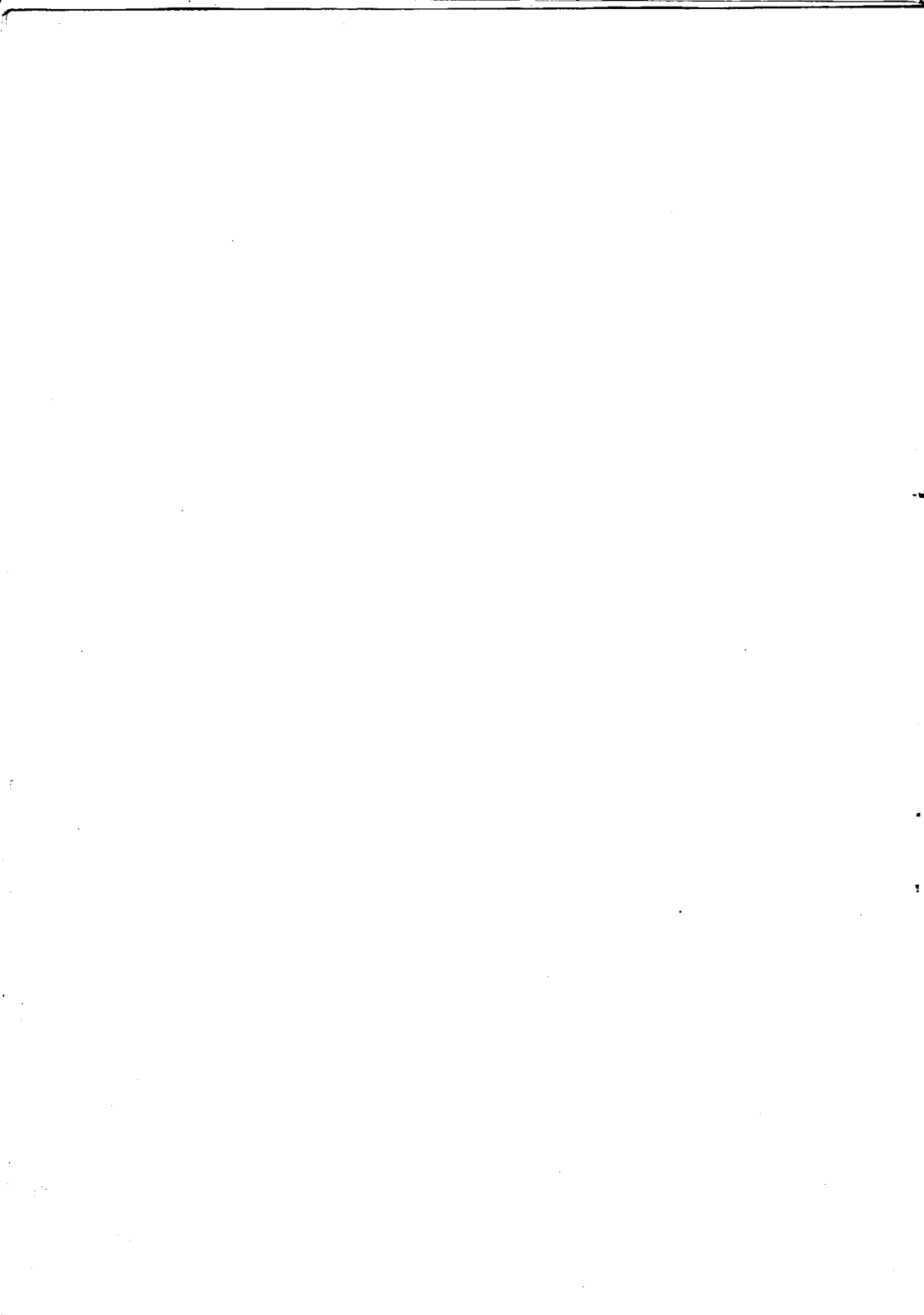


A.B.R.

Advanced Battery-backed RAM

(C) 1987 P.R.E.S. Ltd.



CONTENTS

PRES ABR utilities - instructions.

| | |
|--|----------|
| <u>How to fit the ABR cartridge</u> | <u>2</u> |
| <u>How to use ABR</u> | <u>2</u> |
| <u>Software utilities</u> | <u>2</u> |
| <u>Saving the Programs in ABR</u> | <u>3</u> |
| <u>Saving ROMS onto Tape or Disc</u> | <u>3</u> |
| <u>Loading ROMS into ABR</u> | <u>3</u> |
| <u>Zero RAM</u> | <u>4</u> |
| <u>Lock</u> | <u>4</u> |
| <u>Unlock</u> | <u>4</u> |
| <u>Printer Buffer</u> | <u>4</u> |
| <u>MakeROM</u> | <u>5</u> |
| <u>Using ABR from Machine Code</u> | <u>7</u> |
| <u>Write Protecting ABR</u> | <u>8</u> |
| <u>LOCK and UNLOCK from Machine Code</u> | <u>9</u> |
| <u>The Internal Battery</u> | <u>9</u> |

WARNING

This cartridge contains software. Please read the following instructions carefully before using the cartridge. (See section on 'Saving the programs in A.B.R.' before loading data into cartridge).

