

VOL. 1  
NO. 7

# STATUS 1500

AUGUST  
1983

\* 12 issues £10.50 (UK) \* overseas £14.50 \*  
Published Monthly

\* Edited by RONALD COHEN, 62 BLENHEIM CRESCENT, LONDON W.11 \*

## SHAMBLES AT STATUS 1500

Having hitherto boasted of care and attention in the avoidance of errors, I am obliged to enclose a correction sheet, revised page 42, and revised page 61. I must express my profound - apologies. I must also thank John Kerr and others for pointing these errors out.

\*\*\*\*\*

Please note new subscription rates above. I had not wished to raise rates yet, but (as you realise) the newsletter is almost always thicker than originally intended, thus costs more to produce, and very much more to post. Also the 6 months subscription, which was an introductory offer, is being phased out.

However at present the new rates only apply to new subscribers. \*\* Current subscribers, including those joining before 1st September, may subscribe or renew at the old rates. \*\* Indeed, anyone may renew at the old rates for any period ahead they care to risk. \*\*

\*\*\*\*\*

\*\*UNTIL 1st SEPTEMBER 1983

I am sorry to tell you that all relations with SHARP have now been broken completely, after a stormy interview at which I was accused of being offensive and underhand. This means that I shall no longer be able to bring you various snippets of news as in the past: it also means that I shall be spared the agonies of frustration involved in trying to get some sense out of SHARP. At times like these I almost feel like giving up: and would seriously think of doing so, were it not for so very many kind letters of support from subscribers all over the world. These are indeed appreciated. However if any subscriber feels he would like to take over from me the burdens of this newsletter, I would be happy to discuss this. I am not able to convey to you the contempt with which SHARP have spoken of your enthusiasm and interest in their products, on the grounds of your restricted number and influence. They are naturally interested in volume sales, and refuse to admit that your goodwill, or your opinions, are of the slightest value.

## \*\*\* CONTENTS \*\*\*

63 SHAMBLES AT STATUS 1500  
64 LETTER FROM SHARP  
65 DISTRESS SIGNALS  
66 PEEK, POKE & MEMORY - VII  
67 INTENTIONS  
67 MINDBOGGLE CORNER  
68 AND AND OR

69 DELETE  
69 SUPERSALVAGE  
70 SIXPACK reviewed  
71 MC ROUTINES locations  
71 MARKET PLACE information  
72 MNEMONICS explained  
73 Advertisement

Do not sell this PDF !!!



# SHARP

Registered in England No. 255677

SHARP ELECTRONICS (U.K.) LTD.  
Sharp House, Thorp Road,  
Manchester M10 9BE.  
Telephone: 061-205 2333.  
Telex: 668380 (Sharp - G)  
Telegrams: Sharptron, Manchester

Our Ref. AM/PGW

Your Ref.

Date 20th July, 1983

Mr. R. Cohen,  
62 Blenheim Crescent,  
London W11.

Dear Mr. Cohen,

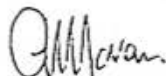
As promised Sharp Electronics (UK) Ltd., have now released the PC-1500 Technical Reference Manual, which will be available through our Calculator Service Department, at a price of £20.00 inclusive.

This manual was specifically written to try to answer the many questions we have had regarding machine Code Language on the PC-1500, you will also find it contains other useful information such as the circuit diagrams for the PC-1500 and its peripherals.

We do hope that the enclosed manual which is being sent to you with our compliments, will prove to be useful and hopefully make your monthly newsletter, of which I am a subscriber even more interesting.

Assuring you of our best intentions at all times.

Yours sincerely,



A. Moran,  
Calculator Service Department.

\*\*\*\*\*

After the incidents described on the front page of this newsletter, I was more than amazed to receive the letter from SHARP reproduced above. The "Technical Reference Manual" is indeed the same as the expected "Programming Manual": I am delighted that I have lost my bet (see page 51).

At first sight the Manual contains a wealth of information; not only Machine Code in great detail, but also a number of detailed Memory Maps, Circuit Diagrams, and some commands missing from the normal Instruction Manual. The explanations of these commands tends to be in inscrutable Janglish, unfortunately, and the price, which is 50% higher than in USA, will deter most would-be purchasers.

A full review will appear in September STATUS 1500.

