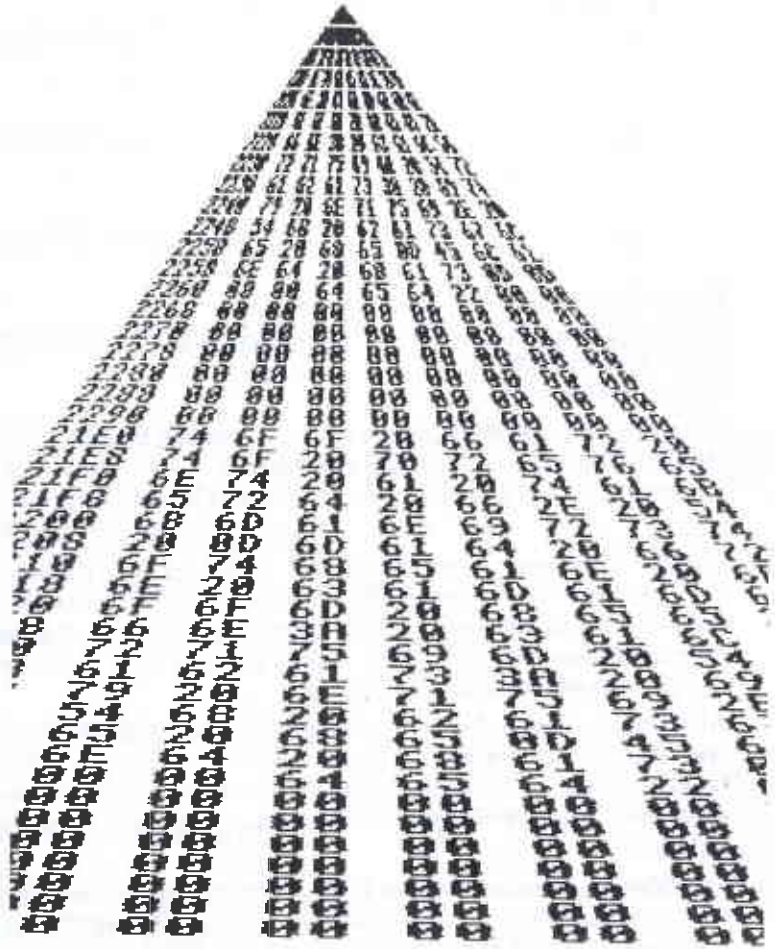


FINGERPRINT FINGERPRINT

A 6502 machine code monitor, memory editor and disassembler for the BBC Micro and Master 128



by David Spencer

DABS
PRESS

Fingerprint

© David Spencer 1987
First published June 1987
ISBN 1-87033-607-0
Edited by Bruce Smith

Advanced Disc Toolkit is produced by Advanced Computer Products, 6 Ava House, High Street, Chobham, Surrey. *Disc Doctor* is produced by Computer Concepts, Gaddesden Place, Hemel Hempstead, Herts HP2 6EX.

Within this book the letters BBC refer to the British Broadcasting Corporation. The terms *BBC Micro*, *Master 128* and *Master Compact* refer to the computers manufactured by Acorn Computers Ltd under licence from the BBC. The terms *Econet* and *Tube* are registered trademarks of Acorn Computers Ltd.

All rights reserved. No part of this book (except brief passages quoted for critical purposes), or any of the computer programs to which it relates may be reproduced or translated in any form or by any means mechanical electronic or otherwise without the prior written consent of the copyright holder.

Disclaimer : Because neither Dabs Press nor the author have any control over the way the material in this manual and accompanying software is used, no warranty is given or should be implied as to the suitability of the advice or programs for any given application. No liability can be accepted for any consequential loss or damage, however caused, arising as a result of using the programs or advice in this package.

Published by Dabs Press, 5 Victoria Lane, Whitefield, Manchester M25 6AL. Telephone 061-766 8423. Fax 061-766 8425.

Designed and typeset in 10 on 11 pt Palatino by Dabs Press. Printed and bound in Great Britain.

Contents

	Page
1 Introduction	4
2 Installation	6
3 Using Fingerprint	9
4 Ranges and breakpoints	10
5 Output stream	12
6 Keyboard commands	13
7 Disc based commands	20
8 Hints and Tips	24
9 Summary of commands	26
10 About Dabs Press	28

1 : Introduction

Fingerprint is a machine code monitor program for the BBC and Master Series micros. It is fully compatible with all versions of BASIC and MOS and will work from main memory, from Sideways RAM (SRAM), Shadow RAM and across the Tube when run on a 6502 second or co-processor. *Fingerprint* is the ideal development tool for the machine code programmer and can be used to examine and control the execution of machine code programs resident in main memory, SRAM, or even sideways ROM. Whilst being a sophisticated machine code utility program, it is easy to use, and this manual contains full details. It will not teach you to program in machine code. However the Dabs Press Assembler Bundle (see Section 10) provides a complete assembly language tutorial and is the ideal foil for *Fingerprint*.