How to Fit the ABR Cartridge

The ABR cartridge is designed to fit into either of the two slots in the Electron Plus 1 or Master 128 (For other versions consult the relevant fitting instructions) The cartridge should be inserted with the label facing the keyboard, and under normal circumstances should not be inserted or removed with the power switched on.

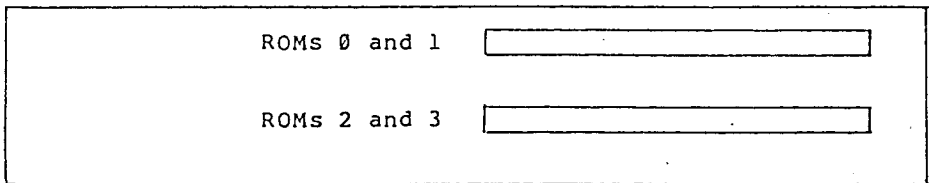
IF YOU REMOVE OR INSTALL ABR WITH THE POWER TURNED ON THE CONTENTS OF THE CARTRIDGE COULD BECOME CORRUPTED.

Master 128

When using ABR in a Master Series computer, the Cartridge Machine Detect/Csync Link (link 12) on the main computer board should be set correctly. This link is located to the left of IC41 above the cartridge slots. The correct position is B (the position Acorn use when shipping) Page F.5-3 in the Master Series Reference Manual Part One contains further information.

How to use ABR

After fitting the cartridge, the RAM in ABR will be mapped into the Computer's memory as two of the 16 sideways ROMs. The ROM numbers of ABR will depend upon which slot in the computer the ABR cartridge has been inserted into. If the ABR cartridge is plugged into the slot nearest the keyboard then ABR will be mapped onto ROM number 2 and 3. If the cartridge is plugged into the other slot then ABR will be mapped onto ROM number 0 and 1. It is important to know which ROM number ABR is in when loading software into the RAM.



Cartridge slots showing the ROM numbers used by ABR

ABR software utilities

The new PRES utilities are supplied in a ROM filing system image loaded in one of the ABR banks when it is purchased. This image contains seven utilities, plus an additional program to transfer the utilities onto tape, DFS or ADFS to free the ABR bank for other uses.

Saving the Programs in ABR

To transfer the programs you should first prepare your media, eg making sure a tape is in the cassette recorder, or that a correctly formatted disc (DFS or ADFS) is in the drive.

When the media is ready type *ABR and press RETURN. This will cause the ROM filing system to be selected and the transfer program to be executed. It will ask you which filing system the programs are to be saved on. Press T, D or A as appropriate.

You should also keep a copy of the whole suite of utilities, as supplied in the ABR. This may be achieved by using SaverOM which is one of the programs which has just been transferred.

If using ADFS it can be a good idea to put some of the utilities into the library directory then they can be used from any directory, assuming the library directory has been set up with the *LIB command.

There are seven utilities supplied with the ABR, SaverOM, LoadROM, Zero, Lock, Unlock, PrntBuf and MakeROM which are described in the following sections.

SaverOM

This program is used to save a RAM/ROM image to the current filing system.

SYNTAX:-

*SaverOM <filename> <bank> (S). (OR from TAPE */SaverOM <etc.>)
The bank number must be included, and must be in hexadecimal.

The program defaults to a fast algorithm which corrupts main memory. If you want to preserve the memory you should type S after the bank number, which uses a slow algorithm leaving memory intact.

LoadROM

This program is used to load a RAM bank with an image stored on the current filing system.

SYNTAX:-

*LoadROM <filename> <bank>. (OR from TAPE */LoadROM <etc.>)
Bank number is optional, but must be in hexadecimal if present.

If no bank number was given the program will test each bank, working down from bank F to find ones which are RAM. If the bank is occupied it will ask if you want to overwrite the image currently loaded into it. If the bank is empty the image will be loaded into it.

If the bank number specified is already in use you will also be asked whether you want the present occupant overwritten.

Zero

This program is used to wipe a RAM bank, i.e. to tell the OS that it is no longer present, and to zero the bank.

NOTE: In some cases CTRL BREAK will be necessary after using this program to completely disengage links to some programs. This applies especially to filing systems.

