There are two areas available to the operator:-

1). The first 6 lines: These are used for entering data information for the utility or for working programs to be punched and/or entered into store;

2). The control line. At present only the left hand of Store Display area is functional although access to the Right hand, or Trace control area, can be obtained.

CONTROL LINE.

Accessed by TAB key to begining of line.

Only HEX digits may be entered.

Cursor movement keys :- → normal shift right

← shift left to next field marked (L)

↙ move to beginning of control line

↖ Return cursor to top left hand corner of screen.

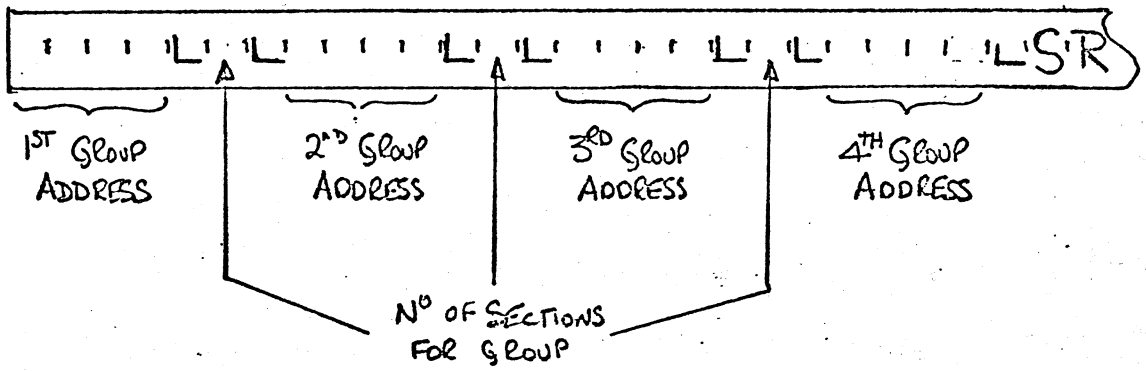
↓ inoperative

Return of cursor to first 6 lines by either :-

↖ as above, or.

New line. This key returns the cursor to it's former position on first 6 lines.

Further depressions of the TAB key move the cursor to the next field (or begining of line if in last field).



The Store Display consists of 8 sections of sixteen consecutive address locations. These sections display 128 consecutive locations unless divided into groups by entering addresses on the control line. e.g.

800␣2.1000.2 3800 2 ⊣700

will display 32 consecutive store locations beginning at *800, *1000, *3800 and *700 (note leading or trailing spaces). Both 'group address' and 'Nº of Sections' must be entered for a valid group address. Invalid addresses are ignored.

800␣ ϕ 1000 2 3800 2 ␣700

will display 128 locations from *800

800␣ 2 1000 ␣ 3800 2 ␣700

Invalid address.

When amending group addresses or 'Nº of Sections' the address is invalidated by the presence of the cursor.

WRITING PROGRAMS.

Enter instruction sequences in machine code anywhere on first 6 lines. eg.

* 14FB ␣ 2151 ␣ 6201 ␣ F012 ␣ 2159 ␣ A006

* 14E6 ␣ B1F1 ␣ 8101 ␣ B2F1 ␣ A006

* NNNN forms an address.

Terminators are ␣ * # or after 4 consecutive hex digits.

Addresses or data ... written omitting leading zeros must be followed by a terminator eg.

* 1Φ ␣ Φ ␣ 1 ␣ 23 ␣ 4 ⇒ equivalent to

* 0010 ␣ 0000 ␣ 0001 ␣ 0023 ␣ 0004

All screen data preceding initial * is ignored.

#NNNN indicates program entry address and also acts as a sequence terminator (see PUNCHING PROGRAMS). Carriage position is also a sequence terminator. Any data after the carriage of #NNNN is ignored.

AMENDING PROGRAM/DISPLAY STORE

Enter amendment data as before – this can be a program – then press

INTERLOCK followed by END PRINT.

Data from first * to, but not including cursor position written to store.

PUNCHING PROGRAMS

(a) From screen data

Enter program on first 6 lines then press START PRINT

Data from first * is punched in 'Autoload format'

(b) From several areas of store

Initial screen data must be :–

*A/B␣*C/C␣*D/E where A to E are 4 digit address characters. Press CONTROL followed by START PRINT.

