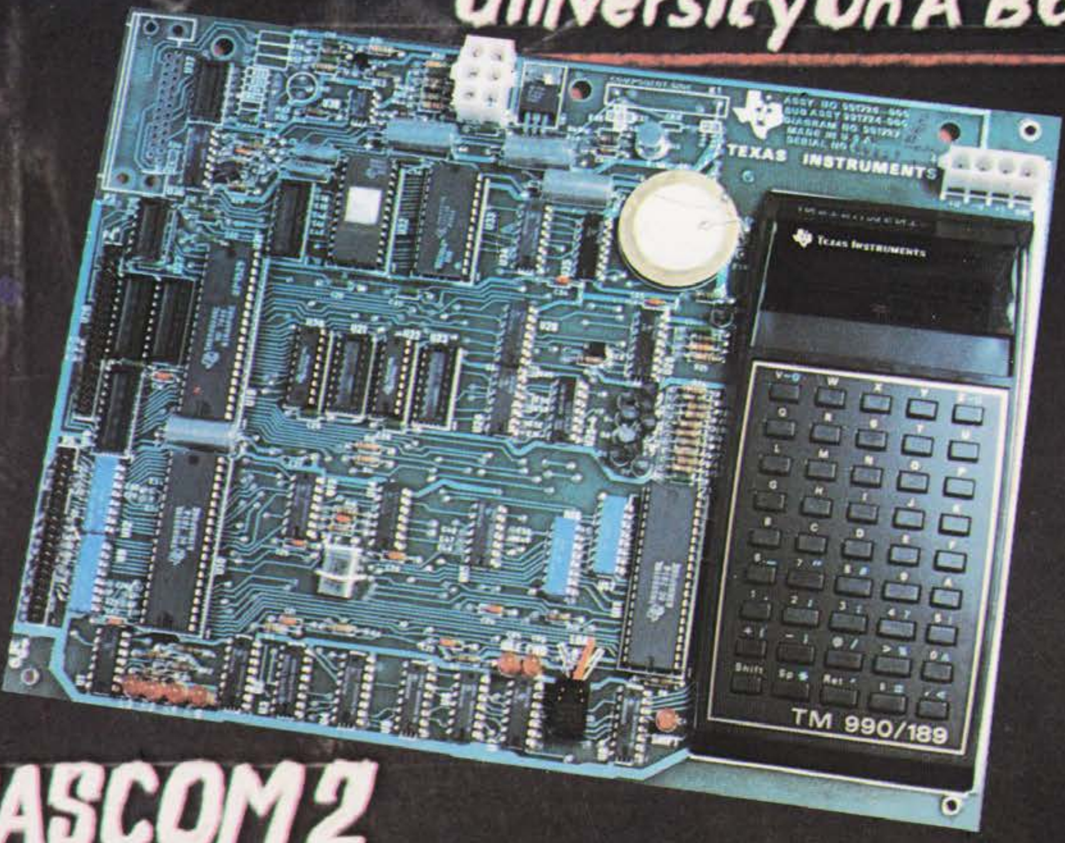
computing today

ISSN 0142-7210
FEB 1980
50p

TEXAS TM990/189
University On A Board?



NASCOM 2 REVIEWED

MICROLINK

TRS80 HINTS

LOGIC EMULATOR

Cabinet size 19.0" × 15.7" × 3.3". Television by courtesy of Rumblelows Ltd., price £58.62

8K ON BOARD MEMORY!
5K RAM, 3K ROM or 4K RAM, 4K ROM (link selectable). Kit supplied with 3K RAM, 3K ROM. System expandable for up to 32K memory.

2 KEYBOARDS!
56 Key alphanumeric keyboard for entering high level language plus 16 key Hex pad for easy entry of machine code.

GRAPHICS!
64 character graphics option — includes transistor symbols! Only £18.20 extra!

MEMORY MAPPED
high resolution VDU circuitry using discrete TTL for extra flexibility. Has its own 2K memory to give 32 lines for 64 characters.

KANSAS CITY
low error rate tape interface.

NEW LOW PRICE!

2 MICROPROCESSORS
Z80 the powerful CPU with 158 instruction, including all 78 of the 8080, controls the MM57109 number cruncher. Functions include +, —, *, /, squares, roots, logs, exponentials, trig functions, inverses etc. Range 10^{-99} to 9×19^{99} to 8 figures plus 2 exponent digits.

EFFICIENT OPERATION
Why waste valuable memory on sub routines for numeric processing? The number cruncher handles everything internally!

RESIDENT BASIC
with extended mathematical capability. Only 2K memory used but more powerful than most 8K Basics!

1K MONITOR
resident in EPROM.

SINGLE BOARD DESIGN
Even keyboards and power supply circuitry on the superb quality double sided plated through-hole PCB.

COMPLETE KIT NOW ONLY £249 + VAT

Kit also available as separate packs: e.g. PCB, Keyboards, Cabinet, etc.

POWERTRAN

PSI Comp 80.Z80 Based powerful scientific computer
Design as published in Wireless World April — September 1979

The kit for this outstandingly practical design by John Adams being published in a series of articles in Wireless World really is complete!

Included in the PSI COMP 80 scientific computer kit is a professionally finished cabinet, fibre-glass double sided, plated-through-hole printed circuit board, 2 keyboards PCB mounted for ease of construction, IC sockets, high reliability metal oxide resistors, power supply using custom designed toroidal transformer, 2K Basic and 1K monitor in EPROMS and, of course, wire, nuts, bolts, etc.

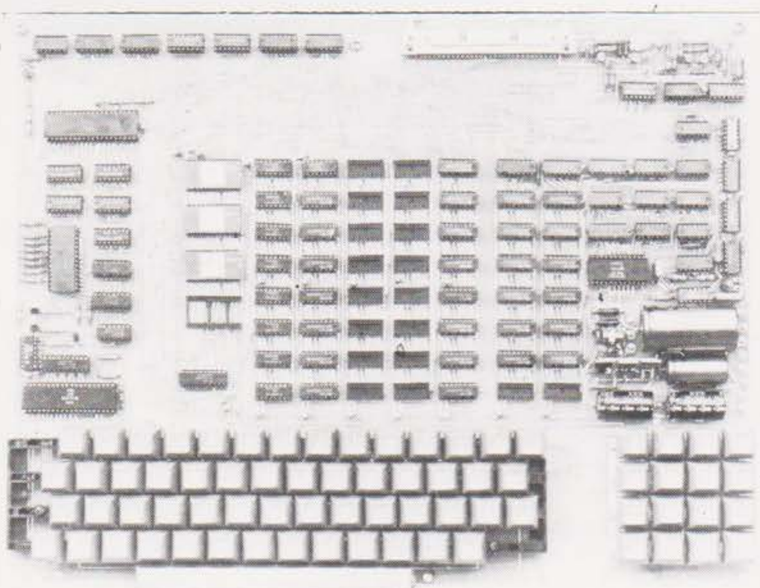
PSI COMP 80 Memory Expansion System

Expansion up to 32K all inside the computer's own cabinet!

By carefully thought out engineering a mother board with buffers and its own power supply (powered by the computers transformer) enables up to 3 8K RAM or 8K ROM boards to be fitted neatly inside the computer cabinet. Connections to the mother board from the main board expansion socket is made via a ribbon cable.

Mother Board	Fibre glass double sided plated through hole P.C.B. 8.7" x 3.0" set of all components including all brackets, fixing parts and ribbon cable with socket to connect to expansion plug	£39.90
8K Static RAM Board	Fibre glass double sided plated through hole P.C.B. 5.6" x 4.8"	£12.50
	Set of components including IC sockets, plug and socket but excluding RAMs.	£11.20
	2114L RAM (16 required)	£5.00
	Complete set of board, components, 16 RAMS	£89.50
8K ROM Board	Fibre glass double sided plated through hole P.C.B. 5.6" x 4.8"	£12.40
	Set of components including IC sockets, plug and socket but excluding ROMs	£10.70
	2708 ROM (8 required)	£8.00
	Complete set of board, components, 8 ROMs	£78.50

Floppy Disk, PROM programmer and printer interface coming shortly!



PCB size 16.0"x12.5"

Value Added Tax not included in prices

PRICE STABILITY: Order with confidence. Irrespective of any price changes we will honour all prices in this advertisement until April 30th, 1980. If this month's advertisement is mentioned with your order. Errors and VAT rate changes excluded.

EXPORT ORDERS: No VAT. Postage charged at actual cost plus 50p handling and documentation.

U.K. ORDERS: Subsequent to 15% surcharge for VAT. NO charge is made for carriage. Or current rate if changed.

SECURICOR DELIVER: For this optional service (U.K. mainland only) add £2.50 (VAT inclusive) per kit.

UK Carriage FREE

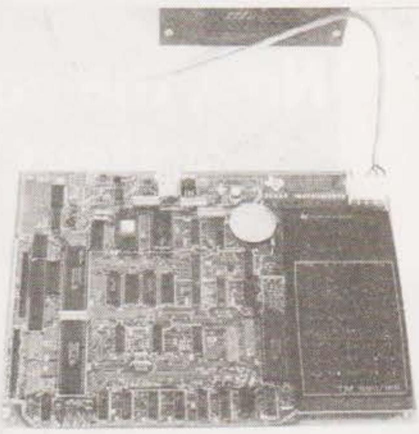
POWERTRAN COMPUTERS
(a division of POWERTRAN ELECTRONICS)

PORTWAY INDUSTRIAL ESTATE
ANDOVER HANTS SP10 3MN

ANDOVER
(0264) 64455

computing today

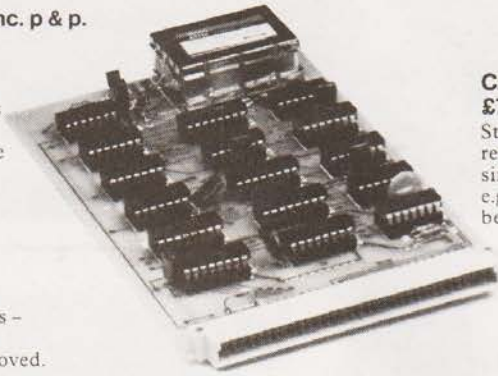
VOL.1, NO 12
FEB 1980



Distributed by Argus Distribution Ltd. Printed by LSG. Limited, Lincoln.

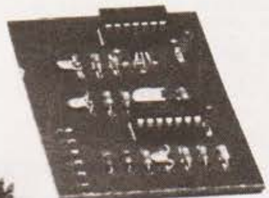
EDITORIAL AND ADVERTISEMENT OFFICE
145 Charing Cross Road, London WC2H 0EE. Telephone 01-437 1002/3/4/5

Now, the complete MK 14 micro-computer system from Science of Cambridge



VDU MODULE. £33.75
(£26.85 without character generator) inc. p & p.

Display up to ½K memory (32 lines x 16 chars, with character generator; or 4096 spot positions in graphics mode) on UHF domestic TV. Eurocard-sized module includes UHF modulator, runs on single 5 V supply. Complete ascii upper-case character set can be mixed with graphics.

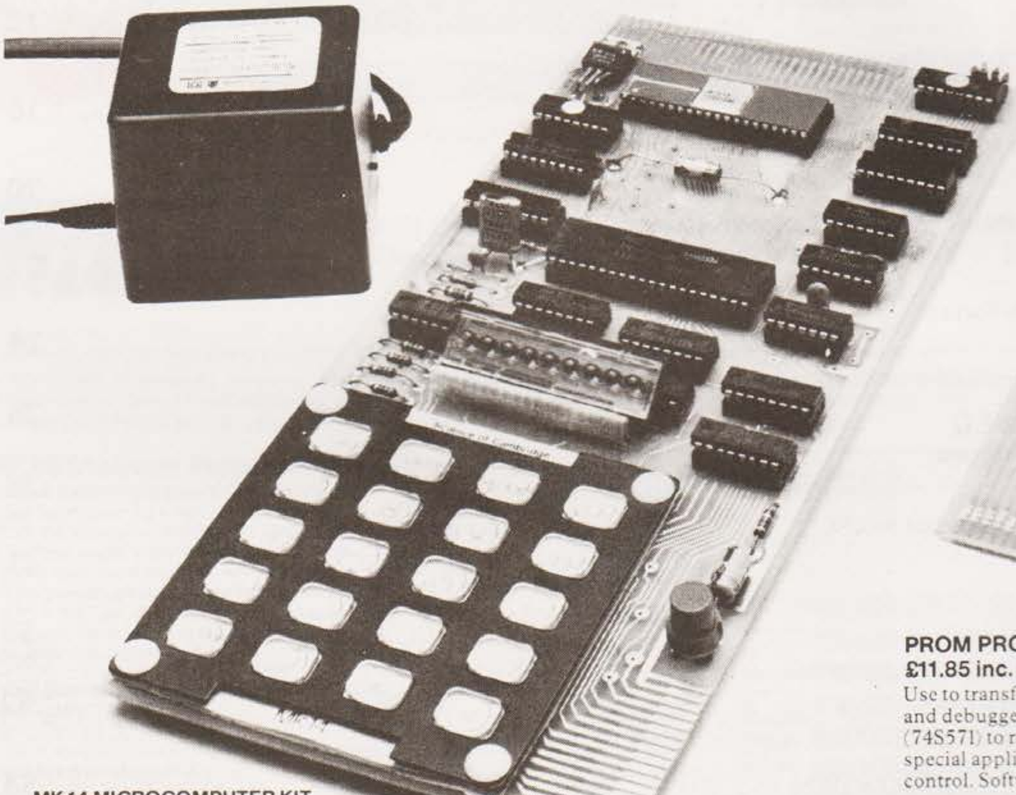


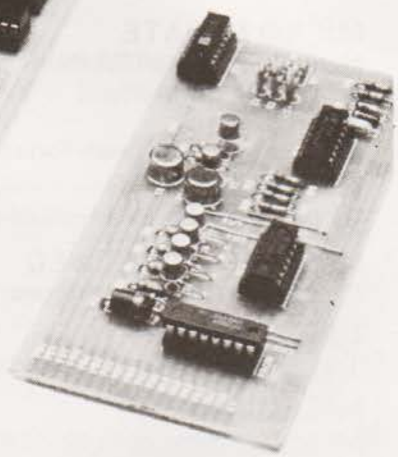
CASSETTE INTERFACE MODULE.
£7.25, inc. p & p.

Store and retrieve programs on any cassette recorder. Use for serial transmission down single line at up to 110 baud (teletype speed), e.g. over telephone line, and to communicate between two or more MK 14s.

POWER SUPPLY. £6.10 inc. p & p.

Delivers 8 V at 600 mA from 220/240 V mains – sufficient to drive all modules shown here simultaneously. Sealed plastic case, BS-approved.





PROM PROGRAMMER.
£11.85 inc. p & p.

Use to transfer your own program developed and debugged on the MK 14 RAM to PROM (74S571) to replace SC10S monitor for special applications, e.g. model railway control. Software allows editing and verifying.

MK 14 MICROCOMPUTER KIT
£46.55 inc. p & p.

Widely-reviewed microcomputer kit with hexadecimal keyboard, display, 8 x 512-byte PROM, 256-byte RAM, and optional 16-lines I/O plus further 128 bytes of RAM.

Supplied with free manual to cover operations of all types – from games to basic maths to electronics design. Manual contains programs plus instructions for creating valuable personal programs. Also a superb education and training aid – an ideal introduction to computer technology.

Designed for fast, easy assembly; supplied with step-by-step instructions.

Science of Cambridge Ltd

6 Kings Parade, Cambridge, CAMBS., CB2 1SN.
Tel: 0223 311488.

To order, complete coupon and post to Science of Cambridge
Return as received within 14 days for full money refund if not completely satisfied.

To: Science of Cambridge Ltd, 6 Kings Parade, Cambridge, Cambs., CB2 1SN.

Please send me:

- ☐ MK 14 standard kit @ £46.55.
- ☐ Extra RAM @ £4.14 per pair.
- ☐ RAM I/O device @ £8.97.
- ☐ VDU module including character generator @ £33.75.
- ☐ Cassette interface module @ £7.25.
- ☐ PROM programmer @ £11.85.
- ☐ Power supply @ £6.10.
- ☐ Full technical details of the MK 14 System, with order form.

All prices include p+p and VAT.

I enclose cheque/MO/PO for £ ______ (total).

Name ______

Address (please print) ______

CT/2/80

A CASE FOR KEYS

A new range of keyboard cases has been announced by Vero. They are available in a variety of sizes to hold numeric pads or full ASCII keyboards and they are easily dismantled for servicing. Special versions will also be available to order. For more details contact Vero at Industrial Estate, Chandlers Ford, Eastleigh, Hampshire SO5 3ZR.

3 LINE CAT

Not literally I'm glad to say, this one is from 3 Line Computing of 36 Clough Road, Hull HU5 1QL. It contains details of all their software for the TRS-80 such as DOS 3.0, FORTRAN, Pascal and many others. Software prices range from £5.95 up to £276 and the specimen documentation certainly looks good. They also do Verbatim disks at £26.45 for 10, storage boxes at £2.19 and a Z80 full colour poster for £3.45.

BOOKED Z80

A useful volume of Nascom and general Z80 Routines has been published by Sigma Technical Press at £7.50, the programs are all available on a cassette for £10. Useful inclusions are listings of all Nascom's monitor routines so any Z80 based system can be used. Programs included, there are over 30, are routines for music generation, numeric handling, screen displays, I/O routines and many others. All the programs are documented with line by line commenting and it would be a worthwhile addition to your library. Either order direct from Sigma at FREEPOST, 23 Dippons Mill Close, Tettenhall Wood, Wolverhampton WV6 7BR or try your local store.

GLITCH STOPPER

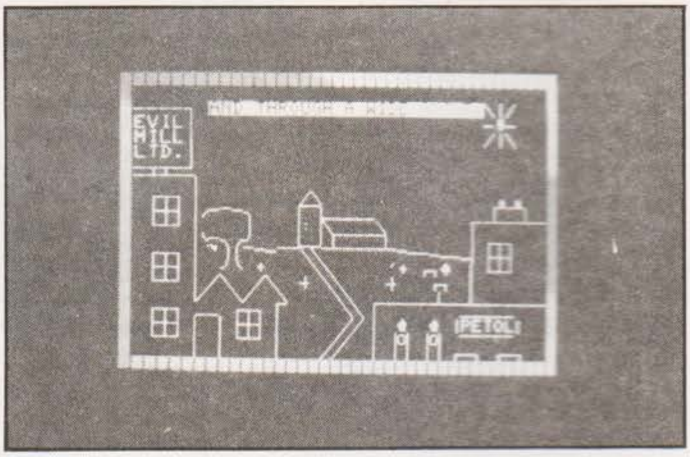
If you want to stop your micro going down when your fridge switches on the L.E.A. Kleanpower may be the thing for you. Two models, MB5 and MB10 are available which simply plug in between your equipment and the power socket. The unit is designed to remove all surges whether of high or low energy and the resultant is then filtered before being fed to your equipment. In the event of catastrophic occurrence the unit will fail safe. For more details contact Lightning Elimination Associates at Vine Cottage, Moreton, Thame, Oxon.

DATA PILE

Data books are the flavour of the month. RCA have released a 440 page book on COS MOS memories etc etc designated SSD–260 and it includes details of the 1802 micro and support chips. Well recommended this one, my copy is well thumbed already. Details from RCA at Sunbury on Thames, Middlesex. A slightly slimmer book from Intel, available free, is called Intelligence and covers details of their popular micros and memories. Get yours from Intel at 4 Between Towns Road, Cowley, Oxford OX4 3NB. Rapid Recall, famous for their bumper bundles, have brought out a new catalogue and price list. Covering everything from chips to systems via peripherals and including details of their PROM programming service it's well worth a look. Details from Rapid Recall at 6 Soho Mills, Woodburn Industrial Park, Wooburn Green, Bucks.

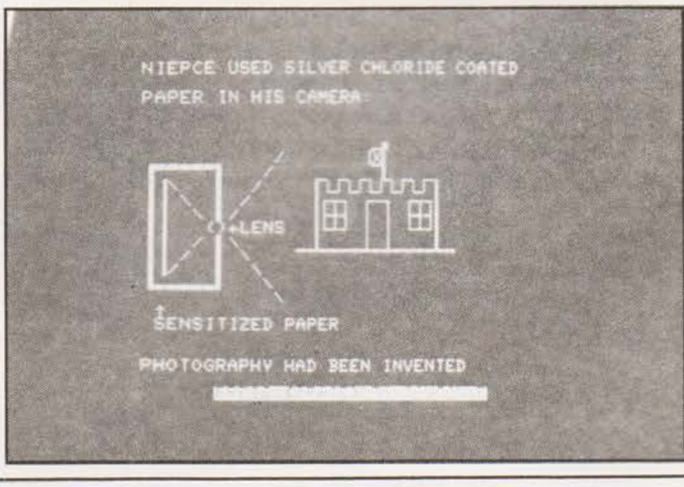
COLOUR 4 S100

Hi-tech Electronics have produced a full colour VDU board which is compatible with IEEE S100 system computers. Without the need for special monitors the plug-in board outputs a range of grey-scales and colours for both alphanumerics and graphics, both stand-alone or compatible with Prestel and Teletext. Features such as separate background and foreground colours, flashing, and double-height are standard, whilst optional sync inputs allow PAL video caption generation. The one-off price of £295 includes a software driver giving both full cursor control and page and scroll mode which can be booted from disc-based systems. Hitech Electronics are at 1 Richmond Gardens, Highfield, Southampton.



INSTANT PHOTOGRAPHY

Whether you require a refresher course for your Instamatic or a detailed set of instructions for your SLR you may find that the Petsoft Photography course will help. Written in eight parts it uses PET's graphics to demonstrate the workings of various camera systems and tests you on what you've learnt. Each part takes 7K and the whole course costs £12 + VAT. Details from Petsoft at 66–68 Hagley Road, Edgbaston, Birmingham B16 8PF.



TAKING THE COURSE

Several micro courses will be run in 1980 and here are the details that we have to date. The London Chamber of Commerce and Industry are running a couple, the first is an Introduction to Computers and their applications which will take place on 13 Feb, 7 May and 16 July between 9.30 and 5.00. The cost is £60 + VAT and the course reference is POL(1). The second is a two day course on Microcomputer Programming running on 12/13 March and 11/12 June between 9.30 and 5.00. The cost is £110 + VAT and the course reference is POL(2). Information on both can be obtained from Miss C.A. Measures at 69 Cannon Street, London EC4N 5AB, or ring 01-248 4444.

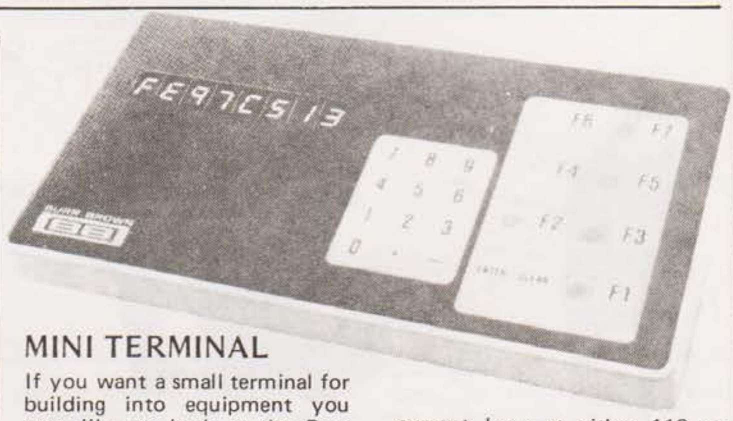
Parwest are running 2 day courses on 23/24 Feb and 24/25 March on microcomputers. These assume you know nothing and spend the first day introducing you to microprocessors and the second day concentrates on BASIC. Cost is £65 including refreshments and details are available from Parwest at Cotstone Bungalow, Brinkworth, Wiltshire or ring 066-641-537.

The Reading branch of the BCS are running Spring Schools on micro's from Feb 19 to March 25 at 8pm in Reading University on Tuesday evenings. Contact Mrs A.E. Haworth at 33 Alexandra Rd., Reading for details, the cost of the course is £25 to non BCS members.

Finally Cambridge Micro Computers are running five day courses which are heavily biassed towards practical implementation of micro based systems. The cost is £240 + VAT and details can be obtained from CMC at Cambridge Science Park, Milton Road, Cambridge CB4 4BN or ring 0223-314666.

FLOPPY DISCO

No we didn't leave our 8" model on top of a fan heater, this is a new filing system for your floppy disks. The box holds up to 20 in a fan file format allowing easy access. For mini disks the cost is £12.34, for 8" versions it goes up to £16.10. For Apple users you can now have a synthesiser card for a mere £215. Capable of producing 3 voices simultaneously, you can have up to 3 cards, it offers direct music entry from the screen, pitch envelope and volume control and eight octaves of range. The unit is crystal controlled and you can store tunes on tape or disk. For details on both these products contact Microsense Computers Ltd at Finway Road, Hemel Hempstead, Herts HP2 7PS or ring 0442-41191.



MINI TERMINAL

If you want a small terminal for building into equipment you may like to look at the Burr Brown TM25. It consists of an eight digit hex display, a numeric or hex keypad, nine function keys and indicators. Connection is via an RS232 serial or 20 mA current loop at either 110 or 300 Baud. Cost is £176 for one off and more details can be obtained from Burr Brown at Cassiobury House, 11–19 Station Road, Watford, Herts.

PROM BURNER

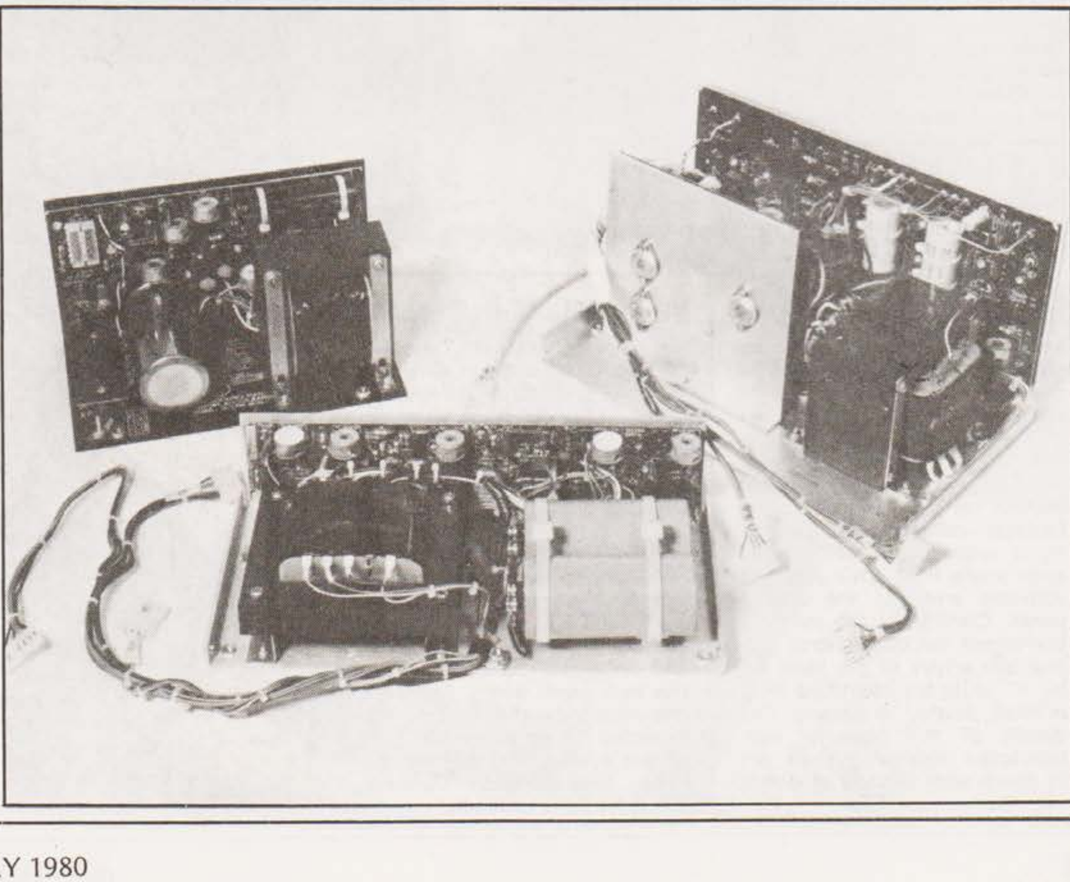
Fancy a cooked PROM for tea? With the new UV eraser from Microdata you can have it quicker than before. Capable of cooking up to 14 at once it can erase a 2708 in about seven minutes. Timing is handled by an internal clock and it bleeps when it's done. Cost is £97 + VAT and details can be had from Microdata Computers Ltd, Belvedere Works, Bilton Way, Pump Lane Industrial Estate, Hayes, Middlesex.

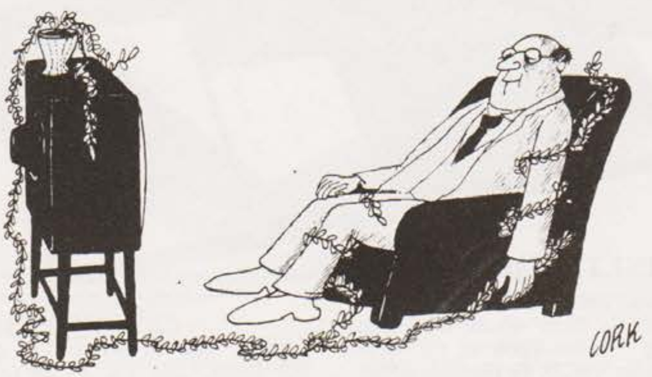
SOFT ON OHIO

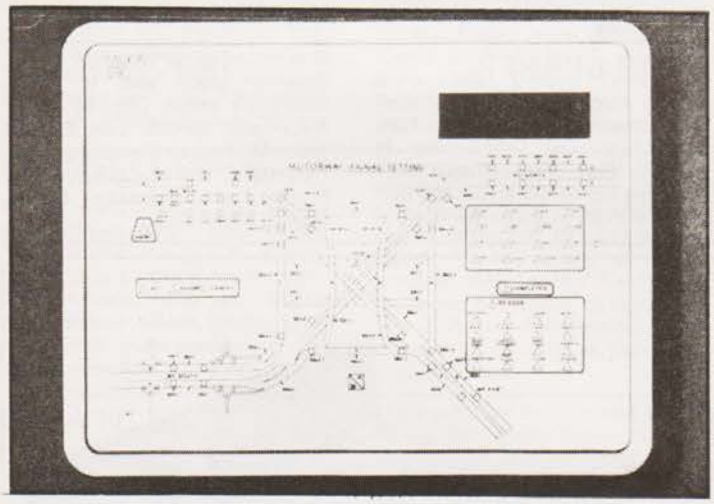
Mutek of Quarry Hill, Box, Wiltshire have produced a software catalogue for Ohio Scientific's range of machines. All the software is original and is fully documented. Programs range from Utilities such as Renumber, Search and Auto Loader through Games which include Chess, Starfighter and Battlefleet to Data sheets on interfaces, joysticks and others. The full catalogue costs £1 and includes a listing for the LIFE game.

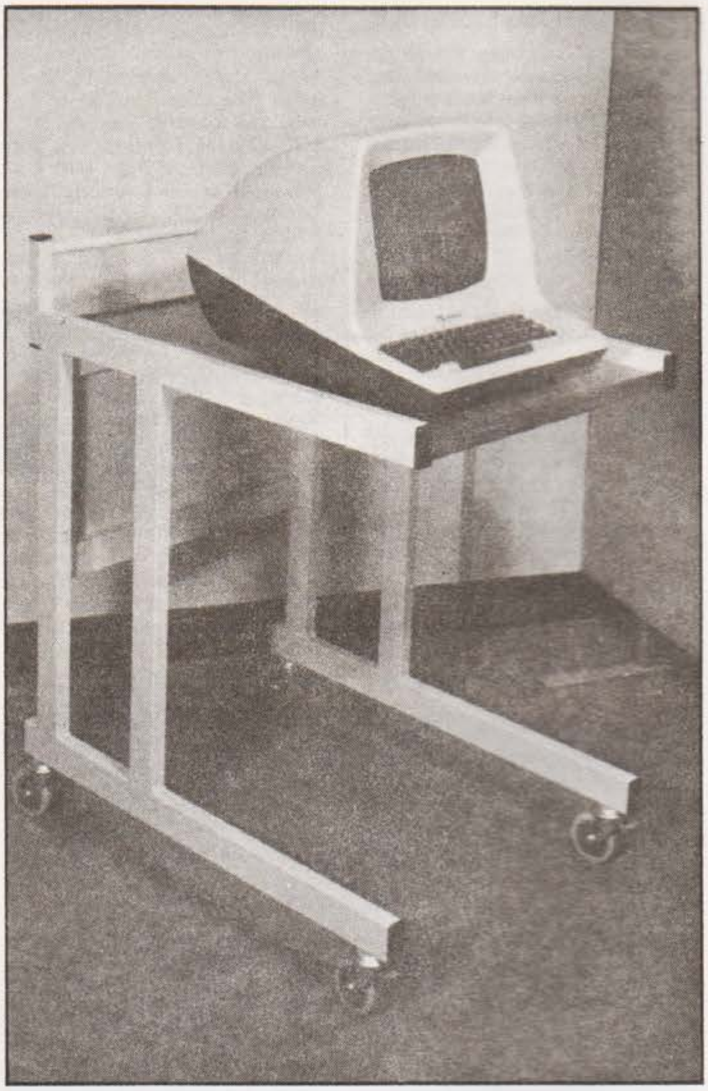
MICRO POWER

HAL Computers are now stocking a range of quad output power units suitable for Intel, National and Motorola based systems. Each gives ±5 and ±12 volts with a choice of current capacities, all outputs have overvoltage protection and can maintain power for up to 7.5 mS after "brown-out". Prices start from £285 which carries an 18 month warranty. Details from HAL at 133 Woodham Lane, New Haw, Weybridge, Surrey or ring Byfleet 45421.









EYES ON WHEELS

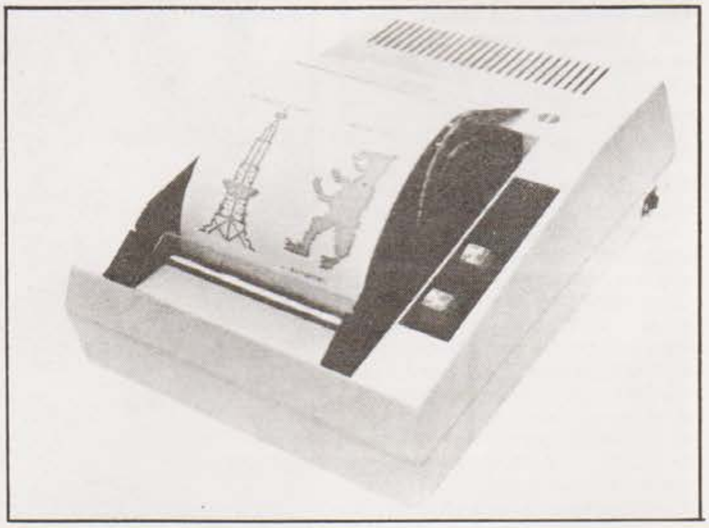
If your VDU has the roaming urge then give it a trolley, or that's what Data Efficiency say. Designed to take a wide range of terminals in sumptuous comfort it will slot over your desk when needed or can simply roam the confines of your room until it is needed. Finished in Pearl Grey(I thought that was a kind of tea) and Teak laminate it is complete with brakeable 3" wheels at £108.24. For details of this and all their other office and computer room furniture write to them at Maxted Road, Maylands Avenue, Hemel Hempstead, Herts HP2 7LE and ask for your free catalogue.

LED DOWN THE M4

Midos, the display system from Grundy and Partners, has found a home on the motorways of olde England. The Department of Transport has chosen the system for a trial at the Almondsbury Control Centre for signal control on the M4 and M5. It replaces conventional teletype input with a quicker and less error prone fibre optic pen that activates areas of the display panel. Control of the panel is performed by dual micros and multiple arrays of the basic 8" by 4" units are identified by a printed overlay as shown. For details of this powerful new interactive display system get in touch with Grundy at Bonds Mill, Stonehouse, Glos.

PRESTEL PRINTER

Newly announced by Dataplus of 39-49 Roman Road, Cheltenham, GL51 8QQ is a Viewdata printer.Using the NMP 40 mechanism it will be sold in cased or OEM forms by Olympia International.The mechanism,supplied by Dataplus, uses metallised paper and is capable of full alphanumerics and graphics reproduction, a full page can be printed in about 3 seconds. The paper feed is of the friction type and the printhead is made up of 240 electrodes spaced across the five inch paper width. Long life and simplicity of operation are expected to be major benefits of this system over the moving head type.Contact Dataplus direct for further details.



STAR TREK, THE FILM

By now the film of our program should be on general release, or rather Paramount's multi mega-buck production of the long running TV series. It seems incredible that the first one was made over ten years ago but in true Mc Arthur fashion they have returned. Aged they may be but these heroes of the small screen are well and living in the 23rd century. As we find our friends Admiral Kirk is taking a drop in rank to get his hands on the refitted Enterprise-much to new recruit Decker's annoyance-Bones has grown a beard, Spock is undergoing re-Vulcanisation on his home planet and Scotty has been practising his accent.
Most of our regular acquaintances, Mr Sulu, Uhara, Checkov, Chapel and Rand are also there in the new improved Enterprise along with the second new recruit Ilia, a bald female navigation officer from Delta.The nameless or to be more exact mis-named threat from outer space that is being problematical to all and sundry zaps a couple of innofensive Klingons and has a few goes at the Enterprise is only trying to do what it has been told.
In true Startrek format the story is just a little too weak and there is just a little too much moralising, more action and less words would have been better in my view, but in general the special effects make up for this. I say in general because there are one or two occasions when I wondered how much of the budget went on cardboard cut-outs, still the American effects people were never really up to our standards. It's nice tc see Alan Dean Foster's hand in the script after his work on Alien and I was a bit suprised to find that he didn't make an appearance. On the whole it is an entertaining film but not up to the standard of Alien or Silent Running, perhaps they'll use British effects for the inevitable follow-up.

DOWN ON THE FARM

The ITT 2020 has been mooving into agriculture recently. One of the distributors of the system, Farmplan, have been given an award by Barclays Bank for their innovative herd monitoring software. Designed to give data on dairy herds or even the performance of a single cow the system has been implemented by twenty farmers. Milk some more details from ITT at Chester Hall Lane, Basildon, Essex.

CASED AIM

As we mentioned last month in our News Portable Microsystems specialise in casing single board computers such as the Nascom family and the AIM 65. Other enhancemants that they offer for the AIM 65 include a range of Motherboard-expanders.
These include an AIM to S100 unit, an AIM bus extension that gives access to the Rockwell System 65 and the Motorola Exorciser range of boards and an AIM to KIM expansion unit. As well as stocking these they can also supply a wide range of boards to plug in. Contact them at 18 Market Place, Brackley, Northants NN13 5SF or ring on 0280-702017.

GENUINE EX-PENTAGON ICBM TARGETS, MATE!

ANY PROGRESS ON THE BINARY-TO-CHINESE DISPLAY CONNVERTOR YET ARNOLD?

AH WELL – AS THEY SAY IN THE TRADE – GARBAGE IN : GARBAGE OUT!

Trevor Lusty

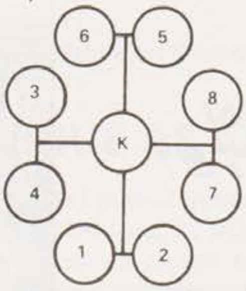
Sobered up from Christmas Knight? Wait no longer, we have the solution!

The program shown in Fig. 2 can find all possible (providing you can wait that long!) Knight's Tours of a chess board. The knight starts at a corner square but the program can easily be modified to start at any desired square.

Method For Solution

The program uses a modified tree search technique. There are eight possible jumps that a knight may make, and these may be arranged in any cyclic order. The starting position within the cycle may also be different for different squares of the board. It is therefore possible to search for Knight's Tours which fulfil, as closely as possible, any given pattern.

eg. ——— The search for a tour in which the knight circles the outside of the board as often as possible in an anti-clockwise direction would have the following pattern of jumps :—



starting anywhere in this cycle :—

The starting position for each square being :—

1	1	3	3	3	3	3	3
1	1	3	3	3	3	3	3
1	1	*	*	*	*	5	5
1	1	*	*	*	*	5	5
1	1	*	*	*	*	5	5
1	1	*	*	*	*	5	5
7	7	7	7	7	7	5	5
7	7	7	7	7	7	5	5

The numbers on the grid give the starting positions within the cycle for the first jump, those not shown in the centre being less important. If a jump is not possible, the next jump in the cycle is tried. If no jump is possible, the program back-tracks and tries a different position for an earlier move.

