

The Solidisk

Twomeg and Fourmeg

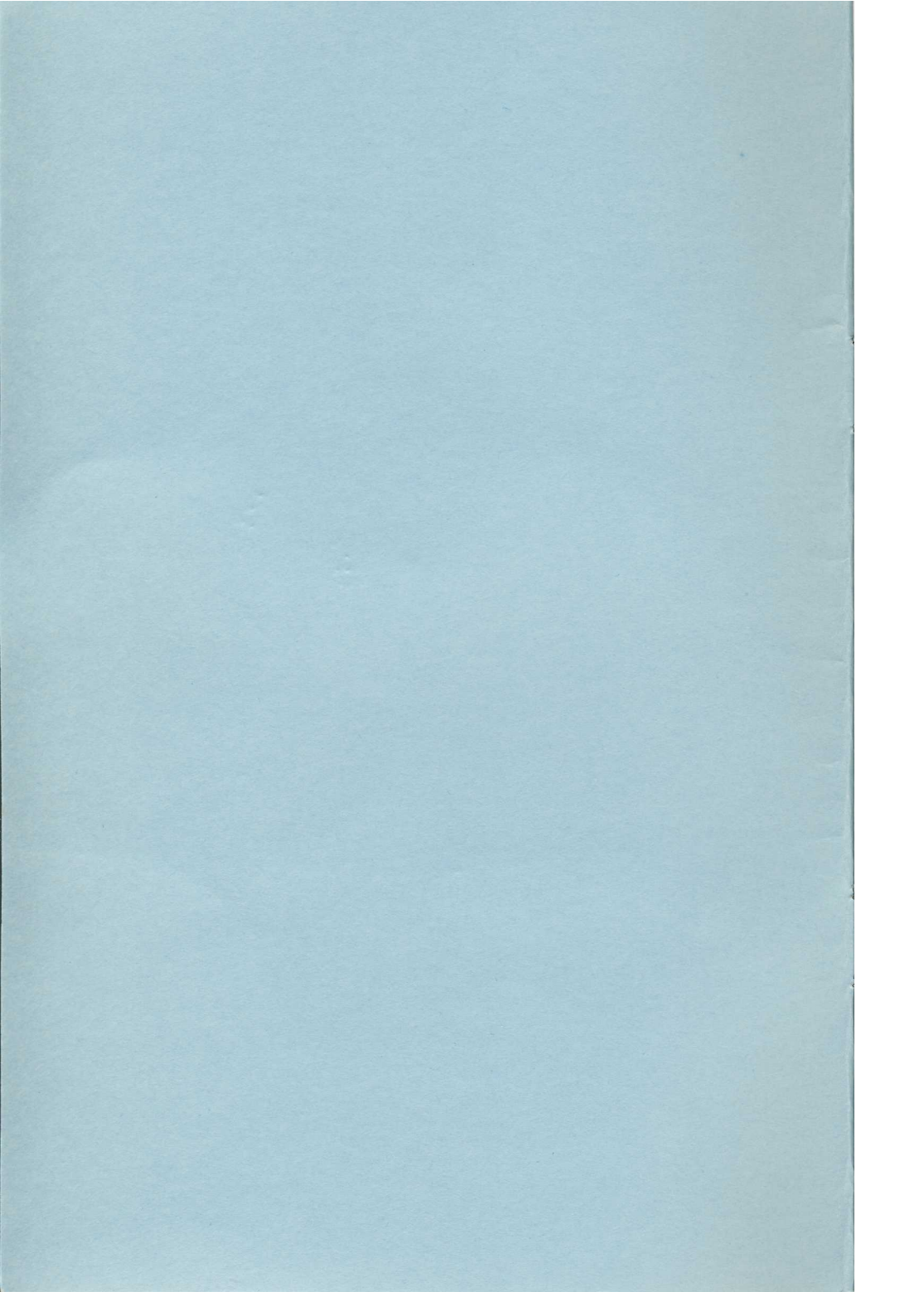
RAM/ROM Expansions

SOLIDISK TECHNOLOGY

17 Sweyne Avenue

Southend-on-Sea

Essex SS2 6JQ



CONTENTS

I. Installation

2. The *ROMS command
3. The *MENU command
4. The *RAMDISK command
5. The *SHADOW command
6. The *PRINTER command

Appendices

- A1—The MANAGER commands
- A2—The TOOLKIT commands
- A3—The LINEMASTER commands
- A4—Software on disc: VSTORE and WINDOS
- A5—Technical details of the Twomeg and Fourmeg boards

Release date: 4th August, 1986
Copyright Solidisk Technology
17 Sweyne Avenue
Southend-on-Sea
Essex

1. INSTALLATION

If you feel that you would have any difficulty in installing the Solidisk fourmeg Board, please please call our offices for the name and telephone number of your nearest local expert, who will carry out the work free of charge.

The fourmeg board is connected to the BBC computer by:

- two 40-pin sockets (blue) that support the board on the 6502 socket;
- three colour coded wires to be pushed on to the pins of S20, S21 and S22.
- two spring clips to be clipped on to points in the BBC micro.

PROCEDURE

1. Open the computer case by undoing the four large self-tapping screws which may be labelled "fix".
2. Remove the 6502 (IC1). Fit the two blue sockets in its place and replace the 6502 on top and switch on the ocomputer. Switch it off. It should work normally. If not, then check for bent pins, repeating the last stage until it does.
3. Remove the three links (short bars) on S20, S21 and S22. S21 has two links; remove only the one on the South (the keyboard)
4. Install the three colour coded wires as follows:
Grey to S20, middle pin
Blue to S21, South-West corner pin
Maue to S22, middle pin
Do not replace the original links, but keep them in a safe place.
5. Clip the spring clip attached to the pad on the fourmeg board marked "ROM select" to pin 9—South-East corner—of IC76 (7LS163N). An alternative point for this connexion is pin 5 of IC26 (74LS139N).
6. Clip the sprung clip attached to the pad on the fourmeg board marked "4MHz clock" to the leg of the resistor marked R106 on the BBC board. It is located a couple of centimetres South of the cassette socket (see picture).
7. Remove the 6502 chip again. Install the fourmeg board in its place.
8. Switch on the computer. You should see:
Solidisk ZZZK Expansion (XMHz)¹
BBC Computer 32K
DFS 2.1M (1770)²
BASIC
>