In addition, the *Fingerprint* disc also contains several extra disc based enhancements of three of the standard *Fingerprint* commands - see section 7 for full details.

Fingerprint is supplied in two different disc formats, for DFS users on a 5.25" 40 track disc, and for ADFS users on a 3.5" disc. See page 27 for a list of the files that should be on your disc.

DFS Format and 80 track discs

The disc is in DFS 40 track format. If you have a 40 track disc drive you will be able to use the disc immediately. If you have a 40/80 switchable disc drive you must first select the 40 track setting. If you have an 80 track disc drive and a BBC B+ or Master 128 you must first enter the following command:

```
*DRIVE 0 40
```

you will then be able to use the disc in drive 0. If you have a BBC B and 80 track drive then the contents of the disc must be copied onto a newly formatted 80 track disc. A special program exists on the disc which can be read by an 80 track and then copy the *Fingerprint* programs across. To use this program place the *Fingerprint* disc into drive 0 and enter:

```
CHAIN"COPY40"
```

and follow the on screen instructions.

ADFS Users

If you have an ADFS version of *Fingerprint* place the disc in the default drive and type:

```
*MOUNT
```

The programs are now ready to use. If you have a Master 128 and twin 80-track drives, and you have bought the 5.25" DFS version and wish to transfer the files onto an ADFS disc, you should format a blank ADFS disc and place it in Drive 1. Put the *Fingerprint* disc in Drive 0, and either set your switchable drives to 40-tracks on Drive 0 only, or type:

```
*DRIVE 0 40
```

then run the COPYFILES program from your Master Welcome disc, or individually copy each file using *MOVE, in which case you should type:

```
*MOVE -DISC-:0.$.$SMON -ADFS-:1.$.$SMON
```

and so on, for each file, substituting the filename for SMON.

Copyright

For your convenience, the *Fingerprint* disc is not copy-protected in any way. We would remind you, however, that the *Fingerprint* programs and manual are protected under copyright. You are authorised to make backup copies of the disc for personal use *only*. Should you wish to use more than one copy of *Fingerprint*, Dabs Press can offer a discounted site licence. Contact us for more details. Details on making backup copies of disc can be found in your Disc User Guide.

About the author

David Spencer, a keen BBC Micro user for many years, co-authored "Toolbox 2" published by BBC Software, and was a contributor to *Acorn User* magazine. David is now Technical Editor at Beebug magazine.

2 : Installation

The disc contains three versions of *Fingerprint* as follows:

- A. Main memory version - Filename "MON"
- B. Sideways RAM version - Filename "SMON"
- C. Tube memory version - Filename "TMON"

The method for loading and running *Fingerprint* will depend on which version you intend to use. The installation procedure for each is now described.

A. Main Memory Version

The main memory version of *Fingerprint* requires 5.5k of continuous memory to run in. The program contains a special relocating mechanism which means that you can load it anywhere you choose. Care must be taken to ensure that the memory block does not interfere with the target program (ie the program you wish to examine) nor corrupts the computers own workspace. For most purposes this will be memory above PAGE and below HIMEM. The value of both of these can be found by typing:

```
PRINT ~PAGE  
PRINT ~HIMEM
```

It is worth making sure that you are aware of the *Fingerprint* boundaries by making a note of its start and end addresses. If, for example, *Fingerprint* was being installed on a Master 128 at &E00 then it would end at the address given by:

```
PRINT ~(&E00+5632)
```

The first step is to *LOAD the *Fingerprint* code as follows:

```
*LOAD MON <addr>
```

where <addr> is the load address. In the Master 128 example it would be &E00. The target machine code should be installed by *LOADing the object code or re-running the assembler source code (making sure that this does not corrupt the *Fingerprint* code).

The next step is to tell *Fingerprint* to re-locate itself and to inform it of the address of the target program. This is done by pointing the X and Y index registers to the target code and calling the *Fingerprint* load address. The syntax then is as follows:

```
X%=<target program address> MOD &100  
Y%=<target program address> DIV &100  
CALL <Fingerprint address>
```

For example, if the target code is to be traced from &A23 and *Fingerprint* is located at &E00, it would be started with the following three lines entered in BASIC direct from the keyboard:

```
X%=&A23 MOD &100
Y%=&A23 DIV &100
CALL &E00
```

The main RAM version of *Fingerprint* may also be run from Shadow screen memory which is fitted as standard on the BBC B+, Master 128 and Master Compact. If a shadow screen mode is in use (ie MODEs 128 to 135 inclusive) then the MON version can be loaded and run as already described. If a non-shadow mode is in use then you must first issue a *FX108,1 command from the keyboard before loading the MON program into shadow memory and executing it as outlined above.