The flowchart (figure 1) and the REMark statements in the program listing help to further explain the basic algorithm.

Outputting The Solution

The output is in the form of an 8 x 8 matrix representing the layout of a chess board. The number on any given square being the n th. position of the knight. The program was written for a RM 380 Z using DBAS9 Ver. 3.0B and graphics are used to show the tree search in action.

Modifications For Other Machines

As the program uses POKE rather than PLOT, it may be adapted for other machines by changing the screen and line pointers S9 and S8. (eg. for the 'new' PET S9=32768–80 and S8=80, also remove lines 1160 and 1240). The PRINT statement in line 2780 should either be removed or directed to a printer. For machines without memory mapped VDUs remove line numbers 1160, 1240, 1260, 1280, 1800, 2180, 2200, 2220, 2380, 2400, 2420, 2440, 2460 and 2480. To see intermediate positions of the board change line 2580 to :— 2580 IF K<n THEN 1860

where n may be any number between 1 and 64 ; and add line 2930 :– 2930 IF K < 64 THEN 1860

To change the starting position of the knight change the numeric constants in lines 1720, 1740, 1760, 1780 and 1820. Remember that the board occupies 3 to 10 of array B as the outer elements are used as out of bounds detectors.

To change the search pattern the data must be changed. It is obviously possible to cheat and enter a search pattern which works first time. A better test of program efficiency is to time over the first, say, 10 Knight's Tours; this takes about 6 minutes with the given search pattern.

Glossary Of Stores Used

- B ––– 12 x 12 array to simulate the board.
- S ––– 12 x 12 array to hold the search pattern.
- X & Y 16 element arrays to hold possible knight jumps, the second 8 elements are used to facilitate efficient programming and may duplicate the first 8 elements.
- P & Q 64 element arrays to hold the position of the knight's n th. move.
- U & V 64 element arrays to hold the tree search position.
- N$ String variable for print routine.
- T1 & U1 hold tens and units digits of knight's move.
- S9 Screen pointer.
- S8 Line length pointer.
- S7 POKE address.

The other variables I,J . . . X2,Y2 . . . X3,Y3 . . . Z1,Z2 represent various co-ordinates for positions on the board.

Solve The Format Problem

The trouble with 'simple to learn' programming languages like BASIC, is that you cannot always get the output in the form you would like. One of the main differences between BASIC and other high level languages, such as FORTRAN, is the lack of a FORMAT statement. Some BASICs do have a PRINT USING statement, but these are usually extended BASICs and are only found on large machines.

Now here's the problem. Write a BASIC program, or better still a subroutine, to print Pounds and Pence in the way we normally write them.

example:– Two Pounds should be printed as £2.00 and not as £2.

Fig. 1

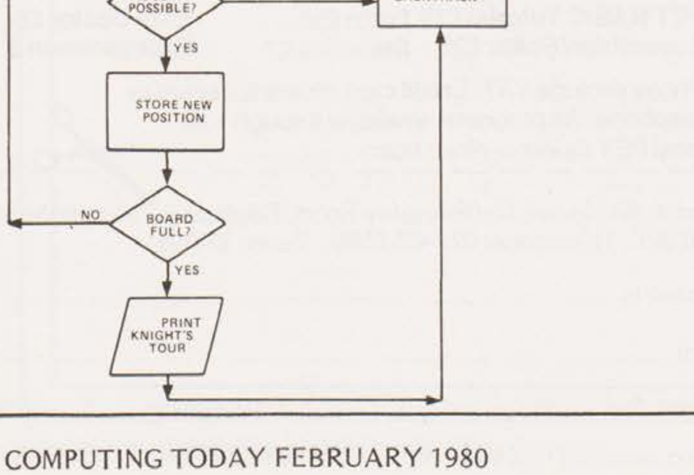


Fig. 2

```
1000 REM     *****************************************
1020 REM     *****                               *****
1040 REM     *****  KNIGHT'S TOUR BY TREVOR LUSTY  *****
1060 REM     *****                               *****
1080 REM     *****  PROGRAMMED IN BASIC 22/01/79   *****
1100 REM     *****                               *****
1120 REM     *****************************************
1140 REM
1160 CLEAR 120
1180 REM
1200 REM     *****  DIMENSION VARIABLES  *****
1220 DIM B(12,12),S(12,12),X(16),Y(16),P(64),Q(64),U(64),V(64)
1240 GRAPH 1
1260 LET  S9 = 61440-256
1280 LET  S8 = 128
1300 REM     *****  INITIALIZE N$ FOR PRINT ROUTINE  *****
1320 LET  N$ = "0123456789 "
1340 REM     *****  SET UP OUT OF BOUNDS FOR BOARD  *****
1360 FOR  I = 1 TO 12
1380 FOR  J = 1 TO 12
1400 LET  B(I, J) = -1
1420 NEXT J
1440 NEXT I
1460 REM     *****  SET UP BOARD AND READ SEARCH PATTERN  *****
1480 FOR  I = 3 TO 10
1500 FOR  J = 3 TO 10
1520 LET  B(I, J) = 0
1540 READ S(I, J)
1560 NEXT J
1580 NEXT I
1600 REM     *****  READ IN POSSIBLE KNIGHT JUMPS  *****
1620 FOR  I = 1 TO 16
1640 READ X(I), Y(I)
1660 NEXT I
1680 REM     *****  SET STARTING POSITION  *****
1700 LET  K = 1
1720 LET  X3 = 3
1740 LET  Y3 = 3
1760 LET  P(1) = X3
1780 LET  Q(1) = Y3
1800 POKE S9+S8*Q(1)+4*P(1)-8, 49
1820 LET  B(3, 3) = 1
1840 REM     *****  FIX START AND END OF JUMP CYCLE  *****
1860 LET  U(K) = S(Y3, X3)
1880 LET  V(K) = U(K)+7
1900 REM     *****  FIX KNIGHTS PRESENT POSITION  *****
1920 LET  X2 = P(K)
1940 LET  Y2 = Q(K)
1960 REM     *****  SEARCH FOR POSSIBLE JUMP POSITION  *****
1980 LET  S1 = U(K)
2000 LET  F1 = V(K)
2020 FOR  J = S1 TO F1
2040 LET  X3 = X2+X(J)
2060 LET  Y3 = Y2+Y(J)
2080 IF   B(Y3, X3) = 0   THEN 2340
2100 NEXT J
2120 REM     *****  JUST IN CASE WE EVER FINISH  *****
2140 IF   K = 1   THEN 3220
2160 REM     *****  WE ARE STUCK !!!  TRY NEW POSITION  *****
2180 LET  S7 = S9+S8*Q(K)+4*P(K)-8
2200 POKE S7, 32
2220 POKE S7-1, 32
2240 LET  B(Q(K), P(K)) = 0
2260 LET  K = K-1
2280 LET  U(K) = U(K)+1
2300 GOTO 1920
2320 REM     *****  MOVE KNIGHT TO NEW POSITION  *****
2340 LET  U(K) = J
2360 LET  K = K+1
2380 LET  T1 = INT(K/10)
2400 LET  U1 = K-10*T1
2420 LET  S7 = S9+S8*Y3+4*X3-8
2440 POKE S7, 48+U1
2460 IF   T1 = 0   THEN 2500
2480 POKE S7-1, 48+T1
2500 LET  B(Y3, X3) = K
2520 LET  P(K) = X3
2540 LET  Q(K) = Y3
2560 REM     *****  CKECK TO SEE IF BOARD IS FULL  *****
2580 IF   K<64   THEN 1860
2600 REM     *****  PRINT OUT KNIGHT'S TOUR  *****
2620 FOR  Z1 = 3 TO 10
2640 FOR  Z2 = 3 TO 10
2660 LET  T1 = INT(B(Z1, Z2)/10)
2680 LET  U1 = B(Z1, Z2)-10*T1
2700 LET  U1 = U1+1
2720 LET  T1 = T1+1
2740 IF   T1>1   THEN 2780
2760 LET  T1 = 11
2780 PRINT MID$(N$, T1, 1);MID$(N$, U1, 1);"  ";
2800 NEXT Z2
2820 PRINT
2840 PRINT
2860 NEXT Z1
2880 PRINT
2900 PRINT
2920 PRINT
2940 GOTO 2240
2960 REM     *****  DATA FOR SEARCH PATTERN  *****
2980 DATA 1, 1, 3, 3, 3, 3, 3, 3
3000 DATA 1, 1, 3, 3, 3, 3, 3, 3
3020 DATA 1, 1, 3, 3, 3, 5, 5, 5
3040 DATA 1, 1, 1, 1, 4, 4, 5, 5
3060 DATA 1, 1, 7, 7, 5, 5, 5, 4
3080 DATA 1, 1, 7, 7, 5, 5, 5, 5
3100 DATA 7, 7, 7, 7, 7, 7, 5, 5
3120 DATA 7, 7, 7, 7, 7, 7, 5, 5
3140 REM     *****  DATA FOR THE EIGHT JUMPS  *****
3160 DATA -1,2,1,2,-2,-1,-2,1,1,-2,-1,-2,2,1,2,-1
3180 DATA -1,2,1,2,-2,-1,-2,1,1,-2,-1,-2,2,1,2,-1
3200 REM     *****  IF WE EVER GET HERE IT WILL BE A MIRACLE !!
3220 END
```

computing today

WHAT TO LOOK FOR
IN THE MARCH ISSUE.
ON SALE FEB 8.

HAPPY BIRTHDAY TO US!

With our next issue CT is one year old.

So we are celebrating by giving YOU presents! As well as a superb issue of Computing Today—as usual—we are holding a fabulous competition to find out who wins the unbelieveable prizes we are giving away.

If you think we're being vague,you're absolutely right! After all all birthday presents come best as a suprise, even ones as excellent as these.

Don't miss out on our celebrations next month, it's all on us!

SPACE WAR

It is the year 5180 AD. Earth sits securely in the centre of an expanding Empire. The Silverberry FTL (Faster Than Light) Drive has taken the Terran Dreadnoughts up to 100 light years from their home planets, sweeping all before them. Up to now only two intelligent races have been discovered that have space travel capability and both have been unable to offer much resistance to the mile long battleships. Seemingly all is well with the Empire.

And then the fleet reaches the Trivax system, and is resoundingly defeated in a battle which sees the enemy using their own moons to smash the Earth forces. With the main fleet in ruins there is little to stop the invaders who sweep towards the Sol planets and Earth's last defences within her own Solar System......the battle of Terre is on........

CT offers two superb programs to simulate the result of this assault upon the outer fortresses, and the desperate planet defense against the Trivax ships that get through. Makes Captain Kirk look like the Fairy Queen!

PET ACCOUNTING

Developed so the domesticated PET could take care of the household finances (better than those little bendy cards). The concept behind it is to set up 17 long running budget accounts and inject each with a monthly allowance, the PET does the rest.

The BASIC program can be easily adapted to suit machines with different dialects as long as they have string handling capability.

M.J. Bell & M.J. Bick.

Assist your logical functions with this program. It's designed to help you design, how's that Mr Spock?

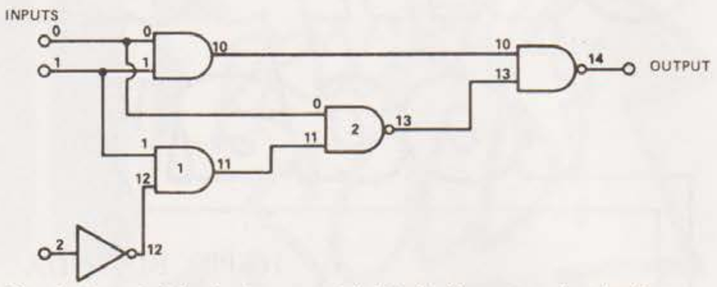


Fig. 1. A typical logic diagram with labelled inputs and outputs.

A logic emulator is a very useful tool to own if you are involved in either designing or analysing circuits that use large quantities of gates. It's primary function is to calculate TRUTH TABLES for combinations of gates, or give the output result for a network given any specific combination of inputs (See CT, OCT 79 – 'MPU's by EXPERIMENT' for further explanation on TRUTH TABLES). The emulator described here can analyse circuits comprising of AND, OR, NAND and NOR gates with additionally the inverter function where this is represented as a NOR or NAND gate with one input.

Using The Program

To use the emulator the circuit to be analysed must be labelled so that the input/output leads can be referred to as circuit point labels. The labels 00-09 have been reserved as inputs and 10-19 as outputs, all other points on the circuit can use 0A – 0F Hex and 1A – 40 Hex (See the typical circuit point labels).

The maximum circuit size that this emulator can handle is difficult to quantify, but the number of gates, plus the number of inputs, plus the number of outputs, must not exceed FF Hex (255 decimal). Should this occur then an overflow message will be printed and the circuit must be split in two and re-entered.

To enter the circuit the gate 'type', followed by the output label, followed by the input labels, must be typed in as shown in Fig.2. Any illegal entry will be ignored and a message will be screened. It is only necessary to re-enter the error line. Immediate errors can be corrected by a backspace, but once the display has been scolled retrospective changes are not possible. To re-enter the entire circuit type 'E' for EXIT and begin again. Once the circuit has been correctly entered type 'RUN' followed by 'New Line' and the result of the initial run with all inputs zero will be displayed with the 10 designated inputs on the left, and the 10 designated outputs on the right. (See Fig.3). To modify the inputs enter a '1' or '0' as appropriate until all 10 inputs have been modified then a re-run will automatically take place and the result displayed. Fig.3 shows the truth table for the circuit in Fig.1 (i.e. the output states for all possible input states). The output only occurs when input '0' is high and inputs '1' and '2' are low.

If, as is sometimes possible, a combination of inputs to a circuit gives an unstable situation, (as with a NAND gate that has its output coupled back to its spare input) then this condition is recognised and stated on the CRT.

To check if a circuit has been correctly entered type 'L' for LIST when modifying the input and the first gate will be displayed. The second and so on will be displayed by pressing the 'space bar' until all gates have been listed when a re-run is made for all inputs zero. To enter a new circuit type 'E' for EXIT.

Long Term Storage Of Circuits

After entering a lengthy circuit or embarking on a protracted development, it might be advisable to store the data on tape so that it can be reloaded at any time. This has been made possible by keeping all the circuit information in one block '0EA0 – 0FA0'. By storing this block, using the monitor commands of 'L' (for T2) or 'W' (for T4) the data can be re-loaded at any time. Under this arrangement there are two execution addresses that can be used. 0DC6 where it is required to list the circuit stored, or 0CC6 for the emulator to give an initial run.

Error detection is provided in the following ways

'Input Error Entry Ignored' – An incorrect gate description. Re-enter correctly.

'Circuit Overflow Re-enter it' – The total number of gates and inputs + output exceeds 255 (FFH) and the storage area has been exceeded.

'Circuit Unstable with this Input' – This is not really an error, but an indication that after 256 (100H) attempts at solving the circuit it will not stabilise.

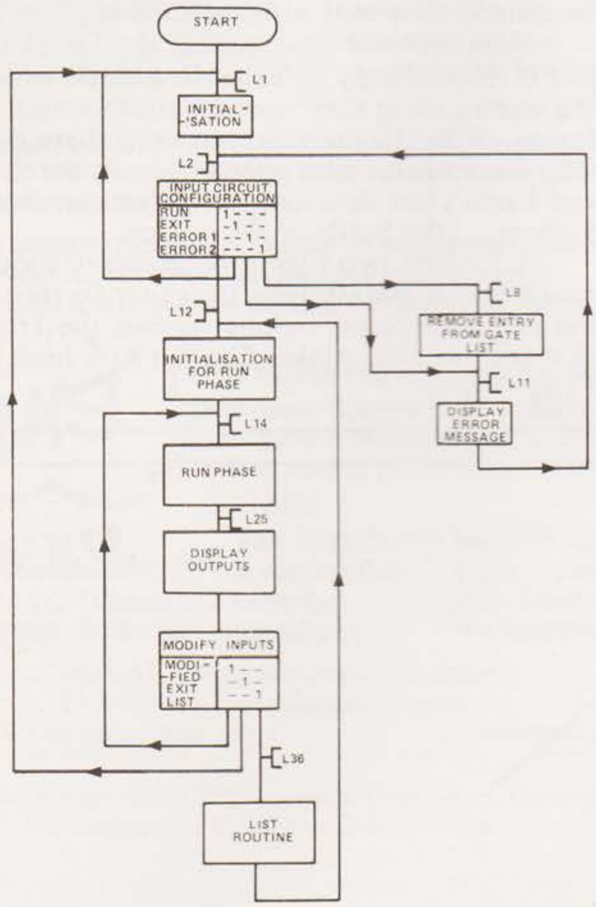


Fig. 4. The main program flowchart with the Input and Run phase routine flow charts.

```
AND    1 0   0  1
OR       1 1   1   1 2
NAND    1 2   2
NAND    1 3   0    1 1
NOR     1 4   1 0  1 3
```

Fig. 2. The format for the gate list data.

```
0 0 0 0 0 0 0 0 0 0 0        0 1 1 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0        0 0 0 1 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0 0 0 0        0 1 1 1 0 0 0 0 0 0 0
0 1 1 0 0 0 0 0 0 0 0        0 1 0 1 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0 0 0 0        0 1 1 0 1 0 0 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0        0 0 0 1 0 0 0 0 0 0 0
1 1 0 0 0 0 0 0 0 0 0        1 1 1 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0        1 1 0 0 0 0 0 0 0 0 0
                                   ▲
                                 OUTPUT
                                   14
```

Fig. 3. Truth table display for the previous circuit and data.

Program Description

The data for this program is held in two arrays, a Gate List and a State List. The gate list is used to hold the circuit topology and the state list stores the conditions existing at each point in the circuit.

The Gate List has a free format as follows:—

GATE TYPE	O/P	I/P	I/P	GATE TYPE	O/P	I/P	I/P	I/P	GATE TYPE	O/P	I/P	I/P	END
1st GATE				2nd GATE					3rd GATE				TERMINATOR

Each gate used a minimum of 3 bytes, a gate type, an output and at least one input. A further byte is used for each additional input. The gate list is terminated with an end statement. The form in which the above information is stored is as follows:—

GATE TYPES
AND = 4EH
OR = 52H
NAND = 41H
NOR = 4FH

INPUT & OUTPUT STATES — 64 States are permissible numbered in the range. 00H to 40H

TERMINATOR — 055H

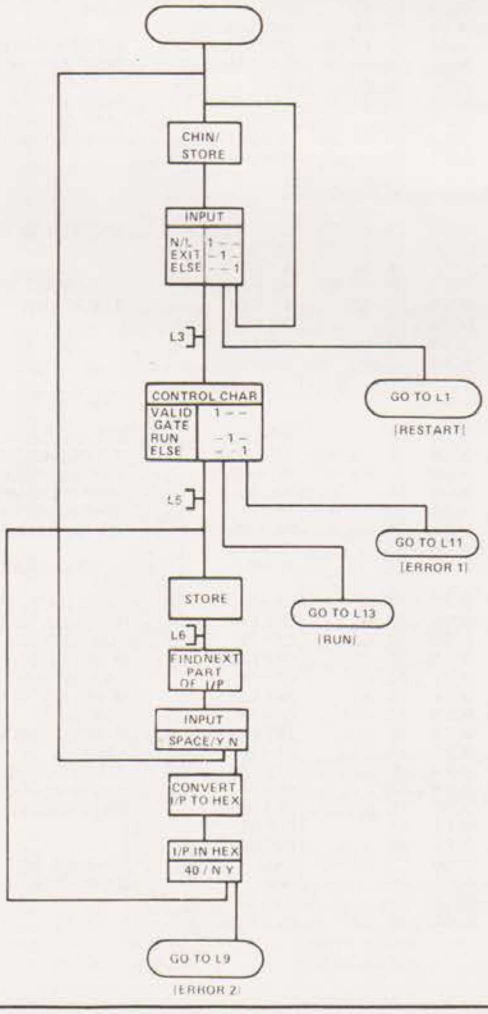
There are 40H locations in the State List, one for each permissible state. 30H is used to signify a '0' state and 31H a '1' state.

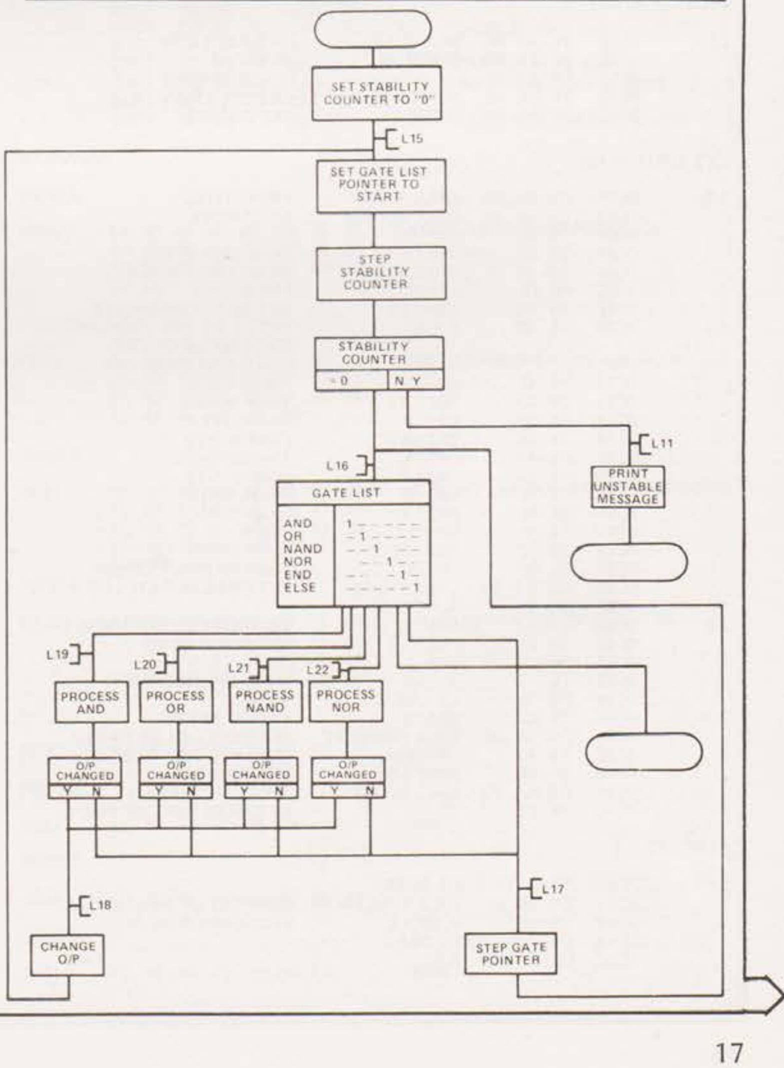
The Initialisation Phase is used to set the gate pointer byte (0EA0) to the beginning of the Gate list. The screen is cleared and the title inserted at the top of the screen.

Circuit Input Phase

The gate entry is received from the keyboard and entered on the screen. Any entry is initially accepted providing it does not contain an 'E' which is an EXIT command and used to jump to initialisation.

When 'New Line' is pressed the entry is scrolled and the second character is used to determine the gate type. If the character is valid it is stored in the gate list as the gate type, otherwise an error message is displayed. For the rest of





the entry spaces are regarded as de-limiters and a double space as an end of entry. A search is made for each space and then the next character is checked to see if that too is a space, if not the two characters following the space are decoded from ASCII to HEX and the range checked to see if it is within 00—40H. If it is the number is stored in the gate list and the next number searched for. On detecting a double space the input of the next line is commenced. Should the result of the ASCII to HEX conversion be outside the range 00—40H the gate list pointer is decremented thus searching back to the last gate type entry. This has the effect of deleting the entry, an error message is also displayed. On detecting a 'RUN' input control is passed to run phase initialisation.

The Run Phase

To initialise the program a terminator is placed at the end of the gate list and the state list is set to '0'. The Run Phase consists of a set of routines which for each gate take the input conditions and produce the output condition, this is then compared with the stored output condition. If they are found to be the same the next gate in the list is processed. If not the stored gate condition is updated and processing is re-started from the first gate. When all the gates have been processed in turn without any changes in state being found. the circuit is said to be stabilised and the program jumps to the Display outputs phase. A counter is maintained which is incremented each time the gate list is re-started. The counter starts a 0LH and on reaching 00H (after FFH) it is assumed that the circuit will not stabilise and an error message is displayed.

The Various Routines

The display is done by two block moves from the state list to the screen. States 00—09H (inputs) and 10—19H (outputs) are copied to the screen.

The input modification routine enables the 10 inputs (state 00—09H) to be over written from the keyboard. An 'E' causes a re-start from the beginning of the program, and an 'L' causes a jump to the List routine.

The list routine was included so that the circuit could be checked for accuracy if an unexpected result occurred. The gate list is scanned looking for one of the gate-type characters. When this has been detected the appropriate gate label is displayed and the following output and inputs are copied onto the display. Each gate can be inspected in turn, but *not* modified. When the list is complete the operation is passed to the initialisation routine and continued as before.

```
EXECUTE 0C50

INITIALISATION

L1 :    0C50  21 A0 0E   HL = 0EA0        )SET GATE LIST
        0C53  22 FC 0D   (0BC0). HL       )POINTER
        0C56  EF 1E 00                    CLEAR SCREEN
        0C59  21 D6 0B   HL = 0BD6        CRT LOCATION
        0C5C  22 18 0C   (0C18), HL       SET CURSOR

CCT INPUT PHASE

L2 :    0C5F  CD 00 0E   CALL MESS 1      PRINT TITLE
        0C62  CD 3E 00   CALL M/CHIN      GET ENTRY
        0C65  FE 45      CP = 'E'         ? EXIT
        0C67  28 E7      JRZ L1 :         DO AGAIN IF YES
        0C69  CD 3B 01   CALL M/CRT       PRINT CHARACTER
        0C6C  FE 1F      CP = N/L         ? NEW LINE
        0C6E  20 F2      JRNZ L2 :        GET NEXT CHARACTER
L3 :    0C70  2E 4B      L = 4B           SET HL TO 2ND LOCN ON
                                          2ND LINE UP OF CRT
        0C72  7E         A, (HL)          LOAD THIS CHAR IN A
        0C73  FE 4E      CP = 'N'         ? AND GATE
        0C75  28 12      JRZ L4 :         JUMP IF YES
        0C77  FE 52      CP = 'R'         ? OR GATE
        0C79  28 0E      JRZ L4 :         JUMP IF YES
        0C7B  FE 41      CP = 'A'         ? NAND GATE
        0C7D  28 0A      JRZ L4 :         JUMP IF YES
        0C7F  FE 4F      CP = '0'         ? NOR GATE
        0C81  28 06      JRZ L4 :         JUMP IF YES
        0C83  FE 55      CP = 'U'         ? RUN
        0C85  28 3F      JRZ L12 :        JUMP IF YES
        0C87  18 30      JR L10 :         JUMP TO INPUT ERROR
L4 :    0C89  CD 8A 0D   CALL STORE       PUT CHAR IN GATE LIST
L5 :    0C8C  3E 20      A = 'SPACE'
L6 :    0C8E  ED A1      CP1              )SEARCH FOR NEXT SPACE
        0C90  20 FC      JRNZ L6 :        )DO AGAIN IF NO
        0C92  00         NOP
        0C93  7E         A,(HL)           LOAD CHARACTER
        0C94  FE 20      CP = 'SPACE'     ? SPACE
        0C96  28 CA      JRZ L2 :         END OF ENTRY
        0C98  CD A4 0C   CALL CONVERT     CONVERT CRT I/P TO HEX
        0C9B  FE 40      CP = 40H         )TOO MANY ENTRIES GO
        0C9D  30 0E      JRNC L8 :        )TO ERROR
        0C9F  CD 8A 0D   CALL STORE       STORE IN GATE LIST
        0CA2  18 E8      JR L5 :          GET NEXT PART OF INPUT

CONVERT

L7 :    0CA4  EB         EX DE/HL
        0CA5  CD 5A 02   CALL M/NEXNUM    CONVERT I/P TO HEX NUM
        0CA8  3A 13 0C   A, (0C13)        STORE HEX NUM IN A
        0CAB  EB         EX DE/HL
        0CAC  C9         RTN
```

```
ERROR ENTRY ROUTINE

L8 :    0CAD  2A FC 0D   A, (0DFC)        GET GATE LIST POINTER
        0CB0  3E 40      A = 40H          )DELETE ERROR
L9 :    0CB2  ED A9      CPD              )LINE
        0CB4  38 FC      JRC L9 :         )
        0CB6  22 FC 0D   (0DFC), HL       SET NEW GATE LIST POINTER
L10 :   0CB9  CD 1A 0E   'CALL MESS 2     PRINT ERROR MESSAGE 2
        0CBC  18 A2      JR L2 :          GET NEW ENTRY
L11 :   0CBE  CD 37 0E   CALL MESS 3      PRINT ERROR MESSAGE 3
        0CC1  E5         PUSH HL          SAVE ADDRESS
        0CC2  C3 62 0D   JP L26 :         GO TO I/P MOD PHASE
        0CC5  00         NOP

INITIALISATION FOR RUN PHASE

L12 :   0CC6  CD 8A 0D   CALL STORE       STORE 'U' IN GATE LIST
        0CC9  EF 1F 00   RST SCROLL
        0CCC  3E 30      A = '0'          )
        0CCE  21 A0 0F   HL = 0F80        )CLEAR ALL STATE LIST
        0CD1  06 40      B = 40H          )LOCATIONS TO '0'
L13 :   0CD3  77         (HL), A          )
        0CD4  23         INC HL           )
        0CD5  10 FC      DJNZ L13 :       )

RUN PHASE

L14 :   0CD7  AF         XOR A,A          SET A = 00H
        0CD8  32 FB 0D   (0DFB), A        SET COUNTER TO 00H
L15 :   0CDB  21 FB 0D   HL = 0DFB        )SET COUNTER ADDRESS
        0CDE  34         INC (HL)         )AND INCREMENT
        0CDF  CA BE 0C   JZ L11 :         JUMP IS UNSTABLE
        0CE2  21 A0 0E   HL = 0EA0        SET HL TO GATE LIST
L16 :   0CE5  E5         PUSH HL
        0CE6  7E         A, (HL)          LOAD CHARACTER FROM GATE
                                          LIST
        0CE7  FE 4E      CP = 'N'         ? AND GATE
        0CE9  28 1C      JRZ L19 :        IF YES GO TO AND
        0CEB  FE 52      CP = 'R'         ? OR GATE
        0CED  28 27      JRZ L20 :        IF YES GO TO OR
        0CEF  FE 41      CP = 'A'         ? NAND GATE
        0CF1  28 32      JRZ L21 :        IF YES GO TO NAND
        0CF3  FE 4F      CP = '0'         ? OR GATE
        0CF5  28 3D      JRZ L22 :        IF YES GO TO NOR
        0CF7  FE 55      CP = 'U'         ? RUN
        0CF9  28 50      JRZ L25 :        IF YES RUN IS FINISHED
L17 :   0CFB  D1         POP DE           WASTE STACK LOCATION
        0CFC  18 E7      JR L16 :         GO FOR NEXT CHARACTER
L18 :   0CFE  E1         POP HL           )
        0CFF  23         INC HL           )
        0D00  CD A2 0D   CALL FINDST.     )CHANGE O/P
        0D03  00         NOP              )CONDITION
        0D04  12         (DE), A          )
        0D05  18 D4      JR L15 :         )

AND
```

```
L19 :   0D07  23          INC HL           STEP TO OUPUT
        0D08  CD A2 0D    CALL FINDST      )FIND STATE AND LOAD
        0D0B  1A          A, (DE)          )
        0D0C  D9          EXX
        0D0D  0E 31       C = '1'          SET C TO '1' STATE
        0D0F  CD AC 0D    CALL PROGT.      PROCESS THE GATE FUNCTION
        0D12  28 E7       JRZ L17 :        IF NO CHANGE REQ'D GO TO L17 :
        0D14  18 E8       JR L18 :         CHANGE O/P STATE

OR

L20 :   0D16  23          INC HL           STEP TO O/P
        0D17  CD A2 0D    CALL FINDST      )FIND STATE AND LOAD
        0D1A  1A          A, (DE)          )
        0D1B  D9          EXX
        0D1C  0E 30       C = '0'          SET C TO '0' STATE
        0D1E  CD AC 0D    CALL PROGT.      PROCESS GATE FUNCTION
        0D21  28 D8       JRZ L17 :        IF NO CHANGE REQ'D GO TO L17 :
        0D23  18 D9       JR L18 :         CHANGE O/P STATE

NAND

L21 :   0D25  23          INC HL           STEP TO
        0D26  CD A2 0D    CALL FINDST      )FIND STATE AND LOAD
        0D29  1A          A, (DE)          )
        0D2A  D9          EXX
        0D2B  0E 31       C = '1'          SET C TO '1' STATE
        0D2D  CD AC 0D    CALL PROGT       PROCESS GATE FUNCT:ON
        0D30  20 C9       JRNZ L17 :       IF NO CHANGE REQ'D GO TO L17 :
        0D32  18 0D       JR L23 :

NOR

L22 :   0D34  23          INC HL           STEP TO OUPUT
        0D35  CD A2 0D    CALL FINDST      )FIND STATE AND LOAD
        0D38  1A          A, (DE)          )
        0D39  D9          EXX
        0D3A  0E 30       C = '0'          SET C TO '0' STATE
        0D3C  CD AC 0D    CALL PROGT       PROCESS GATE FUNCTION
        0D3F  20 BA       JRNZ L17 :       IF NO CHANGE REQ'D GO TO L17 :

INVERT OUTPUT

L23 :   0D41  FE 31       CP = '1'
        0D43  28 03       JRNZ L24 :
        0D45  3C          INC A            CHANGE A TO '1'
        0D46  18 B6       JR L18 :
L24 :   0D48  3D          DEC A            CHANGE A TO '0'
        0D49  18 B3       JR L18 :

DISPLAY

L25 :   0D4B  21 A0 0F    HL = 0FA0        SET TO STATE LIST
        0D4E  11 90 0B    DE = 0B90        SET DISPLAY
        0D51  01 A0 00    BC = '10'        SET NUMBER OF CHARACTERS
        0D54  ED B0       LDIR             PRINT 10 INPUTS
        0D56  0E 06       SET BC = '6'
        0D58  09          ADD HL, BC       SET HL TO
        0D59  1E 9F       DE = 0B9F        SET DISPLAY
        0D5B  0E 0A       SET BC = '10'    SET NUMBER OF CHARACTERS
        0D5D  ED B0       LDIR             PRINT 10 OUTPUTS
        0D5F  EF 1F 00    RST SCROLL

INPUT MODIFIED STAGE

L26 :   0D62  11 90 0B    DE = 0B90        SET CRT TO 1ST INPUT
        0D65  21 A0 0F    HL = 0FA0        SET HL TO STATE LIST
        0D68  06 0A       B = '10'
L27 :   0D6A  CD 3E 00    CALL M/CHIN      GET ENTRY
        0D6D  FE 4C       CP = 'L'         ? LIST
        0D6F  CA C6 0D    JZ LIST
        0D72  FE 30       CP = '0'         ?'0'
        0D74  28 0A       JRZ L28 :
        0D76  FE 31       CP = '1'         ?'1'
        0D78  28 06       JRZ L28 :
        0D7A  FE 45       CP = 'E'         ?EXIT
        0D7C  28 21       JR L30 :         GO BACK TO START
        0D7E  18 EA       JR L27 :         IGNORE ANYTHING ELSE
L28 :   0D80  77          (HL), A          STORE NEW I/P IN STATE LIST
        0D81  12          (DE), A          UPDATE DISPLAY
        0D82  23          INC HL           )STEP TO NEXT INPUT
        0D83  13          INC DE           )
        0D84  10 E4       DJNZ L27 :       D0 FOR 10 INPUTS
        0D86  E1          POP HL           EMPTY STACK
        0D87  C3 D7 0C    JP RUN           RUN EMULATOR

STORE

L29 :   0D8A  EB          EX HL/DE
        0D8B  2A FC 0D    HL (0DFC)        LOAD CONTENT OF GATE LIST
        0D8E  77          (HL), A          )ENTER CHARACTER
        0D8F  23          INC HL           )INC AND STORE GATE LIST
        0D90  22 FC 0D    (0DFC) HL        )POINTER
        0D93  7D          A, L             )
        0D94  EB          EX HL/DE         )?OVERFLOW
        0D95  FE A0       CP = A0          )
        0D97  C0          RNZ              RETURN IF NO OVERFLOW
        0D98  CD 5C 0E    CALL MESS 4      PRINT OVERFLOW MESSAGE
        0D98  CD 3E 00    CALL M/CHIN      WAIT FOR ENTRY
        0D9E  E1          POP HL           EMPTY STACK
L30 :   0D9F  C3 50 0C    JP 0C50          START AGAIN
```

```
FINDST

L31 :   0DA2  06 00       B = 00H          )SET BC TO STATE
        0DA4  4E          C, (HL)          )NUMBER
        0DA5  EB          EX HL/DE
        0DA6  21 A0 0F    HL = 0FA0        SET HL TO STATE LIST
        0DA9  09          ADD HL/BC        )SET DE TO STATE
        0DAA  EB          EX HL/DE         )
        0DAB  C9          RTN

PROGT

L32 :   0DAC  47          B, A             PUT INITIAL STATE IN B
        0DAD  51          D, C             COPY REF. STATE IN D
        0DAE  D9          EXX
L33 :   0DAF  23          INC HL           )STEP TO I/P STATE
        0DB0  7E          A, (HL)          )AND LOAD
        0DB1  FE 40       CP = 40H         )END OF INPUT LIST
        0DB3  30 0C       JRNC L35 :       )
        0DB5  CD A2 0D    CALL FINDST      FIND NEXT STATE AND
        0DB8  1A          A, (DE)          LOAD
        0DB9  D9          EXX
        0DBA  B9          CP, C            COMPARE STATE WITH REF.
        0DBB  28 01       JRZ L34 :        IF YES DON'T INVERT REF.
        0DBD  57          D, A             INVERT REFERENCE STATE
L34 :   0DBE  D9          EXX
        0DBF  18 EE       JR L33 :         GO FOR NEXT INPUT
L35 :   0DC1  D9          EXX
        0DC2  7A          A, D             PUT FINAL STATE IN A
        0DC3  B8          CP, B            HAS IT CHANGED STATE
        0DC4  D9          EXX
        0DC5  C9          RTN

LIST

L36 :   0DC6  21 A0 0E    HL = 0EA0        SET TO GATE LIST
L37 :   0DC9  7E          A, (HL)          LOAD ENTRY
        0DCA  FE 4E       CP = 'N'         ?AND
        0DCC  CC 7C 0E    CZ MESS 5
        0DCF  FE 52       CP = 'R'         ?OR
        0DD1  CC 83 0E    CZ MESS 6
        0DD4  FE 41       CP = 'A'         ?NAND
        0DDC  CC 89 0E    CZ MESS 7
        0DD9  FE 4F       CP = '0'         ?NOR
        0DDB  CC 91 0E    CZ MESS 8
        0DDE  FE 55       CP = 'U'         ?RUN
        0DE0  CA C6 0C    JZ L12 :         END OF LIST
L38 :   0DE3  23          INC HL           STEP TO NEXT ENTRY
        0DE4  7E          A, (HL)          LOAD CHARACTER
        0DE5  FE 40       CP = 40H         ?NEXT GATE TYPE
        0DE7  30 08       JRC L39 :
        0DE9  CD 44 02    CALL M/B2 HEX    DISPLAY AS 2 CHARACTERS
        0DEC  EF 20 00    RST SPACE
        0DEF  18 F2       JR L38 :         GO FOR NEXT REF
L39 :   0DF1  CD 3E 00    CALL M/CHIN      WAIT FOR KEY PRESSED
        0DF4  EF 1F 00    RST SCROLL
        0DF7  18 D0       JR L37 :         PRINT SUBSEQUENT GATES

MESSAGES

MESS 1

0E00    EF 4C 4F 47 49 43 20 45    LOGIC EMULATOR PROGRAM
        4D 55 4C 41 54 4F 52 20
        50 52 4F 47 52 41 4D 1F
        00 C9

MESS 2

0E1A    EF 49 4E 50 55 54 20       INPUT ERROR ENTRY IGNORED
        45 52 52 45 52 20 45 4E
        54 52 59 20 49 47 4E 4F
        52 45 44 1F 00 C9

MESS 3

0E37    EF 1F 43 49 52 43 55 49    CIRCUIT UNSTABLE WITH THIS INPUT
        54 20 55 4E 53 54 41 42
        4C 45 20 57 48 49 53 20
        49 4E 50 55 54 00 C9

MESS 4

0E5C    EF 43 49 52 43 55 49 54    CIRCUIT OVERFLOW RE-ENTER IT
        20 4F 56 45 52 46 4C 4F
        57 20 52 45 2D 45 4E 54
        45 52 20 49 54 1F 00 C9

MESS 5

0E7C    EF 41 4E 44 20 00 C9       AND

MESS 6

0E83    EF 4F 52 20 00 C9          OR

MESS 7

0E89    EF 4E 41 4E 44 20 00 C9    NAND

MESS 8

0E91    EF 4E 4F 52 20 00 C9       NOR
```

Ian Sinclair

After a month or two of playing with his Trusty 80 Ian Sinclair has a few more comments to make.