ROGER SVENSSON, MALCOLM RAY and JOHN KERR all point out that in fact the ON key does not return 0, which was merely returned because no key had been pressed after turning on: it returns 14 (although this is not actually its ASCII code): and send me machinecode programs proving my error on page 54. C.NORTH points out that my program on page 55 can be beaten by keeping a finger on the ON key.

*I shall avoid the ON key for a while.*

SIMON COX recommends the TANDY Service Manual (about £3) as far more informative than the SHARP version.

K.SOUTHGATE has a problem with a program which requires entering a formula, and tabulating the results. When there is an error the original formula is irretrievable.

*You need an EVAL routine. The first line of your program should be*  
10: REM ..... (80 dots)  
*Enter each formula as a string (A\$), and then poke each character of the string into the REM line, and RUN in the usual way. This is the principle of it, but must be written with exactitude.*

BENJAMIN THUM of Kuala Lumpur has written a suite of architectural programs, and would be happy to supply information on these to any reader seriously interested. As well as the PC 1500 he uses a NEWBRAIN, and would like to get in touch with any reader who uses the same combination.

PETE ELDRIDGE asks me whether - if money were no object - I would prefer to extend memory with some CE 159s, or with the WAGNER conversion.

*Its a question of 'horses for courses'. For some applications I would like 26K memory; on the other hand I have a number of 8K programs which I would very much like to plug in. I believe WAGNER do a substitute CE 159 for use with their conversion: on the other hand their Toolkit modules could no longer be plugged into this space. Anyway, if money were no object, I would probably buy a different machine, which was better supported.*

THORKILD RASMUSSEN writes from Denmark to say that the memory in his PC 1500 is full up, and since he has no printer or cassette-recorder he has to buy a new computer, and the new one will probably be a TEXAS CC40 and will I consider writing about other machines as well as the PC 1500 ?

*I cannot help being reminded of the Texas millionaire who bought a new Rolls-Royce whenever the ashtrays were full on his present one. Before getting a new computer for every program, why not consider the WAGNER extensions; or surely the CE 159, or several of them, would serve your purposes. As for writing about other machines in a newsletter dedicated to the PC 1500, I cannot write about computers to which I do not have access; but if you care to send me a few I will be happy to write about them.*

A number of readers write of successes in connecting printers to the PC 1500 via the CE 158. ANDREW HALSEY has connected a Tandy Color Printer, K.SOUTHGATE has connected a Seikosha, and a Smith-Corona printer/typewriter, while an Epson MX80 F/T II has been connected by I.LITTLEWOOD. C.NORTH has also been successful, but does not specify the printer! If any reader wishes to use one of the machines mentioned above, I will forward any queries to the above experts. However N.GORMAN has been trying to make an Olivetti 41 work with the CE 158, and no less than 6 technicians have retired defeated. [LATEST NEWS - the 7th technician succeeded!!!]

These are two very powerful tools, either used separately or in conjunction with each other.

NEW nnnnn (nnnnn stands for an address) can be keyed before writing or CLOADing a program. The program will then start at the address you have specified, and not at the normal start-of-program-area. For instance, if you wished for space to hold a machine-code routine, 100 bytes long, and assuming you have the 8K extension, so that programs start normally at 14533, you could key NEW 14633, and then any BASIC program you write or CLOAD will start at 14633, leaving space for your m/c routine between 14533 and 14633. Your BASIC program will RUN etc in the normal way, and you will not be aware of its strange start address. It can be wiped out by keying NEW, and the space you reserved for the routine will still be protected: the protection will only be removed by NEW 0, or NEW nnnnn, being the original start-of-program-area address.

This technique can also be used in another way. Quite often you have a long program in memory, and wish to retain it: but meanwhile also want to work on a short experimental program; you do not wish to start with enormous line numbers to distinguish it from the first program, nor to wipe out the earlier program in order to work on the new one.

This is now simple. Before starting writing or CLOADing the new program, key NEW STATUS 2. The old program disappears, and the new one is written or CLOADed in the normal way. When you have finished with it, key NEW, then key NEW 0. Everything disappears. Then key CALL 14344 (or wherever you have hidden the Simon Cox "Instant Rescue" described on page 56). And quite miraculously the former long program reappears, and you are back where you started!