B. Sideways RAM version.

This version, which is saved as SMON on the *Fingerprint* disc, should be loaded into an empty Sideways RAM in your computer. On the Master 128, Master Compact and BBC B+128 these banks are designated W, X, Y and Z. The program is loaded with the *SRLOAD command as follows:

```
*SRLOAD SMON 8000 <id>
```

where <id> is the Sideways RAM bank identity letter. See your computer user guide for more details on the *SRLOAD command. On a BBC B+ or BBC B computer you will need to have had a third party sideways RAM module installed. The user guide that accompanies this will contain full details on how to load the sideways RAM version of *Fingerprint* into it. Once in the Sideways RAM bank the ROM image must be initialised by pressing <CTRL-BREAK>. Typing OLD will restore any existing BASIC program. Once the ROM image has been initialised it can be started with the following command:

```
*FINGERPRINT <target>
```

where <target> is the target address for *Fingerprint* to operate from.

Important: Because of the way in which *Fingerprint* works it is not possible to 'blow' the sideways RAM version into an EPROM chip, neither will this version work with sideways RAM systems that have a write protect switch enabled. When using *Fingerprint*, any such switch must be disabled.

For those users who only have 8k of Sideways RAM, we have provided a version of *Fingerprint* which only occupies 8k, but doesn't contain the MEMX, DISASS, and ROMLIST code. This appears on the disc as file "SMON2".

B. Tube version

The Tube version of *Fingerprint* is started in much the same way as the main memory version except that the file to be loaded is TMON instead of MON, ie,

```
*LOAD TMON 00000800
```

Remember that the full 8 digit address must be specified to ensure that the code is loaded at the correct address , &800 in the second processor in this instance. Like the main RAM version the Tube version is fully relocatable Each time the tube version of *Fingerprint* is started it will try to fetch a file called THOOK from the disc. Therefore, the *Fingerprint* disc must be in the default drive each time the program is started.

3 : Using Fingerprint

Once you have started Fingerprint running, you are presented with the Trace display which occupies the bottom of the screen. In 40 and 80 column modes this display will occupy the last two lines. In 20 column modes the bottom four lines are used. The 40/80 column will look something like this.

```
&1300 LDY #&0F          NV-BDIZC
A=&00 X=&ED Y=&FF SP=&ED  00110010
```

Working from left to right we have:

&1300	- the address being traced
LDY #&0F	- the current 6502 instruction
A=&00	- the accumulator contents
X=&ED	- the X register contents
Y=&FF	- the Y register contents
SP=&ED	- the Stack Pointer contents
NV-BDIZC	- the status register flags and directly below
00110010	- the state of the status register (1=flag set, 0=clear)

Important: each instruction is displayed before it is executed, therefore, if the instruction displayed is a register change, then the contents shown for that register will be the old contents and not the new.

Fingerprint has three modes of operation - please ensure that you understand the difference between these modes before continuing further. The three modes are:

Single Step Mode (SSM) : The target program is executed one instruction at a time with the SPACE bar being used to execute the next instruction.

Free Run Mode (FRM) : The target program is executed one instruction at a time automatically at a variable speed set by the user.

Command Mode (CM) : This mode is entered when a command requiring parameters is selected from one of the first two modes. The mode is recognized by the fact that the top line of the display is replaced by a text prompt.

4 : Ranges and Breakpoints

When tracing, ie stepping through, a long machine code program, it is seldom necessary or even desirable to trace through every bit of the code. A prime example is the tracing of an Operating System routine. If you are tracing a program with lots of operating system calls to print characters on the screen you don't want to work through the OS routine each time it prints a character - it will be the same for each one after all.

Fingerprint inhibits the tracing of any machine code located above &C000 (&F800 on Tube version). The default trace range of *Fingerprint* on a standard micro is therefore &0000 to &BFFF.

It is possible to set up to four user definable trace ranges so that only machine code that falls within the trace ranges will be traced on-screen. Code falling outside the defined trace range will be executed but not traced on-screen. When in one of these trace ranges, *Fingerprint* will display each instruction in turn and then either wait for the SPACE bar to be pressed or wait a pre-defined time period before executing the next instruction. During this waiting period keyboard commands can be invoked.

When outside a range *Fingerprint* still monitors each instruction to see if it is back in a range but won't produce any display. To increase the speed of out of range execution the keyboard commands (ie, function key presses as outlined in section 6) are not normally recognized when outside a range. To regain keyboard control press CTRL-TAB this will force a command loop in single step mode, allowing any number of keyboard commands to be entered until the SPACE bar is pressed, at which point execution continues. A typical use of this would be to reset the ranges if execution has got stuck outside one of the ranges.

Breakpoints allow you to execute a section of machine code, stopping it, ie breaking into it, at a defined point and displaying register contents. Imagine the situation when you are tracing a short loop which has many iterations, the one shown below for example:

```
.loop  
LDX #&FF  
STZ &2000,X  
DEX  
BPL loop
```

After the first couple of iterations through the loop it is obvious what is happening and you either have to laboriously single step through 256 passes or set a fast free run mode and risk missing some vital instructions when the loop ends. The ideal solution would be to free run at maximum speed until the loop terminates and then to enter single step mode so that you can decide what to do next. This can be done by inserting a breakpoint

at the point where the loop finishes. Up to eight breakpoint addresses may be set within *Fingerprint*, and when the address of the next instruction matches one of these eight addresses, *Fingerprint* will enter single step mode and force a display, even if the breakpoint was outside a range. To help identify a breakpoint *Fingerprint* will beep when it encounters one and print the breakpoint number (in the range 1 to 8) followed by a 'J' to the left of the next instruction. In Mode 7 the 'J' will be shown as a right arrow. Breakpoints are set using a keyboard command and the technique for this is outlined on page 14 of this manual.