After several months of intense 'playing around' with my TRS-80 I have managed to untangle some of the problems that were encountered when I was writing my review, see November's Computing Today. In an attempt to assist anyone who has bought themselves one here are a few hints.

Cassette Handling

In the review I said one or two unfair things about the cassette file handling capabilities of the machine, or so it appears on further scrutiny. I've learned that it pays to test out any program which uses cassette files, there is a very simple scheme for doing this.

In place of every PRINT#–1 statement put PRINT : STOP, and similarly for each INPUT#–1 you substitute STOP : In this way you will see on the screen exactly what is going to be recorded and you can jump to the step after INPUT#–1 to see what happened on replay. This tends to save a lot of trouble with cassette testing and you can see at a glance if you are using too many bytes, or if the replay procedure is wrong.

Television Or Monitor

After Mr Heller's comments I played around with the TV circuits and concluded that I didn't really need a monitor after all. If you find that the lettering on your screen looks a bit disjointed, in particular double ee's, it's a fair bet that there's too much bias on the modulator. Open up the modulator box and you find a standard ASTEC device, see Fig.1, with the video input taken through a 100 uF capacitor and biassed by two resistors. Try connecting a 10 K pot between the video input and the earthed case, or better still connect a 5 K pot in place of the two resistors. Starting with the voltage at around 2V5 gently twiddle the pot until you have the lettering as you like it (to coin a cliche). The improvement can be quite dramatic and is well worth a couple of minutes of your time. It should also be possible to improve the graphics by doing a DC restore at this point.

Problem Loads

Because BASIC is so much simpler to operate than machine code it occurred to me that system tapes could be entered as part of a BASIC program, the KBFIX being a prime example. Keyboard bounce has always been a problem on the TRS-80, not of major proportions but if you get LLIST instead of LIST and like me you haven't got a lineprinter yet (at £1200 a time who has?) the whole thing hangs up until you use RESET. The manual, and other sources, tell you to remove the offending keytop and clean the contacts but on my TRS-80 they are NOT removeable so where do we go from here.

The answer is a machine code subroutine which slows down the rate at which the keys are scanned, this means that the key isn't read until after the bounce. There is a routine supplied with the machine but it simply wouldn't load, not at any setting, and a quick listen convinced me that it wasn't even made for the same machine, the version on the other side did load fortunately. As I didn't feel like going through all this bother each time I listed the machine code and wrote a short BASIC program which POKE's the values into the correct addresses at the top of memory. Now, whenever I want to enter a long BASIC program I start with this tape which lies in lines 1 to 5 along with entry procedures. I run this and then start entering the new program from lines 10 onwards. Once the new program is entered, I can then delete lines 1 to 5, or if the new program is one which requires a lot of keyboard entries I can keep the first five lines in place to make sure that there is no bouncing on new entries. It is much more satisfactory than using a SYSTEM tape. At the moment, I'm developing a method (which has worked in its first trials) of placing bytes from a system tape directly into a BASIC program without having to note down the values and enter them.

Printer Or Disc

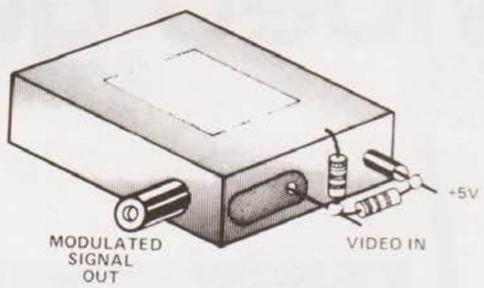
After the first few weeks with the machine, I was convinced that the first major addition would be a disc system. A bit more experience has changed my mind. Useful as disc operation might be, a printer now seems a much more useful addition, because any program needs referring to and unless you're going to spend hours at the keyboard, the referring *has* to be done on a printout. It is decidedly infuriating to spend a long time sorting out a program and then having to hand-copy it from the screen.

The snag at the moment is the silly prices of printers. A printer is a box which is mainly empty and certainly doesn't contain so much expensive equipment as a £150 Japanese electric typewriter. It looks very much like the pocket calculator story again, and production levels must surely be getting to the stage where prices will drop. I'm just not going to be tempted by printers which give only 40 columns on 'peculiar' paper, what's needed is at least 80 columns on plain paper – *not* sprocketed, since the price of these holes is just ridiculous. Any genius who can convert a £100 electric to use bog roll and interface it to the TRS-80 should earn a fortune!

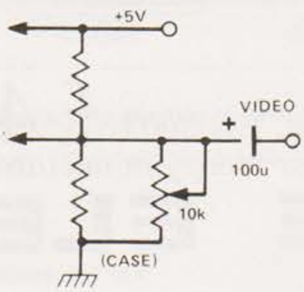
Self Instruction

All-in-all, then, I've had an instructive time – and I'm still learning fast. I've seen some good software (from A.J. Harding) and I now have most of the programs I need for keeping track of my books and my accounts. There's one thing I think should be stressed to all prospective computer owners – time. It takes a long time to enter a program from the keyboard, it takes longer to get it running the way you want it, and longer still to tidy up the printing. The factor which, more than anything else, distinguishes a 'professional' program from an amateur one is foolproof operation. If each act on your part is prompted by clear instructions on the screen, if each mistake results in a rescue operation, rather than a blank screen, if answers are printed legibly with explanations of what they represent, then you have a reasonably professional program. Don't kid yourself that you can get by with much less. The acid test is to go back to a program you last used over a month ago. Can you run it right away without reading a listing to see what it's all about? If you can't or if it's not immediately obvious what you should do, then it needs a lot of work – and it all takes time. This time factor is one which must be explained to any prospective user of a microcomputer. A business user expects to be able to switch on, load a program and start operations. We may find it more interesting to develop our own programs, but there's no need to settle for less once we have them running. From that point of view, the excelling editing facilities of the TRS-80 have been worth their weight in gold. From the adverts I read, it appears that the unfortunate buyers of a system costing twice as much have to spend £80 odd on a software package which lets their machine have some of the features which are completely standard on the TRS-80. Now that more than one supplier is selling TRS-80 at £399 + VAT, it's a better buy than ever.

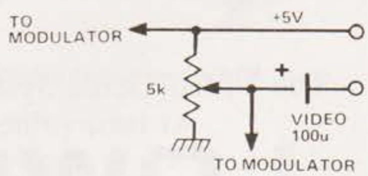
The debated TRS 80 monitor. If you modify the modulator circuit as shown below you can use your telly instead and save yourself a few pennies.



(a)



(b)



(c)

Alistair Smith.

KIM CLOCK PROGRAM

The program converts a standard KIM I board to a digital clock using the seven segment displays to show hours, minutes and seconds in the usual way. Any time can be entered as a start value, and the program commenced at address 0000_{16}.

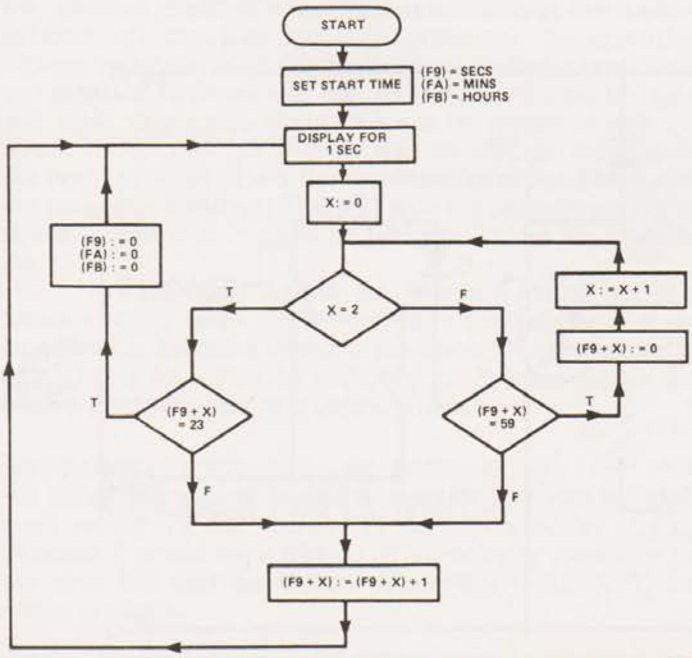
Program Function
The program comprises three parts:
1) Initialisation. The start values are read into address locations F9 (seconds), FA (minutes), FB (hours); these being the memory locations accessed by the SCANS display routine (in the monitor ROM). Unfortunately, values cannot be directly read into these locations from the keyboard, hence this part of the program is required.
2) Delay. The current time is displayed by calling subroutine SCANS from a loop, the duration of which causes a delay of one second before the display is incremented.
3) Logic. The display is incremented such that a 24-hour clock cycle is emulated. The processor is set in decimal mode and the program determines the values of the display memories.

Running The Program
After the program has been stored in RAM, the following procedure sets the start time:

Address	Data
0001	seconds
0005	minutes
0009	hours

The program must be started from address 0000_{16}. To change to a 12 hour cycle, put value 11_{16} in address 001E. This however displays zero hours at twelve o'clock.

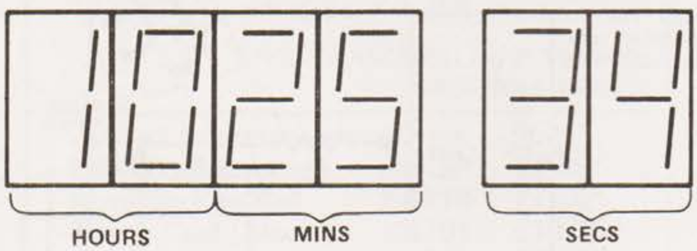
"Fine adjustment" of the delay loop is affected by varying normal value of EB_{16} in address $000D_{16}$, in fact a 24 hour cycle takes about six minutes if this value is reduced to 01_{16}!




```
0000  A9  00             LDA% 00
0002  85  F9             STAF9
0004  A9  00             LDA% 00
0006  85  FA             STA FA
0008  A9  00             LDA% 00
000A  85  FB             STA FB
000C  A9  EB      START  LDA% EB
000E  85  47             STA 47
0010  20  1F  1F  DISP   JSR SCANS
0013  C6  47             DEC 47
0015  D0  F9             BNE DISP
0017  A2  00             LDX% 00
0019  E0  02      NEXT   CPX% 02
001B  D0  11             BNE TEST
001D  A9  23             LDA% 23
001F  D5  F9             CMP F9,X
0021  D0  19             BNE INCREM
0023  A9  00             LDA% 00
0025  85  F9             STA F9
0027  85  FA             STA FA
0029  85  FB             STA FB
002B  4C  0C  00         JMP START
002E  A9  59      TEST   LDA% 59
0030  D5  F9             CMP F9,X
0032  D0  08             BNE INCREM
0034  A9  00             LDA% 00
0036  95  F9             STA F9,X
0038  E8                 INX
0039  4C  19  00         JMP NEXT
003C  18          INCREM CLC
003D  F8                 SED
003E  B5  F9             LDA F9,X
0040  69  01             ADC% 01
0042  95  F9             STA F9,X
0044  4C  0C  00         JMP START
```

Note: % denotes direct addressing.

TIME DISPLAY



Above:- An example of the time display format on the KIM.

Left:- The flowchart for the KIM time program.

M.J. Bell.

We take a close look at a powerful monitor for Nascom

The NASBUG T4 monitor program is a 2K (two ROM's) package which when fitted into a NASCOM 1 microprocessor controls all the basic monitor functions. Documentation is supplied detailing both the commands and their uses together with an object code listing. The T4 is a third generation of Nascom monitors improving on the facilities of its predecessors. It is downwards compatible which means that the hours spent writing programs using a T2 monitor need not be wasted because all existing monitor subroutines have been retained at the old start addresses. This is very commendable on the part of Nascom as it must have presented accommodation problems.

Where To Put It

The two EPROMS are plugged into the mainboard, one in place of the T2 the other in the spare socket so ensure that the correct EPROM is put into the correct socket. The address range is 0000 to 07FF with 0C00-0C5F being used as a workspace area to contain reflected address and temporary registers used by the monitor subroutines.

Many new commands have been incorporated which take the NASCOM into the elevated class of business machines where intercommunication by modem or acoustic links is required. Other new commands facilities the use of printers, tape punches or teleprinters. The tape loading speed can be increased by 4 times and a generate command will enter a start address onto tape for automatic execution of program once read from cassette. Another new useful feature is the various keyboard options, and the ability of entering text into program from the keyboard.

A detailed list and explanation of all the commands follows: •

Table 1. T4 COMMANDS

	A	Arithmetic	Calculate the sum, difference and relative jump between two hex numbers.
*	B	Breakpoint	Halts a program at a predetermined address and prints SP, PC, AF, HL, DE, BC, I, IX, IY and flags set.
*	C	Copy	Copies a block of data downwards to a new starting address. Can also be used for wiping a block of memory clean.
*	D	Dump	Dumps programs onto a cassette or punched tape in the T2 'slow' format. (224 band).
*	E	Execute	Executes a program from the entered address.
*	G	Generate	Dumps a program onto cassette with a prefix and suffix start address. This facilitates loading and execution of a program without entering any monitor commands.
	I	Intelligent Copy	Copies a block of data either up or down a new location. Supersedes the copy command.
	K	Keyboard	K0 sets keyboard function normal. K1 reverses the effect of the shift key. K2 with the space bar held down the ASCII value of the keypressed is entered and displayed K3 is the function of K1 and K2. K4 reverses the 7th bit of the ASCII code for graphics use (when available).
*	L	Load	Loads a program from cassette or punched tape in the T2 format.
*	M	Modify	Displays the address entered for modification if required. Two extra facilities have been added: ":" will step back to the previous address. "/" followed by an address will jump to that address. "." still returns to the monitor for another command.
	N	Normal	Resets the keyboard and CRT, when using a printer or the 'Z' command.
	O	Output	This command is used to output any value to any specified port.
	Q	Query	This is the converse of "Output" in that it inputs value present in a predetermined port.
	R	Read	Loads a program from cassette or paper tape in a new format that operates at 4 times the "T2" standard rate.
*	S	Single Step	Steps the program through one operation at a time printing the breakpoint display format.
*	T	Tabulate	Tabulates onto the screen the program locations between two specified addresses in rows of 8 bytes in length. The improved appearance of the table is offset from the margin two spaces.
	W	Write	Writes a specified block of data onto a cassette of paper tape in the fast format used by the "Read" command.
	X	External	This command followed by a suffix enables the computer to communicate with peripherals such as teletypes, mainframe computers, intelligent printers etc. The capability of selecting half or full duplex working, even or odd parity and optional line feed make this command very powerful indeed.
	Z	Change Command Table	Allows the command table to be changed or disabled from the keyboard.
	?	Command	Lists the command letters for reference.

Those with * are also provided on the T2 monitor.

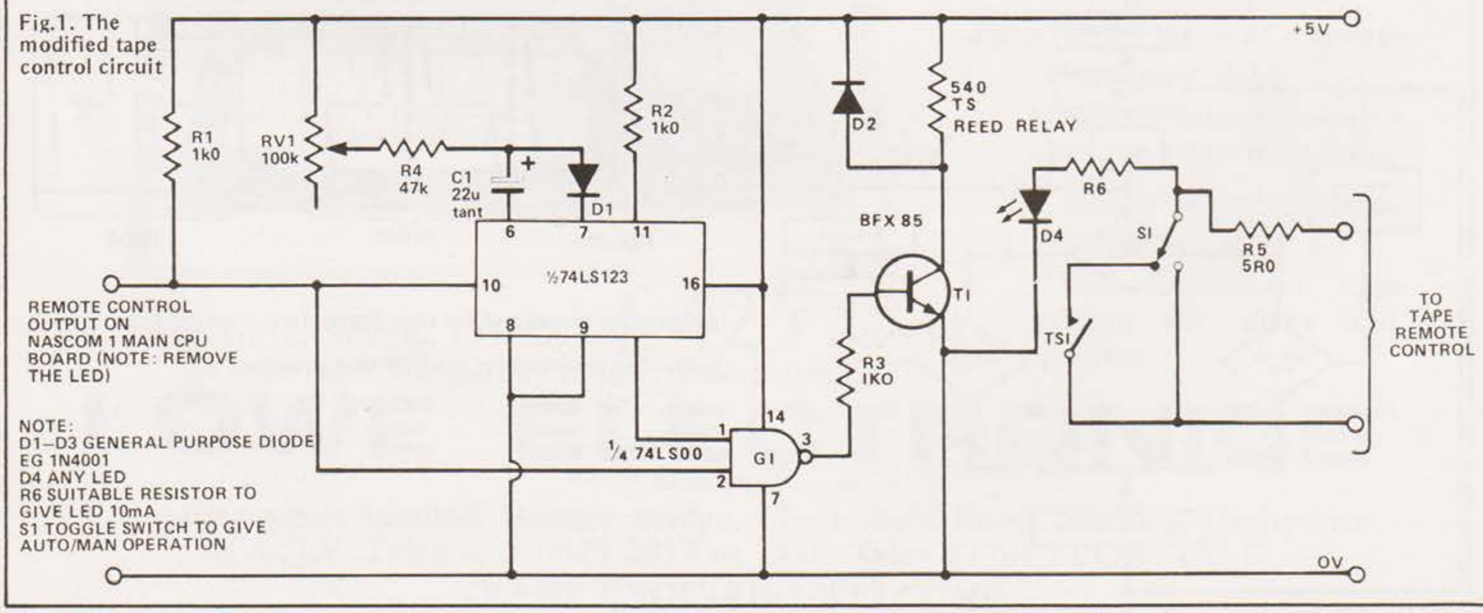


Fig.1. The modified tape control circuit

T4 REVIEW

Observations

In addition to the extra commands "T4" has attempted to clean-up some of the problems of the "T2" in the area of reset and tape loading. The new cursor control character of "IC" to home the cursor without scrolling is welcome.

One feature very worthy of comment is the reorganisation of the program around the restart vectors, as follows:

Table 2. Vector assignment

Hex Code	Vector	Function
C7	RST 0	Restart the system.
CF	RST 8	End program and return to monitor without clearing screen.
D7	RST 10	Relative call.
DF	RST 18	User subroutine call.
E7	RST 20	Breakpoint.
EF	RST 28	String.
F7	RST 30	Call CRT display routine.
FF	RST 40	Delay timer.

The monitor as received by the author was well documented and with few exceptions worked very well, although it must be said that facilities were not available to test all the peripheral options. The arithmetic and keyboard commands were immediately recognised as most useful, and the faster read and write format was a joy to behold. There are however a few problems that need further consideration. If you are in the habit of dumping a program and then reading it back to check if it was loaded correctly *BE WARNED!* Any errors will be read back into the store at the expense of losing the existing program. What is needed is a VERIFY command that will compare the contents of the tape with the memory. The 'GENERATE' and 'READ' commands are also in need of attention to be of full use to those that have their tape recorder under automatic control (see CT November 1978). The generate command will not start the tape until after the generate prefixes have been output and at the end, it switches off the tape too soon preventing the execution address being recorded. Similarly when reading a tape back, the generate function will not take place because the tape will again stop early. Whilst this can be overcome with the circuit shown in Fig.1 it must also be pointed out that in the event of input errors the monitor will still go ahead with the program execution. The text entering capability is not as straight forward as it could be although it does the required job.

An additional facility that I would like to see in a future monitor is the ability to specify the number of times an address is executed before a breakpoint routine is carried out. Thus a loop may be executed say 9 times and on the tenth a predetermined breakpoint effected.

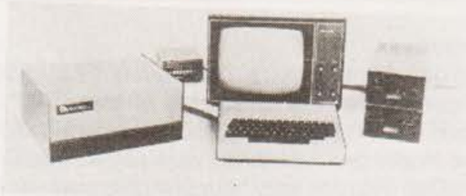
Conclusions

All things being considered this monitor is a vast improvement on the T2 and well worth the capital outlay of £25. However I would hope that its deficiencies be overcome by the time the next generation of monitor (NAS–SYS) becomes available.

NEW UNBEATABLE 1980 PRICES NOW!

EXPLORER/85

FEATURES INTEL 8085 cpu WITH ON BOARD S-100 EXPANSION



FLEXIBILITY: Real flexibility at LAST. The EXPLORER/85 features the Intel 8085 cup 100% compatible with all 8080A and 8085 software. Runs at 3Mhz. Mother Board (Level A) with 2, S-100 pads expandable to 6 (Level C).
MEMORY: 2K Monitor ROM — 4K WORKSPACE/USER RAM — 1K Video RAM — 8K Microsoft BASIC in ROM or Cassette.
INTERFACES: STANDALONE FULL ASC11 Keyboard Terminal, 32/64 characters, 16 lines. Cassette interface (with motor control and cassette-File Structure). RS-232/20Ma loop. 4, 8 bit: 1, 6 bit I/O ports, programmable 14 bit binary counter/timer. Direct interface for any S-100 Board. FULL Buffering Dicoding for S-100n Bus pads. Wait state generator for slow memory. Each stage has separate 5v 1A regulator for improved isolation and freedom from cross talk. P.S.U. requirements:— 8v, 6.3v AC. Runs with North Star controller and Floppies/CPM. EXPLORER/85 is expandable to meet your own requirements with easy to obtain S-100 peripherals. EXPLORER/85 can be purchased in individual levels, kit form or wired and tested. OR as a package deal as above.

£275 + VAT
Microsoft BASIC on Cassette

£295 + VAT
Microsoft BASIC in ROM

AVAILABLE NOW!

We are killing inflation with ELF II

The tried and tested Microcomputer System that expands to meet your needs

STARTS AT £59.95 +VAT

ELF II
Board with video output
FEATURING THE RCA COSMAC 1802 cpu

STOP reading about computers and get your "hands on" an ELF II and Tom Pitman's short course. ELF II demonstrates all the 91 commands which an RCA 1802 can execute and the short course speedily instructs you how to use them.
ELF II's VIDEO OUTPUT makes it unique among computers selling at such a modest price. The expanded ELF II is perfect for engineers, business, industry, scientific and educational purposes.

ELF II EXPANSION KITS — Ex VAT	
Power supp 6.3v AC for ELF II	£5.00
ELF II Steel Cab	£19.75
Giant Bd	£25.50
4K RAM Bd	£57.50
Expansion power supply	£19.00
ASC II KeyBd	£39.95
ASC II Cab	£12.75
Kluge Bd	£11.00
86 Pin Con	£3.75
LIGHT PEN	£6.00
Video Graphics Bd	£61.50
ELF II Tiny Basic cassette	£9.75
ELF-Bug	£9.75
Short course on progrm	£3.00
Short course on Tiny basic	£3.00
RCA 1802 manual	£3.00
Tex Editor Assembler, etc.	£12.75

ELF II BOARD
SPECIFICATION
*RCA 1802 8-bit micro-processor with 256 byte RAM expandable to 64K bytes.
*RCA 1861 video IC to display program on TV screen via the RF Modulator Single Board with Professional hex key-board – fully decoded to eliminate the waste of memory for key-board decoding circuits. Load, run and memory protect switches. 15 Registers. Interrupt. DMA and ALU. Stable crystal clock. Built in power regulator. 5 slot plug in expansion bus (less connectors)

NEWTRONICS KEYBOARD TERMINAL
AT £114.20 + VAT

The Newtronics Keyboard Terminal is a low cost stand alone Video Terminal that operates quietly and maintenance free. It will allow you to display on a monitor 16 lines of 64 characters or 16 lines of 32 characters on a modified TV, (RF Modulator required).
The characters can be any of the 96 ASC 11 alphanumerics and any of the 32 special characters, in addition to upper/lower case capability it has scroll-up features and full X-Y cursor control. All that is required from your microcomputer is 300 baud RS232-C or 20ma loop serial data plus a power source of 8v DC and 6.3v AC. The steel cabinet is finished in IBM Blue/Black. And if that is not enough the price is only **£114.20 + VAT as a kit**, or **£144.20 + VAT** assembled and tested. Plus £2 P & P (Monitor not included).

RACAL AP12, C12 TAPES: 10 for £4.50 + VAT

NOW AVAILABLE 8K FULL BASIC FOR ELF II

NEWSOFT GAMES FOR ELF II: 4 for £5 + VAT

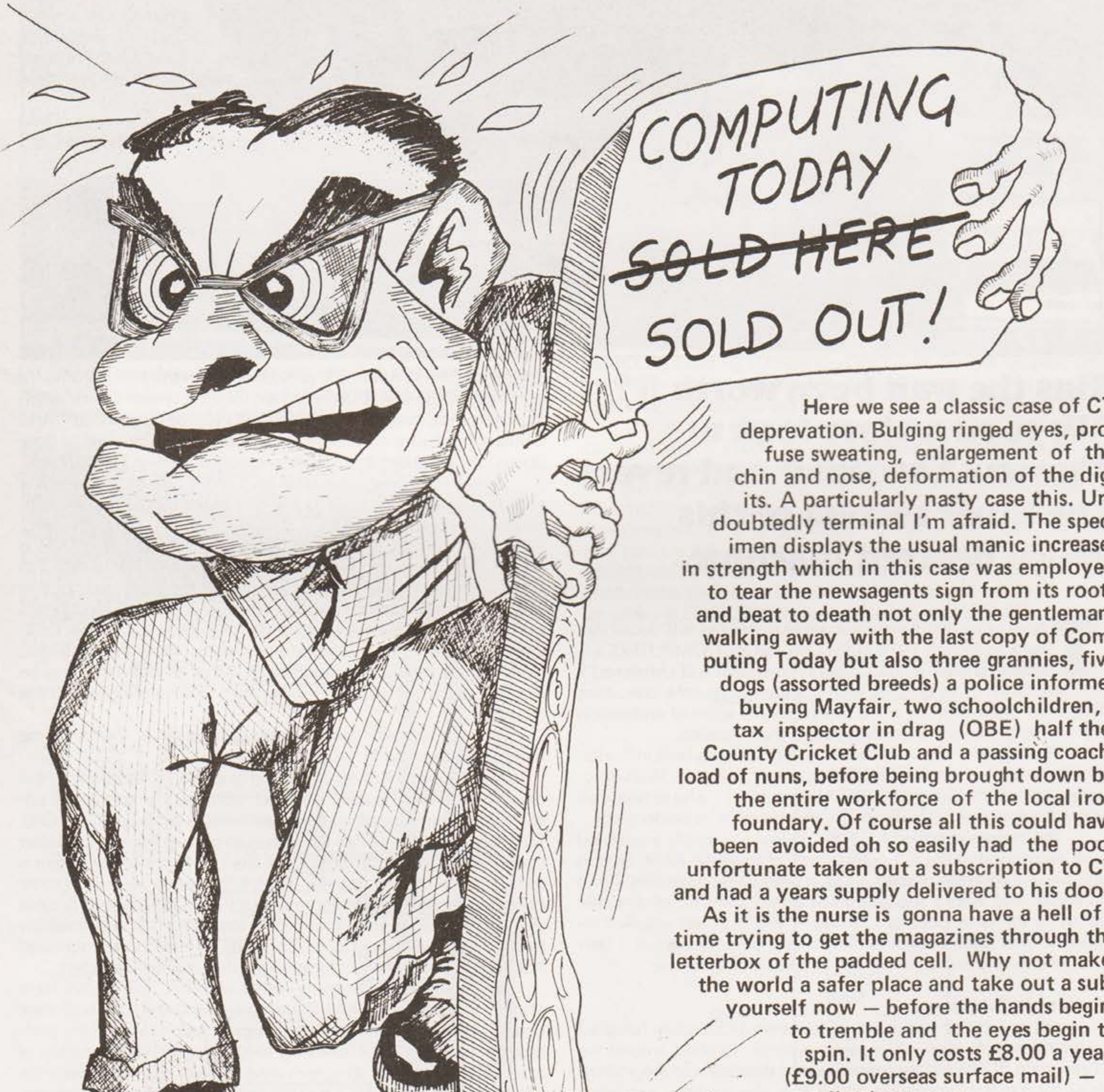
SEND SAE FOR COMPREHENSIVE BROCHURE

Please add VAT to all prices (except manuals) P&P £2. Please make cheques and postal orders payable to **NEWTRONICS** or phone your order quoting BARCLAYCARD, ACCESS number.

NEW ADDRESS: H. L. AUDIO LTD
BIGGER PREMISES 255 ARCHWAY ROAD, LONDON N6 5BS
NEW PHONE No. 01-348-3325

WE ARE NOW OPEN FOR DEMONSTRATIONS AND SALES MON-SAT 9.30-6.30pm NEAR HIGHGATE UNDERGROUND ON MAIN A1 INTO LONDON

ARRRGGGHHH.....



Here we see a classic case of CT deprivation. Bulging ringed eyes, profuse sweating, enlargement of the chin and nose, deformation of the digits. A particularly nasty case this. Undoubtedly terminal I'm afraid. The specimen displays the usual manic increase in strength which in this case was employed to tear the newsagents sign from its roots and beat to death not only the gentleman walking away with the last copy of Computing Today but also three grannies, five dogs (assorted breeds) a police informer buying Mayfair, two schoolchildren, a tax inspector in drag (OBE) half the County Cricket Club and a passing coach load of nuns, before being brought down by the entire workforce of the local iron foundary. Of course all this could have been avoided oh so easily had the poor unfortunate taken out a subscription to CT and had a years supply delivered to his door. As it is the nurse is gonna have a hell of a time trying to get the magazines through the letterbox of the padded cell. Why not make the world a safer place and take out a sub yourself now — before the hands begin to tremble and the eyes begin to spin. It only costs £8.00 a year (£9.00 overseas surface mail) — a small price to pay.

Computing Today
Subscriptions Dept.,
M.A.P. PO Box 35,
Bridge Street,
Hemel Hempstead,
Hertfordshire.

Has the wait been worth it? We took a close look at the new Nascom 2 and reveal the inner secrets of this much delayed machine.

Those of us that have elephantile memories will recall that back in March 1979 NASCOM MICROCOMPUTERS announced their second generation system and christened it the NASCOM 2. In keeping with tradition, 9 months later, their baby has been born in all its glory and the team of enthusiasts (turned professional) are duly proud of their creation.

This high density single board computer has a built in flexibility that no other system in the same class can match. With its integral 2K NAS/SYS monitor, 8K BASIC and 8K (+ a bit or two) user RAM it represents a significant improvement on its predecessor.

The system supplied to us for review was ready assembled and incorporated the full compliment of 8K static RAM. Due to the present shortage of static RAM in production quantities the kit is being sold with a separate RAM board and 16K of dynamic RAM for the same price of £295.00 + VAT + power supply if required. This additional RAM board will be reviewed in a later article if we are fortunate enough to take delivery of one.

System Architecture

It comes as no great surprise that the system architecture has great similarities with most other micro-systems. Centred around the three main BUSSES (the control bus, the data bus and the address bus) the central processing unit (CPU), the memory and interfaces are appropriately interconnected to give a 16-bit address and 8-bit data capability. The published architecture diagram is shown in Fig 1.

Nascom have sensibly retained the use of the Z80 family of chips and have uprated to the Z80A, which has the same machine codes and facilities of the Z80 but will operate at twice the speed, viz 4MHz. With the provision of MK4118 static RAM chips the whole system can be operated at this speed although an option for 2MHz running is provided. When the 16K RAM board or the promised 48K RAM board are used the system can be run in a compromise mode, whereby the CPU will be operated at 4MHz whilst a hardware controlled WAIT period can be used to slow down the operations that require memory accessing to take place.

A welcome new innovation is POWER-ON-RESET, which is switch adjustable for reset to any one of the thousands hex addresses (typically 0000H, 2000H, A000H, E000H etc). With the monitor located at 0000H and BASIC at E000H it means the system can be reset to either the monitor or direct to BASIC. This sort of flexibility is a feature of the entire architecture and accordingly options are also available to replace all onboard user RAM by PROM allowing the board to be as suitable for dedicated systems as for development systems. Even the 8K BASIC ROM can be replaced by another 2K, 4K or 8K ROM that may be desired, particularly useful if other high level languages become available.

The usual method of memory allocation is shown in the memory map of Fig 2.

The 1K video RAM is organised to give a 16-line by 48-chars display with large wasted offscreen borders. The top line is unscrolled so theoretically can be used for titles etc (see MONITOR). This video RAM area can be written to or read from as any other user RAM providing an extra 1K of workspace area if the system is to be used without a monitor display, but when used in its more common memory mapping role it facilitates precise and flexible display formatting. In this mode it is important that any character to be displayed should conform to the standard to the standard ASCII code, with bit 7 set for use with the graphic characters.

In common with many other systems, NASCOM have chosen to ignore the plight of the many users who have built their systems around the non-standard aspect ratio (5 × 4) of the portable TV resulting in the tendancy for the first and last characters of a line to be lost in the off-screen area. Whilst I appreciate the technical problems for NASCOM I am also very aware of the operational problems for their customers.

A fundamental requirement of any new system is its capability in communicating with peripheral equipment. The keyboard is the most important and will be dealt with separately, but printers, floppy disk systems, modems etc are becoming increasingly more popular as their prices fall, and we must not forget the humble cassette recorder. Here again the designers have considered every possibility and have provided options for

Malcolm Bell

TTY/cassette, RS232/20mA interface, half or full duplex working with all the combinations of single/double stop bit, odd/even parity etc. In practice, as with most 'all-singing-all-dancing' machines the biggest problem becomes one of selecting which options are required for each application.

The cassette interface has now been standardised to the KANSAS CITY format and will run happily at 1200 baud (and faster if high quality tape and cassette recorder are used). With the addition of a relay the cassette can be made to start or stop under the control of the CPU, the relay contacts being wired to the remote input to the recorder.

Also provided is a 16 bit Programmable INPUT/OUTPUT port organised as two 8 bit ports with handshake controls, which can be used for interfacing to a wide variety of user controlled circuits such as relay boards, A-D convertors, floppy disc controllers etc. The outputs of these ports are automatically reset by the power-on-reset controls and all outputs and inputs are well buffered. The technical manual for this device is included in the documentation.

The VDU interface takes two forms: A video monitor output of 1 V at 75 ohms for direct connection to a monitor or video section of a TV, and this gives a very sharp and stable display; or a UHF output from an ASTEC modulator which has proved a little disappointing. The one provided on the review system proved to be slightly unstable. Nascom were asked to comment and they suggest that it is a 'one-off' fault. It is true that other NASCOM users have not reported this problem.

Expansion

All of the Z80A control leads, data bus and address bus leads are fully buffered and are available on the 77-way edge-connector in the NASBUS format. This is fully documented in the system manual. There are a few spare locations that have been reserved for the future but these could be used in the interim by the user if required. The bus is capable of supporting the full 64K memory and/or input/output port boards. All memory addressing is carried out on the expansion boards.

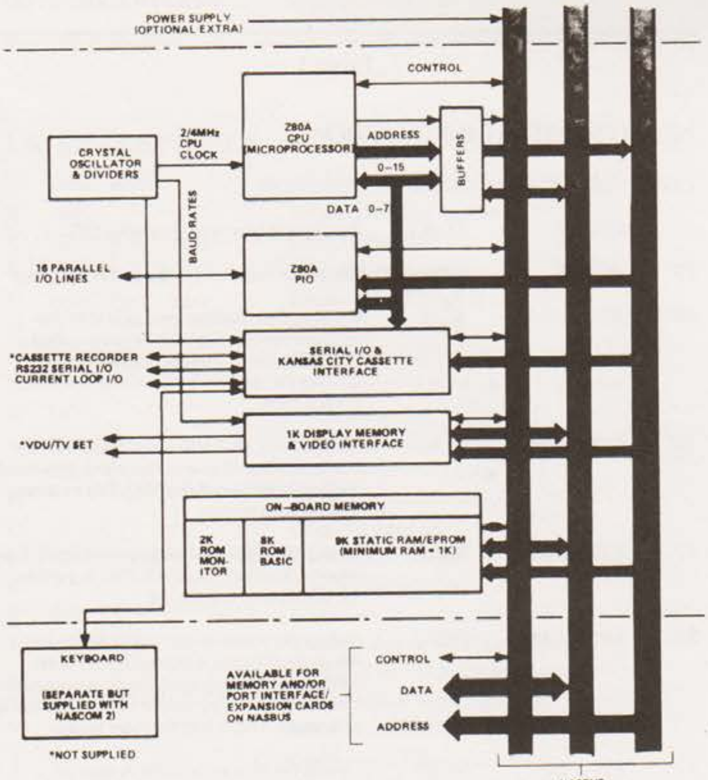
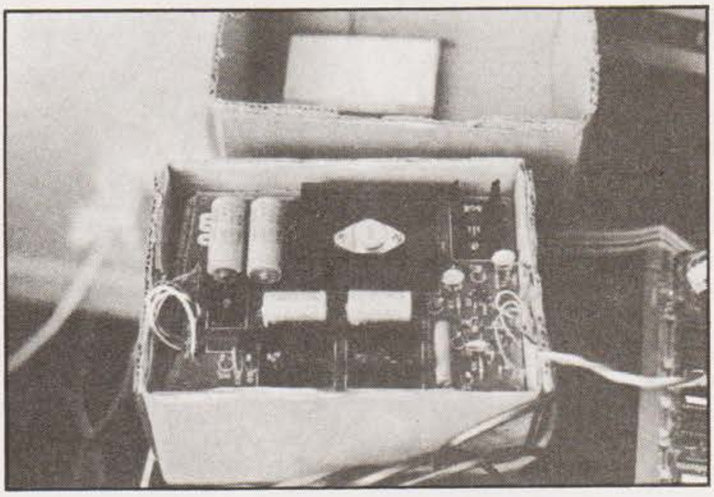


Fig 1. The Nascom 2 architecture diagram revealing sensible design ideas and wide flexibility.



The 3 A power supply unit. The cardboard box it came in makes a useful case for it until we get our racking system. Construction is simple and it performs perfectly, although a little warm.

Keyboard

The keyboard is an expanded version of the one used on the NASCOM I adding 10 new keys for cursor control, graphic control and some extra characters. It comes ready built to the same high quality standards that are characteristic throughout.

The board is connected to the CPU board by a ribbon umbilical cord which **MUST** be connected to the correct socket first time or permanent damage will take place. Nascom comment that this could not be made mechanically foolproof for technical reasons. The keys are of a pulse transformer type which makes them very reliable and robust to the unsympathetic user. The only bad point is the incredibly poor fixing of the RESET key; ideally this should be removed from the board entirely and resited separately in the keyboard cabinet. A suitable cabinet is manufactured by VERO ELECTRONICS and is priced about £17.