Unlike MERGE, there is no interference with RUN or editing, and the only disadvantage is that (after NEW STATUS 2) the expression STATUS 2 - STATUS 1 can no longer be used as reference point for the start-of-program-area address.

CSAVE M is used to lift a section of code from a specific address, and the converse CLOAD M to place that chunk of code anywhere you like!

The forms are a) CSAVE M "FILENAME"; nnnnn, mmmmm

*(these being the start and end addresses of code, inclusive)*

b) CLOAD M "FILENAME"; nnnnn

*(being the new address at which you want the code to start)*

The forms must be followed exactly. Using these facilities, you could store your machinecode routines on cassette, and then put any one of them into memory anywhere convenient, such as template area, variable areas, or space reserved by the NEW nnnn technique described above.

The CSAVE M and CLOAD M techniques could also be used to lift a chunk of a program, such as a mass of DATA statements, and place it at the end of another. In this case you would have to be very careful of your addresses, and since this is not the same as CLOADing a program, the system pointers are not automatically adjusted, nor is the end-of-program-marker, so you would have to adjust these manually, and exactly.

At first sight, this all seems rather complicated. But work through it slowly and carefully, and it will soon make sense: and is a wonderful bonus for anyone who wishes to have power over the machine he is using.

## INTENTIONS

=====

Several very interesting communications from readers have called in question the function of this newsletter, and therefore I take this opportunity to re-affirm what I believe its function should be. This is to help readers to know the full capabilities of the computer they are using: to point out elementary tricks they may have missed: to publicise what hardware and software are available: to put readers with similar interests in touch with each other: to criticise, and to entertain.

It has been suggested that too much space is devoted to exotica such as STATUS CODES and not enough to useful programs such as DISCOUNTED CASH FLOW. This is backed up by the suggestion that since there is no big-screen access, the PC1500 is not games-orientated, and therefore it is programs for serious use that are required. I would answer this with the fact that most minor business applications can be discovered in a number of books, whose BASIC is minimally different from ours: conversion is easy. In any case, a definition of <serious use> can hardly be limited to purely managerial use: doctors, pilots, divers, engineers and scientists have other interests besides cash flow.

On a more philosophical level, it has been suggested that all these PEEKS and POKES and so on are abhorrent to a classical programmer, who wishes to write programs which are clearly comprehensible, and easily maintained by another user. I must reply that simple structured programs are beyond the capabilities of most of us. Our programs tend to be experimental, and grow like cultures. Perhaps they should be rewritten, but we do not have time.

Two further objections: firstly, BASIC does not lend itself easily to structured programming: and <clever> programming is often necessitated to cram programs into the very limited memory. Slowness of CLOADing is another motive for keeping ones programs short at any cost. Secondly, the exotica dealt with in the newsletter are needed because of the enormous gap between the capabilities of the machine, and the limited BASIC delineated in the manual.

A newsletter such as this is intended to help you get the best out of your computer, and to enjoy it. It is not intended as a cheap source of business software for managers who are too busy to write their own programs, and too economical to purchase them. It is my job to help you find the tools, and to help you use them. It is not my job to dig your ditches for you.

---

### MINDBOGGLE CORNER

The winner of the JUNE competition is C.J.NORTH (see page 45). Here is his solution. The ingenious statement COLOR 3 AND N is not strictly necessary, but makes it clearer how the pattern is developed.

```
1:POKE 31220,36:
  FOR N=64 TO 72:
    COLOR 3 AND N:
    LPRINT CHR$ N:
  NEXT N
```

The clue was on page 42 (ASC \* is 42).

No competition this month. MINDBOGGLE will resume in September.



## OR AND AND AND AND AND OR

By which, of course, we mean : "OR and AND" and "AND and OR".

But the computer cannot hear the tone of voice in which the above are pronounced, and does not understand at all. A difficulty with AND and OR is that they are used in a way very different to their ordinary conversational meanings. In particular, AND does not mean +. In fact, its Boolean equivalent is \*. This seems strange, and the rest of this page will try to explain.