1. Depending on the board and position of the changeover switch (if fitted). If the message does not correspond to the size of the RAM on the board, or you get the message 'no 4MHz clock', this may be due to variations in the 4MHz supply between BBC micros, then it may be necessary to use a different pick-up point for the supply of the 4MHz clock. If it shows less than the correct number of bytes, switch off the computer and try attaching the clip labelled "4MHz clock" to a different point; first tray pin 5 of IC44 (74LS74N) and switch on again. If this still does not work, try pin 2 of IC44, or pin 12 of IC43 (74LS04N). If none of these 4MHz supplies passes the memory test, please ring the Solidisk office for advice.
2. Or any other Disk Filing System fitted.

***ROMS**

When the *ROMS command is issued, the machine will list all the ROM software (firmware) in residence.

A typical display might be thus:

>*ROMS

```
00 STL ADFS (S) 16k
01 DFS 2.1 M (1770) (S) 16k
02 Clock and Desk (S) 16k
03 Linemaster (SL) 16k
04 MANAGER (S) 32k
05 SOLIMON (SL) 16k
06 UVIPROM (D) 16k
07 BASIC (LR) 16k
```

*** Back up Rom store ***

```
21 INTER-CHART (SL) 16k
22 VIEW (SL) 16k
23 DFS,NET (S) 16k
```

>

Where (in parentheses after the name of the ROM) 'S' means Service, 'L' means Language, 'D' means Disabled, and 'R' means Second Processor Relocatable.

The various items of firmware which appear in the backup store (which are blown into 32k EPROMS-27256 devices) are available from the *SELECT command (q.v.).

THE *MENU COMMAND

*MENU issued on its own will produce a menu of drive 0, although when followed by a parameter (between 0 and 3) will, subject to DFS or ADFS firmware, produce a menu of the relevant drive.

The menu will accept various commands from the keyboard.

These are:

- <RETURN> or <ESC> terminate
- <TAB> switch to alternate catalogue (ADFS mode only)
- <1> to <5> switch drives (but see above)
- "<" or ">" change currently selected RAM bank
- "?" new disc
- "*" perform hard break
- "^" return to previous directory (ADFS only)

On the MENU screen, in DFS mode, the resident ROMS are displayed and numbered; those that are disabled are shown in red. Below this display is shown the catalogue information of the disc. In ADFS mode this is available as a second screen merely by pressing <TAB> ; a second depression of the <TAB> key will revert to the original screen.

The files shown all have letters next to them, and pressing the relevant key will CHAIN, *RUN, *EXEC, *TYPE, *LOAD, etc., the selected file. In ADFS mode, if the file is in fact a directory, this will be selected, and can be displayed by pressing <TAB>.

THE *RAMDISK COMMAND

The RAMDISK is created by typing in:

*RAMDISK (<drive number>) <RETURN>

The drive number can be from 0 to 3 using the normal BBC Disc Filing System (DFS) or from 0 to 7 using the Advanced Disc Filing System (ADFS) although in ADFS mode, if a Winchester disc drive is not connected, only drive numbers 0 to 3 will be accepted.

The RAMDISK created will be in the same format as the current disc filing system, and of 200k in either case. In DFS, this is equivalent to a single sided 80-track (200k diskette).

If the RAMDISK is already present, *RAMDISK <drive number> will simply change the drive number and leave the contents unaffected.

The RAMDISK will survive the normal BREAK key, but CTRL+BREAK will turn it off. However, it may be recovered by typing in *RAMDRIVE <number>.

N.B. This command does not initialise a RAMDISK, and therefore should only be used for recovery.

If you wish to create a RAMDISK in a specified format, then you should use:

*RAMDFS (<drive number>) <RETURN>—for DFS

*RAMADFS (<drive number>) <RETURN>—for ADFS

Once created, the RAMDISK will behave exactly like a normal disc. For example COPY, BACKUP, DELETE, DZAP, etc., will work as usual. However, there is no need to FORMAT the RAMDISK before use.

Should the RAMDISK be no longer required, it is disabled by "
*RAMOFF".

Because protected software usually relies upon special formatting of the disc which contains the programs, and the RAMDISK only uses pointers to the stored programs, it is impossible to transfer software from protected discs to RAMDISK.

Notes:

1. The minimum acceptable abbreviations are:
 - *RA. for RAMDISK
 - *RAMA. for RAMADFS
 - *RAMD. for RAMDFS
 - *RAMDR. for RAMDRIVE
2. The Solidisk DFS or ADFS ROMs must be present to allow a RAMDISK to be created. If not, they must be loaded into Sideways RAM bank F. Solidisk will upgrade your present DFS ROM at nominal cost if required, or free if it is a Solidisk DFS.

You can find out which version of DFS you have in your machine by typing:

*HELP <RETURN>

The following releases will support RAMDISK

DFS 2.2 F DFS 2.2 M DFS 2.2 J

DFS 2.1 J issue 1 or issue 2

DFS 2.1 M issue 1 or issue 2

ADFS 2.1 D issue 1 or issue 2

ADFS 2.1 M issue 2

ADFS 20/2 issue 2

THE SHADOW COMMAND

When used as Shadow RAM, it shadows the BBC screen memory, allowing BASIC (and other languages) more workspace, up to &8000 in any screen mode.

The commands to toggle Shadow memory on/off are the same as on the BBC Plus. You can use *SHADOW or *FX114 either directly from the keyboard or in BASIC and other programs (including machine language).

To turn the Shadow RAM on, type in:

```
*SHADOW <RETURN>
```

or:

```
*FX114. <RETURN>
```

followed by a mode change (even to the same mode).

These steps must be included in any !BOOT file requiring Shadow mode.