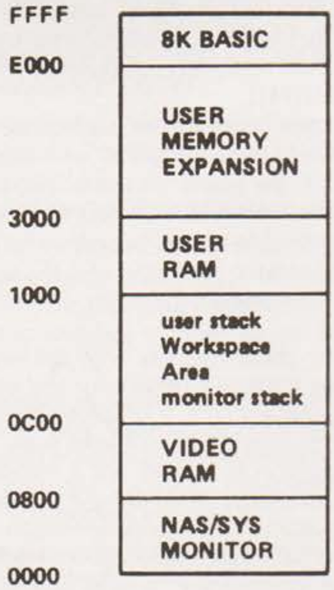
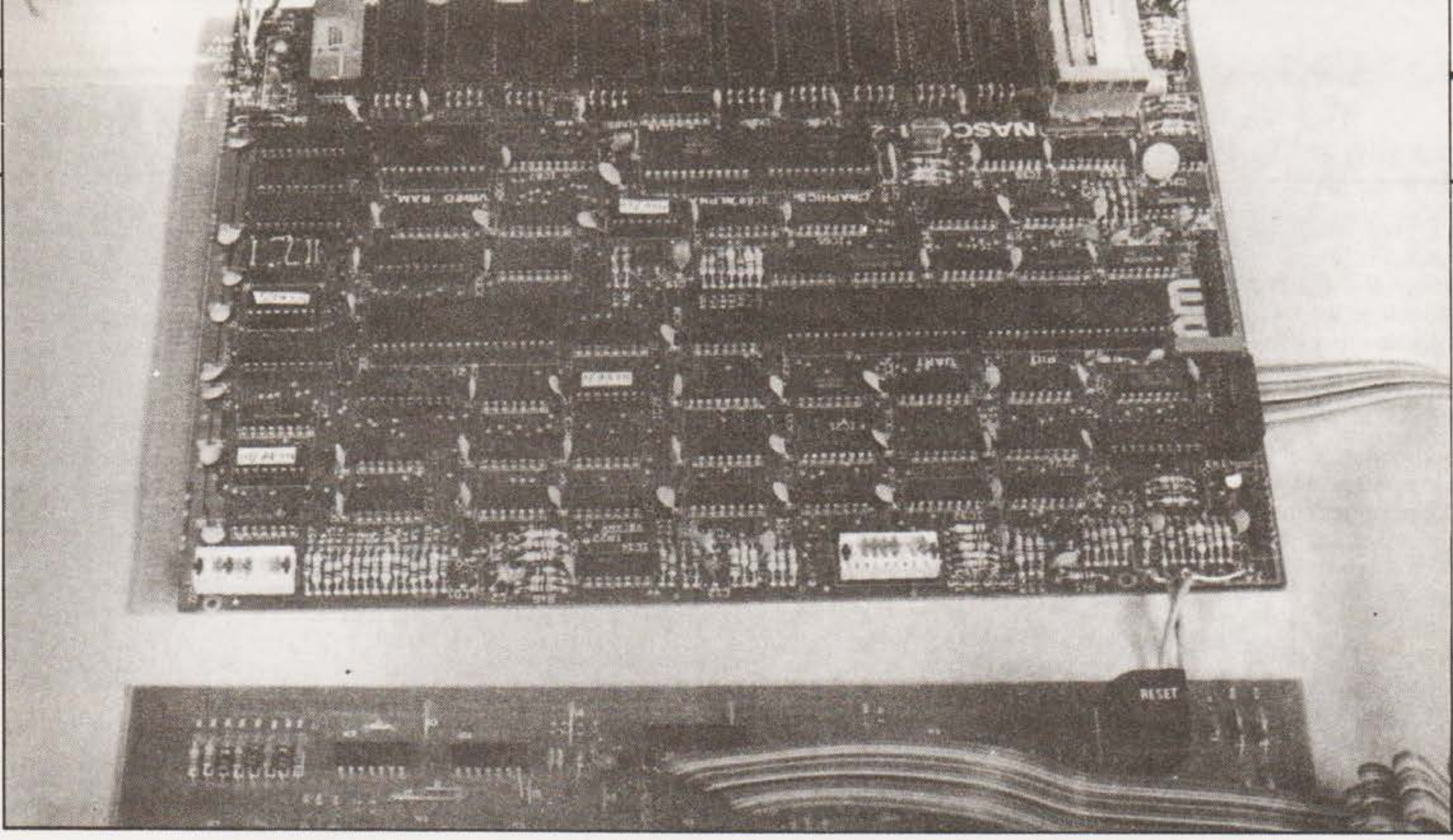


Fig 2. The typical, and recommended, memory map for the Nascom 2 with 8K of user RAM and 8K of BASIC.



Physical Realisation

The glass fibre printed circuit board has been laid-out and manufactured to industrial standards with all integrated circuits orientated so that pin 1 is at the bottom left-hand corner for horizontal positioning, or top left-hand corner for vertical positioning, thus reducing to a minimum the likelihood of ICs being plugged in incorrectly.

The assembly instructions are clear and precise with numerous check lists backed up with circuit diagrams, overlays and printed circuit drawings. A few documentation errors have come to light and these have been passed on to NASCOM for immediate correction. For the experienced constructor there should be no real problems, the secret being 'take your time and triple check everything'. Inspect every soldered connection for being 'dry' and when it comes to inserting ICs engage the assistance of wife, dad or friend to help. Be especially careful with the larger chips not to buckle any pins, and take the suggested precautions against static. Any mistakes could prove to be costly. For the novice, **DON'T START!!!**

If troubles are experienced there is adequate documentation available in the manual, which together with the unprecedented practice of including test points on the board, fault finding has been greatly simplified. Should it be necessary to call in the experts there are two possible routes; Nascom offer a flat rate repair and 'get it working' service for a cost of £35.00, or Jason Twell of the INUC in Lancaster will repair for cost + marginal handling charge.

NASCOM have chosen to stay with the single board construction which has immense benefit for the enthusiast whose main interest is in programming. It simplifies the boxing arrangements and eliminates infuriating problems from interconnecting leads. (Where expansion is carried out a motherboard can be used). The real disadvantage when using it as a dedicated system is the inherent redundancy of components and space, together with a restriction on hardware modification. However the cost of providing modular construction would probably overshadow its advantages.

Power Supply Requirements

The power supply is an optional extra with this kit and for the 3A version is priced at £29.50 + VAT. To my knowledge the 8A version is not yet available. I am assured that the 3A model, which has already been well proven in service, is adequate for all envisaged expansion. Its output ratings are: + 12V at 1A, + 5V at 3A, –5V at 0.5A and –12V at 0.5A.

The electrical design is fairly standard with all the outputs clamped to prevent voltage crossover when the mains input is removed. (If a home-brew power supply is used it must be protected in this way and full details are given in the manual). Like the main board it is constructed on a glass fibre PCB and is clearly annotated. The mains transformer is included in the price of the kit.

The Monitor

The 2K NAS/SYS monitor is supplied in one ROM package which **MUST** be strapped to memory locations 0000H - 07FFH. It is the next generation of monitor following the T4 and contains virtually all of the features offered in that package. But that is where the similarity ends. NAS/SYS is **NOT** downwards compatible and programs that have been written for use with the T2, T4 or B-BUG

Table 1

NAS-SYS RESTART INSTRUCTIONS

CODE	ASSEMBLER	NAME	FUNCTION
C7	RST 9	START	Reset computer. Initialise NAS-SYS.
CF	RST 8	RIN	Obtain an input character in the A register
D7	RST 10H	RCAL	Relative Call. Follow this code with the displacement to the routine to be called. This is similar to the Z80 Jump Relative instruction, and it allows relocatable code to be written
DF	RST 18H	SCAL	Subroutine Call. Follow this code with the number of the routine to be called. This is the method used to call the NAS-SYS routines. See the next section
E7	RST 20H	BRKPT	Store and display the program registers, then return control to NAS-SYS. This is used by the Breakpoint command.
EF	RST 28H	PRS	Output the string of characters following this code until a 0 is encountered. Then continue execution with the next instruction. This provides a very simple way of displaying a message. The A register is set to 0.
F7	RST 30H	ROUT	Output the character in the A register.
FF	RST 38H	RDEL	Wait for a period of time dependent on the value in the A register. A is set to O.

Left: The complete Nascom 2, we actually got the 4118s! The croc clips in the top right corner are connected to our video monitor, see below. The superb board layout and high packing density mean that this is not really an amateur project.
Below: Our trusty monitor connected up. The manual makes a useful shade!

monitors will have to be ammended. However the manual includes a very detailed user section and a complete machine code/ mnemonic listing. The most important differences are those of the display format, which will now write down from line 2 to line 15 before scrolling, the special control characters for N/L,B/S, clear screen etc have also been changed to conform to the ASCII standard, and the monitor subroutines have been relocated. All monitor subroutines can be accessed by using a system restart and a vector. This usage of the restart control of the Z80 dominates the philosophy of this monitor. The common routines of INPUT, OUTPUT, STRING and DELAY can all be called by a single machine code instruction and user subroutines can be called by relative addressing thereby saving one byte per CALL. The organisation of the system restarts is shown in Table 1, with the monitor commands in Table 2.

Table 2

LIST OF COMMANDS

A	xxxx	yyyy		Arithmetic - in hexadecimal
B	xxxx			Breakpoint set or cleared
C	xxxx	yyyy	zzzz	Copy - move data
E	xxxx			Execute a program
G	xxxx	yyyy	zzzz	Generate a self loading cassette tape
H				Half duplex terminal
I	xxxx	yyyy	zzzz	Intelligent copy - move data safely
J				Jump to address FFFA - BASIC cold start
K	xx			Keyboard option
L				Load a paper tape
M	xxxx			Modify or examine data
N				Normal I/O to be resumed
O	xx	yy		Output data to port
Q	xx			Query data from port
R				Read a cassette tape
S	xxxx			Single step
T	xxxx	yyyy	zzzz	Tabulate data or write a paper tape
U				User specified I/O routines activated
V				Verify cassette tape is readable
W	xxxx	yyyy		Write a cassette tape
X	xx			External serial device activated
Z				Jump to address FFFD - BASIC warm start

(D, F, P, Y commands do not exist)

Some of the commands are worthy of more detailed description. Typically the 'X' command is multipurpose and can set, from user program if required, to any one of the output options mentioned earlier. The VERIFY corrects the deficiency in the T4 and permits verification of a program being correctly stored onto tape without the possibility of correcting the program held in RAM. 'K' sets the keyboard to allow upper case or lower case direct entry from the keyboard by reversing the function of the SHIFT key, or sets the direct entry to graphics. 'I' & 'O' are very useful, they will permit the interrogation of a port, or the output control of a port direct from the keyboard.

The main additional feature of this monitor over all the others is its cursor control and screen editing facilities. Using the four directional arrows the cursor can be moved into any location on the CRT ready for character entry. Control characters can further be used to insert or delete characters, or delete whole lines, return cursor to the beginning of a line, or move it to the start of the next line. When used in conjunction with the BASIC it is most impressive.

It is perhaps unfortunate that this monitor was planned so soon after T4 that the deficiencies in the breakpoint and generate commands have been perpetuated but this in no way should detract from what is a well thought out monitor program.

The BASIC Story

For the benefit of those who are not familiar with the term BASIC I will explain that it is the name given to the high level language that was developed in America to enable a programmer to communicate with a computer in a manner that is nearer to English than machine code. It is very versatile and considerably simplifies program writing. BASIC comes in many sizes and styles and the one chosen for the NASCOM II is an enhanced 8K version that is based on the increasingly popular MICROSOFT package. There are already several computers on the market using this package so there is a wealth of published programs that can be used without too much alteration. I said that it was an enhanced 8K and this is because it utilises the monitor subroutines and particularly the cursor control, leaving space available for additional commands. It is contained on one MK36000 64K bit ROM and is normally addressed in locations E000-FFFF.

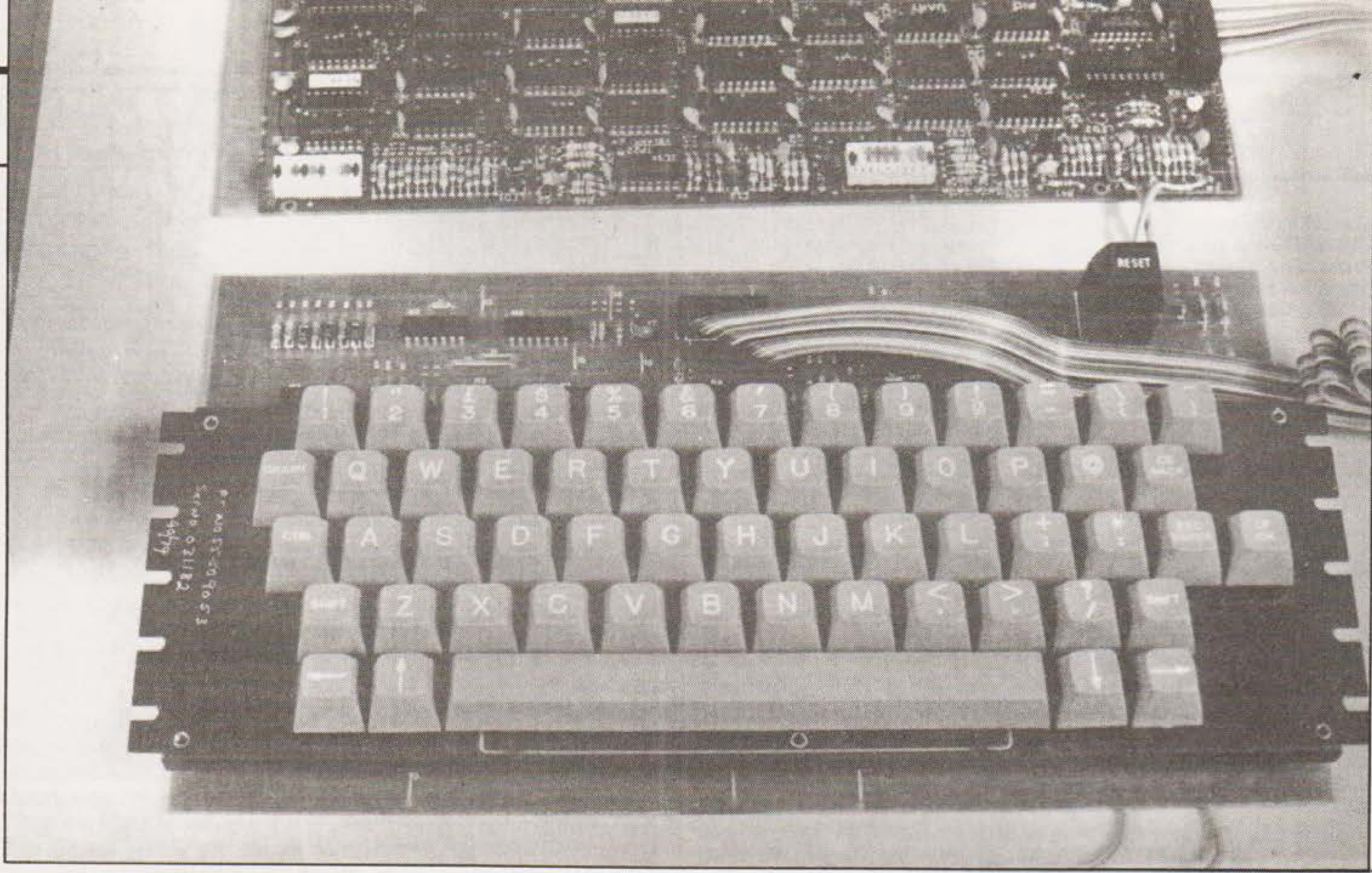
The arithmetic capability offers a 7-digit floating point accuracy in the range of 1.70141 E38 to 2.9387 E-38 with all the usual mathematical and trigonometrical functions. In addition the three extra functions of AND, OR and NOT are included.

In addition to the more common COMMANDS there is MONITOR, for passing system control directly to the NAS/SYS monitor, WIDTH, for adjusting line length on printers, and LINES for selecting the number of lines to be listed under the LIST command.

A novel feature is the first appearance of DEEK and DOKE. These are double byte versions of PEEK and POKE and allow a two byte specified number to be read from or inserted in a chosen double byte memory location. Especially useful when user machine code subroutines are used.

There are also commands designed to make the most of the memory mapped display and particularly the graphic display capability. CLS - to clear the screen, SCREEN (X,Y) - to set the cursor to a specified screen position and SET, RESET & POINT for very sophisticated picture work.

The string handling facilities are also an important feature of BASIC and these are adequately supported by all the usual functions, including 'FOLLOW ON', 'NEXT LINE' and 'NEXT ZONE' punctuation. Positioning of strings is aided by being able to specify the number of spaces to be omitted or by setting a TABulation control.



The new extended keyboard with full cursor control, RESET is still in a bad position though.

A full list of the BASIC commands and the intrinsic functions is given in Table 3. In addition there are 19 ERROR messages to assist in program debugging.

The system variables can be one or two characters wide with the first character always alphabetical and the second alphanumerical. There may be as many subscriptions as can fit onto one display line. Strings can be up to 255 characters in length and can be comprised of literals, variables or functions. As with all finite things a compromise has to be made and the more useful commands that have been omitted are RENUMBER and the MATRIX set.

On the whole a very comprehensive compliment of commands and functions and one which outclasses other BASICS of the same size, however there is one problem that ought to be highlighted, and that is the lack of an ESCAPE facility from any of the CASSETTE INPUT or OUTPUT functions. Whilst this may not appear on the surface to be very important the implications can be catastrophic. If for example the option of 'power-on reset to BASIC' has been selected then an error in the tape loading that caused the 'Finish' signal to be missed would result in a continuous load. Push reset to abort the command and you lose your entire program without a tape back-up. With the 'power-on reset to monitor' the program is still saved as long as the WARM START (Z) command is used to get back to BASIC. The two other facilities that I would have liked to see are line display immediately on error detection and the shift key + cursor control key to create a repetitive shift of the cursor.

Functional Tests

The system reviewed functioned well under both the monitor and the BASIC, although it must be said that it took some time to ascertain which options should be used despite the detailed explanations in the handbook. There are to date two known problem areas which can easily be put right. First is the dreaded 'memory plaque' which NASCOM tell me is unlikely to occur, but they do devote a whole page to describe its causes and cures. The second is a corruption of the characters on the display which manifests itself in two ways, either the whole display is shifted one character to the right revealing the left border, or segments are missing from a character. The recommended cure is shown in

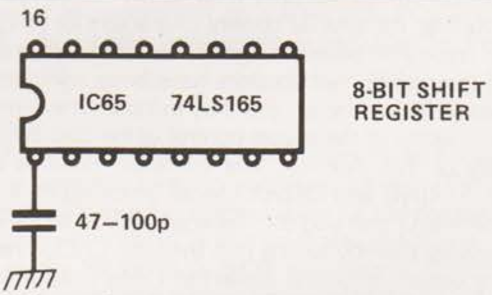


Fig 3. A quick cure for the video problem, see text, we had no trouble with our board fortunately.

Fig 3, but to be fair the review system did not have this fault and NASCOM are modifying their kits now that they are aware of the problem.

To assess the efficiency of the BASIC the BENCHMARK tests were carried out for all three operation speeds and a comparison of these against the Commodore PET are given in Table 4.All timings are in seconds.

Summing Up

At a time when the advances of technology are so rapid that the most modern equipment becomes obsolete before it even hits the production lines NASCOM can be well pleased with their achievement. This system at under £400.00 up and running presents a challenge to the rest of the market, paving the way for the next few years at least. Its flexibility allows expansion and interconnection to most innovations that can be envisaged.

As a kit this may well be the last of its kind and I look forward to the day when the computer enthusiast with little hardware experience can buy one ready built. Nascom are planning a printer and floppy disc system as back-ups and the age of connecting home systems to modems and using large mainframe central computers is probably not far off. The advantage of this machine is that it is ready and waiting for these trends.

It has not been possible to comment on everything due to the lack of time and peripheral equipment but as more people adopt this system the better proven it will be. Certainly it presents excellent value for money and in the end that's what counts.

N2 REVIEW

Table 3

COMMANDS

NEW	LIST	CONT	MONITOR	RUN
CLEAR	NULL	SCREEN	LINES	WIDTH

STATEMENTS

DEF	DIM	REM	WAIT	END
POKE	DOKE	LET	OUT	STOP
IF...THEN	ON...GO TO	ON..GO SUB	GOTO	NEXT
SET	RESET	GOSUB	RETURN	FOR

INPUT/OUTPUT

PRINT	DATA	INPUT	READ	CLS	RESTORE

OPERATIONS

=	−	+	×	/
NOT	AND	OR		

ARITHMETIC FUNCTIONS

ABS	ATN	LOG	SIN	PEEK
INP	INT	SGN	TAN	SPC
POS	RND	USR	COS	DEEK
SQR	TAB	EXP	FRE	POINT

STRING FUNCTIONS

ASC	CHR	FRE	STR	RIGHT
LEFT	LEN	MID	VAL	

CASSETTE INPUT/OUTPUT FUNCTIONS

CSAVE (array or program) CLOAD (array or program)
CLOAD? (to check a array or program is stored accurately)

Table 4.

BENCHMARKS TESTS

TEST	4MHz	4MHz+WAIT	2MHz	PET
1	1.1	1.3	2.4	1.7
2	5.4	6.7	13.2	9.9
3	11.1	14.0	28.0	18.4
4	11.7	14.9	29.5	20.4
5	12.8	16.1	31.9	21.7
6	19.4	24.7	49.21	32.5
7	27.9	35.3	69.8	50.9
8	5.2	6.5	12.9	12.3

computing today

technical book service

What Is A Microprocessor ?
2 Cassette tapes plus a 72 page book deal with many aspects of microprocessors including Binary and Hexadecimal counting, Programming etc. **£12.00**

Adams, C. BEGINNERS GUIDE TO COMPUTERS AND MICROPROCESSORS WITH PROJECTS £5.60
Understanding building programming and operating your own microcomputer.

Ahl, BASIC COMPUTER GAMES £5.25

Albrecht, B. BASIC FOR HOME COMPUTERS. A self teaching guide £5.30
Shows you how to read, write and understand basic programming language used in the new personal size microcomputers.

Albrecht B. BASIC. A self teaching guide (2nd edition) £5.30
Teach yourself the programming language BASIC. You will learn how to use the computer as a tool in home or office and you will need no special maths or science background.

Alcock, D. ILLUSTRATING BASIC £2.60
This book presents a popular and widely available language called BASIC, and explains how to write simple programs.

Altman, I. MICROPROCESSORS £10.65
Gives a general overview of the technology design ideas and explains practical applications.

Altman, L. APPLYING MICROPROCESSORS £12.00
Follow volume which takes you into the second and third generation devices.

Aspinall, D. INTRO TO MICROPROCESSORS £6.40
Explains the characteristics of the component.

Barden, W. Z-80 MICROCOMPUTER HANDBOOK £7.65

Barden, W. HOW TO BUY AND USE MINICOMPUTERS AND MICROCOMPUTERS £7.75
Discusses these smaller computers and shows how they can be used in a variety of practical and recreational tasks in the home or business.

Barden, W. HOW TO PROGRAM MICROCOMPUTERS £7.00
This book explains assembly language programming of microcomputers based on the Intel 8080, Motorola MC6800 and MOS Technology MCS6502 microprocessor.

Barna, A. INTRODUCTION TO MICROCOMPUTERS AND MICROPROCESSORS £8.15
Provides the basic knowledge required to understand microprocessor systems. Presents a fundamental discussion of many topics in both hardware and software.

Bibbero, R. J. MICROPROCESSORS IN INSTRUMENTS AND CONTROL £12.45
Introduces the background elements, paying particular regard to the dynamics and computational instrumentation required to accomplish real-time data processing tasks.

Lancaster, D. TV TYPEWRITER COOKBOOK £7.75
An in-depth coverage of tv typewriters (tv's) the only truly low cost microcomputer and small display interface.

Lancaster, D. CHEAP VIDEO COOKBOOK £6.50

Lesea, A. MICROPROCESSOR INTERFACING TECHNIQUES £8.50

Leventhal. INTRO TO MICROPROCESSORS £16.70

Lewis, T. G. MIND APPLIANCE HOME COMPUTER APPLICATIONS £4.75

Libes, S. SMALL COMPUTER SYSTEMS HANDBOOK £5.75
The Primer written for those new to the field of personal home computers.

Lippiatt. ARCHITECTURE OF SMALL COMPUTER SYSTEMS £4.35

Moody, R. FIRST BOOK OF MICROCOMPUTERS £3.85
(the home computer owners best friend).

McGlynn, D. R. MICROPROCESSORS — Technology, Architecture & Applications £9.00
This introduction to the 'computer-on-a-chip' provides a clear explanation of the important new device.

McMurran, PROGRAMMING MICROPROCESSORS £5.50
A practical programming guide that includes architecture, arithmetic/logic operations, fixed and floating point computations, data exchange with peripheral devices computers and other programming aids.

Monro, INTERACTIVE COMPUTING WITH BASIC £3.65

Nagin, P. BASIC WITH STYLE £4.00
Programming Proverbs. Principles of good programming with numerous examples to improve programming style and producing.

Ogdin SOFTWARE DESIGN FOR MICROCOMPUTERS £7.00

Ogdin. MICROCOMPUTER DESIGN £7.05

Peatman, MICROCOMPUTER BASE DESIGN £5.45

Peatman, J. B. MICROCOMPUTER BASED DESIGN £20.40
This book is intended for undergraduate courses on microprocessors.

Peckham, HANDS ON BASIC WITH A PET **£8.70**

Peckham, BASIC — A HANDS ON METHOD **£6.85**

Bursky, D. MICROCOMPUTER BOARD DATA MANUAL **£5.40**

Bursky, D. MICROPROCESSOR DATA **£5.40**

Includes complete description of the processor. Support circuits, Architecture, Software, etc.

Coan, J. S. BASIC BASIC **£7.50**

An introduction to computer programming in BASIC language.

Coan, J. S. ADVANCED BASIC **£7.30**

Applications and problems.

Ditlea, A SIMPLE GUIDE TO HOME COMPUTERS **£4.00**

Freiberger, S. CONSUMERS GUIDE TO PERSONAL COMPUTING AND MICROCOMPUTERS **£5.50**

Frenzel, L. GETTING ACQUAINTED WITH MICROPROCESSORS **£7.10**

This is an invaluable book for those who want to know more about hobby and personal computing.

Gilmore, C. M. BEGINNERS GUIDE TO MICROPROCESSORS **£4.75**

Grossworth, BEGINNERS GUIDE TO HOME COMPUTERS **£3.10**

Gosling, R. E. BEGINNING BASIC **£3.25**

Introduces BASIC to first time users.

Graham, N. MICROPROCESSOR PROGRAMMING FOR COMPUTER HOBBYISTS **£7.00**

Haviland, N. P. THE COMPULATOR BOOK **£6.20**

Building super calculators and minicomputer hardware with calculator chips.

Hartley, INTRODUCTION TO BASIC **£2.40**

Heiserman, D. L. MINIPROCESSORS FROM CALCULATORS TO COMPUTERS **£4.85**

Hilburn, J. L. MICROCOMPUTERS, MICROPROCESSORS, HARDWARE, SOFTWARE AND APPLICATIONS **£16.95**

Complete and practical introduction to the design, programming operation, uses and maintenance of modern microprocessors, their integrated circuits and other components.

Klingman, E. MICROPROCESSOR SYSTEMS DESIGN **£16.95**

Outstanding for its information on real microprocessors, this text is both an introduction and a detailed information source treating over a dozen processors, including new third generation devices. No prior knowledge of microprocessors or microelectronics is required for the reader.

Kemeny, J. G. BASIC PROGRAMMING **£6.55**

A basic text.

Korn, G. A. MICROPROCESSOR AND SMALL DIGITAL COMPUTER SYSTEMS FOR ENGINERS AND SCIENTISTS **£21.00**

This book covers the types, languages, design software and applications of microprocessors.

Duncan. MICROPROCESSOR SOFTWARE ENGINEERING **£13.50**

Rao, G. U. MICROPROCESSOR AND MICROPROCESSOR SYSTEMS **£20.50**

A completely up-to-date report on the state-of-the-art of microprocessors and microcomputers written by one of the leading experts.

Rony, P. H. THE 8080A BUGBOOK: Microcomputer Interfacing & Programming **£8.15**

The principles, concepts and applications of an 8-bit microcomputer based on the 8080 microprocessor IU chip. The emphasis is on a computer as a controller.

Scelbi. 6800 SOFTWARE GOURMET GUIDE AND COOKBOOK **£8.80**

Scelbi. 8080 SOFTWARE GOURMET GUIDE AND COOKBOOK **£8,80**

Scelbi. UNDERSTANDING MICROCOMPUTERS **£8.60**

Gives the fundamental concepts of virtually all microcomputers.

Spencer, GAME PLAYING WITH BASIC **£4.70**

Schoman, K. THE BASIC WORKBOOK **£3.70**

Creative techniques for beginning programmers.

Sirion, D. BASIC FROM THE GROUND UP **£6.00**

Soucek, B. MICROPROCESSORS AND MICROCOMPUTERS **£19.00**

Here is a description of the applications programming and interfacing techniques common to all microprocessors.

Spracklen, D. SARGON **£9.75**

A computer chess program in Z-80 assembly language.

Titus, MICROCOMPUTER ANALOGUE CONVERTER **£7.45**

Titus, 8080/8085 SOFTWARE DESIGN **£7.45**

Tracton. 57 PRACTICAL PROGRAMS & GAMES IN BASIC **£6.40**

Programs for everything from Space war games to Blackjack.

Waite. M. MICROCOMPUTER PRIMER **£6.25**

Waite, YOUR OWN COMPUTER **£1.50**

Introduces the beginner to the basic principles of the microcomputer.

Ward. MICROPROCESSOR / MICROPROGRAMMING HANDBOOK **£6.00**

Authoritative practical guide to microprocessor construction programming and applications.

Veronis. MICROPROCESSOR **£12.85**

Zaks, R. INTRODUCTION TO PERSONAL AND BUSINESS COMPUTING **£8.50**

Zaks, R. MICROPROCESSORS FROM CHIPS TO SYSTEMS **£7.50**

Note that all prices include postage and packing. Please make cheques, etc, payable to Computing Today Book Service (Payment in U.K. currency only please) and send to:

Computing Today Book Service,
P.O. Box 79, Maidenhead, Berks.

Mr I.J. Nicolle

MISSILE SHOOT

The following program is designed to be run on the Mk14. The object of the game is to launch all eight missiles, if you launch a missile into a space already occupied by another one you simply shoot the first one down and replace it with the second.

0F1D controls the speed of the missiles, to start the program enter 0F12 GO and launch your missiles using the GO key. The program takes a total of 45H bytes.

0F12	CA 0D	0F29	1E	0F3E	E4 FF
0F14	35	0F2A	C8 E4	0F40	98 08
0F15	C4 00	0F2C	94 04	0F42	C8 CE
0F17	31	0F2E	C4 88	0F44	C0 CA
0F18	C4 02	0F30	90 02	0F46	E4 80
0F1A	C8 F4	0F32	C4 00	0F48	C8 C6
0F1C	C4 10	0F34	C9 80	0F4A	40 03
0F1E	C8 F1	0F36	8F 09	0F4C	FC 01
0F20	C4 00	0F38	C0 D8	0F4E	94 D6
0F22	C8 EE	0F3A	9C 0E	0F50	B8 BF
0F24	C4 08	0F3C	C1 80	0F52	98 C8
0F26	01			0F54	C4 07
0F27	C0 E7			0F56	90 CE

Stephen Draper

SAFEBREAK GAME

Safebreak is a simple game of logic and skill which is played against the computer. The computer generates a random code consisting of five variables in the range 0 to 30 which must be found by asking the computer a limited number of allowable questions. Legal questions are statements (EG. A=B) which the computer will answer either yes or no. Any statement which uses any of the comparitive operators =, > or < found) or any number in the range +–32,767, is allowable. However, only one comparitive operator may be used in any one statement:

EG–A+B*27=C is allowable, whereas

A=B=C is not.

When the player is ready to make a guess he must tell the computer so when it asks and then type in what he thinks the variables (A–E) are.

The game can be made easier or harder by altering the limit number of questions (Z).

```
READY.

 5 PRINT"SAFEBREAK"
 10 LET A=RND(30)
 15 LET B=RND(30)
 20 LET C=RND(30)
 25 LET D=RND(30)
 30 LET E=RND(30)
 35 INPUT"DIFFICULTY FACTOR?"Z
 40 FOR N=1 TO Z
 45 INPUT"DO YOU WISH TO HAVE A GUESS,YES OR NO"Q$
 50 IF Q$="YES" THEN 185
 55 INPUT"YOUR STATEMENT PLEASE"M$
 60 LET L=LEN(M$)
 65 FOR X=1 TO L
 70 S$=MID$(M$,X,1)
 75 IF (S$=">")OR(S$="<")OR(S$="=") THEN 95
 80 NEXT X
 85 PRINT"INVALID STATEMENT,TRY AGAIN"
 90 GOTO 55
 95 LET W$=LEFT$(M$,X-1)
 100 LET V=VAL(W$)
 105 LET W$=RIGHT$(M$,L-X)
 110 LET P=VAL(W$)
 115 IF S$="=" THEN 135
 120 IF S$=">" THEN 150
 125 IF S$="<" THEN 160
 130 GOTO 85
 135 IF V=P THEN PRINT"YES":GOTO 170
 140 PRINT"NO"
 145 GOTO 170
 150 IF V>P THEN PRINT"YES":GOTO 170
 155 GOTO 140
 160 IF V<P THEN PRINT"YES":GOTO 170
 165 GOTO 140
 170 NEXT N
 175 PRINT"YOU HAVE HAD"Z"GOES THEREFORE YOU HAVE LOST"
 180 GOTO 220
 185 INPUT"A"F
 190 INPUT"B"G
 195 INPUT"C"H
 200 INPUT"D"I
 205 INPUT"E"J
 210 IF (A=F)AND(B=G)AND(C=H)AND(D=I)AND(E=J) THEN 235
 215 PRINT"WRONG,YOU HAVE LOST"
 220 INPUT"DO YOU WISH TO CONTINUE?"T$
 225 IF T$="YES" THEN 10
 230 END
 235 PRINT"CORRECT,YOU HAVE WON"
 245 GOTO 220
READY.
```

Paul B. Kaufman

SCAMPSCOPE ROUTINE

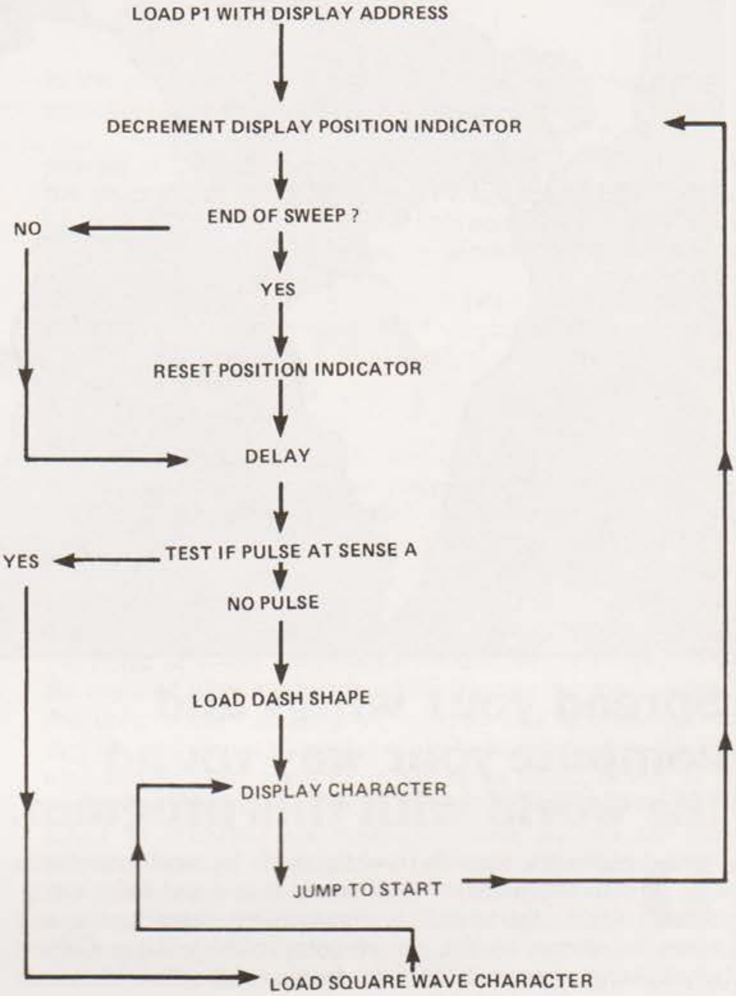
This program enables an Mk14 or similar SC/MP based system to perform as a simple Digital oscilloscope. Many recently published programs for SC/MP machines have tended to fall into one of two categories: 1) Simple games, 2) Hardware test routines. This program is intended to add a third category; Genuinely useful programs.

Program Function

Pointer Register 1 is initialised with the address of the display , '0D00 (see listings). The display position indicator is decremented (SHOW +1) and checked to see if it is –1 ('FF), if it is, it is set to '09, otherwise processing continues from SYNC. There is a short delay, then the Status Register is tested for Sense A going high. A 'square wave' shaped character is stored in the display if Sense A is high, if not then a 'dash' is displayed instead. The program then loops back to BEGIN and this processing is repeated. Thus while the display characters are being scanned from left to right, a high pulse at Sense A will cause a 'square wave' to appear on the display. If the speed of scan matches the speed that Sense A is being toggled, the display will appear to stand still. The rate of scan is determined by the delay constant at SYNC, the lower the constant the faster is the scan.

Using The Program

Load the program into any free area of memory e.g. 'F12 and 'G0' at this address. Immediately a line will show on the display, by increasing the value at SYNC the motion can be observed. If a logic pulse (max 5 V) is sent to Sense A a square wave will be displayed for its duration. If a train of pulses is sent e.g. A square wave signal generator, the waveform can easily be observed up to about 2 kHz. If the cassette interface is used it is possible to play back a tape and watch the waveform as each character is read in. Thus the Scamposcope can be used as a very useful logic probe. The program can be easily modified to freeze the display after one pulse, or with a few diodes connected to Sense A and Sense B, observe pulses of either polarity.



SCAMPOSCOPE

```
F12   C40D             LDI    '0D      ; Load Display address
F14   35               XPAH   (1)
F15   C400             LDI    '00
F17   31               XPAL   (1)
F18   B813             DLD    SHOW +1
F1A   E4FF    BEGIN    XRI    'FF      ; End of sweep ?
F1C   9C04             JNZ    SYNC     ; if not go to SYNC
F1E   C409             LDI    '09
F20   C80B             ST     SHOW +1  ; Reset sweep pointer
F22   8F22    SYNC     DLY    '22      ; Delay
F24   06               CSA
F25   D430             ANI    '30      ; Test Sense A
F27   9C06             JNZ    PULSE    ; Pulse detected ?
F29   C440             LDI    '40      ; Display 'dash'
F2B   C900    SHOW     ST     (1)
F2D   90E9             JMP    BEGIN    ; Start again
F2F   C423    PULSE    LDI    '23      ; Display 'square wave'
F31   90F8             JMP    SHOW
```

E.A. Parr.

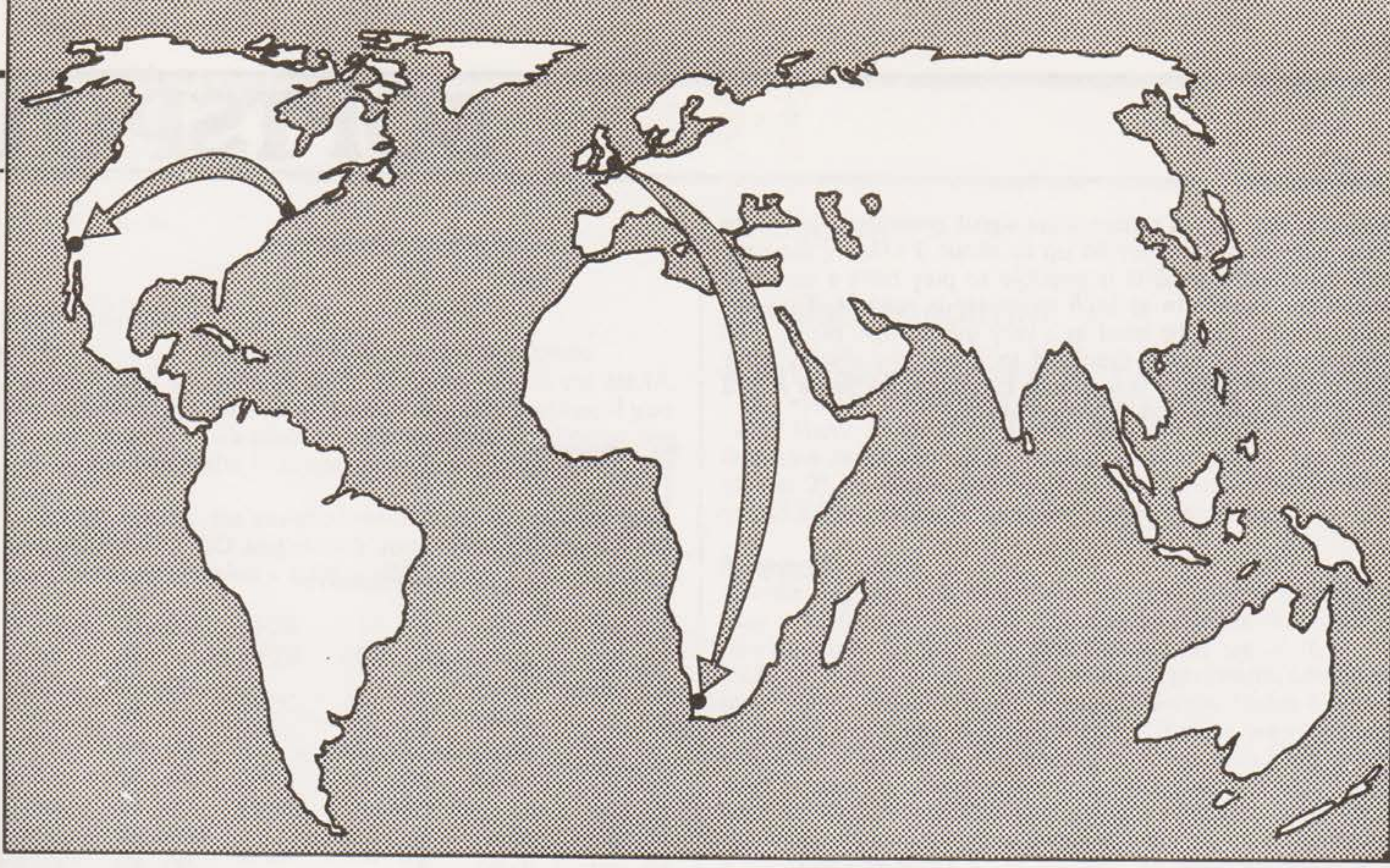
REM FOR TREKKIES

This modification to the Star Trek program (Oct '79) adds a command "8" to the command functions. It provides a history and map of the parts of the Galaxy explored to date. The display is an 8 by 8 array using the same format as command 2 (Long range scan) for explored regions. Unexplored regions are displayed as ***.