You remember that if a statement is true, it returns 1, and if false it returns 0. (see page 53). So

```
10: IF (A=X) AND (B=Y) .....
```

has given 1 for each of the expressions in brackets. For the IF to be fulfilled, both must be true, i.e., both must return 1, so that the whole IF statement returns 1, and this is the same as

```
10: IF (A=X) * (B=Y) .....
```

In both cases, if either statement is false, and returns 0, then the whole IF statement returns 0 (false).

It is OR that works the same as +

```
10: IF (A=X) OR (B=Y) .....
```

needs only one true expression. Thus

```
10: IF (A=X) + (B=Y) .....
```

will return 1 if either expression is true.

But the computer has in fact not multiplied or added, with AND and OR. It has done a "logical AND" or a "logical OR" with the 1s and 0s returned.

We must here return to considering Binary. There is a hint of it on page 42. Take any two numbers in their binary form

```
      11001110
AND   10010100
gives 10000100
```

Imagine that the computer puts 1 only where 1 exists in the same position in both numbers. Whereas when using OR the computer gives 1 where there is a 1 in that position in either number.

```
      11001110
OR    10010100
gives 11011110
```

Try this on a few numbers. Key 85 AND 99. Then try 85 OR 99. Work out what they are in Binary, and the decimal equivalent of the Binary results, and you will see how it works.

It is particularly useful with Screen Graphics. Try keying the following:

- a) GPRINT 85; 99
- b) GPRINT 85 AND 99
- c) GPRINT 85 OR 99

Take any value you like for X up to 127. Key Y=127 - X. Then key

- a) GPRINT X; Y
- b) GPRINT X OR Y
- c) GPRINT X AND Y. ( This gives a blank screen. Can you see why?)

MERGE this program with the subject program and execute by DEF D

Time: to delete 8K from 9K program: about 3 minutes  
to delete 1K from 9K program: about 7 minutes

```

100:"D"INPUT "1st
      line=";A
110:INPUT "last li
      ne=";B
112:S=STATUS 2-395
115:CLS :WAIT 0
117:F=STATUS 2-
      STATUS 1-1:
      GOSUB 500
120:FOR F=STATUS 2
      -STATUS 1+3TO
      S
130:IF PEEK F=13
      GOSUB 500
140:PRINT F
150:NEXT F
160:L=K-J:BEEP 1,2
      00,200
170:FOR F=KTO S
180:POKE F-L,PEEK
      F
185:PRINT F
190:NEXT F

210:T=S-L-1:U=INT
      (T/256):V=T-25
      6*U
215:POKE 30825,
      PEEK 30821,
      PEEK 30822
220:POKE T,255:
      POKE 30823,U,V
230:BEEP 3:END
500:Y=256*PEEK (F+
      1)+PEEK (F+2)
510:IF Y=ALET J=F+
      1:BEEP 1
520:IF Y=BLET K=F+
      1:BEEP 1:K=K+3
      +PEEK (F+3):F=
      S
525:F=F+PEEK (F+3)
530:RETURN

STATUS 1
395

```

In September : superfast Machine Code version of this program! [if debugged]

#### SUPERSALVAGE

This is a ~~speedier~~ automatic version of SALVAGE on page 25.  
In certain rare cases it can fail; but what have you got to lose?  
Normally it will salvage up to the end of the last uncorrupted line.  
FROM THE KEYBOARD

- a) CLEAR [ENTER]
- b) POKE STATUS 3-4, PEEK 30821, PEEK 30822 [ENTER]
- c) NEW STATUS 3-300 [ENTER]
- d) CLOAD the program below, and execute by RUN

(The time taken to salvage a 4K cassette which had ERROR 44 in the last 2 lines, was 3 minutes - as fast as CLOADing the cassette!)

```

5:WAIT 0
10:FOR F=256*PEEK
  (STATUS 3-4)+
  PEEK (STATUS 3
  -3)+3TO STATUS
  3-301
20:IF PEEK F=13
  AND PEEK (F+
  PEEK (F+3)+3)<
  >13GOTO 100
30:IF PEEK F=13
  LET F=F+PEEK (
  F+3)
40:PRINT F
50:NEXT F
60:STOP

100:F=F+1:POKE F,2
  55
110:A=INT (F/256):
  B=F-256*A:C=
  PEEK (STATUS 3
  -4):D=PEEK (
  STATUS 3-3)
120:POKE 30821,C,D
  ,A,B,C,D
125:POKE 256*C+D,0
130:BEEP 3:END

STATUS 1
247

```

I am trapped in an underground cavern, being bombarded with rocks by an unspeakable SCRAMBLOID. This is one - but only one - of my excuses for the newsletter being late this month.