SYNTAX:-

*Zero <bank> (OR from TAPE */Zero <bank>)

It has one parameter, a bank number in hexadecimal, which must be present.

Lock

The RAM in ABR can be made to behave like the memory in ROM, i.e. the memory can only be read and not written back to. This can be done by write protecting the RAM. This is useful when programs which normally run from ROM try to write back to themselves as a means of software protection.

This program is used to turn on the write-protect facility in the ABR. It's parameters are one or more bank numbers in hexadecimal. If no parameters are given all ABR banks will be locked.

NOTE : Although the program lets you enter any of the sixteen permissible bank numbers the design of the ABR is such that the only thing of consequence is whether it is an odd or even number. The effect of this is that if you have two ABRs fitted the odd banks will always be locked or unlocked together (whichever odd bank number you entered), and the same will apply to the even banks.

SYNTAX:-

*Lock <rom or bank no.> or from TAPE */Lock <rom or bank no.>

NOTE

ABR will normally be locked at switch on:-
although this cannot be guaranteed.

Unlock

This program is used to turn off the write-protect facility in the ABR. It's parameters are one or more bank numbers in hexadecimal. If no parameters are given all ABR banks will be unlocked.

SYNTAX:-

*Unlock <rom or bank no.> or from TAPE */Unlock <etc.>

See the note above, which applies here also.

Printer Buffer

This program is a sideways RAM image, which provides a printer buffer, to supplement the meagre one used normally. Before it can be used it must be loaded into a RAM bank by using LoadROM.

The program responds to *PB, or *PPB to avoid any possible clashes.

There are five commands, which may be entered on the same line as a *PB or *PPB :

- F - Turn off buffer program (only allowed if buffer is empty)
- N - turn on buffer program
- C - Clear buffer's contents (only allowed if on)
- P - Purge buffer if <ESC> is pressed (only allowed if on)
- K - Keep buffer intact if <ESC> is pressed (only allowed if on)

Any number of parameters may be included on the same line, eg *PB FK. If a parameter is not recognised a list of the parameters will be displayed. If an unknown parameter occurs in a multi-parameter line all preceding parameters will be processed.

*PB and *HELP PB, or *HELP PPB produce a buffer status report, and the list of commands. The buffer status report gives you four pieces of information about the program, a) whether it is on or off (the remaining reports will be omitted if the buffer is off), b) the buffer size (14K in 16K image, or 6K in 8K), c) whether the buffer is empty, and d) whether the ESCAPE key will cause the buffer to be cleared.

MakeROM

This program is used to frame one or more files in a Rom Filing System image, for loading from a RAM bank (like the way these utilities were supplied).

The Rom Filing System is activated by *ROM, and behaves like the Tape system, except that only reading is permitted.

This program is a sideways RAM image, and must be loaded into a RAM bank using *LoadROM.

Start the program with *MakeROM or *PMakeROM. Either form of the name may be abbreviated by putting a full stop after the characters.

The program can only be run in the host processor, so Second Processor users should run this program with the Tube off. If a Tube is active the program will refuse to respond to *MakeROM/*PMakeROM.

Once the program has been started it will ask you for a title for the image about to be created. This name will appear on *ROMS as the title of the ROM, and will also appear on *HELP. It can be up to 20 characters long, and should only include alphanumeric characters.

You will then be presented with a list of options

to choose from, which are :

| | |
|---|--|
| T | Select Tape and then load a file |
| D | Select DFS and then load a file |
| A | Select ADFS and then load a file |
| R | Load a file from current filing system |
| * | Operating system command |
| F | Start a fresh image |
| S | Save the current image |
| Q | Quit the program |

Options <T>, <D> and <A> will select the relevant filing system, and then ask for a filename (<R> just asks for the name, keeping the current filing system). The filename must be just a filename, of one to ten characters. It cannot include any kind of directory specification or other special characters.

The file will then be read and converted into ROM filing system format. An error message will be displayed if the file would cause the image to overflow, and the file will not be included.

Once the conversion has been completed or aborted the option list will be displayed again.

<*> is provided for Operating System commands to be entered. Care should be taken that memory is not corrupted, otherwise the image being built may be overwritten. When the OS command has completed press the RETURN key and the option list will be displayed again.

<F> starts the program again with an empty image. The program will ask you to confirm that you want to start afresh by pressing Y. Press ESCAPE if you have second thoughts.

<S> saves the image to the current filing system. You will be asked for a filename.

<Q> leaves the program. You will be asked to confirm that you want to leave by pressing Y, or ESCAPE if you have had second thoughts. When you leave MakeROM you will be returned to BASIC.

The image produced by this program may be loaded into a RAM bank by using *LoadROM.