In the initial set up of the Galaxy 1000 is added to each sector. When a sector is entered or scanned the 1000 is removed from the corresponding array element. Print out of the galactic map then simply involves a simple test to see if the sector array element is or 1000.

Program Modifications

```
 140 @(I)=X*100+Y*10+Z+1000                        :— Initial set up increased by 1000
 200 IF @(Q) > 1000 @(Q)=@(Q)—1000                 :— Sector has 1000 removed if entered
 205 Z=@(Q)                                        :— Displaced instruction
 610 IF(B > 8)+(B < 1) GOTO 600                    :— Command range increased 1 to 9
2027 IF @(U) > 1000 @(U)=@(U)—1000                 :— Sector has 1000 removed if scanned
3070 GOSUB 8002                                    :— Changed destination
8000 GOTO 9000                                     :— Jump to History Print Out
8002 Z=H—F                                         :— Displaced instruction
9000 PRINT "MAP OF EXPLORED GALAXY AT
          STAR DATE",T
9010 FOR I=0 TO 7
9020 FOR J=0 TO 7
9030 IF @(8*I+J) > 1000 PRINT " *** ",; GOTO 9050  :— Prints *** if unexplored
9040 PRINT #4, @(8*I+J)                            :— Prints data if explored
9050 NEXT J
9060 PRINT
9070 NEXT I
9080 GOTO 605
```



Spread your wings and compute your way round the world with this program.

Despite the inroads recently made by small computers into the fields of synthesized music and voice recognition, this article is unrelated to these topics, and poses no threat to the record companies or Patsy Gallant whose song prompted the title. Instead this article describes a trigonometric calculation to determine the shortest distance between the two places in the title – or any other two places on the earth's surface. This will be of special interest to readers who fly their own aircraft, or who are planning to build their own intercontinental missiles.

The Program Options

The program DISTANCE is written in an elementary sub-set of BASIC which should be implmented without difficulty on all mainframes and microcomputers which support floating point BASIC. Informative messages are printed out at all stages to prompt the input of data, and as far as possible data are checked to ensure that they are physically possible.

First the program asks if you prefer to work in degrees, minutes and seconds, or in decimal degrees. Next you are asked for the name of the first place, followed by its latitude, and provided it is not on the equator whether it lies in the northern or southern hemisphere. Then, provided you have not chosen the North or South Pole you are asked for the longitude and if necessary whether this is east or west of the Greenwich meridian. The place name and position of the second place are then input in a similar manner.

```
10 DIM A$[20], B$[20], D$[10], Q$[10], L[4]
20 PRINT "PROGRAM TO CALCULATE THE SHORTEST DISTANCE BETWEEN TWO POINTS"
30 PRINT "ON THE EARTH."
40 PRINT
50 PRINT "WOULD YOU LIKE TO WORK IN DEGREES, MINUTES AND SECONDS OR"
60 PRINT "DECIMAL DEGREES. TYPE DMS OR DD AND PRESS RETURN."
70 INPUT D$
80 IF D$ = "DMS" THEN 120
90 IF D$ = "DD" THEN 120
100 PRINT "REPLY '"; D$; "' NOT UNDERSTOOD. RE-TYPE DMS OR DD"
110 GOTO 70
120 LET T = 0
130 FOR I = 1 TO 2
140 PRINT "TYPE IN THE NAME OF PLACE"; I; "AND PRESS RETURN."
150 INPUT A$
160 PRINT "TYPE IN THE LATITUDE OF "; A$
170 GOSUB 670
180 IF L[2 * I - 1] = 0 THEN 270
190 PRINT "TYPE NORTH OR SOUTH AND PRESS RETURN"
200 INPUT Q$
210 IF Q$ = "NORTH" THEN 270
220 IF Q$ = "SOUTH" THEN 260
230 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD"
240 PRINT "RE-";
250 GOTO 190
260 LET L[2 * I - 1] = -L[2 * I - 1]
270 LET L[2 * I] = 0
280 IF ABS(L[2 * I - 1]) = 90 * 2 * 3.14159 / 360 THEN 390
290 PRINT "TYPE IN THE LONGITUDE OF "; A$
300 GOSUB 660
310 IF L[2 * I] * (L[2 * I] - 3.14159) = 0 THEN 400
320 PRINT "TYPE EAST OR WEST AND PRESS RETURN"
330 INPUT Q$
340 IF Q$ = "WEST" THEN 400
350 IF Q$ = "EAST" THEN 390
360 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD"
370 PRINT "RE-";
380 GOTO 320
390 LET L[2 * I] = -L[2 * I]
400 IF I = 2 THEN 420
410 LET B$ = A$
420 NEXT I
430 DEF FNC(X) = ATN((SQR(1 - X * X) / X)) * 360 / (2 * 3.14159)
440 LET D1 = SIN(L[1]) * SIN(L[3])
450 LET D2 = COS(L[1]) * COS(L[3]) * COS((L[2] - L[4]))
460 IF D1 + D2 <> 0 THEN 490
470 LET D = 60 * 90
480 GOTO 520
490 LET D = 60 * FNC(D1 + D2)
500 IF D1 + D2 > 0 THEN 520
510 LET D = 60 * (180 - FNC(-D1 - D2))
520 PRINT
530 PRINT
540 PRINT "THE GREAT CIRCLE DISTANCE BETWEEN "; B$; " AND "
550 PRINT A$; " IS"; INT(D + .5); "NAUTICAL MILES."
560 PRINT
570 PRINT
580 PRINT "WOULD YOU LIKE ANOTHER RUN (YES/NO)"
590 INPUT Q$
600 IF Q$ = "YES" THEN 130
610 IF Q$ = "NO" THEN 640
620 PRINT "REPLY '"; Q$; "' NOT UNDERSTOOD. RE-TYPE YES OR NO."
630 GOTO 590
640 PRINT "JOB COMPLETED"
650 STOP
660 REM *** SUBROUTINE TO CHECK INPUT DATA
670 LET M = 90
680 IF D$ = "DMS" THEN 740
690 INPUT X
700 IF X < 0 THEN 720
710 IF X <= M THEN 1020
720 PRINT "THIS VALUE SHOULD BE BETWEEN 0 AND"; M; ", RETYPE CORRECTLY"
730 GOTO 690
740 IF T = 0 THEN 770
750 PRINT "DEGREES,MINUTES,SECONDS"
760 GOTO 800
770 PRINT "AS THE NUMBER OF DEGREES, A COMMA, THE NUMBER OF MINUTES,"
780 PRINT "A COMMA, AND THE NUMBER OF SECONDS THEN PRESS RETURN"
790 LET T = 1
800 INPUT X1, X2, X3
810 IF X1 <> INT(X1) THEN 850
820 IF X1 < 0 THEN 850
830 IF X1 < M THEN 890
840 IF ABS(X1 - M) + ABS(X2) + ABS(X3) = 0 THEN 1010
850 PRINT "THE NUMBER OF DEGREES MUST BE A WHOLE NUMBER BETWEEN 0 AND"; M
860 PRINT "RETYPE THE NUMBER OF DEGREES CORRECTLY"
870 INPUT X1
880 GOTO 810
890 IF X2 <> INT(X2) THEN 920
900 IF X2 < 0 THEN 920
910 IF X2 <= 60 THEN 960
920 PRINT "THE NUMBER OF MINUTES MUST BE A WHOLE NUMBER BETWEEN 0 AND 60"
```

FROM N.Y. TO L.A.

John D. Lee and Timothy D. Lee.

The Theory Of Distance

The form of trigonometry familiar to most people involves right angled triangles in two dimensions. At least one published program for calculating distances uses this approach, but this takes scientific thinking back to the days of the ancient Greeks who believed that the earth was flat! Clearly this is an unacceptable approximation unless the distances involved are so small that the curvature of the earth has an insignificant effect. This program makes the assumption that the earth is spherical. Whilst it is known that the earth is slightly flattened at the poles (equatorial radius = 6378.2 km and polar radius = 6356.8 km) the difference in radii of 21.4 km accounts for a maximum error of one third of one percent. Spherical trigonometry is more complicated, and using this the shortest distance between two points is no longer a straight line but is the distance along the great circle which passes through them. (A great circle is any circle round the earth whose centre is coincident with the centre of the earth). The equation used calculates the minimum angular separation of the two places measured from the centre of the earth. Since one minute of angle corresponds to one nautical mile, the angle can easily be converted into a 'distance'. Note that this method avoids even the maximum one third of a percent error mentioned above! The implication of this is that a nautical mile is not a constant, and an American nautical mile is about four feet smaller than a British one! The extreme distances are 6045.6 feet per minute of latitude at the equator, and 6108.1 feet at the poles.

The equation to calculate the distance is:

distance = 60 arc cos $[\sin\theta_1 \sin\theta_2 + \cos\theta_1 \cos\theta_2 \cos(\phi_1 - \phi_2)]$ where θ_1 and θ_2 are the latitudes of places one and two respectively and ϕ_1 and ϕ_2 are their longitudes. The program empirically assigns + and − signs to latitudes which are north and south respectively, and to longitudes which are west and east respectively. Trigonometric functions provided on computers require the angles to be measured in radians, so the values of θ and ϕ are converted after input.

Since many compilers and interpreters do not provide an arc cos function, its use is avoided by using the art tan function which is generally available.

$$\text{arc cos } x = \text{arc tan } \left[\frac{\sqrt{1-x^2}}{x}\right]$$

In the program this is done by defining FNC, which also converts the result from radians to degrees.

Some interpreters – notably the SWTP 8K BASIC provide no inverse trigonometric functions. To implement the program on such a machine, the arc cosine function may be evaluated by summing the polynomial expression given below taking sufficient terms to provide the required accuracy.

$$\text{arc cos } x = \frac{\pi}{2} - \left[x + \frac{1}{2*3}x^3 + \frac{1*3}{2*4*5}x^5 + \frac{1*3*5}{2*4*6*8*9}x^7 + \frac{1*3*5*7}{2*4*6*8*9}x^9 + \dots\right]$$

For angles greater than 45° the accuracy may be improved by evaluating arc sin x (by omitting $\pi/2$ – from the above equation) and using $\cos^2 + \sin^2 = 1$ hence arc cos x = arc sin $(\sqrt{1-x^2})$.

Internal Checking

The program includes a number of checks. Latitudes outside the range of 0–90° and longitudes outside the range 0–180° are rejected with a message asking they be input correctly. If the units chosen are degrees, minutes and seconds then the degrees and minutes must be whole numbers, and the minutes and seconds must be in the range 0–60. If any of the replies to questions DMS/DD, NORTH/SOUTH, WEST/EAST or YES/NO are mistyped, these are rejected and a suitable message requests their re-input. Internal checks are also performed to prevent failure through dividing by zero when attempting to take the arc cos of zero – corresponding to an angle of 90°. Evaluating the arc cos by using the arc tan may give rise to an angle outside the range 0–180° if the value of x is negative, and to avoid this negative x values are handled differently.

Finally the program calculates and prints the distance between the two points in nautical miles, and asks whether you would like another run or wish to finish.

```
930 PRINT "RETYPE THE NUMBER OF MINUTES CORRECTLY"
940 INPUT X2
950 GOTO 890
960 IF X3 < 0 THEN 980
970 IF X3 <= 60 THEN 1010
980 PRINT "RETYPE THE NUMBER OF SECONDS IN THE RANGE 0 TO 60"
990 INPUT X3
1000 GOTO 960
1010 LET X = X1 + X2 / 60 + X3 / 3600
1020 LET L(2 * I - 2 + M / 90) = X * 2 * 3.14159 / 360
1030 LET M = 180
1040 RETURN
1050 END
PROGRAM TO CALCULATE THE SHORTEST DISTANCE BETWEEN TWO POINTS
ON THE EARTH.
WOULD YOU LIKE TO WORK IN DEGREES, MINUTES AND SECONDS OR
DECIMAL DEGREES. TYPE DMS OR DD AND PRESS RETURN.
? DMS
TYPE IN THE NAME OF PLACE 1 AND PRESS RETURN.
? L.A.
TYPE IN THE LATITUDE OF L.A.
AS THE NUMBER OF DEGREES, A COMMA, THE NUMBER OF MINUTES,
A COMMA, AND THE NUMBER OF SECONDS THEN PRESS RETURN
? 33,50,0
TYPE NORTH OR SOUTH AND PRESS RETURN
? NORTH
TYPE IN THE LONGITUDE OF L.A.
DEGREES,MINUTES,SECONDS
? 118,22,0
TYPE EAST OR WEST AND PRESS RETURN
? WEST
TYPE IN THE NAME OF PLACE 2 AND PRESS RETURN.
? NEW YORK
TYPE IN THE LATITUDE OF NEW YORK
DEGREES,MINUTES,SECONDS
? 40,45,0
TYPE NORTH OR SOUTH AND PRESS RETURN
? NORTH
TYPE IN THE LONGITUDE OF NEW YORK
DEGREES,MINUTES,SECONDS
? 74,0,0
TYPE EAST OR WEST AND PRESS RETURN
? WEST

THE GREAT CIRCLE DISTANCE BETWEEN L.A. AND
NEW YORK IS 2135 NAUTICAL MILES.

WOULD YOU LIKE ANOTHER RUN (YES/NO)
? YES
TYPE IN THE NAME OF PLACE 1 AND PRESS RETURN.
? LONDON
TYPE IN THE LATITUDE OF LONDON
DEGREES,MINUTES,SECONDS
? 51,30,0
TYPE NORTH OR SOUTH AND PRESS RETURN
? NORTH
TYPE IN THE LONGITUDE OF LONDON
DEGREES,MINUTES,SECONDS
? 0,5,0
TYPE EAST OR WEST AND PRESS RETURN
? WEST
TYPE IN THE NAME OF PLACE 2 AND PRESS RETURN.
? CAPE TOWN
TYPE IN THE LATITUDE OF CAPE TOWN
DEGREES,MINUTES,SECONDS
? 33,59,0
TYPE NORTH OR SOUTH AND PRESS RETURN
? SOUTH
TYPE IN THE LONGITUDE OF CAPE TOWN
DEGREES,MINUTES,SECONDS
? 18,30,0
TYPE EAST OR WEST AND PRESS RETURN
? EAST

THE GREAT CIRCLE DISTANCE BETWEEN LONDON AND
CAPE TOWN IS 5222 NAUTICAL MILES.

WOULD YOU LIKE ANOTHER RUN (YES/NO)
? NO
JOB COMPLETED
```

NASCOM COLOUR GRAPHICS BOARD

only £22.50 in kit form

*Runs on minimum Nascom under T2, T4 or B-Bug.
*Genuine PAL encoder.
*8 background and 8 foreground colours.
*3072 addressable colour cells.
*Alpha numerics and colour graphics can be mixed.
*Simple to build, connect and implement.
*Complete with construction and software notes.
For use with a standard unmodified colour TV.

Just send £22.50 inclusive for a complete kit to:

INTRACEPT ELECTRONICS LTD.

203 Picton Road, Liverpool. L15 4LG
Tel: 051-733-3042

AVALON SOFTWARE
of England

ZEN

a Z80 Editor Assembler

Zen is a complete system for the production and Assembly of Z80 code. Several versions are available. All are the same price and consist of an Object cassette, a User manual, a Symbol Sort and a complete Listing. Zen is designed with the small system in mind. It is fast, convenient and makes no special demands on the system console or cassette recorder.

- Full set of Editor commands, including string search.
- Compact source files, no line numbers. Free format, no tabbing.
- Named source files to cassette in fast binary format.
- Standard ZILOG syntax assembler, 4000 lines/minute.
- Decimal, hex, octal constants & command parameters
- Expressions using math operators – add sub. mult div, AND, OR.
- Pseudo-ops for strings, bytes & words.
- Symbols any length, symbol table easily extended.
- LIST and SYMBOL SORT to Video or printer.
- OBJECT to cassette or memory.

NASCOM 1, origin 0C50H

- Runs under Nasbug, etc in 3·5k.
- Object to cassette in DUMP format.

STANDARD, origin 100H

- Runs under any Monitor with character I/O.
- No special Terminal needed, CR only control character used.
- Object to cassette in INTEL HEX format.

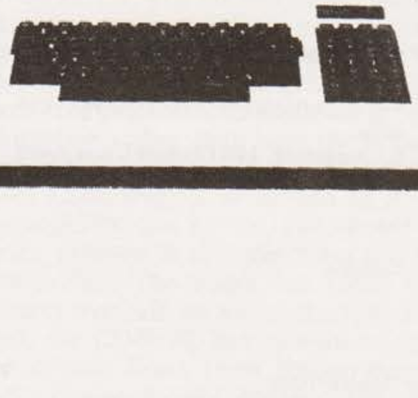
TRS-80 LEVEL II, origin 4300H

- Includes MONITOR/DEBUGGER superior to TBUG.
- All commands on one level, no reloading.
- SINGLE BYTE breakpoint.
- ALL Z80 user registers visible.
- Display/Modify memory in Decimal, hex, octal or ASCII
- Runs in 4·5k.
- Object to cassette in LEVEL II format.

ZEN costs £14·50 inclusive of Carriage & VAT. **ZEN** is **BRITISH**.

distributor: NEWBEAR COMPUTER STORE
40 Bartholomew Street
Newbury
Berks.

We've pioneered the west!



We've opened up the west of London and if you're local to the Ealing area that can only be good news for you.

We stock all the big names – Commodore, PET, Apple and Nascom.

We have the knowledge and expertise to provide a software package or business system that's right for you plus a comprehensive maintenance and engineering back-up service.

Call in for more information or a demonstration.
(We also sell a wide range of tapes and books.)

Adda Computers, 17–19 The Broadway, Ealing, London W.5.
(between W.H. Smiths and Burtons)
Telephone 01-579 5845
Open 09.00–18.00 Monday to Friday;
10.00–16.00 Saturdays.

adda we add up to a great deal.

CT MARKET PLACE

1 CLOCK RADIO

How about a round clock radio which can double as a very smart desk clock — as we can testify!
To time, rotate one end of the cylinder to display the frequency selected. Most of the functions are controlled by a push-button panel and the display is a large, clear LCD affair.
Made by Hanimex, the battery clock radio comes in white, white or white. It will lull you to sleep and then turn itself off an hour later and waken you to the sound of Radio 1, or music if you prefer.

£17.95

To: CLOCK RADIO offer
CT Magazine, 145 Charing Cross Road
London, WC2H 0EE

Name . . . .

Address . . . .

. . . .

2 DIGITAL ALARM

This mains-only Hanimex alarm has a large 12-hour display incorporating AM/PM and alarm set indicators. You can have a dim or bright display at the touch of a switch. Fast and slow setting buttons make time setting simplicity itself. You can forget about knocking these accidentally in the morning scramble to turn off the alarm, as a locking switch is fitted under the clock. A 9-minute snooze switch completes the list of all mod. clock cons.

£10.60

To: DIGITAL ALARM offer
CT Magazine, 145 Charing Cross Road
London, WC2H 0EE

Name . . . .

Address . . . .

. . . .

3 LCD CHRONO

Our Chrono comes complete with a high grade adjustable metal strap and is fully guaranteed.
The LCD display shows seconds as well as hours and minutes. Press a button and you get the date and day of the week.
Press another button and you have an accurate stopwatch with hundredths of seconds displayed, giving the time up to an hour. There's a lap time facility, too — and of course a back light.

£11.95

To: LCD WATCH offer
CT Magazine, 145 Charing Cross Road
London, WC2H 0EE

Name . . . .

Address . . . .

. . . .

4 LCD ALARM CHRONO

This is no ordinary watch. It's a slim, multi-function, dual time LCD alarm chronograph.
This model will show hours, minutes, seconds, date, day of the week, stopwatch, split time, alarm and alternate dual time zone — not all at once of course. There's a night light, too.
Hours, minutes, seconds and day of the week are displayed continuously, while the date will appear at the touch of a button. The alarm is beefy enough to wake you up in the morning and get you to work on time (or wake you up when it's time to go home).

£16.95

To: ALARM/CHRONO LCD WATCH offer
CT Magazine, 145 Charing Cross Road
London, WC2H 0EE

Name . . . .

Address . . . .

. . . .

All prices include 15% VAT and postage

Examples of Marketplace offers can be seen at our Charing Cross Road offices.
Please mark your envelope with the offer that you want and order separately from offers shown elsewhere in CT.

Owen Bishop

MICROLINK

The first part of our new series on connecting your machine to the outside world

In this series we are looking at various practical ways of interfacing a simple microprocessor system, such as Acorn or the Mk-14, with other electronic devices and with the environment in general. Emphasis is on applications in the home and office. Interfacing concerns both software and hardware. For the software enthusiast, who knows only a little about electronics, the series includes full constructional details of the interfaces and suggests how they can be used. For the hardware expert, the series explains in full the short simple programs needed to operate the interfaces, and suggests how to modify the programs to suit individual circumstances.

Outputs To The World

The stock excuse for buying a microprocessor system is that 'it can be used to control the central heating' – implying that this will bring about enormous savings of fuel, easily covering the initial outlay on the system. Leaving aside the question as to whether anyone will be able to afford to run a central heating system at all if fuel prices continue to rise, there still remains the problem – how actually do you go about connecting an MPU to an oil-fired boiler? This is not a question that will be answered here – too much depends on the exact nature of your heating installation – but there are lots of other devices around the home that can easily be put under microprocessor control. When you have played around with the programs in the manufacturer's handbook and eventually have become bored with shooting down ducks, then is the time to make the system do something *useful*, for a change.

In order to do something useful the MPU must know when there is something useful to be done. It needs an *input*. This can be by way of the keyboard, as you enter instructions manually, or by an input interface which operates automatically. Several input interfaces will be described in later parts of this series, including interfaces responsive to electrical signals, to sound, to light intensity and to temperature. Having been informed that there is something useful to be done (such as 'turn on the porch light') the MPU must then have some means of taking the necessary action. It needs an output interface. This is the subject of this month's article.

LED Interface

This may seem somewhat trivial but, if you can get the MPU to turn on a LED, you are more than half-way toward getting it to turn on the porch light, the central heating boiler or even the Blackpool Illuminations. So let's keep to LEDs for the moment, for the LED interface illustrates the principles fully and it is preferable to work out programs first using the LED interface rather than have the house lights flashing on and off in apparently uncontrollable fashion. The interface has three LEDs (Fig.1), which can be all of the same colour or, if you prefer, can be red, yellow and green. A good programming exercise is to make them run through the traffic-lights sequence. The board has room for more LEDs and other items that will be added at a later stage. Only one IC is required, the CD4050, hex non-inverting buffer. The LEDs can be driven direct from the outputs, without need of resistors. Power supply comes from the microprocessor board. Note that we are using only 3 of the 6 buffers, keeping the other 3 in reserve for use later. In the meantime their inputs *must* be tied to the positive rail (or to the negative rail – but the main point is that they must not be left unconnected). The layout of the board is shown in Fig.2. It is preferable to use a socket so that a different IC could possibly be used later instead of the 4050 (with suitable changes in the wiring, of course). The 4050 is very unusual in that the positive supply goes to pin 1. Input to to the interface is by a 5-pin PCB plug; a second plug is provided so that further devices can be connected. Then the LEDs indicate the state of each of the 3 output lines that are controlling the attached device. The components are restricted to the front left-hand region of the board so as to leave room for additions later.

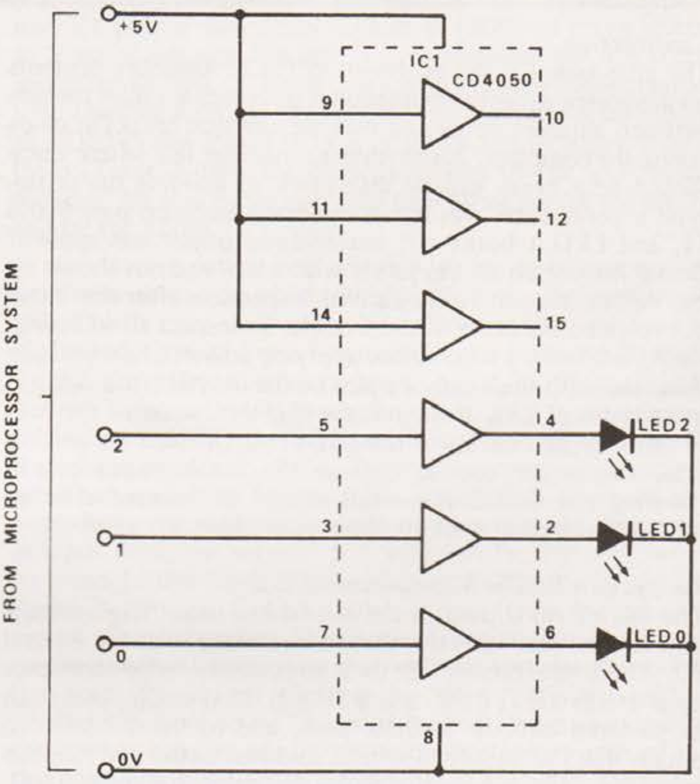
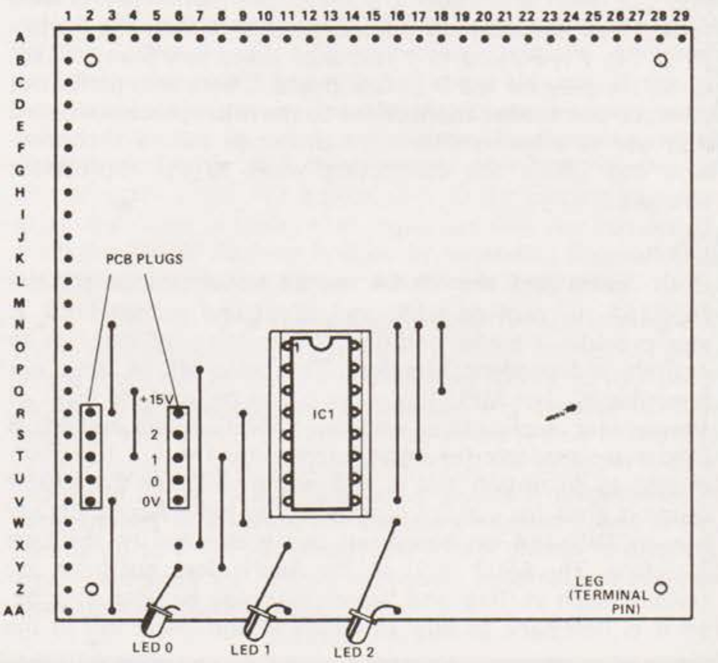


Fig. 1. The circuit diagram for the LED port.

Fig. 2. The veroboard layout for the circuit.



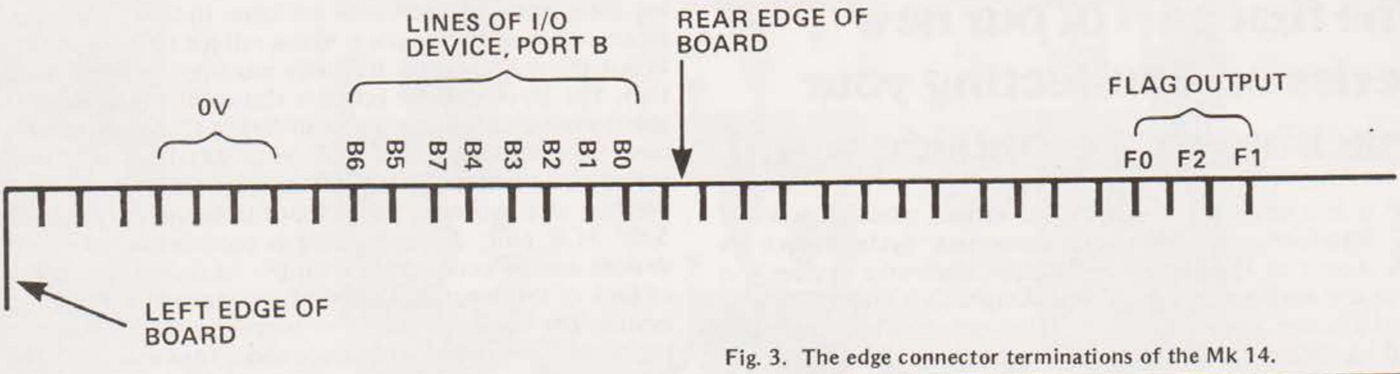


Fig. 3. The edge connector terminations of the Mk 14.

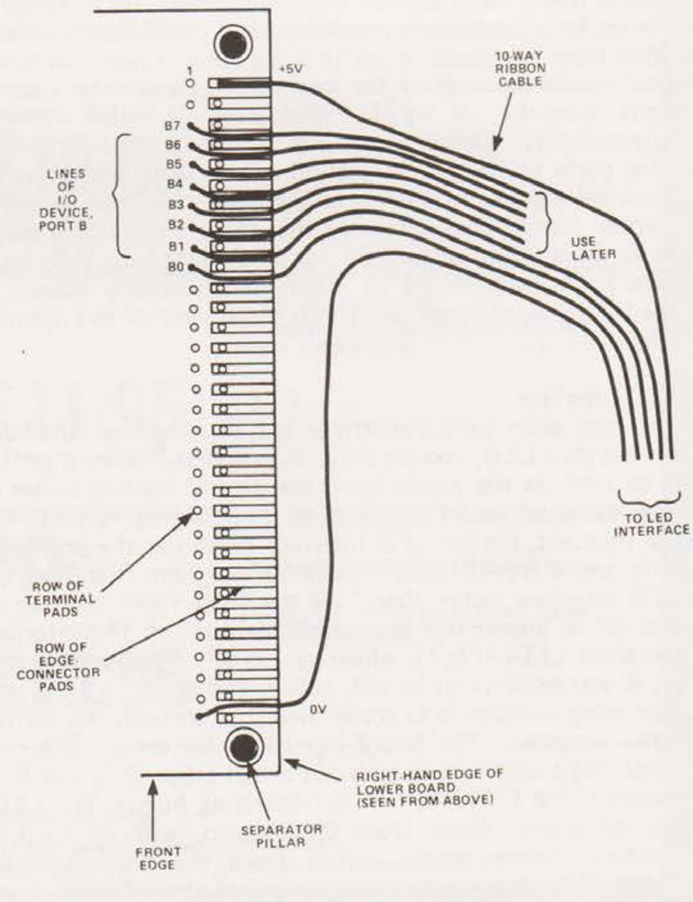
Construction
The strip-board is cut as shown in Fig.2. Assembly presents no problems, except the usual ones of avoiding solder threads between adjacent strips and making sure that breaks in strips really are complete. A hair-thin connection left where there should be a break can be disastrous: in building the prototype a connection was left accidentally between pins 6 and 11, and LED 1 burnt out immediately power was applied! Casual inspection of the board with a lens had not shown up the defect, though really careful inspection after the blow-out revealed the cause of the trouble. So inspect all soldering, breaks etc. with a lens, before applying power. LEDs must be mounted with their cathode pins to the 0V rail (strip AA). In most types of LED this pin is the slightly shorter of the two; in other types the rim of the LEDs body is flattened on that side. Remember too to observe the usual precautions in handling the CMOS IC – this should be inserted after all other construction work has been completed.

Connection To The Microprocessor Board
The SC/MP MPU used in the Mk-14 has three 'flag' outputs that can be used directly; these are referred to as F0, F1 and F2. These are connected to 3 pads of the edge-connector strip at the top of the board (Fig.3). Connecting wires can be soldered directly to these pads, and to the 0V pad. To obtain the 5 V supply, a wire is soldered to the 5 V rail; this runs down the left-hand side of the board (upper surface); close to the voltage regulator where there are some holes in the strip. A wire can be soldered into one of these holes, or you can insert and solder a terminal pin (same type as used for 0.1" stripboards) and solder the wire to this. The 5 wires (0 V, F0, F1, F2 and +5 V) are then taken to a 5-way socket to fit the plug on the interface board. Those who prefer not to make permanent connections to the microprocessor board may use an edge-connector for all except the +5 V connection and solder the connecting wires to the appropriate terminals.

I/O Device
Both Acorn and the Mk-14 use an Input/Output IC, the INS8154, to provide additional input and output lines. It also provides a useful addition to the memory space as an entirely independent function. The basic Mk-14 does not have this IC, but MPU flag outputs can be used for most of the simpler applications, and the SENSE A and SENSE B inputs are available for input interfacing. The I/O IC is purchased as an option and is well worthwhile for the greater scope it gives for control purposes. The basic Acorn already has an INS8154 on board but this is devoted to the tape interface. The 6502 MPU of the Acorn does not have any outputs such as 'flag' and 'sense' that may be used directly, so it is necessary to buy an INS8154 and insert this in the socket provided (IC8). Pin connections from this are taken almost to the edge of the board, but not to the edge-connector. Fig.4 shows where connections should be made.

Programming For Output
The kind of program used depends on whether we are using flag outputs (SC/MP) or the I/O device. We will consider each in turn. Programming flag outputs:– The flags are three locations in the status register of the MPU (Fig.5), and can be high (1) or low (0) depending on how they are set. Setting is simply a matter of loading accumulator with a byte in which there is a '1' for each flag that is to be high, and a '0' for each flag that is to be low. We then transfer the byte from accumulator to status register and the flags immediately assume the required state. Program A shows how this is done. You can alter byte 0F21 to determine which flags are to be set and which to be reset; for example, to set flags 0 and 2 (and thus light LEDs 0 and 2), alter the byte to '05' (= 0000

Fig. 4. The connections required for the Acorn. Do not solder wires to the edge connector pads.



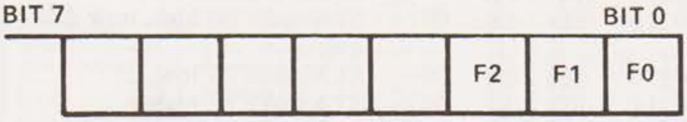


Fig. 5. The status register of a microprocessor. The flag locations are the three least significant bits.

0101). A little experimenting in varying the program and seeing what happens at the LED interface will soon make the procedure clear. Later we shall see how to extend this switching to items a lot more powerful than LEDs – in fact, to any kind of electrically powered device. Once a flag has been set, it remains set until 'reset' button is pressed or the appropriate bit is made low (0) by programming. For example, if byte 0F21 of Program A is made '00', all flags are reset, and all LEDs go out. This leads us to Program B, in which a LED is turned on, left on for about a quarter of a second and then turned off again. The ability to flash a warning lamp or make a buzzer emit a string of bleeps is very useful in alarm systems. The program is a loop, causing continuous flashing (or bleeping). If you want just a single flash, change 0F2A to '3F'. The length of flash and the length of period between flashes can be adjusted by altering the value of bytes 0F24 and 0F29 respectively.

Table 1: addressing the INS8154 I/O device (low byte: see text for high byte)

Operation	Location	Address (low)
CLEAR (or reset) single bit, to make it low (0)	Port A; lines A0 to A7	00 to 07
	Port B: lines B0 to B7	08 to 0F
SET single bit, to make it high (1)	Port A: lines A0 to A7	10 to 17
	Port B: lines B0 to B7	18 to 1F
PARALLEL (8-bit) setting or resetting	Port A	20
	Port B	21
OUTPUT DEFINITION REGISTERS	Port A (0DA)	22
	Port B (0DB)	23

Programming The I/O Device

This has 16 lines each of which can be independently programmed to be either an input or output (but not both at the same time). When the system is reset, all lines become inputs. In the input condition, interface LEDs attached to the line glow slightly but are not fully on or off, so it is necessary to program their lines as outputs. This is done by sending a byte to one of the output definition registers. There are two of these; one deals with the group of 8 lines known as Port A (individual lines are numbered 0 to 7, e.g. Ao, A1, A2. . . .A7; the other deals with the remaining 8 lines known as Port B (B0, B1, B2. . . .B7). Our LED interface is connected to the Port B lines so we need to instruct output definition register B (0DB) to make lines 0 to 2 act as outputs. We send a byte in which the bits corresponding to lines 0, 1 and 2 are high (1) and the remainder low. Thus we send the byte 0000 0111 from the accumulator to 0DB; in the Mk-14 0DB is at 0A23 and in Acorn it is at 0923. Table 1 lists the other addresses of the I/O device, showing the low bytes only; the high bytes are 0A for Mk-14 and 09 for Acorn.

Having determined which lines are to be outputs, we next have to decide which outputs are to be high and which low. This is done very easily by simply sending any byte to the appropriate address. For example, to make bit B1 high, we use the instruction 'store (anything) at 0A19' (or 0919 in Acorn). To make bit B1 low again, we address the instruction to 0A09 (or 0909). At this stage it is worthwhile trying out these programs with the LED interface connected and see what happens when various bytes are altered. There are also procedures for reading the state of lines that are designated as input lines, but we will deal with this facility later, when we need to use it. Another variation in the use of the I/O device is to write (into outputs) or read (from inputs) all 8 bits of a port together. This is parallel operation, in contrast to the single-bit operation that we have just described. The first stage is as before – inform the output definition register whether lines are to be inputs or outputs, Programs F and G show what happen next. The required state of each of the three outputs is set up in a byte that is stored in the Port B register (at 0A21, 9021). When this is done all three outputs change together. Since microprocessors work exceedingly fast, Program E and Program G appear to have the same action, yet in fact while Program E changes each of the lamps in turn – though only a few microseconds apart – Program G changes them simultaneously. Although this may not make any visible difference in this demonstration, it could make a lot of difference in other applications. Furthermore, if we have to deal with all 8 lines, parallel operation requires far less program steps. It's a good idea to try running this program and make it flash other sequences that you can design. With the SC/MP flashing is done by extending Program F to set or reset the output bits, as was done in Program B.

Program G is unnecessarily long for the storing and loop-counting routines are repeated for each change of lights. These steps can be made a sub-routine, with a further jump to the WAIT subroutine in monitor – a nesting of subroutines one within the other. The result is that it requires only 5 bytes (LDA followed by JSR) to program a new change of lights, so that long and complex sequences can be programmed in a very small amount of memory, and it can handle up to 8 lines at once. Now is your change to progress from hum-drum traffic-light sequences to something more in the nature of disco-lighting!

PROGRAMS

Program A: to set a flag bit (or more than one) and turn on an LED. For SC/MP (relocatable).

0F20	C4	01		LDI '01' (= 0000 0001)
0F22	07			CAS; sets bit 0 (F0) to '1'; bits 1 and 2 are '0'
0F23	3F			XPPC P3; return to monitor

Program B: to set and reset a flag bit (or more than one) and flash an LED. For SC/MP (relocatable)

0F20	C4	01		A:LDI '01'
0F22	07			CAS; sets bit 0; LED 0 turned on.
0F23	8F	FF		DLY to let you see LED is on.
0F25	C4	00		LDI '00'
0F27	07			CAS; all flag bits reset; all LEDs off.
0F28	8F	FF		DLY to let you see LEDs are off.
0F2A	90	F4		JMP back to A to repeat sequence.

Program C: to program lines B0, B1 and B2 as outputs, then set B1 high and the other two lines low. For SC/MP (relocatable).