There are - as you may have guessed - six programs on this excellent cassette, and all of them run on the unexpanded machine. Some of them I found less valuable to me than others, but even discounting the programs which I liked least, this cassette is still first-class value.

The manual is utilitarian rather than glossy: the real presentation comes on the cassette itself. One side consists simply of the programs: on the other each program is prefaced by a finely presented and organised introduction, which tells you (almost) all you need to know.

Since there are 6 programs to review, my comments on each are necessarily brief. 3D allows the input of a series of points, and you specify what lines you want joining them. The image is then drawn, 3-dimensionally, at a slight angle. You can then TILT, ROLL, and ROTATE the image, and it will be redrawn in accordance with your instructions. A great help is the fact that optionally you may impose the X,Y,Z axes on the image. I found some difficulty making the best use of the facilities provided. I understand an 8K version of this program, from the same source, will appear later this year, with additional explanations and controls.

SCRAMBLOID has been mentioned above: it is a simple screen arcade game, probably within the capacity of an average child - although it defeated your editor. NOUGHTS AND CROSSES needs the printer. It answers 1st degree threats, but some 2nd degree threats are ignored: so it is possible to beat the computer. An intelligent child should have a slight edge; I just about broke even.

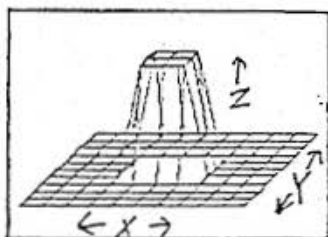
LURGO - the author's favourite - allows the creation of images on the screen by cursor movements, without complex mathematics: and the formulae for these images can then be printed out. Most useful for anyone trying to develop a screen game, or characters.

The COLOR DEMO demonstrates the color possibilities of the machine by combining colors. Very clever and decorative, but I doubt if one would use it often.

My own favorite is GRID, which requires a little more explanation than is provided. A grid is shown, with X and Y axes. Input a formula, in the terms of Z= (values of X and Y), and the grid is distorted in terms of Z. Below is the result of

$$Z=100*(ABS\ X \leq 20) * (ABS\ Y \leq 20)$$

For the mathematically-minded it is most valuable, and for the less mathematical it is still great fun.



SIXPACK is written by CHARLIE SIMPSON, and is distributed by DDD SOFTWARE, 150 High Street, Twerton, Bath, - price £6.95



The ideas suggested so far have been:

- 1) In the part of memory devoted to fixed variables, taking care not to use those variables whose normal home is otherwise occupied.
- 2) In the Reserve Template Area starting at STATUS 2-STATUS 1-189.
- 3) In a REM line starting the BASIC program. If the routine is POKed into the address immediately following the REM, the BASIC program will not be interfered with, but the routine can still be called from this address.

Here are 3 further ideas:

- 4) DIMension a dummy variable such as DIM X\$(1)\*80, and this would give 167 bytes free for m/c routines. (Suggested by John Kerr). The address of the routine could be referenced by STATUS 3.
- 5) In the no-mans-land between the end of BASIC program, and the beginning of the area you have used for your DIMensioned and 2-chr. variables. Such addresses could be referenced from STATUS 2.
- 6) At the beginning of the normal program area, by delaying the start of the program area actually used, with the NEW nnnnn facility described on page

Provided you make use of the CSAVE M and CLOAD M techniques also described in this month's PEEK POKE & MEMORY, there is no real problem, except for the necessity of planning exactly what you wish to do.

---

MARKETPLACE - INFORMATION

RAYMOND MINTOFF has been trying to obtain the PC 1500 service manual. He wrote to SHARP (UK) but, as so often happens, received no reply. However he has found another source:

ATLANTIC NORTHEAST MARKETING, INC  
PO BOX 921  
MARBLEHEAD  
MA 01945 , USA

This cost him \$10, and they also stock the SHARP Technical Reference Manual at \$20, as well as some very interesting hardware and software for PC 1500.