BBC Plus emulation is achieved by following the command the *FX114 by a non-zero number, for example:

```
*FX114, 1 <RETURN>
```

This will select Shadow RAM when the mode change is to a number over 127: MODE 128=MODE 0 with Shadow; MODE 135=MODE 7 with Shadow, and so on. A mode number between 0 and 7 will turn Shadow off.

To turn the Shadow RAM off, type in:

```
*SHADOW OFF <RETURN>
```

Notes:

1. The simplest way to run a program in Shadow mode is to type in *SHADOW <RETURN>, press <BREAK> then load and run your program.
2. You should not change the Shadow status inside your programs or after loading them (unless you know what you are doing) because this may result in a bad program error.
3. *SHADOW will survive the action of the <BREAK> key alone but will be turned off by <CTRL> + <BREAK> or switching the computer off and on again.
4. *SHADOW will use the first RAM bank (bank 8, which shows as Shadow 0). If you need to assign any other bank number to the Shadow, then *FX115, <bank number> must be used afterwards to allocate the Shadow RAM bank for the computer.
5. Only bank 0 is available when the RAMDISK is active.

SHARING RAM BETWEEN SHADOW AND SIDEWAYS SYSTEMS

You cannot use the same areas of RAM twice over, but the 256k board can be shared between the Shadow and Sideways system. The RAM on the board is physically arranged as eight banks of 32k, each mapped to occupy a VARIABLE but CONTINUOUS block somewhere between &800 and &FFFF in the memory map. Each bank is normally used solely as Shadow RAM or Sideways RAM but not without some compromise; the total requirement must not exceed 32k.

1. If used as Shadow RAM, the block will normally extend from &3000 to &B000. In this case the top 12k (between &8000 and &AFFF) is free and can be used as Sideways RAM.
2. If your program needs a very large Shadow area you can position your block lower, from &800 to &8800. In this case, the bottom 30k is available as Shadow RAM and the top 2k can be used as Sideways RAM. To put the block in this position:

```
?&FE36=&40 <RETURN>.
```

3. If your program can be restricted to screen mode 3, the block can be positioned higher, from &4000 to &C000. In this case the top 16k (between &8000 and &C000) is available to hold a full 16k ROM image. To put the block in this position you type in ?&FE36=&20 <RETURN>. This may be done either from the keyboard or from a program.
4. The block can also be positioned from &8000 to &FFFF. In this case no Shadow RAM is available, but the Sideways RAM is 32k. To put the block in this position, &10 should be poked into &FE36—but you must not type it in from the keyboard or the computer will crash. It must be done in machine code, as in the Manager ROM. Solidisk will supply the "PATCH" program to software authors wishing to develop ROMs to occupy this position.

THE *PRINTER COMMAND

The *PRINTER command is used to copy the Printer Buffer software from the MANAGER ROM into a sideways RAM bank.

Its syntax is •PRINTER <RAM bank>

When the printer buffer is required, the <BREAK> key must be depressed in order to initialise it. The Printer Buffer software is only available for a parallel printer, -and the current state of buffer may be obtained by "PRINT ADVAL(-4)". As bank switching is dangerous when the possibility of NMI occurring is likely, the buffer software is written to address only 16k, which should be more than adequate for most purposes.

THE MANAGER COMMANDS

The MANAGER ROM is concerned with the control of the RAM board's various functions.

The commands available are:

PLUS

SHADOW

RAMDISK

RAMOFF

RAMDFS

RAMADFS

TESTRAM

RAMDRIVE

ROMS

SELECT

TOOLKIT

***PLUS**

This command provides BBC + type Shadow RAM emulation, most notably for firmware packages such as Wordwise Plus, Inter-Word, Inter-Sheet, etc. This is also obtainable by *FX114, 1. Used with a change of mode, this will give Shadow RAM or not, as per the mode number. For example, MODE 0 with *BASIC gives PAGE at &3000; MODE 128 with *BASIC gives the same screen mode but leaves PAGE at &8000.

***SHADOW**

*SHADOW is the command used to invoke Shadow RAM where all screen modes selected are shadowed by the inbuilt RAM. When followed by "OFF" it turns off Shadow RAM, and the machine is returned to its normal state from the next change of language (even to the same one) .

Syntax:

*SHADOW OFF

THE *RAMDISK SERIES OF COMMANDS

These commands are thoroughly documented in chapter 4, so we shall not go into them in detail.

***TESTRAM**

*TESTRAM does exactly as it says—and tests the sideways RAM on the fourmeg board, non-destructively.

***ROMS**

This command is documented in chapter 2, so we will not go into it.

***SELECT**

The *SELECT command is available to copy the contents of a backup store on a 32k ROM to sideways RAM for use. The *ROMS command (q.v.) enables the identification of suitable ROM software for this.

Syntax:

*SELECT <RAM bank> <ROM name—or an abbreviation> <RETURN>

An example of this command, using the list of ROMS in chapter 2, would be:

*SELECT F DFS, NET <RETURN> (or *SELECT F DFS <RETURN>)

TOOLKIT

This is not a command as such, but merely a reminder that the complete list of commands available from the toolkit can be listed by "*HELP TOOLKIT".

RECVDU

Description

*RECVDU copies all VDU commands to an allocated area in sideways RAM. This area may be varied by copying the VDU recorder into a higher or lower RAM bank. The higher it is, the smaller the RAM available will be. Once installed, <BREAK> must be pressed in order to initialise the ROM image.

Syntax

*RECVDU <RAM bank>

Commands

*RECORD records the VDU output

*STOP stops the above process.

*ADD appends a sequence to one already recorded, and is halted by

*STOP

*PLAY shows the result of your work !

PRINTER

Description:

*PRINTER causes a printer buffer to be created in the SWR in the specified RAM bank. <BREAK> must be pressed in order to initialise it.