0F20	C4	0A		LDI '0A' Pointing P1 to address
0F22	35			XPAH P1 of I/O device (0A00)
0F23	C4	00		LDI '00'
0F25	31			XPAL P1
0F26	C4	07		LDI '07' (= 0000 01111)
0F28	C9	23		ST P1+23 ;0DB instructed to define lines 0, 1 and 2 as outputs
0F2A	C9	08		ST P1+08 ;makes B0 output low (actual data stored does not matter)
0F2C	C9	19		ST P1+19 ;makes B1 output high (set)
0F2E	C9	0A		ST P1+0A ; makes B2 output low (clear)
0F30	3F			XPPC P3 return to monitor

Program D: same function as Program C, but for 6502 MPU (Acorn) (relocatable)

0200	A9	07		LDA#'07' (=0000 0111)
0202	8D	23	09	STA at 0923 (0DB instructed to define lines 0, 1 and 2 as outputs)
0205	8D	08	09	STA at 0908; makes B0 output low (actual data stored does not matter)
0208	8D	19	09	STA at 0919; makes B1 output high (set)
020B	8D	0A	09	STA at 090A; makes B2 output low (clear)
020E	4C	04	FF	JMP return to monitor

Program E: to program lines B0, B1 and B2 as outputs, then flash LEDs controlled by these lines. For 6502 (relocatable)

0200	A9	07		LDA#'07' (= 0000 0111)
0202	8D	23	09	STA at 0DB
0205	8D	08	09	A:STA make B0 low
0208	8D	19	09	STA make B1 high
020B	8D	0A	09	STA make B2 low
020E	A0	30		LDY#'30'; Y used as loop counter
0210	20	CD	FE	B:JSR to WAIT
0213	88			DEY decrement Y by 1, counting loops
9214	10	FA		BPL to B, if Y still positive
0216	8D	18	09	STA make B0 high, now delay is over.
0219	8D	09	09	STA make B1 low
021C	8D	1A	09	STA make B2 high
021F	A0	30		LDY#'30'; restoring loop counter.
0221	20	CD	FE	C:JSR to WAIT
0224	88			DEY ; counting loops
0225	10	FA		BPL to C, if Y still positive
0227	4C	05	02	JMP to A to repeat sequence

Program F: parallel output from Port B. For SC/MP (relocatable.

0F20-0F29	as in Program C			
0F2A	C4	02		LDI '02' (= 0000 0010)
0F2C	C9	21		ST at P1+21; Port B register; makes B1 high, B0 and B2 low.
0F2E	3F			XPPC P3 return to monitor

Program G: same effect as Program E, but by parallel outputs, For 6502 (relocatable)

0200	A9	07		LDA#'07'
0202	8D	23	09	STA at 0DB
0205	A9	02		A:LDA '02' (= 0000 0010)
0207	8D	21	09	STA at Port B register; makes B1 high, B0 and B2 low.
020A	A0	30		LDY#'30'; setting loop counter
020C	20	CD	FE	B:JSR to WAIT
020F	88			DEY counting loops
0210	10	FA		BPL to B, if Y still positive
0212	A9	05		LDA#'05' (= 0000 0101)
0214	8D	21	09	STA at Port B register; makes B1 low, B0 and B2 high.
0217	A0	30		LDY#'30' restoring loop counter.
0219	20	CD	FE	C:JSR to WAIT
021C	88			DEY counting loops
021D	10	FA		BPL to C, if Y still positive
021F	4C	05	02	JMP to A to repeat sequence

Program H: as Program G but with nested subroutines: For 6502 (relocatable)

0200	A9	07		LDA#'07'
0202	8D	23	09	STA
0205	A9	02		A:LDA#'02'
0207	20	40	02	JSR to B
020A	A9	05		LDA#'05'
020C	20	40	02	JSR to B
020F	4C	05	02	JMP to A to repeat sequence
0240	8D	21	09	B:STA at 0DB
0243	A0	30		LDA#'30' loop counter
0245	20	CD	FE	C:JSR to WAIT
0248	88			DEY counting loops
0249	10	FA		BPL to C, if Y still positive
024B	60			RTS back to main program

Next month: We give your micro an eye with a light-sensitive interface providing an input to the system and having many applications.

All these only from HENRY'S

IN STOCK

NASCOM-2 + FREE 16K RAM

NASCOM-2 + FREE 16 K RAM **£295** VAT P&P £1.50

Here's an offer you can't refuse:

Because of the lack of availability of MK 4118 RAMs, Nascom Microcomputers is supplying its Nascom 2 without the 8 spare 4118s but with a FREE 16K dynamic RAM board.

NASCOM–2 with 32K RAM £345 VAT P&P 1.50

When the 4118s become available, Nascom 2 purchasers can have them at the special price of £80 + VAT for the 8K.

So, for £295 plus VAT this is what you get:

MEMORY

- 16K RAM board (expandable to 32K).
- 8K Microsoft BASIC.
- 2K NAS-SYS 1 monitor.
- 1K Video RAM.
- 1K Workspace/ Scratchpad RAM.
- Main board sockets for the 8x4118s *or* 2708 EPROMS.

Buy British It's Best!!

MICROPROCESSOR

- Z80A which will run at 4MHz but is selectable between 1/2/4 MHz.

HARDWARE

- Industrial standard 12" x 8" PCB, through hole plated, masked and screen printed. All bus lines are fully buffered on-board.

INTERFACES • Licon 57 key solid state keyboard.

- Monitor/domestic TV interface.
- Kansas City cassette interface (300/1200 baud) or RS232/20mA teletype interface.

The Nascom 2 kit is supplied complete with construction article and extensive software manual for the monitor and BASIC.

MEMORIES
MK4116 16K **£7.50**
DYNAMIC RAM
8 for **£55.00**
MK 4027 4K
£2.80
8 for **£27**
2708 **£8.50**
2102 **£1.20**
2716 **£32.00**
Full List P.O.R.

No more slaving over a hot soldering iron – the Nascom 1 is now supplied BUILT!

Britain's biggest small system is available fully constructed for you to slot into your own housing for the ridiculously low price of £175 plus VAT (kit price still only £165 plus VAT). **EX-STOCK**

nascom-1

12" x 8" PCB carrying 5LSI MOS ... 1K MOS memory packages ... There is on-board ... unmodulated ... operating ... The 4K ... user RAM. ... standard Z80 which ... executing 158 ... including all 8080 code

Less P10 £125

With NAS-SYS
SCOOP KIT NOW ONLY
FULLY GUARANTEED

NASCOM-1 BUILT **£140** VAT P&P 1.50

NEW

EXPANSION NASCOM-1

★Expansion buffer board	£32.50
MEMORY KITS (inclusive all hardware)	
8K	£85
16K	£140
32K	£200
★I/O board with decoders and all hardware except ICS will accept up to 3 PIOs 1 CTV and 1 UART	£35
NEW. T 4 operating system in (2) 2708 EPROMS UPWARDS COMPATIBLE FROM T2 and B-BUG	£25.00
NAS SYS 1. MONITOR	£25.00
Programming Manual	£4.50
★Power supply for up to 32K expansion Mk II	£24.50
★8A power supply for larger than 32K expansion	£60.00
★Expansion card frame	£29.50
EPROM programmer	£13.95
SMART-1	£74.95
Tiny Basic	£25.00
Super Tiny Basic (with editor and machine utility routined)	£35.00
Zeap assembler editor	£32.00
8K BASIC ROM	£40.00
Naspen Text Handler	£30.00
Disassembler	£9.95

NASCOM IMP **£325** VAT P&P £2.50

NASCOM IMP PLAIN PAPER PRINTER

Fully built and housed in a stylish enclosure for just £325 plus 15% VAT. Interfaces with all micro computers. Deliveries Ex-Stock.

- Optional tractor feed.
- Baud rate from 110 to 9600
- External signal for optional synchronisation of baud

The Nascom IMP (Impact Matrix Printer) features are:

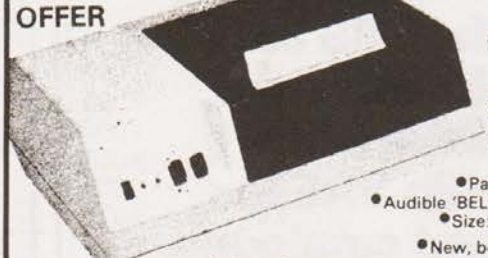
- 60 lines per minute
- 80 characters per line.
- Bi-directional printing.
- 10 line print buffer
- 96 character ASCII set (includes upper/lower case
- Automatic CR/LF.
- Accepts 8½" paper.

NEW

SPECIAL TOUCH ACTIVATE KEYBOARD **£49.50** + VAT 15% +£1 P&P

TASA has full 128 position 8-bit ASCII output plus continuous strobe, parity select. Ex-Stock

HALF PRICE OFFER



80 column
Centronics P1 printer

OUR PRICE £195

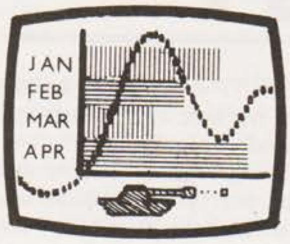
- Software selectable 20, 40 and 80 column using 120mm aluminiumised paper. 1 roll supplied.
- 150 lines per minute.
- Centronics parallel data interface for Nascom, Tandy, etc.
- 240volt mains input. ASCII character set
- Paper feed, and on/off select switches
- Audible 'BELL' signal Weight 10lbs
- Size: 13" x 10½" x 4¼" LIST PRICE **£400**
- New, boxed and fully guaranteed

Our Price £195.00
plus VAT post free
Paper **£12.13** per 3 rolls plus VAT

Your London & National Nascom Distributor.
Export Orders deduct VAT, but add 5% carriage
Official Export & Educational Orders welcome
Our Telex 262284 Mono Ref. 1400 Transonics

COMPUTER BROCHURE FREE **SEND 15p STAMP**

Henry's
Computer Kit Division
404 Edgware Road, London, W2, England
01-402 6822


JAN
FEB
MAR
APR


disk II
apple II

Henry Budgett.

Bogged down with a bug? Write a flowchart!

People who program generally tend to fall into one of two categories, those who use flowcharts and those who don't. I tend to write mine after the program and then correct the bugs, and I'm sure many of you do too! The techniques of flowcharting are of great benefit to those who like to tackle problems logically, they draw vast diagrams, test for all the possible quirks and then code up the result. The result of all this is usually a superb program, it never fails and is always late. The rest of us write and debug our efforts as we key them in, end up with programs that work, fail occasionally and are usually ready on time. In this article I hope to put across some of the ideas behind the writing of flowcharts and demonstrate their useful points.

The Simple Idea

A flowchart is defined as "A diagrammatic representation of a series of events, usually indicating the analysis or solution of a problem[1]." This is similar to, but not quite the same as an Algorithm, this is defined as "A defined process or set of rules for solving a given problem[1]." One usually starts with an algorithm, produces the flowchart and then codes the program. The simplest form of flowchart is shown in Fig.1, it uses no special symbols, and is really an extended version of the basic algorithm.

As regular readers will know, our flowcharts contain lots of pretty little boxes which must mean something, and indeed they do. In Fig.2 I have listed all the common types and their designated functions. This is only a small selection of the available symbols but for most purposes it will be quite adequate. As is only to be expected there is a Standard for these symbols and for those of you who like such things it is BS 4058 Part 1:1966.

The Standard Use

Having taken a look at the available set of symbols we can now re-write our simple flowchart in acceptable form, this is shown in Fig.3. For the actual task of converting it into a given language this will be quite sufficient, regardless of which language is to be used. A problem of this staggering complexity doesn't really deserve a flowchart at all, and indeed most proficient amateur programmers are quite capable of coding up large programs from a simple set of rules, or even the basic algorithm.

In Fig.4 I have attempted to flowchart another everyday problem, that of running a bath. As can be quickly seen it will work but is by no means bug-proof. Never mind, we'll

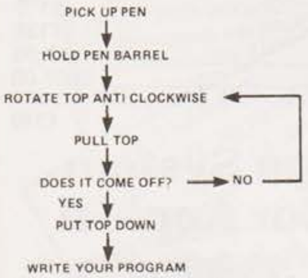


Fig 1. The simplest flowchart format.

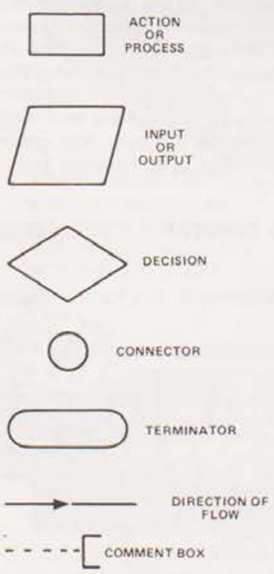


Fig 2. The standard flowcharting symbols.

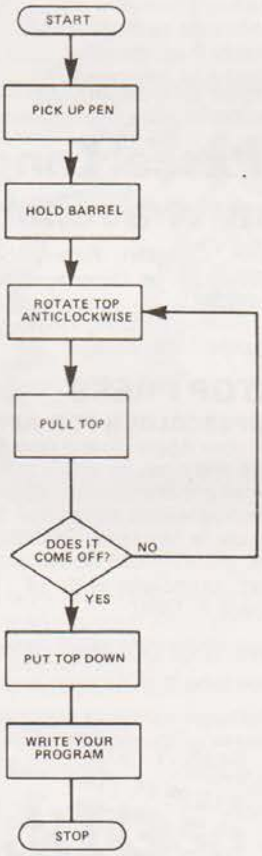


Fig 3. Figure 1 redrawn using the standard symbols.

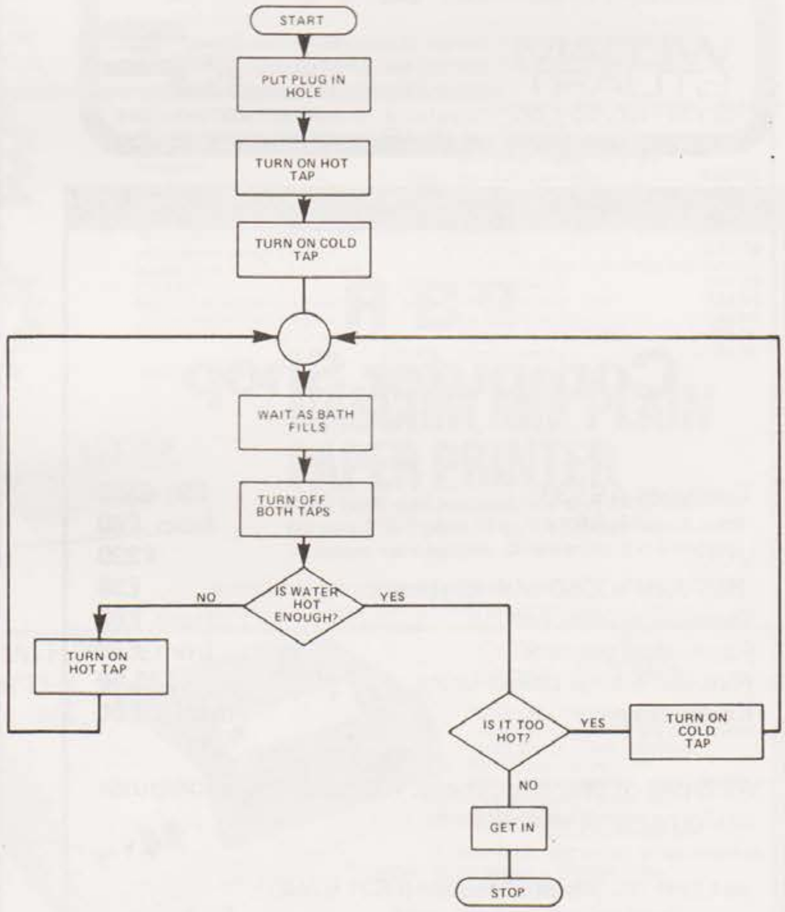


Fig 4. An attempt to flowchart a more complex problem.

sort them out later is the usual reply, in fact it's quite good enough to write a program from. We will take a last look at this program flowchart before we move on, it can be re-written into two parts, a Control section and a single sub-routine sections of the task as subroutines with their own flowcharts one can quickly sort out complex problems, and even write and test the various routines on their own before fitting them into the complete program.

The Real World

Computers being what they are, logical, the previous attempts at flowcharting bear no relation to a true programmers flowchart. A typical example of such a beast can be seen in Fig.6. The task is to produce a set of arithmetic tables for any given number between 1 and 12. The diagram shows all the steps needed and you should be able to follow it through on your own, there are comments!

The ideal of every programmer is to produce not only the ultimate bomb proof program but also to have it lavishly documented. This is the breakpoint between professional programs for a software house, or indeed a magazine for publication, and hopefully payment. It is almost obligatory to include not only a flowchart but a complete description of just what it does. In a case such as this you will find that your first flowchart will be so scrawled on that you have to re-draw it and it is well worth investing in a stencil that gives the standard symbols. It is also essential to keep a duplicate set of all the documentation for security, if you lodge a sealed set with the bank you have got a handy piece of evidence in case anyone rips off your version of S**r W**s and starts selling it and not paying any royalties!

In Conclusion

If you are capable of determining the way you wish to solve any given problem, writing the algorithm, you are capable of producing a flowchart. They are useful for debugging programs but you will find that they soon become covered with modifications and have to be re-drawn. Their most useful function is as a piece of documentation, how often do you remember how a program worked after six months, and as a means of testing out sections of a program such as subroutines.

Flowcharts are not essential as some people would have you believe but they do bridge the gap between successful programs and those which work.

References

Both definitions[1] are taken from The Dictionary of Data Processing from Newnes Butterworths so you can argue with them!

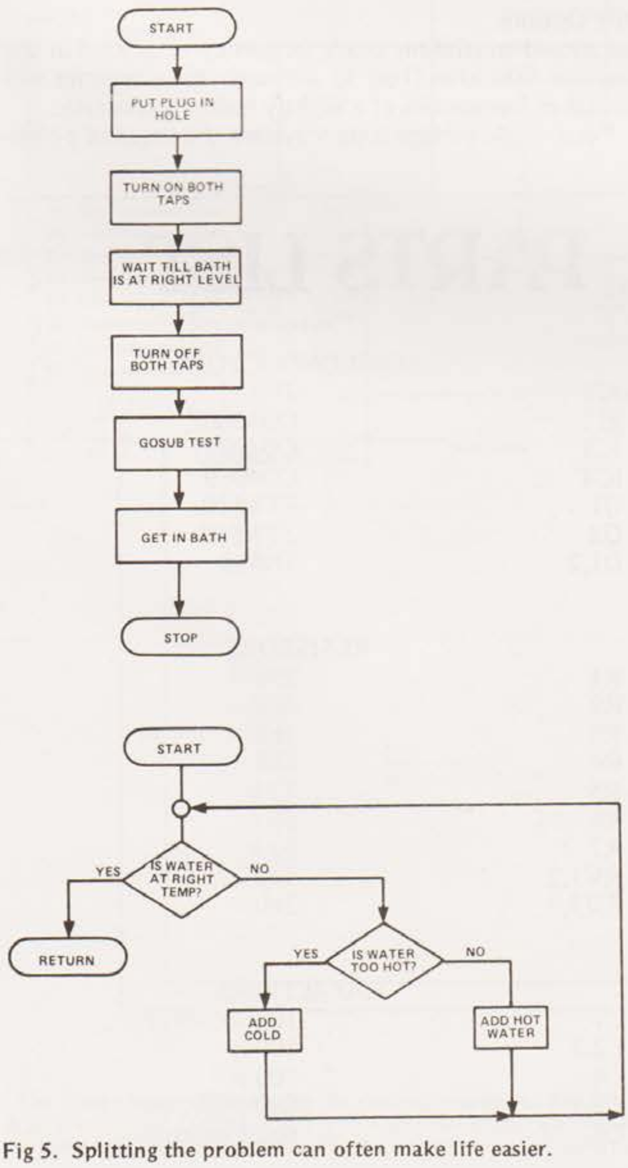


Fig 5. Splitting the problem can often make life easier.

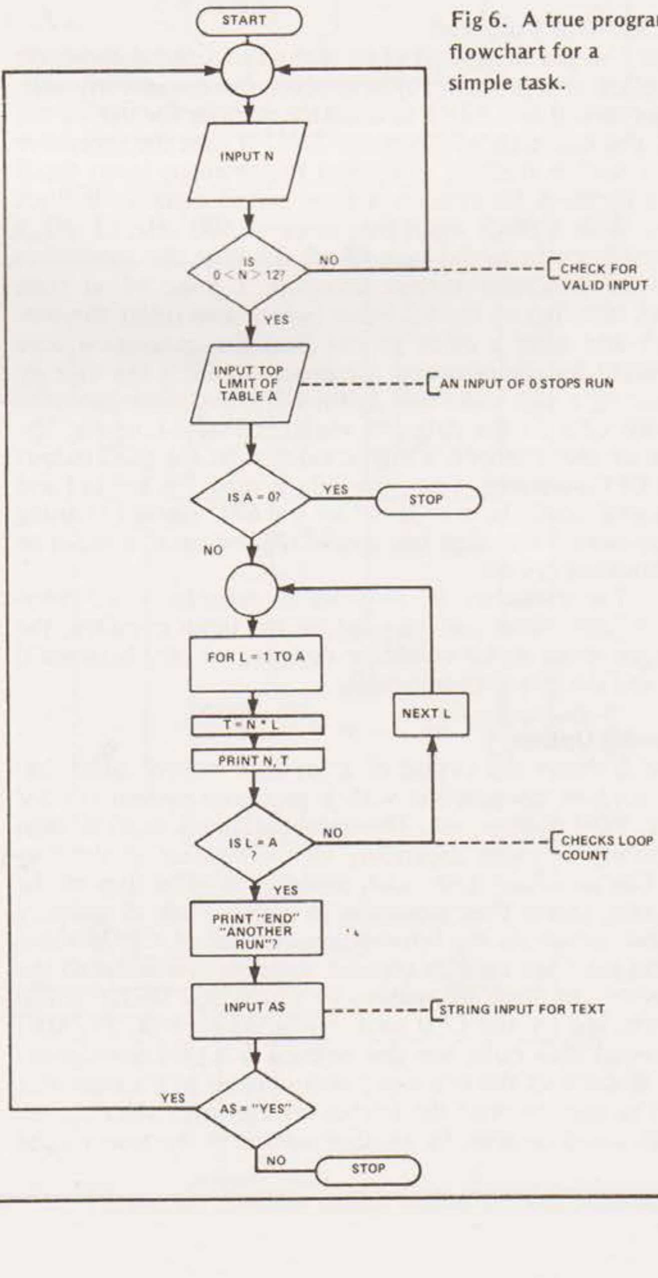


Fig 6. A true program flowchart for a simple task.

Peter Hibbs

Most of the information in the natural world is analogue, this project makes it acceptable food for any micro.

The natural world is full of interesting information that simply cries out to be investigated by the microprocessor owner. This information, such as sound,pressure, temperature and light intensity is in the form of an analogue signal. This is obviously incompatible with the digital signals that a microprocessor requires and some form of conversion must be undertaken prior to the data processing. There are many commercial chips that perform this function, known as Analogue to Digital, and the chip used here was chosen simply because it is one that has been used many times before and is well understood. No printed circuit has been published for this project because of the wide variety of possible applications.

The Electronic Converter

Figure 1 shows the circuit of an analogue to digital converter controlled by a processor system which can convert any voltage between 0 and +2V5 to a binary number for use by the CPU. The converter IC (Ferranti ZN427) uses the successive approximation method to convert the analogue input signal into a digital 8 bit code in a time period equal to 9 clock pulses. With a clock input frequency of 500 kHz, (2 uS), a conversion cycle would take 18 uS. To start the conversion cycle the processor system connects a pulse of at least 500 nS duration to the SC input (which also resets the converter) and after a delay greater than the conversion time (generated by the program) the processor reads the data by connecting a high condition to the OE input which gates the encoded data to the data bus via the tri-state outputs. The converter also connects a high condition to the EOC output (End Of Conversion) when the data is ready to be read and this signal could be monitored by the CPU instead of using a delay period although this would require another input to the processor system.

The converter IC provides an accurate +2V5 reference voltage which can be used by the input circuitry, the analogue input signal should be designed to vary between 0 volts and the reference potential.

A Gaming Option

Figure 2 shows the circuit of a joystick control which has been used in conjunction with a processor system for TV games, VDU control, etc. The potential at the slider of each potentiometer varies depending on the position of the control. The processor gates each potentiometer in turn to the converter, stores their positions in digital form in memory and then processes the information as required. CMOS transmission gates are used to connect the potentiometers to the converter and these are enabled by addressable latches which are switched by the CPU under program control. The start conversion (SC) pulse can also be used to set the appropriate latch to connect the required potentiometer to the converter could be used to reset the latches (as shown) or the required latches could be reset by another output instruction via the data bus which would allow the other latches to be used for other purposes. To read the value of the second potentiometer the process is repeated with the second latch being switched instead of the first. The sequence is repeated as often as is necessary depending on the required response time.

Capacitors C2 and C3 are provided to reduce "jitter" and the preset potentiometers can be used to adjust the 'zero' potential if the joystick potentiometers do not allow the slider potential to go to zero volts. A similar arrangement could also be used at the high voltage end of the potentiometers (i.e. connected to +5 volts) although the input voltage to the converter must not exceed 3V5.

Hardware Options

The guaranteed maximum clock frequency is qutoed in the data sheet as 600 kHz (1u6 S) although the converter will work at higher frequencies at a slightly reduced accuracy.

For a single voltage supply system the negative poten-

PARTS LIST

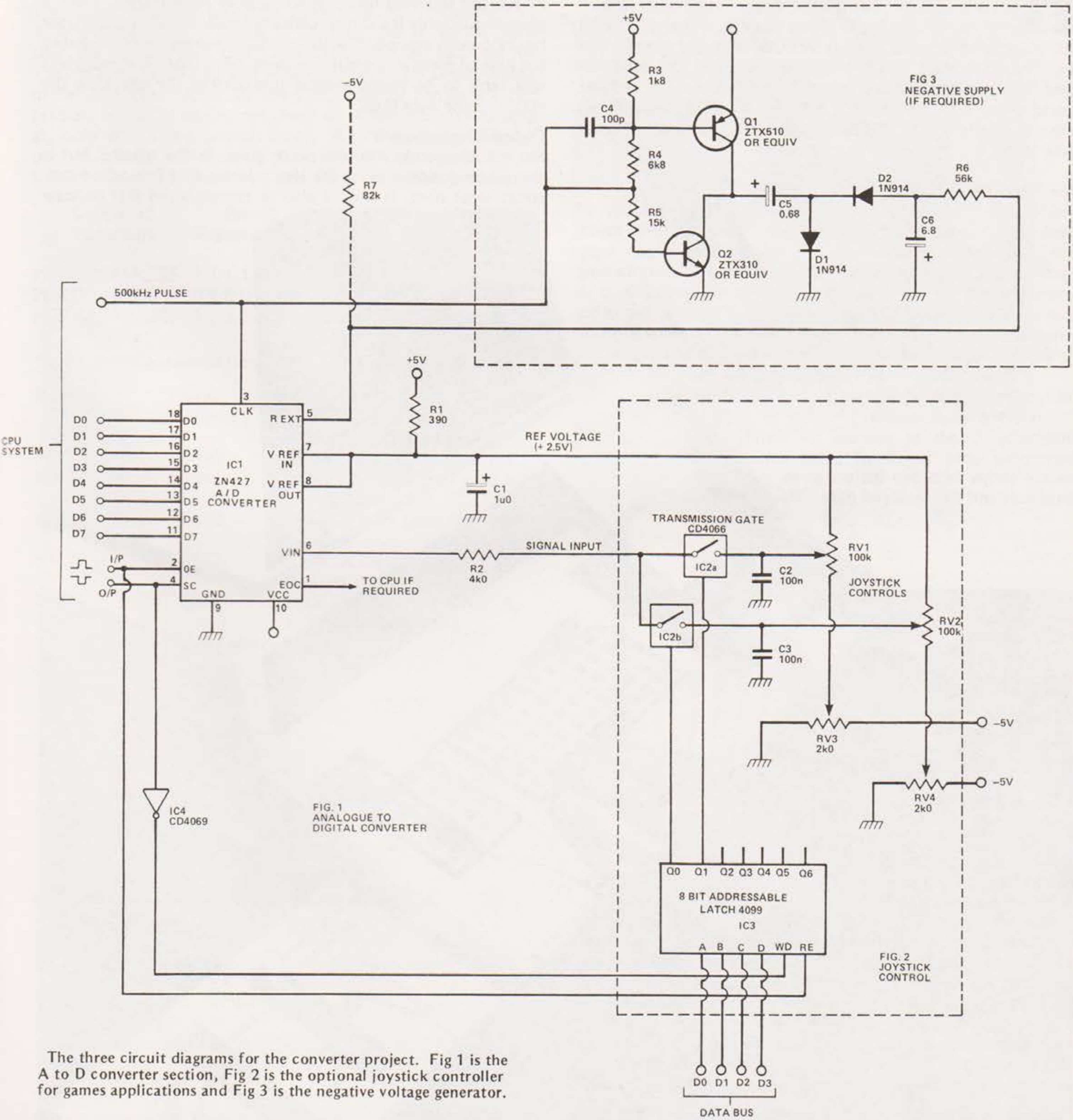
SEMICONDUCTORS	
IC1	ZN427
IC2	CD4066
IC3	CD4099
IC4	CD4069
Q1	ZTX510
Q2	ZTX310
D1,2	1N914
RESISTORS	
R1	390 R
R2	4k0
R3	1k8
R4	6k8
R5	15 k
R6	56 k
R7	82 k
RV1,2	100 k
RV3,4	2k0
CAPACITORS	
C1	1u Electrolytic
C2,3	100 n
C4	100 p
C5	680 p Electrolytic
C6	6u8 Electrolytic

tial for the Rext input can be provided from the positive 5 volt supply using the diode pump circuit of Fig.3 that is published in the Ferranti data sheet. Since the negative supply current is only in the order of 25–150 uA, this circuit could be used to power several converters.

The data sheet also states that the positive going edge of the SC pulse should not occur within 200 nS of an active clock pulse edge and that the first negative going edge of the clock pulse after the SC pulse should not occur until at least 1u5 S after the negative going edge of the SC pulse. Other input configurations are also shown in the data sheet together with timing diagrams, suggested circuits, etc.

The use of an analogue to digital converter with a microprocessor system allows a number of applications which would not otherwise be possible such as light level measurement, accurate temperature control or (with with addition of a suitable 'sample and hold' circuit) audio signals could be processed for use with speech recognition facilities.

The ZN427 is obtainable from Davian Electronics, 13 Deepdale Avenue, Oldham.



The three circuit diagrams for the converter project. Fig 1 is the A to D converter section, Fig 2 is the optional joystick controller for games applications and Fig 3 is the negative voltage generator.

Henry Budgett

To co-incide with the start of our new series "Microlink" we took a look at a commercial interface unit for the PET.

Having exhausted your capabilities as an X-wing fighter pilot it is more than likely that you will wish to turn your programming skills to more useful ends. Whilst the ubiquitous central heating controller is not going to be the first thing that you tackle you will need at least some kind of communications interface to talk to the outside world with. We mentioned the Communicator in our News pages a couple of months back and decided to take a closer look at the beast.

The Heart Of The Matter

Inside the box one immediately finds a large quantity of fresh air, a single PCB of rather poor quality and not much else. The circuit is based around two Darlington pair transistor arrays, there are a grand total of two IC's containing seven arrays each. Twelve of these have been paralleled up to provide a drive capability of 1 A for six channels, the other two channels can drive 500 mA each. Each channel is monitored with an LED which lights for both input or output.

Power for the unit is provided by the user, a maximum of 24 volts, both to drive his external loads and to generate +5 volts internally. Loads or sources are simply connected onto the front panel connector strips between the common terminals and the required data line.

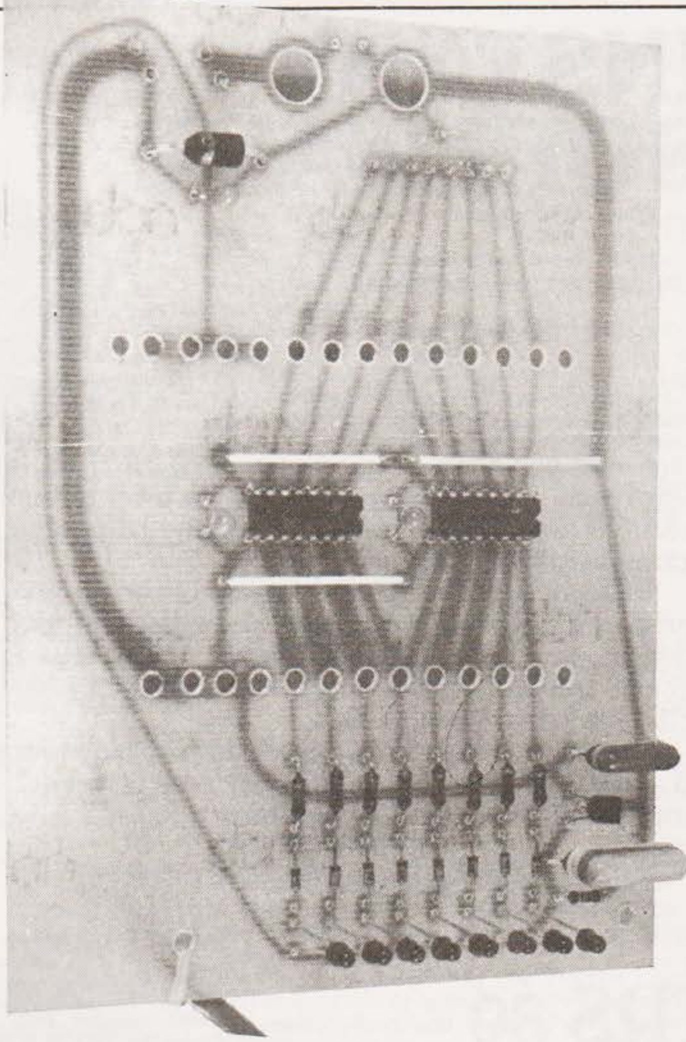
Talking Bi-directionally

The manual that we were supplied with was of a provisional nature but clearly explained the necessary programming techniques required. However, you can do a lot more with the device than the manual tells you as we soon found out. If you have the new PET manual, that's the one with the blue cover, pages 60 to 62 will tell you the rest but in brief you can do the following.

The parallel user port on the PET uses a VIA chip which can be programmed to perform a number of different functions. The available commands are given in Table 1. Having set up the parallel port all one has to do is PEEK or POKE the required location to input or output data, Table 2 gives the useable locations. Unfortunately the Communicator has not been equipped with any handshaking lines, probably for ease of general use but this does mean that your programs will tend to be based around subroutines for checking the status of the data lines.

Coded Requirements

No machine code routines were given in the manual but it should be possible using the Hex addresses in Table 2 to construct your own. It should also be remembered that because



you are using data lines the labelling on the front panel corresponds not only to the actual line in use but also to the decimal code. For example if you wish to ouput a data byte to lines 1 and 7 all you have to do is to add the value of 2ı to lines 1 and 7 all you have to do is to add the value of 2^1 to 2^7, that's 2+128, which gives 130. This value will set lines 1 and 7 on with all the rest off, easy isn't it!

We have given a couple of simple programs to check out the Communicator, you should be able to modify the basic ideas to suit your specific requirements.

Conclusions

The Communicator certainly does what it is supposed to with the minimum of hassle to the user, but it can do more than the manual says. Our main criticism is the fact that for a grand total of £92.85 one would expect to get considerably more than this. After all the CMC adaptor for an RS 232 printer, or its 3D equivalent, both of which we have looked at, only cost a few pounds more. A case of overpricing by our standards. Mektronic Consultants can be found at Linden House, 116 Rectory Lane, Prestwich, Manchester.

Command Statement	Binary Representation	Lines	Mode
POKE 59459,255	1 1 1 1 1 1 1 1	PA0-7	Output
POKE 59459,0	0 0 0 0 0 0 0 0	PA0-7	Input
POKE 59459,240	1 1 1 1 0 0 0 0	PA0-3	Input
		PA4-7	Output

Table 1. POKE commands for setting up the parallel user port.

Decimal	Hexa-Decimal	Addressed Location
59456	E840	Output register for I/O port B.
59457	E841	Output register for I/O port A with handshaking.
59458	E842	I/O Port B Data Direction register.
59459	E843	I/O Port A Data Direction register.
59471	E84F	Output register for I/O Port A, without handshaking.

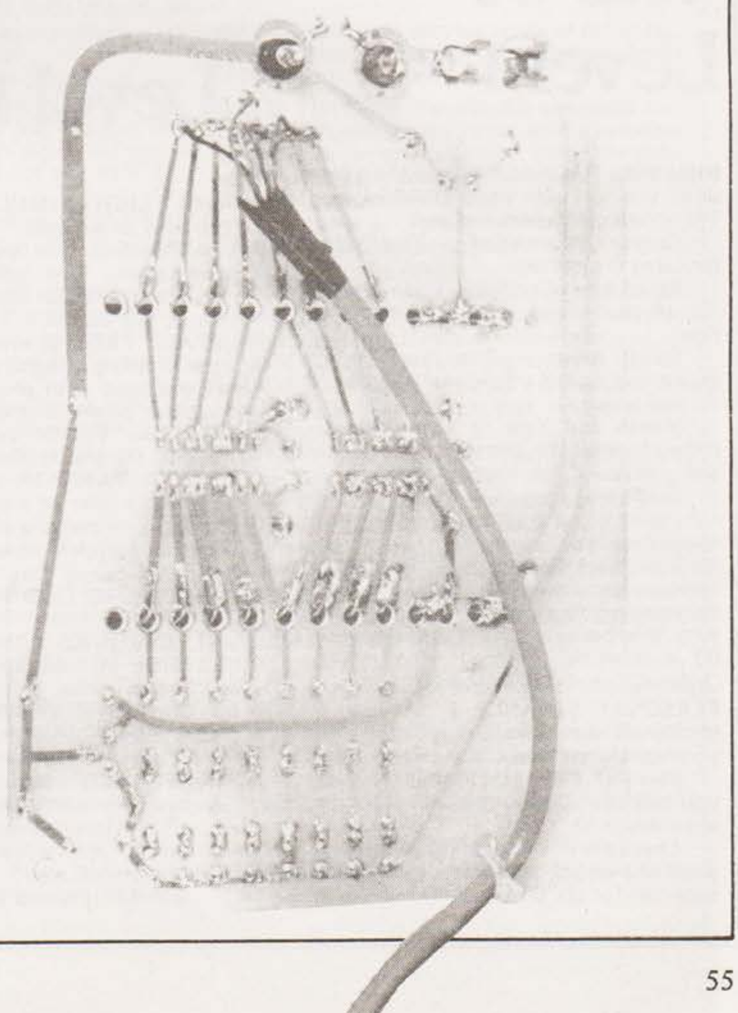
Table 2. Locations for the port registers.

```
READY.
 10 REM OUTPUT CHANNEL TEST
 20 POKE 59459,255
 30 POKE 59457,0
 40 FOR I=1 TO 255
 50 POKE 59457,I
 60 FOR D=1 TO 150:NEXT D
 70 NEXT I
 80 GOTO 30
```

Program to test out all the output channels sequentially.

```
 90 REM INPUT CHANNEL TEST
 100 POKE 59457,0
 110 I=PEEK(59457)
 120 IF I<1 THEN 110
 130 PRINT I
 140 FOR D=1 TO 150:NEXT D
 150 GOTO 110
READY.
```

Program to check out all the input channels sequentially.



INSTANT SOFTWARE

CT Software is a unique service that we offer to our readers. Each program comes on a high quality tape packaged with full documentation.

Read down the list of titles—you'll find a few suprises and some things for which you have offered sacrifices on a stone at dawn before now. All are checked and fully quaranteed, any complaints and we'll replace by return of post.

We think this is a revolution in reader service and one that will change the way you use your computer. So why not try us out ?

All orders and enquiries to:— CT software,
4 Morgan Street,
London E3 5AB.

All programs except 0013R are £6.75.

TRS 80 Level 1

BUSINESS PACKAGE I Keep the books for a small business with your TRS-80 Level I 4K. The six programs included are:

General Information — The instructions for using the package.

Fixed Asset Control — This will give you a list of your fixed assets and term depreciation.

Detail Input — This program lets you create and record your general ledger on tape for fast access.

Month and Year to Date Merge — This program will take your monthly ledger data and give you a year to date ledger.

Profit and Loss — With this program you can quickly get trial balance and profit and loss statements.

Year End Balance — This program will combine all your data from the profit and loss statements into a year end balance sheet.

With this package, you can make your TRS-80 a working partner.

Order Code. 0017R

PERSONAL FINANCE I Let your TRS-80 handle all the tedious details the next time you figure your finances:

Personal Finance I — With this program you can control your incoming and outgoing expenses.

Checkbook — Your TRS-80 can balance your checkbook and keep a detailed list of expenses for tax time.