For WAGNER read SPIEDEL ! It appears that WALTER SPIEDEL has the exclusive rights for the WAGNER extensions for outside Germany. His address:

POSTFACH 1168  
D-7320 GOPPINGEN  
W.GERMANY

I understand that I was wrong about the TOOLKITS. They can in fact be used without the extensions to memory: they come in the form of modules which plug in to the back of the CE 150.

The agent in UK for Walter Spiedel is ELKAN ELECTRONICS (who also now have in stock Centronics Cables for CE 158, at £38.05).

Two other interesting sources of add-ons in Germany:

KAUNE ELEKTRONIK, BERLINER STRASSE 113, 5600 WUPPERTAL 2  
supply a 26K module at DM 449, with reduction to DM 378 if you exchange a CE 155.

RASSO VON SCHLICHTEGROLL , PARKSTRASSE 30, 8000 MUNCHEN 2

have a MACROASSEMBLER(with dissassembler) at DM 178, and a Basic TOOLKIT on Eprom, with 11 new commands, at DM 328.

# MACHINE CODE MNEMONICS - explanations

[Please note that the entry in brackets after each mnemonic does NOT indicate a correct op-code: it is only there to help you find the first appearance of this mnemonic on the chart on page 61]

## ABBREVIATIONS

[also see p.61]

u	UNCONDITIONAL	Z	Z FLAG
c	CONDITIONAL	Q	ACCUMULATOR REGISTER
r	RELATIVE		OR CONTENTS OF REGISTER
b	BRANCH OR JUMP		ACCORDING TO CONTEXT ON
s	SUBROUTINE		PAGE 61 (but not i)
v	VECTOR	@	INTO
	(all v jumps reset Z)	N-M?	COMPARE N TO M
flp	FLIPFLOP	C	IN SUBTRACTIONS, THE
fs	FLAG RESET		INVERSE OF C IS USED
P	PROGRAM COUNTER	C	CARRY
S	STACK POINTER	[BCD]	IN BINARY DECIMAL CODE

ADC (82): A+Q+C @ A	OFF (4C): resets BF
ADI (4F): Q+i @ Q	ORA (88): A OR (Q) @ A
ADR (CA): R+A @ R	ORI (48): Q OR i @ Q
AEX (F1): see note 1)	POP (8A): see note 6)
AM0 (CE): A @ timer, 9th bit	PSH (88): see note 7)
AM1 (DE): .....1@.....	RDP (C8): resets display flp
AND (89): A AND Q @ A	REC (F9): @ @ C
ANI (49): Q AND i @ Q	RIE (BE): @ @ IE
ATP (CC): A @ data bus	ROL (DB): see note 8)
ATT (EC): A @ T	ROR (D1): ROL reversed
BCH (8E): rb: P+i @ P	RPU (E3): @ @ PU flp
BCR (81): crb: if C=@, P+i@P	RPV (B8): @ @ PV ...
BCS (83): ...: if C=1, .....	RTI (8A): return from interrupt
BHR (85): ...: if H=@, .....	to main routine
BHS (87): ...: if H=1, .....	RTN (9A): return to Basic
BII (40): Q AND i @ Z	SBC (88): A-Q-C @ A
BIT (8F): A AND Q @ Z	SBI (81): A-i-C @ A
BVR (80): crb: if V=@, P+i@P	SDE (43): A@ (R), R-1@R
BVS (8F): ...: if V=1, .....	SDP (C1): 1@ display flp
BZR (89): ...: if Z=0, .....	SEC (FB): 1 @ C
BZS (8B): ...: if Z=1, .....	SHL (D9): A bitshift left
CDV (8E): @ @ divider	SHR (D5): A ..... rt.
CIN (F7): A-(X)? ,X+1@X, fs	SIE (81): 1 @ IE
CPA (86): A-Q? see note 2)	SIN (41): A@ (R), R+1@R
CPI (4C): Q-i? see note 3)	SJP (BE): usb, next instruction
DCA (8C): A+(R)+C @ A [BCD]	to stack.
DCS (8C): A-(R)-C @ A .....	SPU (E1): 1 @ PU flp
DEC (42): Q-1 @ Q	SPV (A8): 1 @ PV ...
DRL (D7): see note 4)	STA (08): A @ Q
DRR (D3): converse of DRL	STX (4A): X @ R/P/S
EAL (8D): A XOR i @ A	TIN (F5): (X)@ (Y), X+1@X,
EOR (8D): A XOR (Q) @ A	Y+1@Y
HLT (B1): halts MPU:	TTA (AA): T @ A
resume via IE	VCR (C1): cvsb, if C=@
INC (40): Q+1 @ Q	VCS (C3): ....., if C=1
ITA (8A): INPUT @ A	VEJ (C8): uvb
JMP (BA): ub to i,j	VHR (C5): cvsb, if H=@
LDA (84): Q @ A	VHS (C7): ....., if H=1
LDE (47): (R)@A, R-1@R	VMJ (CD): uvb, highbyte FF,
LDI (48): i @ Q	lowbyte i
LDX (88): R/P/S @ X	VVS (CF): cvsb, if V=1, Z reset
LIN (45): (R)@A, R+1@R	VZR (C9): cvsb, if Z=@
LOP (88): see note 5)	VZS (C8): ....., if Z=1
NOP (38): no operation	