Syntax:

*PRINTER <RAM bank>

Solidisk Toolkit

CONTENTS

Preface

Introduction	
Starting to use Toolkit	
Toolkit Commands	
Memory Usage	
Command Summary	
Glossary	
Index	

by Ian Homan

COMMAND SUMMARY

BFREE	Show Free memory.
CURSOR	Turns on or off the blinking Cursor.
CHECK	Allows you to check a program in memory with a file on disk.
EDIT	Will display all the function key definitions. EXCHANGE Allows you to change one string for another one.
EXPAND	This command will display the program in an easily read form.
FIND	Will search for a string and print the line number for each occurrence.
HARDBREAK	This command will fool the computer into thinking it has just been turned on.
JOIN	The Join command will GLUE a file on to the end of the program in memory.
KEYLOAD	Will load a set of Function key definitions from Disk.
KEYSAVE	Will save a set of Function key definitions to Disk.
KILL	Will disable the Toolkit rom.
LVAR	Will show all variables o rprocedures.
MENU	Will produce a menu of all the files on the drive and all the SWRs.
MSEARCH	Will hunt through from a specified address for a certain string.
PAGE	Will remove a program and PAGE to the specified address.
RCAT	Displays a catalog of all Roms in the machine.
NUMBER	Allows you to renumber a program in parts or whole.
REPORT	Returns a report of the last error.
RLOAD	Allows a rom to be loaded into a certain SWR hank.
RWIPE	Will step through each rom and allows you to enable or disable them.
SALVAGE	Will attempt to recover a 'Bad Program' from memory.
STATUS	Produces a report based on free memory and FX settings.
SOUND	Allows sound to be turned on or off.

STARTING TO USE THE TOOLKIT

Toolkit is a Rom for the BBC containing a set of machine code utilities enhancing the basic language. All commands must start with an asterisk '*' in order so as to be passed to the paged Roms installed in your BBC. All commands sent to the Toolkit Rom can be in 'UPPER' or 'lower' case—for example:

*STATUS or *status

Furthermore, commands can be abbreviated in the usual way by entering the first few letters—for example:

*ST.

The only problem with abbreviating commands is they can clash with other Roms. In the event of two Roms clashing you can prefix Toolkit with a lower case s.

TOOLKIT COMMANDS

This chapter lists all the different commands in alphabetical order with a description of each. Most of the commands in Toolkit can be used in direct or indirect mode, but it is possible to have unusual effects—for example, a program performing EXCHANGE while running will wipe out all the variables.

INTRODUCTION

Now type in *HELP TOOL. (CR). You should now see:

BFREE	
CURSOR	<ON/OFF>
CHECK	<fsp>
EDIT	
ERASE	
EXCHANGE	<(str) (str)> (G/S)
EXPAND	
FIND	<str>
HARDBREAK	
JOIN	<fsp>
KEYLOAD	<fsp>
KEYSAVE	<fsp>
KILL	
LVAR	(V I S A P F) (H/D)
MENU	(drive)
MSEARCH	<addr> <str>
MZAP	(<addr>)
PAGE	<addr>
PRINTER	<bank>
RCAT	
NUMBER	<S,F:S,I>
RECVDU	<bank>
REPORT	
RLOAD	<bank> <str>
RWIPE	
SALVAGE	
STATUS	
SOUND	<ON/OFF>

This proves that the Toolkit Rom is functioning correctly. If this does not occur, go back to stage 4 and check for bent pins. If the Rom still fails to work, please contact the technical department here at Solidisk.

The following abbreviations mean:

addr	Address in Hexadecimal
bank	Sideways Ram bank
drive	Disc drive surface
fsp	File name
str	String

The () parentheses indicate that parameter inside them is optional and < > arrows indicate that parameters are compulsory.

BFREE

Description:

BFREE will print up all relevant information about the program. It is useful to see how much memory is left and how long the the progra mis.

Syntax:

*BFREE

Example:

Free memory= 25342 bytes
Program size =2 bytes
Vars. size = 0 bytes
Next free location= &1902
PAGE= &1900 LOMEM=&1902
TOP= &1902 HIMEM=7C00
HEAP = &1902 STACK = &7000

CURSOR (ON/OFF)

Description:

CURSOR will turn the cursor ON and OFF until you change mode.

Syntax:

*CURSOR (ON/OFF)

CHECK

Description:

CHECK will perform byte-by-byte check on a basic program in memory, with the file name which you specify. If an error occurs the line will be listed and no further checking will be carried out.

Syntax:

*CHECK (str)

Comment:

CHECK is most useful when it is used for checking if the basic program in memory has been changed from the program previously saved.

Example:

***CHECK**

Syntax:CHECK (fsp)

File not found

LO. "MENU"

***CHECK MENU**

Verification completed ok.

6o PRINT "This is a test"

***CHECK MENU**

Verification fails

6o PRINT "This is a test"

EDIT

Description:

EDIT displays the Function key definitions so as to facilitate the editing of them.

Syntax:

*EDIT

Example:

*K.0Print —PAGE M

*EDIT will give

*K.0Print —PAGE M

ERASE

Description:

ERASE simply clears sideways RAM and performs a power-up break.

Syntax:

*ERASE

Comment:

Holding down the <E> key and pressing <**BREAK**> also executes this routine.

EXCHANGE

Description:

EXCHANGE searches through the program in memory to try to find the first string. If it is found it then replaces it with the second string. The strings will NOT be tokenised unless they are prefixed with a '@' sign before the string.

Syntax:

*EXCHANGE <str> <str> (G/S)

EXPAND

Description:

EXPAND will list the program out in a more reasonable form by performing a carriage return before each semi-colon. To exit from EXPAND just press the 'escape' key.

Syntax:

*EXPAND

Example:

NEW

10A=0:B=0:C=0:D=0:E=0

*EXPAND

10 A=0

: B=0

: C=0

: D=0

: E=0

FIND

Description:

FIND is used to search through an entire program for a specified set of characters (string). If found, the line with the relevant string will be listed.

Syntax:

*FIND <str>

Comment:

Before executing the FIND command it is advisable to type '(CTRL) N' to turn on paged mode in order that the text does not scroll off the screen. To exit from FIND press escape. Machine code programmers wishing to find labels will type in '*FIND.(CR)'.

