Return — must be at the end of each line.
: Separates statements on a line.
CONTROL/C Execution or printing of a list is interrupted at the end of a line. "BREAK IN LINE XXXX" is printed, indicating line number of next statement to be executed or printed.
CONTROL/O No outputs occur until return made to command mode. If an Input statement is encountered, either another CONTROL/O is typed, or an error occurs.
? Equivalent to PRINT

THE EXIDY SORCERER.

SAVE

CPM NOW

SORCERER COMPUTER SYSTEM

The Sorcerer Computer is a completely assembled and tested computer system. Standard configuration includes 63-key typewriter-style keyboard and 16-key numeric pad, Z80 processor, dual cassette I/O with remote computer control at 300 and 1200 baud data rates, RS232 serial I/O for communications, parallel port for direct Centronics printer attachment, 4K ROM operating system, 8K ROM Microsoft BASIC in Rom Pac™ cartridge, composite video of 64 char/line 30 line/screen, 128 upper/lower case ASCII set and 128 user-defined graphic symbols, operation manual, BASIC programming manual and cassette/video cables, connection for S-100 bus expansion.

LOOK!

- *32K RAM on board
- *RS232 interface
- *8K BASIC ROM
- *CUTS interface
- *4K MONITOR
- *KANSAS CITY interface
- *S100 BUS
- *User defined graphic symbols
- *Z80 cpu

16K ~~£860~~ £725* 32K ~~£950~~ £790* Credit facilities available. *+ VAT

THE ATARI video computer system

Atari's Video Computer System now offers more than 1300 different game variations and options in twenty great Game Program™ cartridges!

Have fun while you sharpen your mental and physical coordination. You can play rousing, challenging, sophisticated video games, the games that made Atari famous.

You'll have thrill after thrill, whether you're in the thick of a dogfight, screeching around a racetrack, or dodging asteroids in an alien galaxy. With crisp bright color (on color TV) and incredible, true-to-life sound effects. With special circuits to protect your TV.

Cartridges now available
Basic Maths, Airsea Battle, Black Jack, Breakout, Surround, Spacewar, Video Olympics, Outlaw, Basketball.

Years and years of fun and satisfaction are assured

SAVE £30

~~£169~~ **£138.85**

All at £13.90 each.

IBM GOLF BALL SELECTIVE PRINTER

Refurbished to new specifications. RS232C serial interface. Comes complete on stand — floor mounting. Small and compact. Interfaces to Exidy Sorcerer, TRS80 Apple and ITT 2020, Pet, Compukit and Nascom.
Complete your word processing system with a top quality printer for only £990.50.

All prices exclude VAT. Our VAT rate is 8%. We will pay any extra. Please quote this number P W1000 when ordering

SEND FOR OUR SPRING '79 CATALOGUE - 0·60p

COMP COMPUTER COMPONENTS

14 STATION ROAD, NEW BARNET, HERTFORDSHIRE TEL: 01-441 2922 (Sales)
CLOSE TO NEW BARNET BR STATION — MOORGATE LINE 01-449 6596
OPEN — 10am to 7pm — Monday to Saturday TELEX: 298755

VISIT OUR NEW SHOP AT 1 WALLCOT BUILDINGS, LONDON ROAD, BATH, AVON. All Products Ex-Stock Please check availability (Part of the Compshop Ltd. Group).

The World's Most Powerful 8~bit Microcomputer

Featuring the world's most powerful MPU — the Motorola MC-6809

Two types of central processor are available for use in your system. The standard /09 has a maximum random access memory (RAM) capacity of fifty six thousand (56K) bytes. It can have as many as eight input/output (I/O) devices such as terminals, printers, etc, attached. This capacity is adequate for business systems requiring up to four terminals and two printers. If the application involves moving large amounts of data, or scientific and engineering calculations, our larger CPU should be used.

The S/09 CPU has a maximum RAM memory capacity of 384K bytes. It is normally supplied with 128K bytes of memory which can be expanded to 256K, or 384K by adding additional memory arrays. This CPU will support up to 16 I/O devices.

Both CPU's are designed around the Motorola MC6809 microprocessor. This is the most powerful eight-bit microprocessor available.

The MC6809 has more addressing modes than any other 8-bit processor. It has powerful 16-bit instructions, and a highly efficient internal architecture with 16-bit data paths. It is easily the most powerful, most software efficient, and the fastest 8-bit general purpose microprocessor ever.

The greatest impact of the Motorola MC6809 undoubtedly will be software related. Ten powerful addressing modes with 24 indexing submodes, 16-bit instructions and the consistent instruction set stimulate the use of modern programming techniques, such as structured programming, position independent code, re-entrancy, recursion and multitasking.

C/09 CPU	£1050 + VAT
CS/09 128K	£2100 + VAT
/09 board (will directly replace existing SWTPC processor board)	£ 195 + VAT

CT-82 Terminal

- ★ Software function controls
- ★ 56-key "Cherry" keyboard
- ★ 12 key numeric or cursor control pad
- ★ 128 control functions
- ★ Graphics capability
- ★ User programmeable character sets
- ★ Software selectable Baud rates (50-38, 400)

£550 + VAT

Write or telephone for latest brochure including 16MB disc and new printer range.

Southwest Technical Products Co.

Computer Workshop

38 DOVER STREET · LONDON · WIX 3RB · Telephone: 01-491 7507 · Telex: 268913