Normally files in a ROM filing system image cannot be executed with *name but must be run with */name or *RUN name. Images produced by this program WILL permit programs to be run by *name.

It is possible to have a !Boot file in an image which can be started by SHIFT SPACE BREAK. However, this program must be in machine code, and is called before Basic is started. To make the facility more useful a program called !Boot is included in the ABR. This program starts up Basic and then CHAINS a program called BOOT. Thus to set up an auto-boot sequence you should include the !BOOT provided and a Basic program BOOT of your own.

Using ABR from Machine Code

Reading bytes from ABR

The RAM in ABR can be read like any other sideways ROM.

There is a routine in the computer's Operating System which can be used to read a byte from any of the ROMs, including ABR. This routine is called OSRDRM and its address is &FFB9. The following assembly instructions could be used to read a byte from ABR.

```
.program
    LDX #address MOD256
    LDY #address DIV256
    JSR read_ABR           byte read from ABR returned in A
    :
    :
    rest of program

.read_ABR                subroutine to read a byte from ABR
    STX &F6                initialise low byte address
    STY &F7                initialise high byte address
    LDY #abrrom           Y=ROM number of ABR
    JMP OSRDRM           call MOS to read a byte from ABR
```

Writing bytes to ABR

Unlike OSRDRM there is not an Operating System routine that can be used to write bytes to sideways RAM. The following assembly program can be used to write a byte to ABR.

```
.program
    LDX #address MOD256
    LDY #address DIV256
    LDA byte
    JSR write_ABR        write byte in A to address X,Y in ABR
    :
    :
    rest of program

.write_ABR              subroutine to write a byte to ABR
    STX &F6                initialise low byte address
    STY &F7                initialise high byte address
    TAX                  save data in X
    LDA &F4
    PHA                  save current ROM on stack
*   LDA #15
*   STA &F4
*   STA romsel          must first deselect Basic
    LDA #ABR_rom
    STA &F4
```

```

STA romsel      now page in ABR ROM
TXA             restore data in A
LDY #0
STA (&F6),Y    store byte to ABR
PLA            get old paged ROM
STA &F4
STA romsel      select current ROM
RTS

```

* Electron only.

romsel - Electron = &FE05, Master and Compact = &FE30

Note

Writing to ABR can only be done after the RAM has been unlocked. See Write Protecting.

Write Protecting ABR

As mentioned previously the RAM in ABR can be made to behave like the memory in ROM. Two memory addresses are used to lock the RAM and two addresses are used to unlock the RAM.

| | | | | | |
|--------|---|-------|---|-------|----------------------|
| ROM | - | 0 | - | 1 | <input type="text"/> |
| LOCK | - | &FCDD | - | &FCDF | |
| UNLOCK | - | &FCDC | - | &FCDE | |
| ROM | - | 2 | - | 3 | <input type="text"/> |

Lock and unlock addresses for different ROM sockets

FOR EXAMPLE

ELECTRON, to lock the RAM in ROM 1 or 3 write to addresses &FCDD or &FCDF eg, ?&FCDD=0 : ?&FCDF=0 or the utility LOCK or UNLOCK can be used.

ABR in the MASTER and MASTER COMPACT can be locked or unlocked from the keyboard, but requires the following entries before and after:-

S%=?&FE34:?&FE34=?&FE34 OR &20: (un/lock ie.&?&FCDF=0) :?&FE34=S%

The reason for this is the ACCON register in the Master series and can be seen more clearly in the routine in the next section

LOCK and UNLOCK from Machine Code

This routine can be used in your own machine code programs.

.program

```
*   LDA &FE34           read ACCCON register
*   PHA                 save
*   ORA #&20            set bit 5 for cartridge access
*   STA &FE34           save to ACCCON register
*   STA &@@@            lock/unlock RAM
*   PLA                 recover
*   STA &FE34           restore ACCCON register
:
:
```

rest of program.

* Master and Master Compact only.

where @@@@ is the address to either lock or unlock the ram, for example, STA &FCDC to unlock ram 0 or 2.

Internal Battery

The ABR cartridge is fitted with a nickel-cadmium rechargeable battery which is recharged when ABR is used in the computer.

Battery Life

If the ABR cartridge is used regularly, for example, at least 2 hours a week, then the life of the battery will be greater than 5 years. If used less than 2 hours a week, recharging should be carried out at intervals of not less than 3 months. A full recharge period takes 28 hours.

IMPORTANT

If your ABR cartridge is to be used continually for periods greater than 24 hours CONTACT P.R.E.S. Ltd. FOR ADVICE

NOTE

THE BATTERY IS NOT USER REPLACEABLE

P.R.E.S. Ltd., 6 Ava House, High St., Chobham, Surrey. 0276 72046

abrdoc_1.1