Data from and including addresses A to B, address C and from D to E will be punched in 'Autoload Format'

Note:– The sum check and program entry will be omitted unless #NNNN is entered after screen data for (a) or control data for (b). where #NNNN is the program entry.

PRINTING STORE CONTENT

Enter address locations to be printed on control line, then press INTERLOCK followed by START PRINT. The whole of the store display area is printed (centralised on 76 column paper).

NOTE:- Tape cannot be punched at 60 cps. Punching and printing is carried out on device '2C' (or Telmi 1). To change to 60 cps for printing, using device '2D' (or Telmi 2), make the following amendment:-

*14E3 602D *1515 702D

To revert to Telmi 1 or device 2C:-

*14E3 602C *1515 702C

PROGRAM PATCHING.

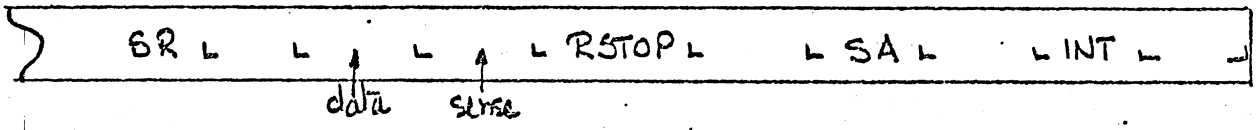
Initial screen characters of:-

*1021␣/*1200␣8201␣6400, when CONTROL followed by END PRINT are pressed will result in *1021 being replaced by a jump to *1200, the original contents of *1021 being placed in *1200 followed by 8201, 6400, Jump to *1022 in locations 1201 to 1203 respectively.

[Caution — Jumps must not be out of range].

READING PAPER TAPE (AUTOLOAD FORMAT ONLY).

load paper tape into reader as for normal loading via "autoload". Press CONTROL/ϕ. Approx 15 seconds is allowed for operator to press RUN/LINE or LINE to move tape and then for first legal character to be read. Data will be read in as if in "autoload" program, except that the program entry point will be ignored.
Blocks of data may be loaded providing the blank tape between blocks does not exceed that generated when punching two separate blocks using the utility. If this run of blank tape exceeds the timeout of the reader program all further data characters will be ignored.
To check that the tape has been read correctly and that the entry address is correct, examine store locations *1A6D for sum check and *1A6E for program entry.
NOTE:- Do not load tapes which overwrite store locations *40 and 41 when using P002 boards or *4 for P010 boards. This will crash the utility.



The following data is required to run a program under the interpreter:-

START ADDRESS (SA) - The address at which the user requires the program to start

INTERRUPT ADDRESS (INT) - The address to which the program will jump to process interrupts. This address must not be #4 or #10 and 41 as these are used by the utility itself.

TO START A PROGRAM:-

Enter SA and INT, press CTRL (control) and 3 then CTRL and 2 for RUN (or 1 for SINGLE SHOT).

Note:- If SA or INT are missing the program will not start

TO ENTER SINGLE SHOT:-

Press CTRL and 1. To advance press SEND.

TO RESET:-

Press CTRL and 3. This aborts the program run pointers and flags. 'Single shot' or 'run' starts from the address in SA in state 0.

©International Computers Limited 1969 Printed in Great Britain by ICL Printing Services Letchworth

TO RAISE SERVICE REQUEST:-

Enter data as follows-

SR - a 3 bit field R nn or W nn where

R is a read

W is a write

nn is a string of two hex digit representing the device number.

Data - enter data as required

Sense - enter sense data as required.

(spaces translate to ϕϕϕϕ).

Then Press CTRL and 4. The device number will flash until the request is honoured.

[Note :- line 6 has been utilised to contain a data string. Every time CTRL/4 is pressed two characters are transferred to the 'data' area and line 6 is shifted left two characters. This is designed for Modem functions. To inhibit this, alter Utility start address (eg *1800, *2000 etc) plus 7D5 and 7D9 to '3000' (ie LFD+3 and LFD+9)]

TO MOVE CURSOR TO CONTROL AREA:-

Press SVC key. Once in this area the cursor tab functions keep the cursor in the area. To exit press NL or ↖. ↙ resets cursor to SR location

TO RESET SERVICE REQUEST:-

Press CTRL and 1 — this enters single shot mode and resets service request flag. To continue press CTRL/2 to run or SEND for single shot

Note:- a service request may be raised again by pressing CTRL/4

REQUEST STOP. (RSTOP).