5 : Output Stream

Fingerprint normally generates a two or four line display at the bottom of the screen. There are some commands which require more space than just a few lines, and it would also be nice to be able to send *Fingerprint's* output to a printer or to a remote computer via the RS423 serial link. These things are catered for in *Fingerprint* by altering the default output stream, and by having a special 'external' compact display format. The compact format fits all the display information onto a single eighty column line, ideal for output to a printer or a remote terminal. The flags on the compact format are shown by using lower case letters for flag clear and upper case for flag set. For example, a flag display of:

```
Nv-bd1Zc
```

means that the N and Z flags are set and the others are clear. ('-' is always shown for the undefined bit as this cannot be cleared).

By using the printer and RS423 toggle commands which are detailed later, it is possible to send the compact format display to the printer, serial port or both, as well as having the main display on the screen.

Commands which need the whole screen for their output work as follows: If neither printer or RS423 output is enabled then the command will use the screen for output, clearing it first, and again after execution once a key has been pressed. This means that any screen output generated to date by the target program will be lost.

If either printer, or RS423, or both are enabled then the commands will send their output to the appropriate device rather than to the screen. In this case the screen will not be cleared and it isn't necessary to press a key at the end. If output to the screen is also required this can be enabled using the screen enable command. If this is done the screen is cleared and a keypress at the end is needed as before.

6 : Keyboard Commands

With a few important exceptions, all the features of *Fingerprint* are controlled by the ten red function keys in combination with the SHIFT and CTRL keys. The use of these keys is in no way affected by any thing the target program does, and the target program can quite happily use the function keys in a totally different way. The following few pages describe the action of all the keys used by *Fingerprint*.

SPACE BAR : Single Step

This has two functions. Firstly, when in free run mode pressing the space bar will cause single step mode to be entered at the next instruction, assuming that tracing is in progress. Secondly, when in single step mode the space bar causes the next instruction to be executed.

ESCAPE : Abort Command

The only use of ESCAPE on its own is to abort a command when in command mode. In general when mid-way through a command pressing ESCAPE will exit the command with no changes taking place. When not in command mode pressing ESCAPE will set the Escape flag as normal and this can be detected by the target program.

CTRL-ESCAPE : Quit

Pressing this combination of keys will result in a soft-reset with both the execution of the target program and *Fingerprint* being terminated. This action is independent of whether *Fingerprint* is in or out of a range. Normally, the program will abort with the message "Target program aborted" and a beep. If however the stack pointer has been reset the results are unpredictable. Note that it may be necessary to hold the two keys down for a short period before *Fingerprint* recognises them.

CTRL-TAB : Force keyboard scan

This 'forces' a keyboard command scan when *Fingerprint* is outside one of the trace ranges. If *Fingerprint* is operating inside a range this command is ignored.

f0 : Free Run Mode

Places *Fingerprint* into the free run mode (FRM) if it was previously in Single Step Mode. If it was already operating in FRM this command has no effect. The free run speed will be that last set, or if unchanged from entry it will be the default.

f1 : Decrease execution speed

Reduces the free run execution speed of *Fingerprint*. The minimum speed is about one instruction every five seconds. If already at the minimum speed this key has no effect. The command is still obeyed when in single step mode but the effect won't be seen until free run mode is next entered.

f2 : Increase execution speed

Increases the speed of free run mode. The maximum speed is when there is no delay between instructions at which point the speed is the same as with the space bar held down in single step mode.

f3 : Set Breakpoint

Allows one of the eight breakpoints to be set. On pressing f3 the top line of the display clears and a prompt for the breakpoint number is printed. At this stage either a number between 1 and 8 should be entered by pressing the relevant number key. Pressing RETURN or ESCAPE will abort the command.

Once a breakpoint number is entered the prompt changes to request an address. The hex address of the breakpoint should be entered, and the RETURN key pressed. Once the breakpoint has been set the display is restored.

f4 : Clear Breakpoint

This command is used to clear a previously set breakpoint. On selection the command prompts for the breakpoint number to be entered in the same way as for setting a breakpoint. Once a valid breakpoint number has been entered, the current value of that breakpoint is printed. If the breakpoint wasn't set then a value of &0000 will be shown. Pressing f4 again will clear the breakpoint, with a beep confirming the fact. Pressing any other key, including ESCAPE,, will abort the command without clearing the breakpoint.

f5 : Set a range

Allows one of the four trace ranges to be defined. The command prompts for a range number (1 to 4), followed immediately by the start address and then an end address. *Fingerprint* rejects an end address lower than the start address.

To de-activate a range the start and end addresses should be set to the same value; we suggest &0000 to avoid confusion. *Fingerprint* will not alter a range until both start and end address have been entered correctly. In this way it is possible to ESCAPE part way through the command.