This handy financial control package for the home requires only a TRS-80 Level I 4K. Order No. 0027R 5.75.

Level 1&2

AIR FLIGHT SIMULATION Turn your TRS-80 into an airplane. You can practice takeoffs and landings wiht the benefit of full instrumentation. This one-player simulation requires a TRS-80 Level I 4K, Level II 16K. Order No. 0002R

SPACE TREK II Protect the quadrant from the invading Klingon warships. The Enterprise is equipped with phasers, photon torpedoes, impulse power, and warp drive. It's you alone and your TRS-80 Level I 4K, Level II 16K against the enemy. Order No. 0002R

SANTA PARAVIA AND FIUMACCIO Become the ruler of a medieval city-state as you struggle to create a kingdom. Up to six players can compete to see who will become the King or Queen first. This program requires a 16K TRS-80 Level I & II. Order No. 0043R

ELECTRONICS I This package will not only calculate the component values for you, but will also draw a schematic diagram, too. You'll need a TRS-80 Level I 4K, Level II 16K to use:

Tuned Circuits and Coil Winding — Design tuned circuits without resorting to cumbersome tables and calculations.

555 Timer Circuits — Quickly design astable or monostable timing circuits using this popular IC.

LM 381 Preamp Design — Design IC preamps with this low-noise integrated circuit.

This package will reduce your designing time and let you build those circuits fast. Order No. 0008R

HAM PACKAGE I This versatile package lets you solve many of the commonly encountered problems in electtronics design. With your Level I 4K or Level II 16K TRS-80, you have a choice of:

Basic Electronics with Voltage Divider — Solve problems involving Ohm's Law, voltage dividers, and RC time constants.

Dipole and Yagi Antennas — Design antennas easily, without tedious calculations.

This is the perfect package fnr any ham or technician. Order No. 0007R

Level 2

TRS-80 UTILITY I Ever wonder how some programmers give their programs that professional look? Instant Software has the answer with the TRS-80 Utility I package. Included are:

RENUM — Now you can easily renumber any Level II program to make room for modification, or to clean up the listing.

DUPLIK — This program will let you duplicate any BASIC, assembler, or machine-language program, verify the data, merge two or more programs into one data block, and even copy Level I programs on a Level II machine. For TRS-80 Level II 16K. Order No. 0081R

TRS-80 UTILITY 2 Let Instant Software change the drudgery of editing your programs into a quick, easy job. Included in this package are:

CFETCH – Search through any Level II program tape and get the file names for all the programs. You can also merge BASIC programs, with consecutive line numbers, into one program.

CWRITE – Combine subroutines, that work in different memory locations into one program. This works with BASIC or machine-language programs and gives you a general checksum.

This package is just the thing for your TRS-80 Level II 16K. Order No. 0076R

SPACE TREK IV Trade or wage war on a planetary scale. This package includes:

Stellar Wars – Engage and destroy Tie fighters in your attack on the Death Star. For one player.

Population Simulation – A two-player game where you control the economy of two neighbouring planets.

You decide, guns or butter, with your TRS-80 Level II 16K. Order No. 0034R

RAMROM PATROL/TIE FIGHTER/KLINGON CAPTURE Buck Rogers never had it so good. Engage in extraterrestrial warfare with:

Ramrom Patrol – Destroy the Ramron ships before they capture you.

Tie Fighter – Destroy the enemey Tie fighters and become a hero of the rebellion.

Klingon Capture – You must capture the Klingon ship intact. It's you and your TRS-80 Level II 16K battling across the galaxy. Order No. 0028R

CARDS This one-player package will let you play cards with your TRS-80 – talk about a poker face!

Draw and Stud Poker – These two programs will keep your game sharp.

No-Trump Bridge – Play this popular game with your computer and develop your strategy.

This package's name says it all. Requires a TRS-80 Level II 16K. Order No. 0063R

HOUSEHOLD ACCOUNTANT Let your TRS-80 help you out with many of your daily household calculations. Save time and money with these fine programs:

Budget and Expense Analysis – You can change budgeting into a more pleasant job with this program. With nine sections for income and expenses and the option for one- and three-month review or year totals, you can see where your money is going.

Life Insurance Cost Comparison – Compare the cost of various life insurance policies. Find out the difference in price between term and whole life. This program can store and display up to six different results.

Datebook – Record all those important dates in your life for fast, easy access. The program has all major holidays already included.

All you need is TRS-80 Level II 16K. Order No. 0069R

FINANCIAL ASSISTANT Compute the figures for a wide variety of business needs. Included are:

Depreciation – This program lets you figure depreciation on equipment in five different ways.

Loan Amortization Schedule – Merely enter a few essential factors, and your TRS-80 will display a complete breakdown of all costs and schedules of payment for any loan.

Financier – This program performs thirteen common financial calculations. Easily handles calculations on investments, depreciation, and loans.

1% Forecasting – Use this simple program to forecast sales, expenses, or any other historical data series.

All you need is a TRS-80 Level II 16K. Order No. 0072R

PET

CASINO I These two programs are so good, you can use them to check out and debug your own gambling system!

Roulette – Pick your number and place your bet with the computer version of this casino game. For one player.

Blackjack – Try out this version of the popular card game before you go out and risk your money on your own "surefire" system. For one player.

This package requires a PET with 8K. Order No. 0014P

CASINO II This craps program is so good, it's the next best thing to being in Las Vegas or Atlantic City. It will not only play the game with you, but also will teach you how to play the odds and make the best bets. A one player game, it requires a PET 8K. Order No. 0015P

CHECKERS/BACCARAT Play two old favourites with your PET.

Checkers – Let your PET be your ever-ready opponent in this computer-based checkers program.

Baccarat – You have both Casino- and Blackjack-style games in this realistic program.

Your PET with 8K will offer challenging play anytime you want. Order No. 0022P

MIMIC Test your memory and reflexes with the five different versions of this game. You must match the sequence and location of signals displayed by your PET. This one-player program includes optional sound effects with the PET 8K. Order No. 0039P

TREK-X Command the Enterprise as you scour the quadrant for enemy warships. This package not only has superb graphics, but also includes programming for optional sound effects. A one-player game for the PET 8K. Order No. 0032P

TURF AND TARGET Whether on the field or in the air, you'll have fun with Turf and Target package. Included are:

Quarterback – You're the quarterback as you try to get the pigskin over the goal line. You can pass, punt, hand off, and see the results of your play using the PET's superb graphics.

Soccer II – Play the fast-action game of soccer with four playing options. The computer can play itself, play a single player, two players with computer assistance, and two players without help.

Shoot – You're the hunter as you try to shoot the bird out of the air. The PET will keep score.

Target – Use the numeric keypad to shoot your puck into the hom position as fast as you can.

To run and score all you'll need is a PET with 8K. Order No. 0097P

ARCADE I This package combines an exciting outdoors sport with one of America's most popular indoor sports:

Kite Fight – It's a national sport in India. After you and a friend have spent several hours manoeuvering your kites across the screen of your PET, you'll know why!

Pinball – By far the finest use of the PET's exceptional graphics capabilities we've ever seen, and a heck of a lot of fun to play to boot.

Requires an 8K PET. Order No. 0074P

ARCADE II One challenging memory game and two fast-paced action games make this one package the whole family will enjoy for some time to come. Package includes:

UFO – Catch the elusive UFO before it hits the ground!

Hit – Better than a skeet shoot. The target remains stationary, but you're moving all over the place.

Blockade – A two-player game that combines strategy and fast reflexes.

Requires 8K PET. Order No. 0045P

DUNGEON OF DEATH Battle evil demons, cast magic spells, and accumulate great wealth as you search for the Holy Grail. You'll have to descend into the Dungeon of Death and grope through the suffocating darkness. If you survive, glory and treasure are yours. For the PET 8K. Order No. 0064P

Apple

MATH TUTOR I Parents, teachers, students, now you can turn your Apple computer into a mathematics tutor. Your children or students can begin to enjoy their math lessons with these programs:

Hanging – Perfect your skill with decimal numbers while you try to cheat the hangman.

Spellbinder – Cast spells against a competing magician as you practice working with fractions.

Whole Space – While you exercise your skill at using whole numbers your ship attacks the enemy planet and destroys alien spacecraft.

All programs have varying levels of difficulty. All you need is Applesoft II with your Apple II 24K. Order No. 0073A

MATH TUTOR II Your Apple computer can go beyond game playing and become a mathematics tutor for your children. Using the technique of immediate positive reinforcement, you can make math fun with:

Car Jump – Reinforce the concept of calculating area while having fun making your car jump over the ramps.

Robot Duel – Practice figuring volumes of various containers while your robot fights against the computer's mechanical man.

Sub Attack – Take the mystery out of working with percentages as your submarine sneaks into the harbor and destroys the enemy fleet.

All you need is Applesoft II with your Apple II and 20K. Order No. 0098A

GOLF Without leaving the comfort of your chair, you can enjoy a computerized 18 holes of golf with a complete choice of clubs and shooting angles. You need never cancel this game because of rain. One or two players can enjoy this game on the Apple with Applesoft II and 20K. Order No. 0018A

BOWLING/TRILOGY Enjoy two of America's favorite games transformed into programs for your Apple:

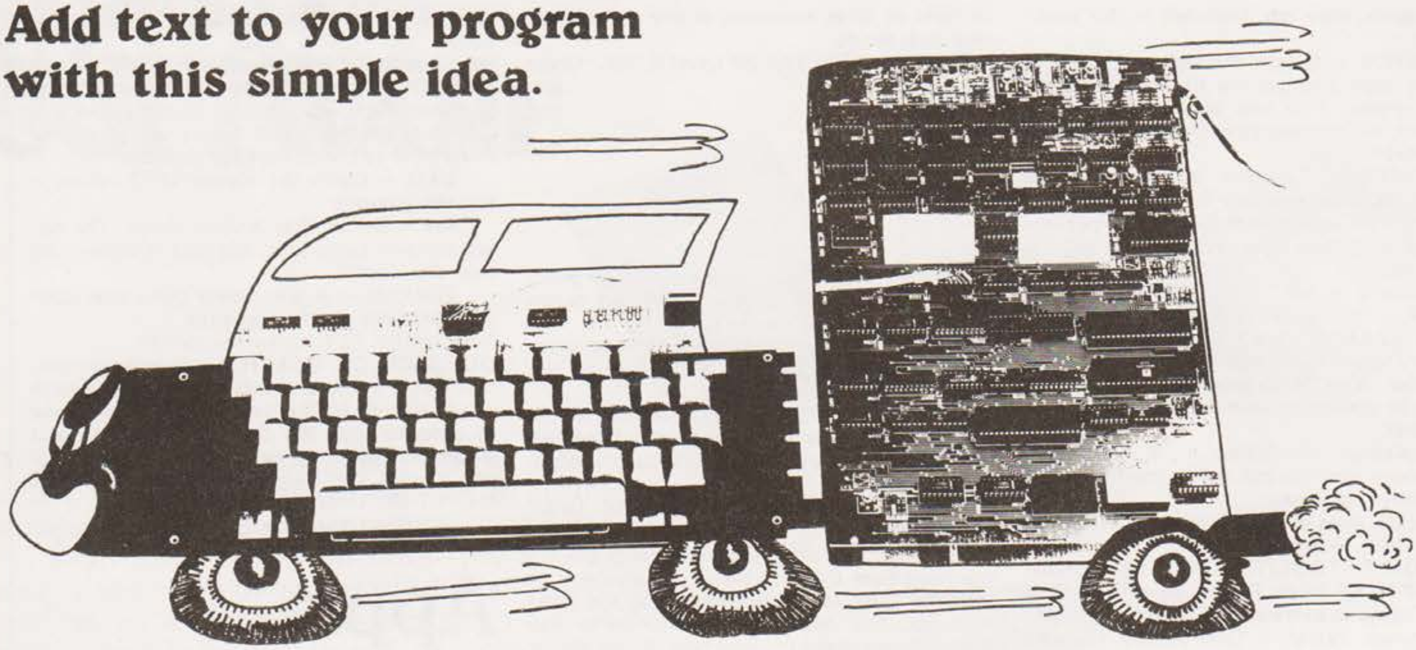
Bowling – Up to four players can bowl while the Apple sets up the pins and keeps score. Requires Applesoft II.

Trilogy – This program can be anything from a simple game of tic-tac-toe to an exercise in deductive logic. For one player.

This fun-filled package requires an Apple with 20K. Order No. 0040A

Mr. S.R. Collins.

Add text to your program with this simple idea.



Films have trailers, TV programmes have them, *your* programs can have them as well. Your trailers will not, of course, proclaim how earth-shattering the program that follows is but will screen the vital information that almost every program needs for execution: information such as the range of memory used, the memory location for execution, how to end the program, what key (if any) has been allocated a special task, and so on.

I accept that – being the good lads we all undoubtedly are – this information is already filed neatly away with the program listing!

Information For Free

But (come on, admit it!) isn't it an awful bind to go rooting for this file? Especially when everything else we need is already on the cassette label. Even if we have a dozen cassettes full of programs it doesn't take more than a few seconds to identify the cassette we need and then locate our program on it.

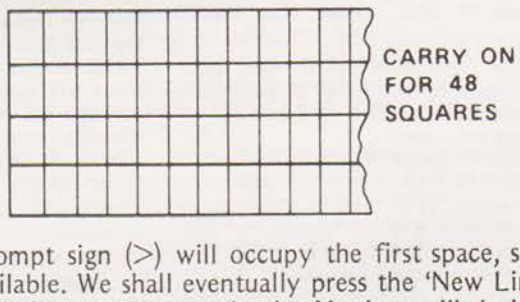
How convenient, to say the very least, if that same cassette could be loaded with the file's vital information about the program and display it on the screen. These trailers do just that.

Before you start moaning and groaning that programming text is tedious and uses acres of precious memory, let me say right now that this method doesn't. You'll know that when your NASCOM is idling – that is, when not actually executing a program or command – anything you type on the keyboard prints out along the bottom line of the screen. What's more, when you get to the end of a line it is scrolled up automatically and you can begin typing your second line, and so on. This is how we get our trailer on the screen. Then we tape the VDU RAM that contains it. So we don't employ any user RAM. Memory-wise our trailer is for free!

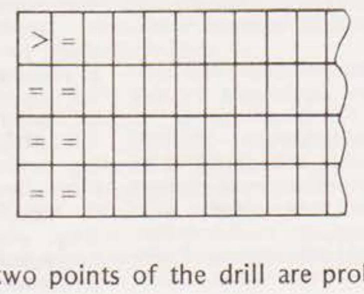
As a matter of fact, it isn't *quite* such a doddle as that. For one thing, we are restricted to 4 lines of the screen (184 characters, actually). For another, we must adopt a simple but rigid drill until the routine becomes automatic. These restrictions are imposed by the scolling up that takes place during the operation of the Write (W) and Read (R) Commands. For the same reason it is not possible to use the Dump (D) and Load (L) Commands, so these trailers are only possible with the B-Bug and T4 Monitors. (It should also work with NAS-SYS.)

How It's Done

Now for the nitty gritty: the drill. The first requirement is that we must write down our trailer on paper before we attempt to put it on the screen. So we need a representation, a map, of four lines of the screen – i.e. a 4-line grid, each line being 48 squares long:



The prompt sign (>) will occupy the first space, so that is not available. We shall eventually press the 'New Line' (NL) key and the moment we do the Monitor will do its damnedest to interpret as a command any character right next to the prompt. To remove this temptation we'll never use that space. And to remove the possibility of any other untoward happenings we'll not use the first two spaces of any other line, either. So let's block these out, as follows, as soon as we've drawn the grid. The 'equals' sign (=) denotes, "Press Space Bar".



The final two points of the drill are probably most important:

. . . we must end every line, except the last, with a

character or space bar so that the line is scrolled up automatically.
. . . the last line on the other hand must be terminated by pressing NL

We are now ready to write our trailer down on the grid and here is an example:

```
 *COLOUR NEGATIVE ANALYSER - PART 1  C50 TO E98*
*-EXECUTE FROM C60  ADJUST ENLARGER THIS UNTIL-*
*-SCREEN READS A STEADY 200  PRESS SPACE BAR &-*
*-PROCEED TO PART 2   @ 3260. SAME CASSETTE NL
```

Type the trailer on the screen, being very careful that your 4 lines end up on the screen exactly as they are written on the grid.

Storing The Trailer

Now it only remains to transfer the trailer from the screen to the tape. Enter the Command 'WA4A B3A', start the cassette recording and press NL. This command is always the same for every 4-line trailer and, provided we have adhered to the recommended drill, takes account of all the scrolling up. Immediately after this we tape the program itself, of course.

When we want to read the trailer and its associated program don't forget to apply the 'R' Command twice — once for the trailer and, immediately afterwards, for the program itself.

You'll soon find the drill becomes automatic and you can stop writing the trailer down. But until then please write down your trailers in the format given. Mind you, if you don't you can have a hell of a lot of fun. You'd never believe the words the Monitor interprets as commands when they immediately follow the prompt. You can fill the screen with such starbursts, star wars, snowstorms and alien encounters of a firework kind that you'll think you are designing backgrounds for the next space spectacular, no doubt to be called 'Son of Alien'.

Getting More For Your Money

I've stressed that we are restricted to 4 lines on the screen. Certainly they should be more than enough for most trailers. But we can actually have 8 lines if we really need them, although this is the absolute maximum. Two modifications to our drill become necessary:
. . . we type and record 9 lines, not 8, but the 5th line will be lost, so fill it with garbage or spaces.
. . . the Write Command becomes W90A B3A'

The same warnings about line endings (including the 5th) apply as for 4-liners.

A little bonus: as soon as you've set the 'R' Command going, out marches your trailer from the wings across the screen, letter by letter, with a staccato precision Busby Berkeley couldn't better. All this whilst the cassette is still turning. In other words, it's an instant indication of whether the 'W' and 'R' Commands are working. If no letters stride immediately across the screen then either they were not recorded correctly or are not being read correctly. Even if you don't want anything to do with program trailers this is therefore an instant, positive — and amusing — routine for testing Read and Write Commands.

SOFTSPOT

D.A.H. Elworthy.

In ETI, May 1976, an electronic game was described, which was based on the reflexes of two players. Here is a version for PET, for up to 5 players. The rules are held in lines 30 to 38. The "light" is graphic shift Q or W, these corresponding to on and off respectively. "Too early", referred to in the rules, means "before the light comes on".

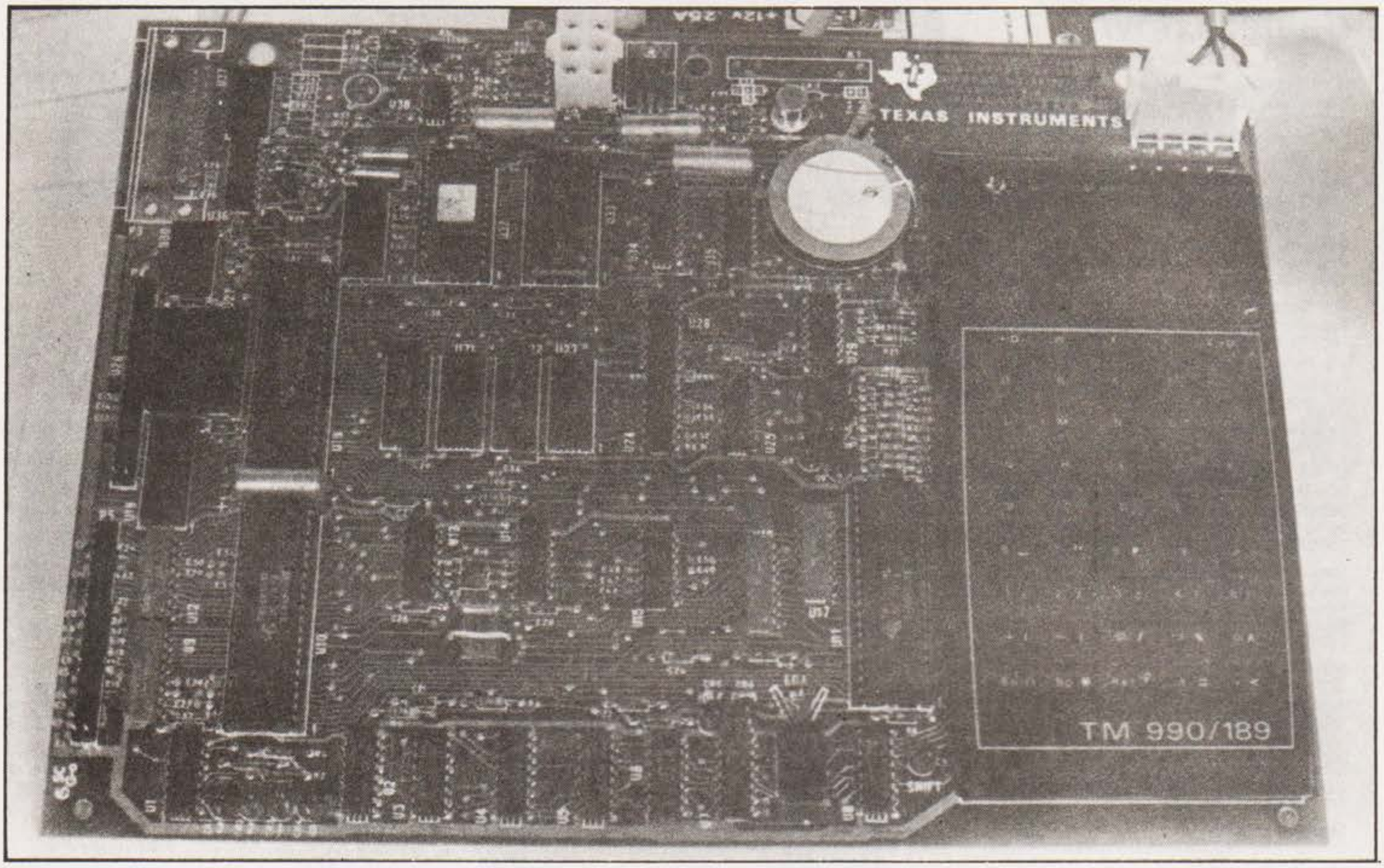
Program Notes

1. In general, the formatting of the program lines is arranged, so as to occupy only one screen line.
2. The bracketed portions in the right column are commentary only.
3. Line 520 will only ever execute if a player presses a key which has not been recorded in A$. If it is executed, player 5 (or the last entered player, if less than 5) will get the point or disqualification:— "the honesty of the player is assumed". If you are playing with a bunch of cheats, then change this line to:
520 NEXT L%;L%=0 and watch out for the result!
4. There are two problems, if the program is to be converted for a different system:
a. GET — some form of non-RETURN input is needed;
b. POKE the screen (lines 150,180). The former of these could be omitted and the latter made into a print statement.
5. Location 33148 is row 10 column 20.

```
10 DIMA$(5,2),A(5)
20 POKE 59468,14:REM GOTO LOWER CASE
30 PRINT"Enter the player's names."
31 PRINT"and the key each will use."
32 PRINT"Up to 5 may play. If more than"
33 PRINT"5 play enter *,* after the last."
34 PRINT:PRINT"When the light comes on,first"
35 PRINT"to press his key wins 1 point."
36 PRINT"If you press a key too early,"
37 PRINT"you lose all your points!"
38 PRINT:PRINT"The first to 10 wins."
40 PRINT:PRINT"Press ANY key when ready."
50 GET B$:IF B$="" GOTO 50
60 PRINT"3":POKE 59468,12:REM UPPER CASE
90 FOR H=1 TO 5:A(H)=0:NEXT H
100 FOR I=1 TO 5
110 INPUT"NAME,KEY";A$(I,1),A$(I,2)
120 IF A$(I,1)="*" GOTO 140
130 NEXT I
140 REM MAIN GAME FOLLOWS
150 PRINT"3":POKE 33148,87:FOR K=1 TO I
160 PRINT A$(K,1);TAB(20);A$(K,1)
170 PRINT"GAME STARTING....."
175 IF RND(TI)<.95 GOTO 175:REM DELAY
180 GET Z$:POKE 33148,81
190 GET Y$:IF Y$="" GOTO 190
200 IF Z$="" GOTO 240
210 Y$=Z$:GOSUB 500
220 PRINT X$;" JUMPED THE GUN"+A(X)=0
230 GOTO 110
240 GOSUB 500
250 PRINT X$;" WINS THIS ROUND"
260 A(X)=A(X)+1
270 IF A(X)=10 GOTO 600
280 GOTO 175
500 FOR L=1 TO I
510 IF A$(L,2)=Y$ GOTO 530
520 NEXT L:GOTO 540
530 X$=A$(L,1):X=L
540 RETURN
600 PRINT"PLAY AGAIN?"
610 GET A$:IF A$="" GOTO 610
620 IF A$="Y" GOTO 90
630 END
READY.
```

Ron Harris B.Sc.

TEXAS T



CT took a course in sixteen bit technology at the Texas University. Did we pass with flying colours......

Somewhat of an oddity this board. As you can see from the photographs, the most prominent feature is a calculator keyboard and display assembly mounted to the right of the main PCB. The pale disc is a piezo 'speaker' providing a sort of sound capability.

The TM 990/189 is one of the series from Texas based upon their unique TMS9980 (16-bit) MPU. It is designed to introduce a complete tyro to the art of assembly language programming and comes complete with a User Guide to the module, and a massive self-teach manual - some five hundred and seventy pages in all - which begins with a run down of computer architecture and hopes to have the reader well into modular programming techniques by Chapter 8.

A PSU is required to run the TM 990, and for £67.82 Texas will supply one. The specification required of the supply is +5V at 2A, and +/−12V at 0A5 or thereabouts.

We used the Texas supply for our review, simply because it saved us building one and we were eager to find out what power lay behind that bleak keyboard.

However we suspect that most of our readers would be able to provide their own for considerably less that £67. Check it before connecting, though, if you intend to follow this course of action through. Regulation should be +/−5% of nominal.

All fairly standard stuff.

Texas have pulled a little string by fitting a cable - reversable

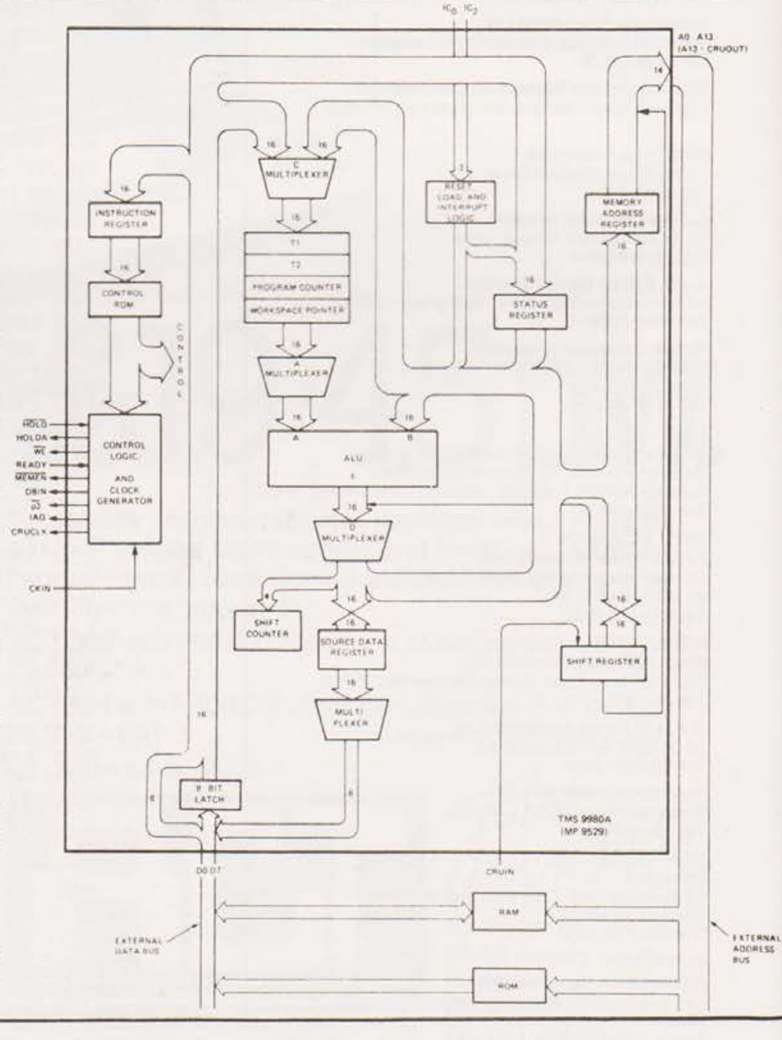


Fig 1. (Right) The CPU architecture of the TMS 9980A

M990/189 UNIVERSITY MODULE

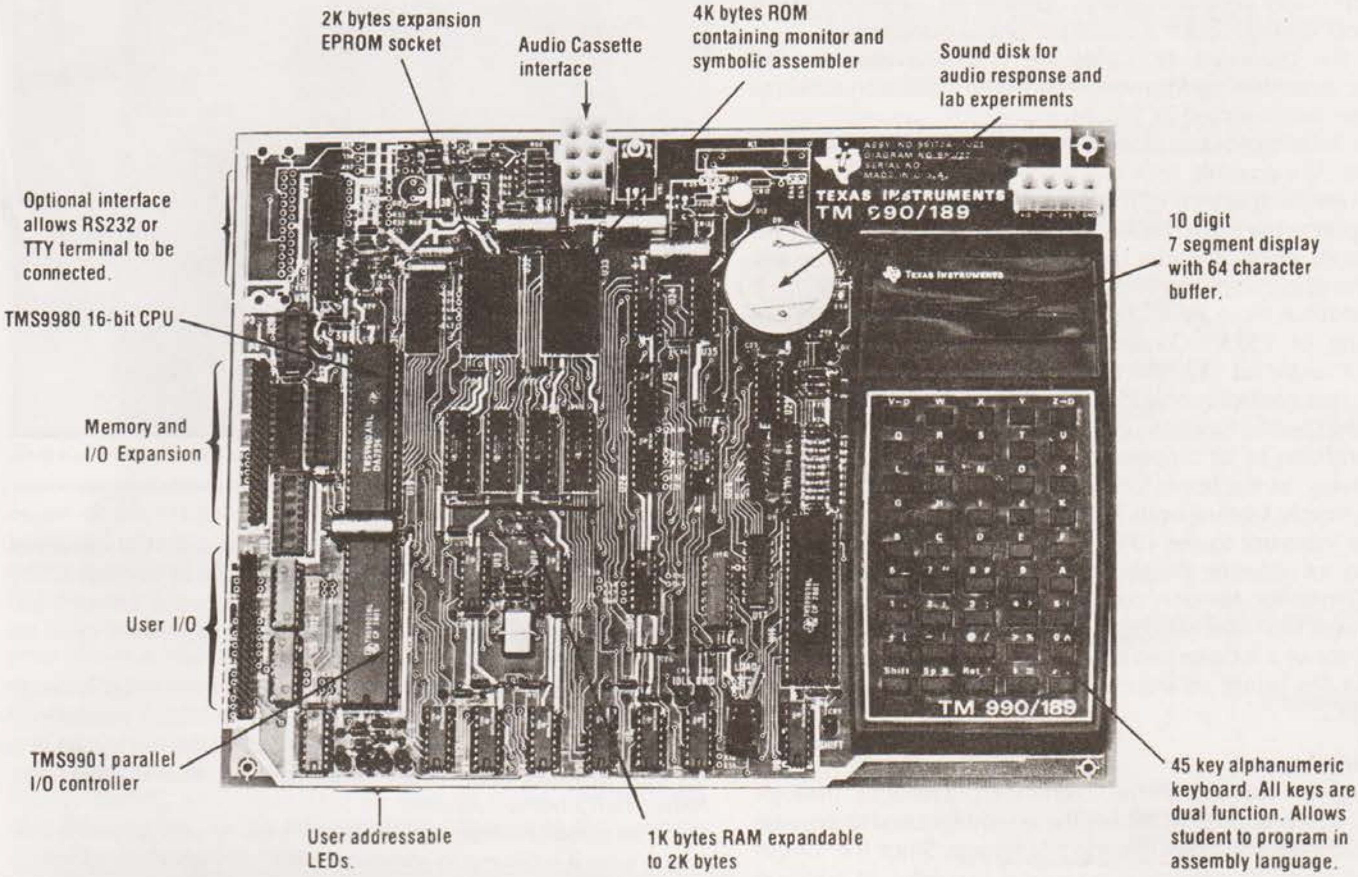


Fig 2. The module PCB and wot's on it!

and idiot proof - with the same weird plug on both ends, that will connect up a TM 990 to their own PSU in a second, but which might cause a few hours wandering around to component shops, vaguely waving plugs in the air in the hope of acquiring a match.

If thine plug offends thee - cut it off and solder 'in' a more common item.

Of the Texas PSU, number TM 990/519, there is little to say - it is superbly constructed, works perfectly and is overpriced. All in all a typical boring power supply!

On Board

The University module, with its 'software' costs a fulsome £256. As this is about £80 more than the likes of a Superboard II, with its BASIC and 8K of user RAM, we are entitled to ask searching questions of the Texas package. For a start what do you get for your £256?

Well, as you can see from our photos, the PCB is well produced and beautifully constructed. Its contents consists of:-

1. Alpha-numeric keyboard (45 keys)
2. Piezo-electric sound output device
3. TMS 9980A 16 bit MPU
4. 4K ROM (expandable to 6K)
5. 1K RAM (expandable to 2K)
6. 2M clock circuitry
7. Cassette I/O
8. 16-bit programmable I/O and interrupt monitor (type TMS 9901)
9. LED display (seven segment)

Keyboard:- 45 keys with a 'shift' facility which allows for 87 ASCII characters to be input.

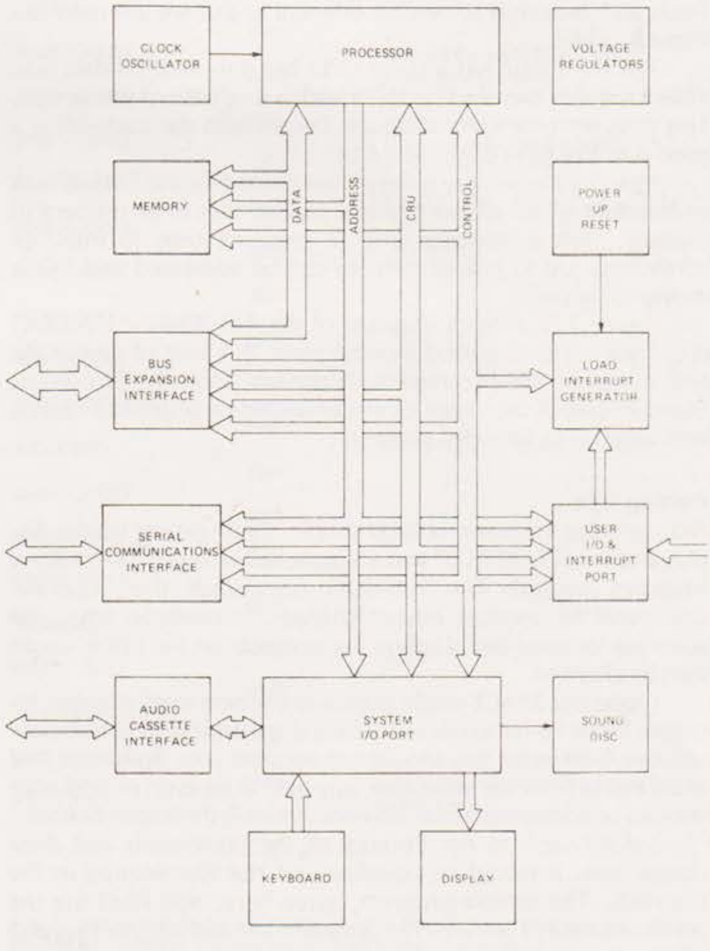


Figure 1-2. System Block Diagram

Fig 3. (Right) Block diagram of the Texas system.

Speaker:- under program control, operates on command. Has a limited sound range, but is a useful peripheral nonetheless.
ROM:- the on-board 4K holds the UNIBUG monitor and 'symbolic assembler' as firmware. There is an expansion socket to hold a user programmed 2K PROM.
Cassette Interface:- use of the TM 990/802 Software Development Board is possible with this, and the cassette I/O is compatible. There is space on PCB for a control relay to be mounted.
LED Display:- the main display shows nine characters out of the 64-character string, and can be shifted left or right to show any nine of the string without affecting store contents.

In addition there are four LEDs on board for general purpose monitoring of CRU, (Communications Register Unit) which allows for single bit I/O, (the CRU is internal to the TM 9980A) and program control monitoring. Three of the four LEDs are for monitoring specific functions (SHIFT etc) under UNIBUG control.

In addition to all this there's a very important little switch hidden away on the board labelled 'LOAD' which is a lot more use than simply loading onto TAPE. The switch generates a non-maskable interrupt to the CPU. This causes discontinuation of execution of current program, and releases control to the UNIBUG monitor. Memory contents are not affected.

A sort of final overide command, which can be used to bring the CPU out of a loop or just generally make it listen to you a bit better! As this brings us around to the monitor, lets take a look at UNIBUG.

Monitoring Around

Table 1 gives the list of the commands available through UNIBUG. In the same EPROM lies the assembler used to provide the TM 990's basic (no pun intended) language. Since the 9980A is a 16-bit beast, its instruction set is very powerful. In addition Texas architecture is somewhat different to that we are used to - to put it mildly.

The TM 9980A has a 16-bit CPU, but only an 8-bit data bus. Thus it requires two read cycles to fetch a single-word instruction. This does limit the chip, although Texas claim the trade-off is a good one. We have our doubts.

Memory-to-memory is the phrase coined for the TMS 9980A architecture which allows multiple register files to be resident in memory, with a resulting drop in response time to interrupt commands. Up to 16K of memory can be addressed and I/O is memory mapped.

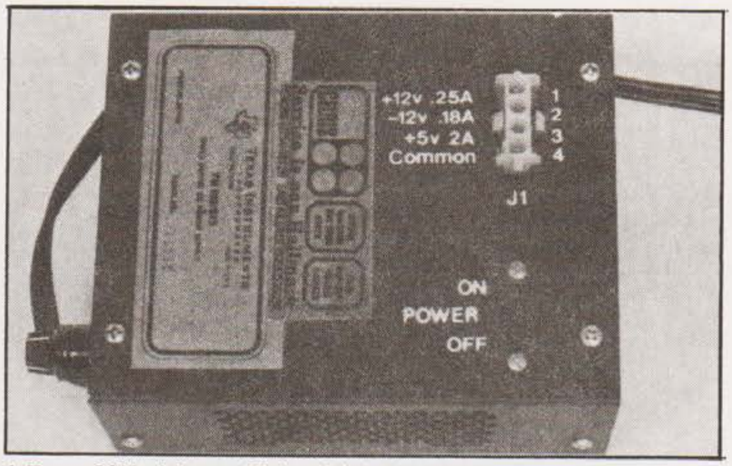
Figure 3 is a block diagram of the TM 9980A. UNIBUG could not fail to be a good monitor given this kind of start of life and it was no disappointment. It confers upon the University Board an ease of use - even given the limited on board I/O - that is well suited to its intended purpose.

Putting It In

Programming the board is fairly simple. Upon power up the display shows 'CPU READY' and a simple RETurn command allows keyboard control. The UNIBUG commands then operate. Command 'M' (memory inspect/change), for example, opens the specified location and displays the contents on the LEDs. It can then be changed.

Operating SPACE single steps into the next even number location. Since 16-bit words are used and are organised as two consecutive 8-bit bytes this should not surprise you. Both byte and word instructions are allowable, any byte at an even or odd address can be addressed by the different modes in the instruction set.

I don't wish to run through all the commands and their usages here, it would be pointless and not illuminating in the slightest. The sample program, given here, will illustrate the points necessary I believe. The program is to add 33_{10} to 15_{10} and display the result.



A Texas PSU. It is so efficient it's boring.

Does It Or Doesn't It?

It is not possible here to do more than simply scratch the surface of the TM 990 board, a detailed description would fill an issue all by itself. The important point, though, remains whether or not it fulfills its design aims and does it in a way which represents value for money to the purchaser.

The aim is to provide an introduction to the MPU technology and to open a door through which some hands-on experience can be gained for serious students. We suspect the pricing level is set thus in expectation of an industrial or academic purchaser rather than a home hobbyist.

The tuition manual is pretty good. Very American and a little vague who it is talking to sometimes, but very good nonetheless. The link to the TMS 990/189 is well forged, and the two complement each other well.

Drawbacks are few, but significant. For a start the keyboard does not have the SHIFTed designations marked on it, and they only exist at all on one page of the manual - incredible! Tsk tsk. Zero for usage there Texas.