Example:

*FIND &FEG0

HARDBREAK

Description:

The Hardbreak command simulates 'power on' of the computer, which will completely reset it.

Syntax:

*HARDBREAK

Comment:

Holding down the<H> key while pressing <BREAK> also executes this routine,

JOIN

Description:

The JOIN command 'glues' the file you specify on to the program in memory. It will only work with Basic programs. Once it has loaded the next program you have the option to 'RENUMBER'.

Syntax:

*JOIN (fsp)

***KEYLOAD**

Description:

The KEYLOAD command will load in any key definitions already saved. Once loaded you can edit the keys with the EDIT command.

Syntax:

*KEYLOAD (fsp)

Example:

*K.OP.'HELLO"

*KEYSAVE keys

*K.O

*KEYLOAD keys

*EDIT

•K.O.P."HELLO"M

***KEYSAVE**

Description:

The KEYSAVE will save any predefined key definitions under the file name specified.

Example:

*K.OP."HELLO"

*KEYSAVE keys

*DUMP keys

KILL

Description:

The KILL command simply kills off the Toolkit Rom until Break. It is most useful for preventing T00lkit dashing with a Rom when you cannot prefix each command by a letter.

Syntax:

*KILL

Example:

*STATUS could go to the Toolkit.

*sSTATUS would go to the Toolkit.

*KILL

*STATUS would be sent to any Roms below Toolkit.

LVAR

Description:

LVAR will alphabetically list all the current variables. The parameters are:

- | | |
|-----------|---|
| *LVAR V | will list all variables in hexadecimal. |
| *LVAR V D | will list all variables in decimal. |
| *LVAR I | will list all integers in hexadecimal. |
| *LVAR I D | will list all integers in decimal. |
| *LVAR S | will list all strings. |
| *LVAR A | will list all variables, strings and integers in hexadecimal. |
| *LVAR A D | will list all variables, strings and integers in decimal. |
| *LVAR P | will list all DEF PROC's. |
| *LVAR F | will list all DEF FN's. |

***MENU**

Description:

The MENU command works in DFS and ADFS. The only parameter is the drive number, an dthat is optional. Commands in the menu are:

'<' select a lower Ram bank
'>' select a higher Ram bank
'?' recatalog disk
'@' save Rom
'*' performs a 'power up' reset run Rom
'0' select drive 0
'1' select drive 1
'3' select drive 3

To exit from the MENU simply press 'RETURN'. To lohad software just press the letter corresponding to it. If its execution address is &801F or &8023 then it is a basic program and it will be changed. If its execution address is above &8000 then it must be a Rom and it will be loaded into sideways Ram. If its execution address is &0000 or &FFFF it is a ASCII file and it will be *DUMPed. Anything else is a machine code program and will be *RUN.

Syntax:

*MENU (drive)

***MSEARCH**

Description:

MSEARCH will search through memory from the specified address to try to find the search string. It will search up to address and &FFFF and then stop.

Syntax:

*MSEARCH <addr> <str>

Example:

MODE 7
\$&4000= "This is a test"
*MS.1900 This is a test
String found at location &4000

MZAP

Description:

This allows examination and editing of memory contents. The <TAB> key switches between entry of ASCII or hexadecimal (refer to top right-hand corner of screen). The cursor keys scroll through memory, in all directions. This effect may be accelerated by the <SHIFT> key. The MZAP mode is exited by pressing <ESC>.

Syntax:

*MZAP (<addr>)

PAGE

Description:

The PAGE command will move a basic program to the address specified. If you move below &1200 type *TAPE first and move the program down. To save the program move it back to being above &1200 and then type *DISK before saving it.

Example:

NEW

LOAD "MENU"

*TAPE

— PAGE E00

OLD

*PAGE 1900

*DISK

SAVE "MENU1"

RCAT

Description:

The RCAT command lists all the Roms in your computer.

*RC.

- 0 (L) SFX Graphics
 - 1 (L) SOLIMON
 - 2 (L) BASIC
 - 3 (S) UVIPROM
 - 4 (S) UVIPROM
 - 5 (S) DFS 2.1 (177o)
 - 6 (S) STL TOOLKIT 1.20
 - 7 (S) Printer Buffer
 - 8 (S) Vdu Recorder
 - 9 (D) Game Master
- L means it is a language Rom
S means it is a service Rom
D means it has been disabled

Syntax:

- RCAT

NUMBER

Description:

The NUMBER command is an enhanced version of the Basic renumber. It has four parameters all of which are compulsory. The parameters are as follows:

NUMBER (S,F:N:I)

- S is the start line
- F is the line which RENUMBER will finish on
- N is the new line number to be assigned to the Start line
- I is the increment

It is possible by pressing 'CTR F0' to 'CTR F9' to insert teletext control codes into your program. Both Basic and Toolkit RENUMBER will try to match these codes with basic tokens and try to renumber them, but a 'RENUMBER failed at line' error will occur.

Example:

*NUMBER 100.300:200,2 will renumber from line 100 to 300, making lines 100 to 300 become 200 onwards. The increment will be in steps of 2.

REPORT

Description:

The REPORT command will report the last error, from any Rom.

Syntax:

*REPORT

RLOAD

Description:

RLOAD will load a ROM into the sideways RAM bank specified.

Syntax:

*RLOAD <bank> <fsp>

Example:

*RLOAD 8 UVIPROM will load UVIPROM into Ram bank 8.

RWIPE

Description:

The RWIPE command will kill off any Rom after break. All the Roms will be listed so you just have to type 'W' to the Roms you want wiped.