Initially, range 1 is set as &0000-&BFFF, with the other three ranges unset. Therefore, it will normally be necessary to reduce range 1 before setting any of the other three ranges. Range changes only come into effect on the next instruction - altering the ranges to not include the current instruction will not destroy the display.

f6 : View memory

This is a simple editor which allows memory contents to be examined and altered if desired. It is useful for editing small sections of memory - for larger changes the disc based memory editor should be used as detailed in section 7. When f6 is pressed the display will change to a prompt for an address, at which point the command can be terminated by ESCAPE.

After entering the address the display will change to show the contents of the given location in hex. At this point there are four options selected by function keys f7, f8 and f9. Pressing any other key will abort the command.

f7 : Move back a byte (View Memory)

Used with the View Memory command, pressing this key will step the memory editor back one byte. If memory is at &0000 then it will wrap back around to location &FFFF.

f8 : Move forward a byte (View Memory)

Used with the View Memory command: on pressing this key the memory editor moves on to the next location which will update the display immediately. If the address was &FFFF it will wrap around to &0000.

f9 : Alter memory contents (View Memory)

When used with the View Memory command you may alter the contents of the current location by pressing f9. The display of the current value will change to a ?, and a two digit hex value can be entered, or escape can be used to cancel the choice. The contents of the location is updated when return is pressed.

f9 : Alter stack pointer display

This key has a second command action when used outside of the View Memory command. By default *Fingerprint* will display the current value of the processors stack pointer register, which can be used to see how deep the stack is getting, or in connection with the View Memory command to examine the stack contents. This mode is indicated by the display showing as SP=&xx where xx is a hex value.

If f9 is pressed, *Fingerprint* will display the stack depth of the target program, i.e. the number of bytes pushed onto the stack by the executing

program. This display is shown as SD=&xx. Negative stack depths are shown in two's complement form (e.g -2 = &FE) although this normally indicates malfunction of the target program as it means that more bytes have been pulled off the stack than have been pushed on. Pressing f9 again will revert back to displaying the stack pointer.

This use of f9 should not be confused with it's use to change memory from within the view command.

Shifted Function Key Commands

SHIFT-f0 Change processor

There are three basic variants to the 6502 microprocessor. These are the original 6502 as fitted to model B's and B+'s, the 65C02 which has some additional instructions and is used on the Master series and newer 6502 second processors, and finally the R65C02 with more instructions still which was used on earlier 6502 second processors and the Master Turbo.

Fingerprint needs to be able to tell which type of processor it is running on so that it knows whether the additional instructions on the 65C02 and R65C02 should be displayed as such or should be shown as illegal. *Fingerprint* will normally automatically detect the type of processor fitted to the machine, but there are some cases when it is desirable to force a different type, for example to make sure that a program written on a Master will run on a model B. This command allows the processor type to be changed.

Each press of SHIFT-f0 steps on to next type in the order 6502, 65C02, R65C02, AUTO and then back to 6502. The new type is shown on the top line of the display for one second when it has been changed. It should be remembered that this command only alters the display of instructions, and doesn't in any way allow non-existent instructions to execute. Some of the legal new opcodes on a 65C02 will crash if tried on a 6502.

SHIFT-f1 : List past instructions

This command will list the last 20 instructions in mnemonic form along with their addresses. The destination of the display (i.e screen, printer or RS423) is determined as explained in section 5 on the output stream. If less than 20 instructions have been executed since the start of execution (or since resetting defaults) then only that number of instructions will be shown. If no instructions have been executed then this command does nothing.

SHIFT-f2 : OS command

Pressing this key gives access to the micro's command line interpreter, ie * commands. The command should be entered at the prompt, without the *. After the command has been executed, *Fingerprint* waits for any key to be pressed before clearing the screen and continuing with normal operation. Pressing '*' has the same effect as SHIFT-f2

SHIFT-f3 : Disassemble

Invokes a simple but effective disassembler that works from the current value of the Program Counter onwards. This command takes up the whole screen so works in the way described in section 5.

Each instruction is shown along with its location and the ASCII equivalent of the bytes. Any key other than ESCAPE can be used to move on to the next instruction. ESCAPE is used to leave the disassembler. For more advanced disassembling use the disc based disassembler described in section 7.

SHIFT-f4 : Call user routine

This is a very simple and flexible way of expanding *Fingerprint*. On pressing SHIFT-f4 *Fingerprint* will prompt for an address and then call the routine at that address. User routines can use locations &38 to &3F as workspace without affecting the target program. Your own program need not worry about preserving registers, but it should obviously return with the stack set to the same level as on entry.

SHIFT-f5 : Printer toggle

This command is used to enable or disable the printer output as described in section 5. The enable/disable state alters with each press of SHIFT-f5, with the new state being indicated by a beep. A high pitched beep means that the printer is enabled, and a low pitched beep means it is disabled.

SHIFT-f6 : RS423 toggle

This command is the same as SHIFT-f5 but controls the serial port instead of the printer.