The main drawback though, we feel, is simply the TMS 9980A itself. There is no doubt as to the power of this processor - indeed it shows very clearly how far these components have come since their introduction - but in this context it may be too a-typical to be generally useful. Use of the board certainly taught me a lot about use of that CPU, and 16-bit hardware in general, but I feel it would be a difficult transition for a student to make from these giddy heights of flexibility and power down to the more usual 8-bit 6502s and the rest.

The TM 990/189 makes a superb evaluation kit though.

Summary

So that is it. A well constructed and thought out package with versatile on board I/O and a powerfu! processor. A board which makes an excellent tutorial tool - but only in teaching its own subject - the Texas Instruments CPUs. Fair enough, I suppose, but be

Input	Results	Paragraph
A	Assembler Execute	3.3.3
B	Assembler Execute With Current Symbol Table	3.3.4
C	CRU Inspect/Change	3.3.5
D	Dump Memory to Cassette	3.3.6
E	Execute to Breakpoint	3.3.7
F	Status Register Inspect/Change	3.3.8
J	Jump to EPROM	3.3.9
L	Load Memory from Cassette	3.3.10
M	Memory Inspect/Change	3.3.2
P	Program Counter Inspect/Change	3.3.11
R	Workspace Register Inspect/Change	3.3.12
S	Single Step	3.3.13
T	"Typewriter" Program	3.3.1
W	Workspace Pointer Inspect/Change	3.3.14
Ret	New Line Request	3.3.15

Fig 4. UNIBUG command set.

T M990/189 UNIVERSITY MODULE

Some of the software which arrived with the TM990/189

aware of the limitation. The tutorial manual is very good and possessed of only a few minor errors. These are two Fig 1-19s for example and no 1-29. Let he who is without printing error cast the first dictionary.

The final question - value for money? I think not compared to what else is available for the price, but then educational courses are are always expensive. This one is good in its own way and in the end you must decide for yourself if it is worth your pounds.

Our thanks to the distributors, Celdis of 37/39 Loverock Road, Reading, Berks RG3 1ED for loaning us the TM 990/189 and PSU for this article. All enquiries concerning the module should be addressed to them.

a. Problem:
Write a program that will add 33_{10} and 15_{10} and display the answer.

b. Program Solution:

LWPI	>0300	Load immediate to workspace pointer.
LI	0,33	Load R0 with first number (33_{10})
LI	1,15	Load R1 with second number (15_{10})
A	1,0	Add, answer in R0 (memory address 300_{16})
XOP	0,10	Display contents of R0
XOP	1,13	Turn display on

c.

Program		Address	Hex Contents
LWPI	>300	0200	02E0
		0202	0300
LI	0,33	0204	0200
		0206	0021
LI	1,15	0208	0201
		020A	000F
A	1,0	020C	A001
XOP	0,10	020E	2E80
XOP	1,13	0210	2F41

d. To enter the previous program:

1. Apply power to the TM 990/189
2. The TM 990/189 will energize in a power up LOAD state and the display will show CPU READY.

DISPLAY	ENTER	COMMENTS
CPU READY_		
	(Ret)	
?_		UNIBUG commands can be entered now
	M	Memory Inspect/Change
?M_		
	200	M.A. 0200
?M 200_		
	(Ret)	
0200 = XXXX_		Current Contents M.A. 0200
	02E0	Enter New Contents
XXXX 02E0_		
	(Sp)	Advance to Next M.A.
0202 = XXXX_		Current Contents M.A. 0202
	0300	Enter New Contents
0202 0300_		
	(Sp)	
0204 = XXXX_		
	0200	
XXXX 0200		
	(Sp)	
0206 = XXXX_		
	0021	
XXXX 0021_		
	(Sp)	
0208 = XXXX_		
	0201	
XXXX 0201_		
	(Sp)	
020A = XXXX_		
	000F	
XXXX 000F_		
	(Sp)	
020C = XXXX_		
	A001	
XXXX A001_		
	(Sp)	
020E = XXXX		
	2E80	
XXXX 2E80		
	(Sp)	
0210 = XXXX		
	2F41	The entire program has been entered at this point
XXXX 2F41		

Fig 5. An example of how easy to use the TM990/189 can be. The UNIBUG monitor cannot be praised highly enough.

BUY nascom-2 NOW AND GET A FREE 16K RAM BOARD

The lack of availability of the MK4118 RAMs has seriously delayed the launch of the Nascom 2, so we have decided to relaunch the product with an offer few will be able to refuse.
The Nascom 2 will be supplied without the optional user 4118s. Instead, we will supply a 16K dynamic RAM board and the interconnect for the NASBUS – absolutely FREE. This board allows further expansion to 32K. Also, when the 4118s become available, customers taking advantage of this offer can have the 8K for just £80 (plus VAT).

Meanwhile, the empty sockets on the Nascom 2 can be filled with 2708 EPROMs allowing dedicated usage, now with 16, or 32K of extra RAM. All the other features of the Nascom 2 are available and these include:

MICROPROCESSOR

Z80A 8 bit CPU which will run at 4MHz but is selectable between 2/4 MHz.

HARDWARE

12" × 8" PCB through hole plated, masked and screen printed. All bus lines are fully buffered on-board. PSU: +12v, +5v, −12v, −5v.

MEMORY

- 2K Monitor-NAS SYS 1 (2K ROM)
- 1K Workspace/User RAM
- 1K Video RAM
- 8K Microsoft BASIC (MK 36000 ROM)

INTERFACES

New 57-key Licon solid state keyboard. Monitor/domestic TV
On-board UART provides serial handling for Kansas City cassette interface (300/1200 baud) or the RS232/20mA teletype interface.
Totally uncommitted PIO giving 16 programmable I/O lines.

£295 PLUS VAT

The Nascom 2 makes extensive use of ROMs for on-board decoding. This reduces the chip count and allows easy changes for specialised industrial use of the board. On-board link options allow reset control to be reassigned to an address other than zero.

The 1K video RAM drives a 2K ROM character generator providing the standard ASCII characters with additions – 128 characters in all. There is also a socket for an optional graphics ROM on-board.

NASCOM UK DISTRIBUTORS

ADDA COMPUTERS Ealing, London W5. Tel:01-579 5845
BITS & P.C.s Wetherby. Tel: 0937 63744
BUSINESS & LEISURE MICROCOMPUTERS Kenilworth. Tel:0926 512127
THE BYTE SHOP Ilford, Essex. Tel:01-554 2177; London W1. Tel:01-636 0647
COMPUTERLAND Nottingham. Tel:0602 40576; Manchester 1. Tel:061-236 4737; Birmingham. Tel:021-622 7149; Glasgow. Tel:041-221 7409
TARGET ELECTRONICS Bristol. Tel:0272 421196
THE CAMERA CENTRE Barrow-in-Furness. Tel:0229 20473
COMP SHOP New Barnet, Herts. Tel:01-441 2922
COMPUTER MANIA Great Milton, Oxon. Tel:Great Milton 729
C. C. ELECTRONICS Torquay. Tel:0803 22699
DATRON MICRO CENTRE Sheffield. Tel:0742 585490
ELECTRONIC SERVICES Sheffield. Tel:0742 668767
ELECTROVALUE LTD Egham, Surrey. Tel:07843 3603
ELECTROVALUE LTD Manchester M19. Tel:061-432 4945
ELEY ELECTRONICS Glenfield, Leics. Tel:0533 871522
HAPPY MEMORIES Southampton. Tel:0703 39267
HENRY'S RADIO London W2. Tel:01-723 1008
INTERFACE COMPONENTS Amersham, Bucks. Tel:02403 22307
A & G KNIGHT Aberdeen. Tel:0224 630526
LOCK DISTRIBUTION Oldham, Lancs. Tel:061-652 0431
MICRODIGITAL Liverpool L2. Tel:051-227 2535
PHOTO ACOUSTICS Watford, Herts. Tel:0923 32006
PIPS COMPUTER SERVICES Whitley Bay. Tel:0632 482359
P & O COMPUTERS Belfast.
STRATHAND Glasgow. Tel:041-552 6731

nm
Nascom Microcomputers

To: Nascom Microcomputers Ltd.,
92 Broad Street, Chesham, Bucks.
Tel:02405 75155. NM/CT/4

Please send me ______ Nascom 2 kits (complete with construction article and extensive software manual for the monitor and BASIC) at £295 plus VAT plus £1.50 p&p. And ______ 3A PSUs at £29.50 plus VAT plus £1.00 p&p. And ______ optional graphics ROMs at £15.00 plus VAT.

Name ______________________

Address ______________________

Access/ Barclaycard No. ______________________

electronics today international

What to look for in the March issue: on sale February 1st

THE ULTIMATE METAL LOCATOR

Calling all treasure hunters. How many times has your metal detector gone ping or buzz or hello sailor and you've shifted half a ton of Surrey only to find a non-biodegradable ring pull tab? Well, next month we have a discriminating metal locator for you.

The magic machine rejects nails, bottle caps, aluminium foil and ring pull tabs. The design also features full ground effect exclusion over normal or high permeability soils. Search for your pot of gold with deepseeking VLF plus three TR discriminating ranges. Instant tuning recall is made possible by a push button memory circuit.

BLACK HOLES

When a massive star reaches the end of its life, uses the last of its nuclear fuel and explodes as a supernova, one of three things can happen. The supernova explosion may destroy the core, or, if a small core remains, it may become a neutron star, or, if it is large enough, it may collapse to form a black hole.

Next month Ian Graham has a bash at explaining that most enigmatic of astronomical propositions — the black hole.

TV SOUNDS GOOD?

Tired of tinny tunes from your telly? The melodic meanderings start out from the transmitter in super-duper hi-fi, but the cost cutting sounds section of your set takes care of that, lowering the fi at the speed of light. Next month Richard Maybury explores the world of TV sound and comes up with a few ideas on improving it.

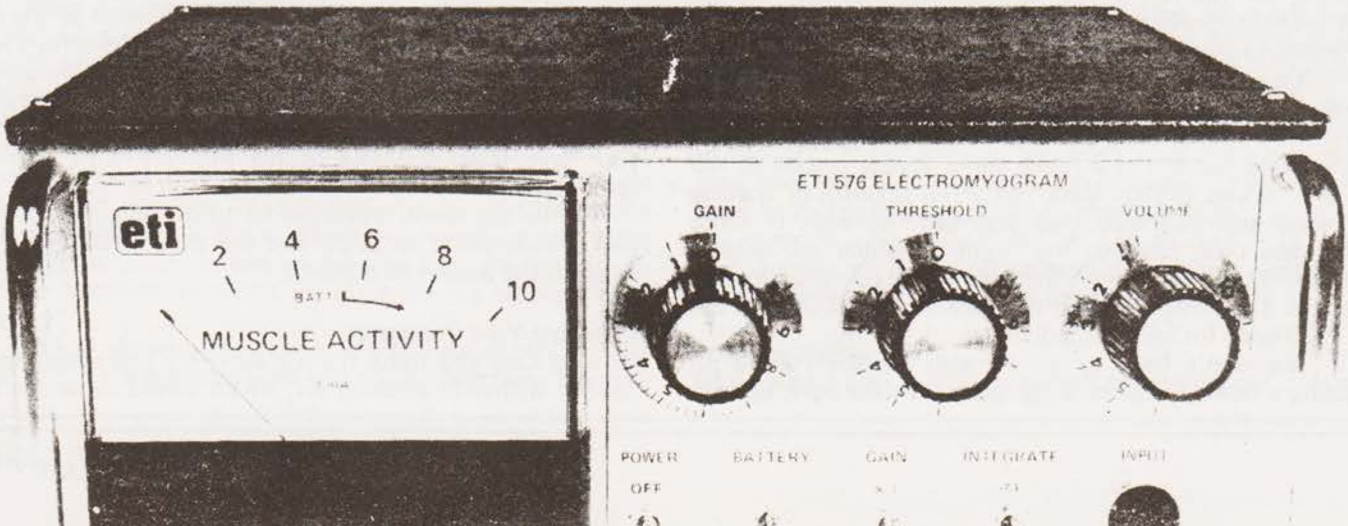
HEATER POWER CONTROLLER

With most heater controllers, your heater is either on or off and the room temperature fluctuates several degrees either side of 'comfy'. Our design will keep your room temperature stable to within half a degree. In addition, by using zero voltage switching, RF interference is avoided.

ELECTROMYOGRAM

The ETI Muscle Meter senses the tiny electrical impulses associated with muscle activity. As Superman flexes his biceps you can hear it all happening and see the activity building up on a meter.

If you're into Biofeedback you can use the ETI Muscle Meter to learn to relax more effectively. On the other hand, if you're into having fun, there's plenty of scope for doing your own thing. Watch this space (give or take a few pages) to find out how the miracle machine picks out the fractions of a microvolt of relaxed muscles from the volts of 50 Hz hum present in the body — induced from power and light wiring.



Ian Sinclair

Having investigated the Mk 14's architecture last month we plunge in with the instruction set.

Human nature being what it is, you've probably been trying some of the programs in the Mk14 booklet. Now if you've been through the book and completely understood what you've been doing, then this series is no longer for you. If, as is more likely, you're more baffled than you were when you started, then read on — this is designed with your needs in mind.

The old adage about walking before you attempt to run holds as true with programming MPU's as in any other activity, so the first exercise we're going to try is a very simple one — adding two one-byte numbers. This is the one we tried very early on with the breadboard unit ; let's see how it's done with the Mk14.

Hex Versus Binary

Two obvious differences emerge right away. One is that we use hexadecimal numbers to represent binary numbers or instruction codes, the other is that we can't start at the lowest address of 0000. Because of the monitor program in ROM and the way in which the RAM addresses are decoded, all the addresses up to 0F12 are spoken for, and just to keep a safe margin, we should start all our programs at 0F20. Why keep a "safe margin"? Answer later, it's all to do with the way we use the memory, folks.

Address	Data	Reminder
0F20	C4	LDI
0F21	1F	first number
0F22	F4	DI
0F23	C2	second number
0F24	C8	ST
0F25	02	at 0F27
0F26	3F	return to monitor

Fig.1. Our first program for the Mk14.

The program is shown in Fig.1, it adds the number 1F (00011111) to C2 (11000010) to produce the answer E1 (11100001). It's a simple enough program, and its importance at this point is that it gets you used to the way in which the Mk14 (and most of its more costly cousins) operates. Switch on and reset. Now tap out the starting address 0 — F — 2 — 0. The dashes aren't part of the number, just a reminder for you not to rush it. Make sure that each key has been properly pressed, and look for the address appearing on the left-hand side of the display.

Once the starting address has been entered, we need to enter data, and to do so we need to press the key marked 'Term'. It's an odd choice of name; the KIM—1 (on which I first cut my teeth) has a much more logical system of AD for address and DA for data, but a few minutes of training will soon convince you that you can live with it. Once 'Term' has been pressed, any numbers which are entered from the keyboard go into memory as data, instructions or numbers. The entry we need to make at address 0F20 is C4, the 8060 code for load-immediate.

We don't have to go through all the routine of selecting a new address now for the next data byte, because the 'Mem' key is a single-step control. Pressing 'Mem' causes the address to change to 0F21 (check that this shows on the address side of the display), and the new data 1F can now be entered. Another jab on the 'Mem' key takes the address to 0F22, and we enter F4, the add-immediate instruction. At 0F23 we then key in C2, the number which is to be added to 1F, and now we have to look for a way to display the answer. Memories of the simple Eurobreadboard system provide a clue, to use the Store command with a memory displacement. At 0F24 then, we input C8, the store instruction, and follow it at 0F25 with 02, so that the result will be stored two places on at 0F27.

Take your time over all this, because it pays to acquire good habits. If you're completely new to it all, you'll probably forget to press 'Mem' at some stage and end up by skipping a step. Microprocessors are utterly unforgiving about errors like this — each step must be 100% correct, so until you really have the hang of it work slowly and think ahead about what you are doing. If you have boobed, press the key marked 'Abort', and then key the address to which you want to return. Press 'Term' again, and you can enter new data at this address, then single-step through the rest of the program, checking what you've entered by using 'Mem'. If this is your first program ever, it's a good idea to use the 'Abort' key to go right back to 0F20, and check each step again.

Running Your Program

Now we've entered a program, how do we run it? There's no point in pressing 'GO' at this stage and expecting something to happen — something might happen, but certainly not what you expect! Why not? The address on the LEDs at the end of the program writing exercise is 0F25, and there's no program starting at 0F25 unless some phantom programmer has been busy. Worse still, there will be a lot of gibberish stored in the memory from 0F26 onwards which could interfere with our program if we let it, so we can't let the program start here. We don't particularly want the machine to run through all the memory steps from 0F26 to the end of the RAM, so it's a good habit to enter 3F as the last byte of each program. Why 3F? That instruction exchanges the program counter with pointer P3, and on the Mk14 and a lot of other 8060-based units, that is a command which causes a return to the monitor program. That way, we don't sweep through all the garbage. The complete program is shown in Fig.1.

How do we run it now? Once again, there's a definite procedure. Press 'Abort'. Despite the name, it doesn't cause everything to clear, it simply lets the keyboard revert to addressing again, so that pressing a key doesn't affect memory. An alternative for this simple program (but not for all others) is to press the red 'RESET' button, which will cause something noticable to happen — it clears the display back to zero. Whichever one you press, the end result is the same; you can now key an address, the starting address for the program which is, of course, 0F20. Whichever way you got there, it's only at this starting address that you can press GO and get the sort of response you expect, unless, of course, you have filled the rest of the memory with NOP (No-operation) instructions.

In the event, the response to pressing 'GO' is fast — the address shifts to 0F27 and the data LEDs display E1, which is the answer to the sum.

Changing Your Program

Now in case you think it's too easy, try this simple modification. Without switching off, which would cause the memory to lose the whole program, press 'Abort' or 'RESET', and dial up the address 0F26. The data byte here is 3F, the return-to-monitor instruction. Press 'Term', and then 0; this has the effect of removing this instruction, leaving the rest of the program unaffected. Now 'Abort' or 'RESET', key in 0F20, and 'GO'. What happens?

The address you end up with is 0022, the starting address of the monitor program, and the data byte is 3F. To get to your answer now, you will have to key in 0F27, the address where the answer is stored. If you use the 3F command at the end of your program, the program *stops* at the step following 3F, displaying your answer. I've put *stops* in italics, because what's actually happening is that the microprocessor is skipping between the monitor program and the last program address, displaying what's stored there.

Address	Data	Reminder
0F20	C4	LDI
0F21	1F	first number
0F22	F4	ADI
0F23	C2	second number
0F24	C8	ST
0F25	03	at 0F28
0F26	3F	return to monitor
0F27	00	to avoid confusion!

Fig.2. The modified programs.

Just rub it in a bit, modify the program again as shown in Fig.2. What's changed? We've simply made the memory displacement 03 instead of 02, so that the answer will now be stored at 0F28. At 0F27, there's now 00 stored, to prevent anything silly happening. What happens when we run this? That's right, the address which is displayed is 0F27, content 00. To get to the answer we have to single-step, using 'Mem', to 0F28, where we decided to put the answer. The *stop* is always one step after the 3F instruction.

To find the answer at the end of a calculation —
1. Store immediately after 3F instruction at the end of the program. Answer is then displayed at end.
2. Leave answer in the accumulator by returning to monitor (for example, use 3F after step C2 in the programs above), then dial up address 0FFD.
3. Store answer at some memory address, and look up this address at the end of the program.
4. Store answer in the extension register, and look up 0FFE.

Fig.3. Where to find your answer.

Now all of this is yawningly obvious to the expert, but you'll have a job to extract it from any of the books which are supposed to help the beginner. Since everyone I've met started as a beginner (even my old mate Sheridan), it all needs to be said. Just in case you've lost track of it all by now, Table 3 sums up all the ways of getting the answer at the end of a program. If you've discovered all of this for yourself, you'll probably be hooked on the Mk14. If, on the other hand you worked it all out for yourself without needing to try it, you're probably a genius, and you'd better emigrate right away. Since I don't write for geniuses (we just telepath) or experts, the next exercise is just one easy step on from the first one. The disadvantage of program number one was pretty obvious — the numbers we are adding are in the middle of the program, and we have to alter the program to alter the numbers. Couldn't we place them a bit more conveniently?

Fig.4 shows a program in which the numbers that are to be added are placed at the end of the program. The pro-

Address	Data	Reminder
0F20	C0	load first number
0F21	06	at 0F27
0F22	F0	add second number
0F23	05	at 0F28
0F24	C8	store. . . .
0F25	04	at 0F29
0F26	3F	return to monitor
0F27	1F	1st number
0F28	C2	2nd number
0F29	. .	Result here.

Fig.4. Locating the data at the end of the program.

cedure for loading this program should be reasonably familiar by now. 'RESET', enter in address 0F20, press 'Term', and then enter in the first data byte C0. From this point, use 'Mem' to single-step the address, and enter each new byte in turn. At the end of the program steps, 'Abort' or 'RESET', key in 0F20 and 'GO'. Why doesn't it stop at the answer?

The reason is that the answer is at address 0F29, but the stopping point is after 3F, and this displays the first of the numbers to be added, not the answer. What's going on?

The difference here is program-relative displacement. Each "do" instruction is followed by a number which refers to a place in memory which is that number of steps on. For example, C0, a load instruction is at address 0F20, and the next byte is 06, at address 0F21. That means that the number which is to be loaded into the accumulator is at a memory address six steps on from 0F21, which is 0F27; it's the number 1F which is one of the numbers to be added. Similarly, the add instruction F0 is followed at 0F23 by 05, meaning that the byte is loaded from address 0F23 + 5 = 0F28, the number C2. The store instruction C8 is followed by 04, so that the answer is at 0F29. To get to this after setting to 0F20 and pressing 'GO', we need to single-step twice.

Address	Data	Reminder
0F20	C0	load. . .
0F21	07	first number
0F22	F0	add. . .
0F23	06	second number
0F24	C8	store. . .
0F25	02	at 0F27
0F26	3F	return to monitor.
0F27	. .	answer displayed
0F28	1F	first number
0F29	C2	second number.

Fig.5. The previous program modified to display the answer.

Could we arrange this more sensibly? Certainly we could, and the modified program is shown in Fig.5. This time, the memory space immediately after 3F is left for the answer, and the input numbers are put in 0F28 and 0F29. Much better — get back to 0F20, press 'GO' and the answer E1 at address 0F27 obediently shows. Why couldn't we just put 3F after the data numbers, and arrange the answer to be in the next byte? Because we don't want the program running over the two data bytes, that's why. These bytes are there to be fetched as data when required. If they are read by the program, one of them at least will be read as an instruction, fouling up the whole scheme. Remember what we said about starting and stopping at the right places?

Working Backwards

Made bold by all this success, let's try placing our data num-

Address	Data	Reminder
0F20	1F	1st number
0F21	C2	2nd number
0F22	C0	load....
0F23	FD	1st
0F24	F0	add....
0F25	FC	2nd
0F26	C8	store...
0F27	02	end
0F28	3F	

Fig.6. Locating the data at the beginning of the program.

bers (to be added) at the beginning of a program, Fig.6. This time we'll have to displace backwards, using negative numbers – in case you've forgotten or never learned, Fig.7 shows how negative numbers are formed. Key in the program in the way which should now be familiar, humming to yourself, reset or abort, ring up 0F20 and 'GO'. What do you get?

Forming a HEX number in easy steps.

Steps	Example
1. Write down the negative number	−12 (decimal)
2. Convert to 8-bit binary, ignore sign	00001100
3. Complement the binary number	11110011
4. Add 1 to lowest place (R.H.S.)	11110100
5. Convert to HEX	F4
6. Write Hex number	F4

Fig.7. How to make negative numbers.

What went wrong? We forgot, didn't we that a program has to start at the beginning, and the beginning of our program is at 0F22, not at 0F20 now. The data byte at 0F20 is a number, 1F, not an instruction, but if we start the program running at this address, 1F will be taken as an instruction. The 8060 is just a chunk of silicon, it doesn't know any better! Reset, and this time make the starting address 0F22. Now when you press 'GO', the correct answer, E1, will appear at 0F29, which is a much simpler way of arranging things.

Doing It Yourself

Now that you've mastered this (you have, haven't you?) you can start on some homework. Turn to page 45 of the S. of C. manual for the Mk14 and you'll see a program for two-byte addition. This program, as the name suggests, is for adding two sixteen-bit numbers. Because the memory stores and the accumulator of the 8060 are only one byte wide, we can read or write only one byte from or two each memory address, so that two-byte numbers have to be split up and stored in two memory addresses. The obvious logical method is to divide each two-byte number into a lower (L) byte and a higher (H) byte. One number is stored with its high byte at 0F20 and its low byte at 0F21; the other number is stored with its high byte at 0F22 and its low byte at 0F23. The two bytes which are the result of the addition are also stored, after running the program, at 0F22 and 0F23, so that subsequent additions can be carried out.

Try running through this program, remembering that to view the answer you'll have to key up address 0F22 to find the high byte of the answer and then press 'Mem' to get the low byte. Now for the challenge. Can you re-design the program so that the high byte of the answer appears at 0F32, and the low byte at 0F33?

A Case Of Amnesia

Now for something completely different, since you're probably fed up with addition by now. You'll have noticed that when you switch on, there are always data bytes in the memory. The reset action does not clear these memory bytes, it only clears the registers of the 8060, and the only way that these memory bytes can be cleared is by writing 00 into each memory space.

Now this would be hard work if we had to dial up each address, set to 00, advance the address using 'Mem', set to 00, and so on through 128 bytes of memory. Fortunately, it's possible to get the microprocessor to do this using the deceptively simple program which is shown in Fig.8. The program starts at 0F12 with the data byte 00 – this needs only one press of the zero key, incidentally, but remember to press 'Term' first, or you'll alter the address instead of entering zero.

Address	Data	Reminder
0F12	00	clears
0F13	CD	stores, auto-indexed to P1
0F14	FF	index, set to decrement
0F15	90	jump...
0F15	FC	back to 0F13

Fig.8. The memory clearing program.

The instruction at 0F13 is to store at an address relative to pointer register number 1 (P1) – but what is the address in pointer 1? If we've just switched on, and that, after all, is when we most need to clear all the memory bytes, then the address in P1 is 0000, so that the store instruction would be relative to this. The index number is at 0F14, and it's FF, equal to −1, so that the address will decrement on each fetch. Since the address is decremented before being fetched, the first address to be put out will be 0000 − 1 = FFFF. This is the first address which will have 00 stored into it, and the next instruction is at 0F15, a jump. At 0F16, the amount of the jump is specified, four places back to 0F13 to carry out the whole operation again. Four places back to 0F13? When there's a jump back, you must make the jump one more number than the number of places you have, because the program counter will increment during the instruction. The result is that when you jump back from 0F16 to 0F13, the program counter is busy going on to 0F17, and four steps of jump, rather than three are needed. Next question? Why did we jump to 0F13 rather than 0F12? This is one of these rare occasions when it doesn't matter too much. The accumulator is cleared by the 00 at 0F12, and ought to stay that way, so that jumping back to 0F13 is quite satisfactory, there's still zero in the accumulator. If you're fussy and you want to go to 0F12, use FB in place of FC at 0F16.

On the next run, the pointer register will hold FFFF, the number which was caused by decrementing 0000 on the first run, and when this is done again, the address will be FFFE, so that zero will be stored at this address. Since the highest order of address lines isn't decoded, the memory positions 0FFF and 0FFE will be the ones which are actually cleared. What stops it? Simple, the program goes down the memory addresses (I nearly said down memory lane) storing 00 until it reaches 0F16. Once it has stored 00 at this address the jump instruction can't work again, and the system returns to the monitor program again, showing the address 0022 and data byte 3F. By this time, every byte of memory from 0F16 upwards has been cleared, and if we now start writing programs at 0F20, we can be sure that we won't encounter any problems from garbage in memory. It's a short program, but an important one to understand, because many operations are based on the idea of auto-indexed loading or storing – more of them later.

MINI-ADS &

CLASSIFIED INFORMATION

Semi-Display:—
1- 3 insertions — £5.00 per single column centimetre
4-11 insertions — £4.50 per s.c.c.
12 insertions — £4.00 per s.c.c.

Classified:—
19 pence per word (minimum 25 words)
Box number £1.00 extra

ALL ADVERTISEMENTS IN THIS SECTION MUST BE PRE-PAID
Closing date:— 2nd Friday in month preceding publication
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request)
Cheques and postal orders should be crossed and made payable to 'Computing Today'

CLASSIFIED ADS, COMPUTING TODAY, 145 CHARING CROSS ROAD, LONDON WC2H 0EE (Tel. 01-437.1002)

INTENSIVE WEEKEND COURSES IN

BASIC

including hands-on mini computer operation.

This short intensive course is intended to instruct from minimal knowledge to an operational capability of computer programming in BASIC high level language. The course is fully residential from Friday evening to Sunday afternoon.
Option of non-residential weekend, weekday evening and weekday courses available if required.

For further details of dates available, fees, etc. **Phone (0401) 43139, or write to:**
Dept CT
CLEVELAND BUSINESS SERVICES
Cleveland House, ROUTH
Beverley, North Humberside

PROMS PROMS PROMS

Blank or programmed with your code. Quick turn-around, low prices, e.g. 2708 (450n Sec)

Unprogrammed	£6.50
Programmed	£9.15
Erased	£0.40
Reprogramming	£3.00
Copied	£1.30

S100 16K Prom board

Kit (less 2708's)	£52.00
Asm and Tested	£61.00

Write for details and price list of selected components.
Add 35p P & P plus V.A.T.

Winchester Technology Ltd.
21 Malibres Road, Eastleigh, Hants. SO5 1DS

J. MORRISON (MICROS)

*BASIC FOR 6800 systems
Powerful arithmetic routines 9 digit. Exp. ± 99
LISTING **£9.00**
DATA 4 x 2708 EPROM **£40.00**
*STANDARD ASSEMBLER (6800) Approx. 2½ K
Supports FCB, FCC, ORG, EQU, RMB, All Motorola mnemonics. LISTING/DATA **£7.50**
*6800 TRACER
Will trace through rom and ram.
Prints out CC, AB, INDX, SP, Address data.
LISTING/DATA **£3.50**
*VARIOUS GAMES FROM ONLY £1.00.
Send S.A.E. for list.
All prices inclusive.
17, Summersell
Bentons Rise,
London S.E.27.
TEL. 01 761 1186

LATEST 8 K WITH LARGE KEYBOARD £480. 16 K **£560.** 32 K **£660.** Dual Floppy **£680**
BASE 2 PRINTER £450
PROGRAMMERS TOOLKIT, £45.
CHALLENGER 1 4K £223, complete with RF converter.
Also Challenger 2 range at very low prices.
***TEXAS TI** 99/4 **£890** with 13" colour Monitor*
The most advanced home computer, delivery March 1980, order now.
AIM65 4K, cased, powered, Basic **£420.**
SEAWELL expansion chassis, memories, etc.
S-100 MEMORY 16K 2MHz FULLY STATIC, assembled, tested — bargain, **£150.**
COMPLETE S-100 SYSTEM WITH Z-80 CPU, North Star double-density disk, 32K memory **£1,200.**
Also other S-100 products.
Full technical support from MAPCON engineers.

INTELLIGENT ARTEFACTS
Cambridge Rd., Orwell, Royston, Herts. Tel. Arrington 689

---

uHEX EPROM PROGRAMMERS

426 2508/2708/2758/2516/2716
Dual and Single supply Eproms, **£95**
416 2704/2708/2716 Dual only, **£65**
480 2704/2708 Kit **£35.** Built **£40**

All programmers require only standard power supplies.
The 426 and 416 are cased and have push-button selection.
Program any length block into the Eprom.
Software included. Range covers Z80, 8080, 6800 and 6500. State machine.

PIO, PIA INTERFACE MODULES
Available for Z80/8080 and 6800/6500.

Prices include carriage. Please add VAT. SAE for further product information.

MICROHEX COMPUTERS
2 Studley Rise, Trowbridge, Wilts.

COLOUR MODULATOR

Kit **£9.95** inc. UHF Modulator
FOR ALL TV GRAPHICS!
Red, Green, Blue inputs (can be mixed)
Coming soon. Nascom Colour Kit **£45.00** inc. Graphics Software

WILLIAM STUART SYSTEMS
Dower House, Billericay Road, Herongate, Brentwood, Essex CM13 3SD. Tel. (0277) 810244
Barclaycard/Access welcome

---

50 Hz SUPERBOARDS BRITISH MODEL
from **£190** plus VAT
Fully built, tested and set up
Authorised dealer backup
Free cursor control, backspace, etc. taps
C.T.S. 1 Higher Calderbrook, Littleborough, Lancs.
Tel. Littleborough 79332 anytime

FOR SALE, Data Dynamics ASR33 on stand, comprising punch, reader and printer. Modem/RS232 interface with cable. Recently serviced and in excellent condition. £300 ono. Ruislip 72852 after 6.30 p.m.

TRITON COMPUTER. Fully built and tested. With full on-board memory, and a number of programs on cassette. £300 o.v.n.o. M. Kuht, 44 Mayfair Road, Cowley, Oxford. 775706.

T.V. TYPEWRITER/REMOTE TERMINAL. For low cost Video Terminals or T.V. Typewriters get NEWTRONICS Video Board kit £70 inc. P & P. Full keyboard terminal kit — £140 inc. P & P. Learn computing with the ELF computer kit, ideal for beginners — £79 inc. P & P. SAE details Penman, 8 Elliothill Street, Dunfermline, Fife.

NASCOM 1, in Verocase, with fan, PSU, Nasbug T2 and CCS level 'A' Basic, in working order, £160 o.n.o. Ring Pipe Gate 063-081-312 after 6 p.m.

CLASSIFIED

ADVERTISEMENT INDEX.

8MHz Super Quality Modulators	£4.90
6MHz Standard Modulators	£2.90
C12 Computer Grade Cassettes 10 for	£4.00
Super Multi-rail P.S.U. +5 −5 +12v	£29.50
Nascom I with Nas-sys	Special Price
Kit } Limited quantities	£125.00
Assembled }	£140.00
ETI Breakout Game — Chip and PCB	£9.90
S100 Expansion Motherboard for Nascom I	£39.00
Anadex Printer Paper — 2000 sheets	£25.00
Floppy Disks 5¼" Hard & Soft Sectored	£3.50
Floppy Disk Library Case 5¼"	£3.50
Lexicon Language Translator	£125.00
Modules for Lexicon	£29.00
Eprom Boards	£63.00
8K Static Ram Boards — S100	£110.00
Grandstand Video Game	£59.00
Cartridges for Grandstand	£11.99
George Risk Ascii Keyboard	£39.00
Cartridges for Atari — Full Range in Stock	£13.90
Interface PET IEEE — Centronics Parallel Not decoded	£49.00
Decoded	£77.00
Interface to Centronics parallel for TRS80	£75.00
Verocases for Nascom 1 & 2 etc.	£22.50
Keyboard Cases	£9.90
Electric Pencil for TRS80	£29.00

HITACHI PROFESSIONAL MONITORS

9" — **£129**
12" — **£199**

• **Reliability** Solid state circuitry using an IC and silicon transistors ensures high reliability. • **500 lines horizontal resolution** Horizontal resolution in excess of 500 lines is achieved in picture center. • **Stable picture** Even played back pictures of VTR can be displayed without jittering. • **Looping video input** Video input can be looped through with built-in termination switch. • **External sync operation** (available as option for U and C types) • **Compact construction** Two monitors are mountable side by side in a standard 19-inch rack.

only **£399** + VAT

TRS80 LEVEL 2 16K

Fully converted to UK T.V. Standard. Comes complete with easy to follow manuals. UK Power Supply — Cassette Leads — Sample tapes. Special box to enable you to plug into your own TV. Recommended for first time-buyers. Just plug in and go. **Full Range of Software Available**

NEW REDUCED PRICES

8K **£449**
16K **£549** + VAT
32K **£649**

RRP £795 for 32K

The PEDIGREE PETS

Very popular for home & business use. 8K Microsoft Basic in ROM. 8K Pet 32K & 16K with new improved keyboard. All with green screen.

Extra cassette deck **£55** Full range of software available

32K **£690** + VAT

48K **£790** + VAT

EXIDY SORCERER

RRP £859 for 32K

For Personal or Business Use.
32K or 48K memory. 8K Microsoft Basic in ROM. Dual Cassette I/O, RS232 I/O. Parallel I/O (Centronics). Expansion available through optional extra S100 Motherboard. 69 Key keyboard including 16 key numeric pad.

S100 EXPANSION — **£199**

RRP £975 for 48K

48K — **£690** + VAT

ITT 2020

Full colour into your own colour T.V. PALSOFT 9 digit floating point Basic in ROM as standard. Low and High resolution graphics. Built in loudspeaker. Compatible with most Apple Hardware add-ons and software.

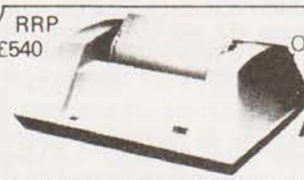
only **£295** + VAT

TRS80 EXPANSION INTERFACE

Expand your TRS80 by 32K **32K**
32K Memory on board. Centronics parallel port. Disk controller card. Real time clock. Requires Level II Basic. Interface for 2 cassette decks complete with power supply.

RRP £540

only **£499** + VAT



ANADEX DP8000

Super Quality — Low cost printer. Tractor Feed with full 96 ASCII character set. Accepts RS232C at band rates between 100 and 9600 and Parallel Bit data.
Attaches either directly or through interfaces to Pet, Apple, TRS80, Sorcerer, Nascom, Compukit etc.

RRP £690 only **£590** + VAT

IBM SELECTRIC GOLFBALL

Refurbished IBM Golfball Printers to new specs. Accepts Centronics parallel data. Friction feed. Prints at 15 cps.
Attaches either directly or through interfaces to Pet, Apple, TRS80, Sorcerer, Nascom, Compukit etc.

video 100

12" BLACK & WHITE LOW COST VIDEO MONITOR

RRP £79

only **£69** + VAT

• Ideal for home, personal and business computer systems • 12" diagonal video monitor • Composite video input • Composite video input • Compatible with many computer systems • Solid-state circuitry for a stable & sharp picture • Video bandwidth - 12MHz + 3DB • Input impedance -75 Ohms • Resolution - 650 lines Minimum In Central 80% of CRT; 550 Lines Minimum beyond central 80%.

NASCOM-2 MICRO-COMPUTER

only **£295** + VAT

Microprocessors Z80A 8 bit CPU. This will run at 4MHz but is selectable between 1/2/4 MHz. This CPU has now been generally accepted as the most powerful, 8 bit processor on the market.

INTERFACE

Keyboard New expanded 57 key Licon, solid state keyboard especially built for Nascom. Uses standard Nascom, monitor controlled, decoding.

T.V. The lv peak to peak video signal can drive a monitor directly and is also fed to the on board modulator to drive the domestic T.V.

I.O. On board UART (Int 6402) which provides serial handling for the on board cassette interface or the RS232/20mA teletype interface.
The cassette interface is Kansas City standard at either 300 or 1200 baud. This is a link option on the NASCOM 2.
The RS232 and 20mA loop connector will interface directly into any standard teletype.
The input and output sides of the UART are independently switchable between any of the options.
i.e. it is possible to house input on the cassette and output on the printer.

PIO There is also a totally uncommitted Parallel I/O (MK3881) giving 16, programmable, I/O lines. These are addressable as 2 x 8 bit ports with complete handshake controls.

Documentation Full construction article is provided for those who buy a kit and an extensive software manual is provided for the monitor and Basic.

Basic The Nascom 2 contains a full 8K Microsoft Basic in one ROM chip with additional features like DEEK, DOKE, SET, RESET for simple programming.
With free 16K RAM board.

A PROFESSIONAL WORD PROCESSING SYSTEM AND IT'S A COMPUTER AS WELL.

FOR ONLY
£3250 + VAT

PET 32K — This is the standard 32K Pet from Commodore. Reverse video and graphics allow the WordPro Package to give simple clear and easy to read displays.

2040 Disk Drives Twin disk drives allow large high speed storage for your letters, or paragraphs. Plugs in the back of the PET.

NEC Spinwriter NEC's high quality printer uses a print "thimble" that has less diameter and inertia than a daisy wheel, giving a quieter, faster, more reliable printer that can cope with plotting and printing (128 ASCII characters) with up to five copies, friction or tractor fed. The ribbon and thimble can be changed in seconds.
55 characters per second bidirectional printing — with red/black, bold, subscript, superscript, proportional spacing, tabbing, and much, much more.

WordPro II The heart of the system — consists of a ROM and diskette. The ROM is inserted into a space socket inside the Pet. One of the most versatile Word Processing Packages around.