Syntax:

*RWIPE

Example:

*RWIPE

Wipe W, Restore R, Leave L

0	(L)	SFX Graphics	:Leave
1	(L)	SOLIMON	:Leave
2	(L)	BASIC	:Leave
3	(S)	UVIPROM	:Wipe
4	(S)	STL ADFS	:Leave
5	(S)	DFS 2.1 (1770)	:Leave
6	(L)	Printer Buffer	:Leave
7	(D)	Vdu Recorder	:Restore

SALVAGE

Description:

The SALVAGE command will recover a bad program or recover it as much as possible. If the start to the program doesn't commence with a carriage return the SALVAGE routine will come back with the error "No CR at start". If this happens just type ?(PAGE)=13 and then *SALVAGE. If the SALVAGE routine ends before it should it has found the negative byte as the MSB of the next line number. To make it continue type '?(TOP=1)=0' and then '*SALVAGE'.

Example:

>LO."MENU"

>NEW

>A=0

>OO. (U.C.)

Bad program

>*SAL.

Addr	Line	No.
------	------	-----

&1900	10	Failed Link
-------	----	-------------

Control char A/F/T

&1904	0	Failed Link
-------	---	-------------

Control char A/F/T.

&1908	38	Failed Link
-------	----	-------------

&1922	20	
-------	----	--

&1958	30	
-------	----	--

&1961	40	
-------	----	--

SOUND**Description:**

The SOUND command will turn off the sound until you turn it back on with a '*SOUND ON' command.

Syntax:

*SOUND <ON/OFF>

Example:

*SOUND OFF will turn off the sound

*SOUND ON will turn on the sound

STATUS

Description:

The STATUS command displays all the relevant information about the computer.

Example:

```
< *STATUS
```

```
FX 3 0 FX 4 0  
FX 5 1 FX 6 10  
FX 9 25 FX 10 25  
FX 5 0 FX 12 8
```

```
@% = &0000090A
```

```
LISTO 0 ERR 0  
WIDTH 0 ERL 0
```

```
SOUND OFF          BRK OFF  
CURSOR ON          BASIC 2
```

```
REPORT: © 1982 Acorn
```

```
Free memory=25342 bytes
```

```
Program size=2 bytes
```

```
vars. size=0 bytes
```

```
Next free location= &1902
```

```
PAGE=&1900 LOMEM=&1902
```

```
TOP=&1902 HIMEM=&7000
```

```
HEAP= &1902 STACK= &7000
```

MEMORY USAGE

The Toolkit Rom uses certain memory locations inside the BBC to operate correctly. These locations are:

&50-&60 General Toolkit Workspace

&90-&9F Permanent Toolkit Workspace

Page &C for switching between the Toolkit and Basic Rom.

Why the Toolkit uses certain memory locations.

&50-&60 These locations are used by the paged Roms as scratch pad.

&90-&9F These locations are the Econet workspace, but are used by the Toolkit for:

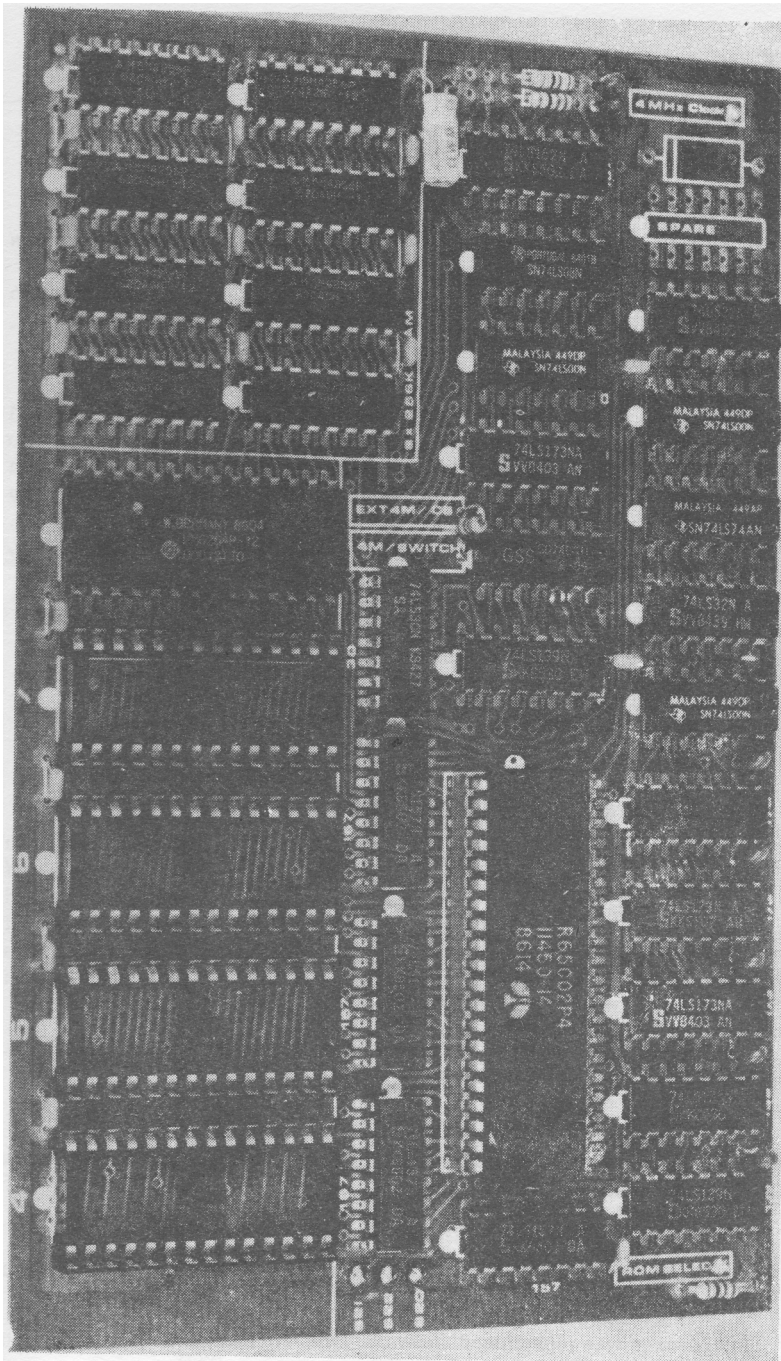
&90 Toolkit Check digit.