## NOTES

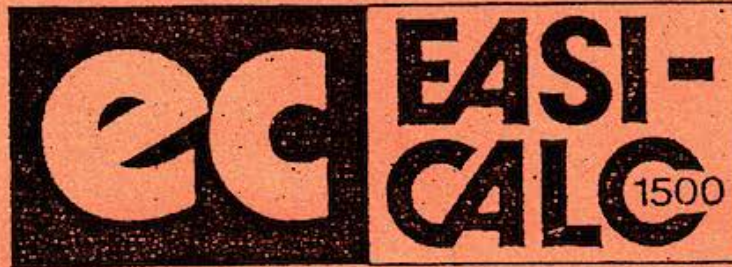
- 1) High order 4 bits exchanged with low order 4 bits
- 2) If A>Q then C=1 and Z=0  
if A=Q then C=1 and Z=1  
if A<Q then C=0 and Z=0
- 3) IF Q>1, etc, as above
- 4) Left rotation between A and (R) in units of 4 bits
- 5) When subtracting 1 from UL produces borrow, then next step.  
If no borrow, relative jump according to i
- 6) Contents of S returned to Q. If Q=A, S=S+1  
but if Q=R then S=S+2
- 7) Contents of Q stacked as directed by S.  
If Q=A then S=S-1, but if Q=R then S=S-2
- 8) Contents of A rotated left through C



# ELKAN ELECTRONICS

POCKET COMPUTER  
SOFTWARE  
SOFTWARE  
SOFTWARE

FOR SHARP PC1500 / TANDY PC-2



## EASI-CALC 1500

"Our most popular program"  
A "spreadsheet" program. Ideal for problem solving and "What if...?" types of questions. Up to 26cols. or 99 rows. Needs 8K RAM, CE150 and tape recorder. Data quality tape and 30page manual.

ELKAN PRICE £19.95

## NEW!! NEW!! EASI-FILE 1500

Data base management program that acts as a "card-index system in your pocket!" Program features rapid sorting techniques searching on 3 levels or on key words. Max. no. of records per tape 256. Ideal for mailing lists, small stock control systems, etc.

ELKAN PRICE £19.95

## SHARP CE-15A PROGRAM TAPE

60 minute software tape contains 14 programs not listed in the applications manual.

ELKAN PRICE £14.95

## TEXT HANDLING PROGRAM

Our first program for the CE-153 software panel. Needs 4K or 8K RAM. Each of the 140 squares can hold up to 45 characters each. Very fast. A unique program.

ELKAN PRICE £19.95



## EASI-TREND 1500

Our very latest program for the 1500 !!!  
A statistical forecasting program using the statistical technique of regression analysis that attempts to fit a series of mathematical curves to your data.

Enables you to see if there is any meaningful relationship between sets of data or to see if something is changing in a predictable way over a period of time.

\* Editing \* Flexibility \* Large Capacity \* High Speed \* Excellent Graphics  
Data quality tape and 40-page manual/reference book.

ELKAN PRICE £19.95

Send cash, cheque, Visa or Access card number for quick delivery.

All ELKAN PRICES include V.A.T. (where applicable) and U.K. post and packing charges (except for books only)

Trade, quantity and dealer enquiries all welcome. Write or phone for details