Enter address at which program is to stop. Full status is not required. (Presence of the cursor will invalidate address).

SCRATCH PADS.

The scratch pad displays are the scratch pad contents that would be if the program were live. Register F contains the full status.

To modify Scratchpads, alter XFE → XFE+15 for SP1, and XFF → XFF+15 for SP2.

ERRORS IN INTERPRETER INSTRUCTIONS.

'1' instruction – transfer acceptable, ie a skip will occur for all '1' instructions except that for reading the service request register ie 1x3ϕ.
A write control will not necessarily terminate a "read service request".
Accessing device ϕ will always skip.

'A' instruction. – When operating in the frozen state, a dump will dump the theoretical content of SP2 only. The count register content is not passed to SP2 during the dump.

'E' instruction (unfreeze) is not incorporated. The instruction will be treated as a 'null'

'D' instruction (freeze) not fully tested

'C' instruction (freeze) will corrupt utility if an attempt is made to read/write data from/to non existent or protected store.

Use of Utility.

To input paper tape :- transparency on / rate at bottom. ↑

Control / O. then RUN + LINE would close. worth having punch off.

To dump from store :- *Add1 / Add2.

Control / print. (loadable format.) — Timer on.

To dump work area of screen from first * to cursor. :- Print. (fatal tol, record of factor)

Punch on, Transparency off, rate in middle.

?

To print display area of store. :- Interlock and print.

At end of load of utility tape - 1B00 should be in R6.
IF load fails - 1 (because of checksum fail) - try 1B00 in F
4000 in T. Go.

International Computers Limited

Memo

Date 3/6/75 | Your ref | Ref. no GD/VR | Tel ext

To

7502 WORKERS,
B. F. BROWNE,
DAL.

From

G. Drew,
DAL.

Interpretive use of the Utility

It is possible to run most E2 code interpretively using the utility. Some instructions, notably the I/O instructions, will not work, but paths can be 'dry' tested. Your program should start beyond *42 and finish before *1800.

Setting up

1. Load utility (see HEW's memo of 2/6/75 ref: HEW/VR).
2. Put PT of your program into Reader, don't press RUN or GO. * Display the program area on screen so that it can be seen to go in.
3. Press Control + 0 (in numeric block).
4. Within 20 secs press RUN + LINE on PTR.

Your program is now in store.

General running

1. Press SCM. This enters the cursor in the interpretive screen.
2. Press TAB twice and enter the address at which you wish your program to stop. (Addresses are assumed as HEX).
3. TAB into START field and enter the start address of your program.
4. TAB again and put 40 for interrupts.
5. Press ↖ to return to TOS.
6. Press CONTROL + 3 in numeric block →
7. Press CONTROL + 2 in numeric block
8. Watch it run.
9. It may be stopped by CONTRL + 1 to enable continuation from step 7, or CONTROL + 3 to reset from step 7.

At step 7 CONTROL + 1 puts it into single shot mode, whence 'SEND' executes the next instruction.

Isolated use

While this facility is of great use in checking inter-modular linkages, it is most useful in single module testing. Tables such as Coretable, DST etc. can be set up in any part of store that is free by writing a set-up and enter routine on the screen. This can be punched out for later use.

I/O instructions will give TRACC but RDC + RD transfer 0 to the register. WT + WTC have no effect.

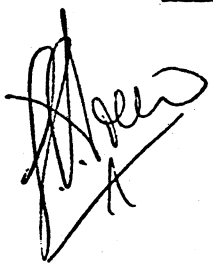
Certain block instructions will stop the interpreter until it is reset. (I haven't a definitive list).

In single shot mode the request stop is ignored.

Altering store in any way causes the interpreter to reset.

'Reset' means reading the parameters from screen and starting from there.

'Registers' are not cleared on reset.



G. Drew