&91 Each bit represents bank &0-&7

&92 Each bit represents bank &8-&F

If the bit is set to 1 then that Rom is disabled, or if the bit is set to 0 the Rom works as normal.

&93-&9F These locations are used by the Solidisk Basic 3 for general workspace.



Twomeg and Fourmeg RAM/ROM Expansions

TECHNICAL DETAILS OF THE FOURMEG BOARDS

The STL fourmeg and two megahertz boards have three special registers for their own use. These are &FE32, &FE34 and &FE36. NB: These registers (like &FE30) are read only, and any attempt to read them will produce the value of the high order address (FE or 254).

THE RAM ACCESS REGISTER &FE32

The RAM access register controls which RAM bank is currently selected for any write operation, thus a RAM or ROM may write directly to itself or to another bank. This is useful when a multi-bank operation is required (as in RAMDISK, or VDU RECORDER). A RAM may also wish to write to itself (as in the PRINTER BUFFER program). In this way, loading of a RAM may be accomplished simply as in, say, a boot file as follows:

```
0001 ?&FE32=9
0002 *LOAD SOLIMON
0003 ?&FE32=
0004 *LOAD UVIPROM
0005 *K .10*SOLIMONIIM
0006 CALL !-4
```

THE SHADOW REGISTER &FE34

The shadow register is associated with whether the RAM bank in question is being used as a SHADOW RAM or not. Earlier versions of the MANAGER ROM did not clear this register correctly on a hard break, thus necessitating poking the register direct to clear it (?&FE34=0), thus returning the display to normal. Simply, if this register holds a negative number (has the top bit set), then screen writes go into the SHADOW RAM bank and are displayed from there.

In order to read the Shadow screen it is necessary to use OSRDRM. This takes as its parameters:

"Call address: &FFB9

Y = ROM number, Locations &F6 and &F7 contain the address of the byte to be read.

On exit, the Accumulator contains the value of the byte to be read.

This routine has not been documented by Acorn, but has been used in applications software."

This passage was reproduced from the BBC Advanced User guide, with acknowledgements.

THE SIZE REGISTER &FE36

The size register controls the size of the sideways RAM, or ROM and is used thus:

- ?&FE36=0: RAM and ROM are configured as normal (i.e. each bank is 16K long)
- ?&FE36=&10 RAM and ROM (where the ROMs are larger than 16K) are configured as 32K banks (wiith the top 1K from &FC01) giving access to the operating system through a hardware 'window'—beware of OS calls with &FE36=10. It is better to set interrupts when using this, taking care not to generate NMI, or the machine may hang, and to restore the contents of &FE36 when the process is over. It is not necessary to set this register when a write operation is being performed, as access is available at all times to the upper portion of RAM (up to &FC00) on write.

?&FE36= &20

?&FE36= &40

?&FE36=&80 This causes ROM programs resident in 32K EPROMS27256 devices—to be 'swapped' with the lower 16K, without affecting the RAM. Thus, in order to *SELECT, a small routine such as this could be used:

```
LDA # &0
STA &80
LDA # &80
STA &81
LDY # &0           ® set index register Y
LDA # &E           ® destination RAM bank number
STA &FE32         ® store A in the access register
LDX &F4           ® get the current ROM number
LDA # &9           ® set source ROM number
SEI               ® set interrupt disable flag
STA &FE30         ® store A in the read register
LDA # &80         ® get ready
STA &FE36         ® and swap ROM halves
.L LDA (&80),Y   ® get byte
STA (&80),Y     ® store byte in designated RAM
INC &80          ® 'bump' pointer byte
BNE L           ® carry on if not zero
INC &81          ® if zero 'bump' pointer hi byte
LDA &81
CMP # &C0        ® make sure all ROM transferred
BMI L           ® if not, then go bac kagain
STX &FE30        ® restore old current ROM
LDA # &80
STA &FE36        ® and size register contents
CLI             ® reset interrupts
RTS             ® and return t ocalling routine
```

The preceding routine has got to be resident in RAM between &0000 and &7FFF, as changing the ROM read (or paging) register can otherwise have disastrous results.

We shall conclude this section with a routine which will assemble into an RS423 output buffer (suitable for those with a serial printer). This code is similar to that used in the resident printer buffer program in the MANAGER ROM, but with adaptation for RS423.

```
10REM Declaration for RS423 BUFFER
20-opt=3
30brkv=3
40romno = &F4
50stack3 = &0103
60INSV=&022A
70buftop= &C0
80romtop = &82
90invers= (romtop-buftop AND &FF)
100start= &8000
110romtyp = &8006
120title$ = "STL "+STR$(1+INT((buftop-romtop)/4) AND
&FF)+"K RS423 Output Buffer"
130DESTRAM=&FE32
140OSEVEN=&FFBF
150OSNEWL=&FFE7
160OSWRCH=&FFEE
170OSBYTE=&FFF4
180FOR opt% =0 TO opt STEP opt
190P% = &8000
200[OPT opt%
210.start
220BRK
230BRK
240BRK
250JMP entry
260.romtyp
270EQUB &82
280EQUB copy—&8001
290EQUB &44
300EQU$ title$
310BRK:EQU$"13":BRK
320.copy
330EQU$© Solidisk 16th. August 1986"
340EQUW &D
350.noteme PHA
360LDA romno
370STA DESTRAM
380STA ramhld
390PLA
400RTS
```

410.restrm
420PHA
430LDA ramhld
440STA DESTRAM
450PLA
460RTS
470.oldvec
480EQUW &800 \ OLD VEC SPACE
490.OLREMV EQUW &800
500.OLCNPV EQUW &00
510.newvec EQUW &FF3F \ EXVEC SPACE
520EQUW &FF42
530EQUW &FF45
540.exvec
550EQUW NWINSV
560.insrom
570EQUB 0
580EQUW NWREMV
590.rmvrom EQUB 0
600EQUW NWCNPV
610.cnprom
620EQUB 0
630.bufblo
640EQUB 0
650.bufbhi
660EQUB 0
670.bufvlo
680EQUB 0
690.bufthi
700EQUB 0
710.NWINSV PHP
720CPX #&02
730BEQ prinst
740PLP
750JMP (oldvec)
760.prinst
770SEI
780PHA
790LDX bufblo
800TXA
810LDY bufbhi
820INX
830BNE roomok
840INY
850BNE roomok
860LDX #&00
870LDY #invers
880.roomok
890CPY bufthi
900BNE instok