SHIFT-f7 : Screen enable

As outlined in section 5, if full screen commands use the printer or serial port as an output stream then they won't also use the screen. This command allows the screen to be used as well. Each press of SHIFT-f7 will alter the screen enable setting, a high pitched beep meaning it is enabled, a low pitched beep meaning it is disabled.

SHIFT-f8 : Restore defaults

Resets *Fingerprint* to the state it was in when it was started, ie, ranges are set to their defaults, all breakpoints are cleared, single step mode is selected, free run speed is set to default, processor is set to 'Auto', the stack pointer is displayed and printer and RS423 are disabled.

SHIFT-f9 : Display status

The current status of *Fingerprint* is displayed consisting of the ranges, breakpoints, processor type, ROM number, speed and output stream.

CTRL function Key Commands

Function keys CTRL-f0 to CTRL-f7 allow the flags and registers to be altered. For flags pressing the relevant key will toggle the flag. With registers the current contents of the register is displayed and you type the new value in hex. Pressing ESCAPE aborts the command.

CTRL-f0 to CTRL-f4 : Flag toggles.

CTRL-f0 : N flag toggle.

CTRL-f1 : V flag toggle.

CTRL-f2 : D flag toggle.

CTRL-f3 : Z flag toggle.

CTRL-f4 : C flag toggle.

Pressing one of these keys will change the 6502 flags—if the flag is one it will change to zero, if zero it will change to one. The screen flags display will alter immediately.

CTRL-f5 to CTRL-f7 : Alter register contents.

CTRL-f5 : Alter accumulator.

CTRL-f6 : Alter X register.

CTRL-f7 : Alter Y register.

These three commands allow you to enter new values in the 6502 registers during a trace. On pressing the relevant key, you are prompted for a new value. Type two hex digits and press RETURN.

CTRL-f8 : Jump

This command allows you to jump to any address within the target program. The command prompts for a new address, and then updates the display to show the new 'next' instruction. As usual ESCAPE can be used to abort the command. Another way to think of the jump command is as a command to alter the program counter (PC).

CTRL-f9 : Change ROM

This command, which is not available on the Tube version of *Fingerprint*, is used to select which sideways ROM is paged into the region &8000 to &BFFF. The current ROM number is displayed, and then a new value between &0 and &F can be entered followed by RETURN. If the next instruction is in ROM the display will change to show the new instruction following the ROM switch.

7 : Disc Based Commands

To help keep the size of *Fingerprint* down so that the RAM version does not eat too greatly into the main memory of the computer, there are several debugging aids which have been implemented in a restricted form in *Fingerprint*, for example the disassembler. It is anticipated that most of the important debugging commands will be available in the machine as part of a utility ROM, for example *ADT* or *Disc Doctor*. However, in case these commands are not available, three of the more popular ones, a disassembler, a memory editor and a ROM lister, are provided on the *Fingerprint* disc as transient utilities which are loaded each time they are used. These commands are described below. To use any one, it is of course, necessary to have the *Fingerprint* disc placed in the default drive.

Note that all of these utilities load in at &1300 and will therefore destroy any code which is already there. The command names are only determined by the filenames on the disc and could if desired be renamed using *RENAME.

The Sideways RAM version of *Fingerprint*, contains these three utilities, and if this is installed, you can use them without having the *Fingerprint* disc in your default drive.

DISASS : Disassembler

This is a scrolling disc-based disassembler. It is invoked with one of the following three command forms:

*DISASS

*DISASS <hex_address>[:rom_id]

*DISASS <file_name>

If a <hex address> is given then disassembly starts from that address in RAM. The optional ROM number allows a sideways ROM socket to be chosen. If, on the other hand, a file name is given (a file name being anything that isn't a valid hex address) then the code to be disassembled will be taken from that file, starting at the beginning. If no argument is given then DISASS will prompt for either an address or a file name.

Once disassembly has started, the display is produced, consisting of the address of the instruction followed by the hex bytes then the mnemonic and finally the ASCII equivalent. The display can be scrolled backwards and forwards using the cursor up and down keys. When disassembling from a file you can't scroll past the end (or beginning) of the file. DISASS can be restarted with the ESCAPE key.

Sideways ROMs may be disassembled by specifying a [:rom id], this will be the hex number of the ROM preceded by a colon.

A second press of the ESCAPE key will abort the DISASS program. The file based disassembly can't be used with the Cassette or ROM Filing Systems. Examples:

- *DISASS 3000 - disassemble memory from &3000
- *DISASS 8000 :8 - disassemble ROM 8 from &8000

ROMLIST : List ROMs

This command simply lists the title of each Sideways ROM in the computer, along with it's version number. This can be useful when trying to locate a particular bit of code to trace. The output is exactly like *ROMS on the Master 128, except that absent ROMs are not listed.

MEMX : Full memory editor

This command provides a hex/ASCII memory editor for all areas in memory on all BBC machines except the non-volatile memory on the Master and Compact. To enter the editor, type *MEMX followed by the required hex address, for example:

```
*MEMX 8000
```

If you don't specify an address, the editor starts at the current value of OSHWM. Within the editor the up, down, left and right cursor keys are used in the normal manner to move from byte to byte. To edit a byte of memory, you simply press a key, and the ASCII value of the key is placed into the current byte (as bracketed by the cursor) and the cursor moves to the next byte. This makes it easy to type text directly into memory. All CTRL keys enter their appropriate value. The only special key, apart from the function keys, is DELETE, which has the same effect as left arrow. If you wish to enter values in hex, necessary for &1B (ESCAPE), &7F (DELETE) and values above &7F, then press COPY. The cursor changes to a square bracket, and only the keys 0-9 and A-F (upper case only) are active. When memory is altered in this mode, the cursor does not move on after each byte.

To move around memory quickly, use SHIFT in conjunction with the cursor keys. SHIFT up and down move a screen page of memory at a time. The number of bytes on a screen page depends on the screen mode in use. SHIFT used with left and right cursor keys moves to the left and right ends of the line of bytes on screen.

f0 - New Address

To move directly to a new location in memory, press f0. MEMX will prompt you for a new address. Enter the address in hex, and press RETURN. The display will then update.

Paged Memory

The most interesting feature of the editor is its ability to examine the various banks of memory on the BBC B+, Master 128 and Compact. There are five controls covering this feature.

f1 - Alter ROM

This allows you to select which of the 16 banks of sideways ROM or RAM is displayed in addresses &8000 to &BFFF on the I/O processor (but see f4). Press f1, then enter a hex digit 0-F. There is no need to press RETURN. If you are currently examining memory between &8000 and &BFFF, the display will instantly alter. The bank selected remains in force until you change it again, or leave MEMX. Sideways RAM on the BBC B+, Master, Compact and Model B (one bank only) can be altered without problems.

f2 - Toggle LYNNE

This determines whether the editor displays the contents of the 20k shadow screen, (20k=IN), or the main RAM between &3000 and &7FFF (20k=OUT). The default is to display main RAM irrespective of which bank is currently being used as the screen. This will not work on non-Acorn shadow systems, and does not appear on the BBC Model B/Electron. The switch works on the BBC B+, correctly toggling bit 7 of &FE34, not bit 0 as on the Master Series.

f3 - Toggle HAZEL

This controls the area between &C000 and &DFFF, and switches between display of Operating System ROM and the private area of RAM where filing systems store data, known as HAZEL. MEMX defaults to display of HAZEL (8k=IN). This private RAM is not available on the BBC Model B/Electron, and will not be displayed.

f4 - Toggle ANDY

ANDY is an 4k area of RAM between &8000 and &8FFF, used to store function key strings and graphics variables and workspace. This area normally shows the currently selected ROM, but pressing f4 with the editor examining memory between &8000 and &8FFF will display the RAM. The default condition is not to display ANDY memory (4k=OUT). On the BBC B+, the command switches in and out the 12k of memory designated as 'spare' by Acorn. The key has no effect on the Model B and Electron.

f5 - Toggle I/O

The three pages of ROM between &FD00 and &FEFF are not usually accessible, as these addresses are occupied instead by I/O ports 'on top' of the RAM. On the Master Series only, it is possible to switch back in the ROM. Pressing f5 does this, and you can see the contents of the ROM in this area. The default is that I/O is read (IO=IN).

8 : Hints and Tips

This section contains a few advanced points about the use of *Fingerprint* which may make life easier.

- a. Where ever possible use the Sideways RAM version of *Fingerprint*. The main memory version is only really provided for users without Sideways RAM.
- b. When tracing code that calls sideways ROMs (e.g filing system calls) it may be helpful to reduce the main range (range 1) to be &0000 to &7FFF. This will stop lots of unnecessary output.
- c. *Fingerprint* cannot trace when IRQs are disabled, that is between an SEI instruction and the corresponding CLI. During this period no display will be generated. Many routines that turn interrupts off for a long period execute the sequence CLI:SEI periodically to let in any pending interrupts. This will cause *Fingerprint* to generate a single display.
- d. Don't let the stack depth become negative. This means that the target program is trying to pull to much off the stack and will probably crash. Routines which for some reason take the return address off the stack, perhaps to locate some following text, shouldn't be started directly by *Fingerprint* as they won't get the right address from the stack. Such routines can however be used without problems from within the target program.
- e. Watch out for language start-ups. These reset the stack pointer to &FF. If this is done *Fingerprint* should still trace okay but you will encounter problems when you try to exit the program and may need to turn the computer off.
- f. *Fingerprint* manipulates the stack depth extensively. Avoid using any address between &180 and &1FF as general workspace as the stack may crash into it. Programs which assume something about the stack depth (e.g the PROC handler in BASIC, or the Macro caller in EDIT) may require a helping hand to run properly under *Fingerprint*.
- g. Since *Fingerprint* intercepts all instructions, some MOS calls can take a long time to execute. In particular be patient over MODE changes.
- h. *Fingerprint* uses the default VDU 7 beep for its beeps and channel 3 for the high and low pitched toggles. Try to avoid using channel 3 in the target program as this can create problems.