910GPX bufvlo
920BEQ ninsrt
930.instok
940JSR noteme
950STA iptrlo
960LDA butbui
970CLC
980ADC #buftop
990STA iptrhi
1000STX bufblo
1010STY bufbhi
1020PLA
1030EQUB &8D \ STA &XXXX
1040.iptrlo EQUB &00
1050.iptrhi EQUB &00
1060JSR restrm
1070LD #&02
1080PLP
1090CLC
1100RTS
1110.ninsrt
1120JMP insfld
1130LDY #1
1140.doevnt
1150JSR OSEVEN
1160PLA
1170PLP
1180SEC
1190RTS
1200.NIVREMV
1210PHP
1220CPX #&02
1230BEQ ppremv
1240PLP
1250JMP (OLREMV)
1260.ppremv
1270SEI
1280LDA bufthi
1290TAY
1300LDX bufvlo
1310CMP bufbhi
1320BNE opchar
1330CPX bufblo
1340BNE opchar
1350LDX #&02
1360PLP
1370SEC
1380RTS
1390.opchar
1400JSR noteme

1410STX oprlo
1420CLC
1430ADC #buftop
1440STA oprthi
1450SEI
1460EQUB &AD ® LDA &XXXX
1470.oprlo EQUB &00
1480.oprthi EQUB &00
1490CLI
1500PLP
1510PHP
1520BVS xambuf
1530PHA
1540IN
1550BNE bufrap
1560INY
1570BNE bufrap
1580LDX #&00
1590LDY #invers
1600.bufrap
1610STX buuftlo
1620STY bufthi
1630CPY bufbhi
1640BNE notmpt
1650CPX bufblo
1660BNE notmpt
1670LDA #&oo
1680LDX #&02
1690JSR OSEVEN
1700.notmpt
1710PLY
1720TYA
1730.xambuf
1740JSR
1750LDX #&02
1760PLP
1770CLC
1780RTS
1790.NWCNPV
1800PHP
1810CPX #&02
1820BEQ prcnpv
1830PLP
1840JMU (OLCNPV)
1850.prcnpv
1860JSR noteme
1870PLP
1880PHP
1890PHP
1900SEC

1910LDA
1920SBC bufvlo
1930TAX
1940LDA bufbhi
1950SBC bufthi
1960BCS wrappd
1970SEC
1980TAY
1990TXA
2000SBC #&00
2010TAX
2020TYA
2030SBC #invers
2040.wrappd
2050PLP
2060BBC ctonli
2070CLC
2080TAY
2090TXA
2100ADC #&00
2110EOR #&FF
2120TAX
2130TYA
2140ADC #invers
2150EOR #&FF
2160.ctonli
2170JSR restrm
2180TAY
2190PLP:CLI
2200RTS
2210.ramhld
2220EQUB 0
2230.insfld
2240PLA
2250PHA
2260TAY
2270LDA #&01
2280LDX #&02
2290JMP doevnt
2300.entry
2310PHA
2320PHY
2330PHX
2340TSX
2350LDA stack3,X
2360CMP #&09
2370BEQ help
2380CMP #brkv
2390BNE notbrk
2400JSR noteme

2410LDY romno
2420STY insrom
2430STY rmvrom
2440STY cnprom
2450LDY #&05
2460.veccpy
2470LDA INSV,Y
2480STA oldvec,Y
2490LDA newvec,Y
2500STA INSV,Y
2510DEY
2520BPL veccpy
2530LDA #&A8 @ exvec table
2540LDX #&00
2550LDY #&FF
2560JSR OSBYTE
2570STX &F8
2580STY &F9
2590.vecptr
2600LDY #&47
2610.morvec
2620LDA exvec-&3F,Y
2630STA (&F8),Y
2640DEY
2650CPY #&3E
2660BNE morvec
2670LDA #&00
2680STA bufblo
2690STA bufilo
2700LDA #invers
2710STA bufbhi
2720STA bufthi
2730JSR restrm
2740.help
2750LDY #&09
2760LDA start,Y
2770.titlpt
2780JSR OSWRCH
2790INY
2800LDA start,Y
2810BNE titlpt:JSR OSNEWL:JSR OSMEWL
2820.notbrk
2830PLX
2840PLY
2850PLA
2860RTS
2870]
2880NEXT

OSBYTE CALLS AND THE FOURMEG AND TWO MEGAHERTZ BOARDS

OSBYTE with A=&72 (114)

X determines the action followed

X=0 all MODE changes produce SHADOW RAM handling

X=1 all MODE changes between 0 to 7 remove SHADOW RAM;
all MODE changes between 128 and 135 produce SHADOW RAM

X=FF all MODE changes remove SHADOW RAM

OSBYTE with A=&73 (115)

X determines the SHADOW RAM bank used:

X= 0 SHADOW 0 (SHADOW bank 8)

X=1 SHADOW 1 (SHADOW bank 9)

. . .
. . .

OSBYTE with A=&74 (116) (bank F)

No entry requirement

X returns the ASCII value of "S", signifying that the STL fourmeg or two megahertz board is in place, included for firmware writers such as Computer Concepts, to use the STL SHADOW RAM.

STL and SOLIDISK are registered trade marks and are the property of Solidisk Technology Limited. BBC in this manual refers to the computer produced by Acorn Computer Limited for the British Broadcasting Corporation. Acorn and Computer Concepts are registered trade marks.

SOLIDISK TECHNOLOGY
17 Swayne Avenue, Southend-on-Sea, Essex SS2 6JQ