- i. Soft keys and windows can be used from within the target program without affecting *Fingerprint*. Note however that soft scrolling under *Fingerprint* is slow.
- j. Don't switch ROMs within a ROM (i.e. don't have LDA #xx:STA &F4:STA&FE30 in a ROM). This won't execute normally and is little improved under *Fingerprint*
- k. To set a particular value in a register before starting the target program do it with the change register commands immediately after starting *Fingerprint*. The same applies to the flags.
- l. When starting *Fingerprint*, the ROM number selected is the current ROM for the main memory version, and either the last used ROM or undefined if none has been used for the Sideways RAM version. To start in a particular ROM change ROM with CTRL-f9 when *Fingerprint* is started.
- m. When *Fingerprint* hits a BRK instruction it will turn off tracing and then continue execution normally. This is so that errors are handled properly.
- n. The Sideways RAM version of *Fingerprint* uses the memory between &7B0 and &7EF to put some patch code in. Do not use this area of memory.
- o. *Fingerprint* uses no user workspace other than for the code itself. The bottom 64 bytes of page zero are swapped in and out as required.
- p. The relocater on the main memory version of *Fingerprint* is overwritten once it has been used. Therefore is not possible to move *Fingerprint* around in memory once it has been executed, or to save it properly after it has run.

9 : Summary of commands

Main *Fingerprint* monitor commands

- SPACE - Next inst / Single step
- ESCAPE - Abort command
- CTRL ESCAPE - Quit *Fingerprint*
- CTRL TAB - Force keyboard scan
- f0 - Free run mode
- f1 - Decrease execution speed (Down arrow)
- f2 - Increase execution speed (Up arrow)
- f3 - Set breakpoint
- f4 - Clear breakpoint / Confirm
- f5 - Set a range

- f6 - View memory
- f7 - Move back a byte (Left arrow)
- f8 - Move forward a byte (Right arrow)
- f9 - Alter memory contents (Copy) / Switch SP (see text)
- sf0 - Change processor
- sf1 - List past instructions
- sf2 - OS command (*)
- sf3 - Disassemble
- sf4 - Call user routine
- sf5 - Printer toggle
- sf6 - RS423 toggle
- sf7 - Screen enable
- sf8 - Restore defaults
- sf9 - Display status
- cf0 - N flag toggle
- cf1 - V flag toggle
- cf2 - D flag toggle
- cf3 - Z flag toggle
- cf4 - C flag toggle
- cf5 - Alter accumulator
- cf6 - Alter X register
- cf7 - Alter Y register
- cf8 - Jump to new address
- cf9 - Change ROM

MEMX Memory editor commands

- f0 - New address
- f1 - Alter ROM
- f2 - Toggle LYNNE
- f3 - Toggle HAZEL
- f4 - Toggle ANDY
- f5 - Toggle I/O

DISASS Disassembler commands

- ↑ - (Up arrow) Disassemble lower memory addresses
- ↓ - (Down arrow) Disassemble higher memory addresses
- SHIFT-↑ - Move back one screen
- SHIFT-↓ - Move forward one screen
- ESCAPE - New address / Exit (see text)

Files on the Fingerprint disc

- !BOOT - Boot file to CHAIN the INTRO program
- INTRO - Simple program to start up *Fingerprint*
- COPY40 - 40 to 80 Track copier for 8271 only.
- MON - The relocatable main memory version of *Fingerprint*.
- SMON - The sideways RAM version of *Fingerprint* including utilities.
- SMON2 - SRAM version without utilities (for 8K sideways RAM).
- TMON - The Tube version of *Fingerprint*.
- THOOK - A file which must be present each time TMON is started.
- DISASS - The disc based scrolling disassembler.
- MEMX - The full screen memory editor.
- ROMLIST - The sideways ROM lister.
- HELP - A short piece of text giving the key functions.

10 : Other Dabs Press products

The products listed are all available now.. All prices include VAT and UK postage. Overseas orders add £2.50 for surface mail, £12 for airmail. Ordering details appear at the end of the list.

VIEW : A Dabhand Guide by Bruce Smith

This 248 page book is a complete tutorial and reference guide to the Acornsoft VIEW wordprocessor, covering everything from introductory wordprocessing to advanced techniques. The accompanying disc contains several utility programs which are printed in the book, and other programs of interest to VIEW users, including a Printer Driver Generator. The book also covers ViewSpell and ViewIndex. Price £12.95 (book), £7.95 (disc), £17.95 (book and disc if ordered direct).

Master Operating System : A Dabhand Guide

At 272 pages, this detailed programmers manual for Master, Compact and BBC B+ owners contains detailed information on all system calls and "*" commands as well as much previously unpublished information on the Tube, CMOS RAM/EEPROM and other new features. The accompanying disc contains the many utility programs in the book, with several extra programs, including a utility ROM image for programming as an EPROM or loading into sideways RAM. Price £12.95 (book), £7.95 (disc), £17.95 (book and disc if ordered direct).