A.S.R.
Advanced Sideways RAM Instructions

(C) Copyright Advanced Computer Products Limited 1986

How to fit ASR

The ASR cartridge can be plugged into either one of the two slots in the Plus 1. The cartridge should be inserted with the label facing the keyboard, and should not be installed or removed with the Electron switched on. If you remove or install ASR with the power turned on, you will need to press **BREAK** or **CTRL BREAK** to restore normal operation.

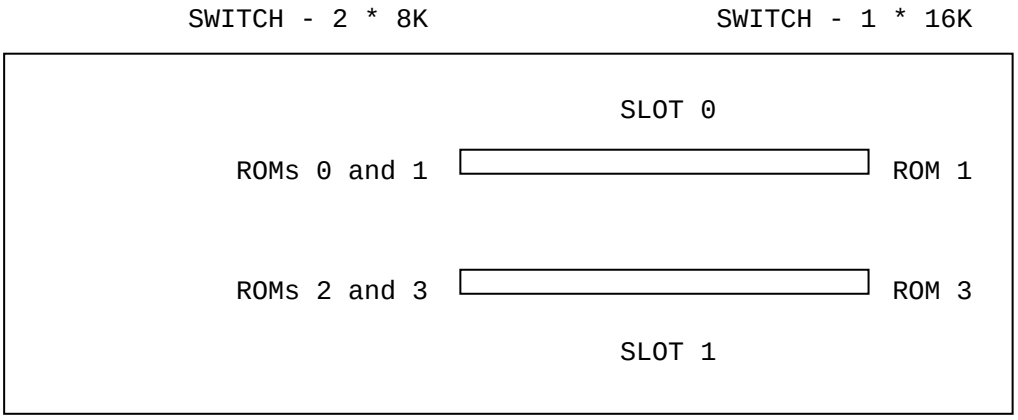
For more information on using cartridges refer to pages 4 and 16 of the Plus 1 User Guide.

How to use ASR

After fitting the cartridge, the RAM in ASR will be mapped into the Electron's memory as one or two of the 16 sideways ROMs. For example, the BASIC ROM occupies ROMs 10 and 11, and the Plus 1 software occupies ROM 12. The ROM number(s) of ASR will depend upon which slot in the Plus 1 the ASR cartridge has been inserted into and the position of the switch on the front of the cartridge.

If the ASR cartridge is pulled into the Plus 1 slot nearest the keyboard then ASR will be mapped onto ROM number 2 and 3. If the cartridge is plugged into the other slot then ASR will be mapped onto ROM number 0 and 1. It is important to know which ROM number ASR is in when loading software into the RAM.

The switch on the front of the cartridge enables the RAM to be configured as one 16K block occupying one ROM socket (1 or 3), or to 8K blocks occupying two ROM sockets (0 and 1 or 2 and 3). The 16K block can be used to run one 16K program like View and Viewsheet, while the two 8K blocks can be used to run two 8K programs like Starmon or Printmaster.



Plus 1 showing the ROM numbers used by ASR for both cartridge slots

Loading programs into ASR

Supplied on the cassette is a program called ASR. This program can be used to load programs such as View or Viewsheet from tape or disc into the ASR cartridge.

To run ASR from tape, type:

*/ASR <fsp> (<rom>) (L/U)

where <fsp> is the name of the program on tape to be loaded into ASR and <rom> is the ROM number containing ASR. If <rom> is omitted, the first ROM found containing RAM will be used. The last parameter decides whether the RAM is left Locked or Unlocked after the file has been loaded into ASR. If the program <fsp> does not follow ASR on the tape then insert the right cassette with the program on it after ASR has been loaded.

To run ASR from disc, type:

*ASR <fsp> (<rom>) (L/U)

The parameters have the same meaning as above. If the program ASR is put into the library directory then a *ASR command can be used from any directory, assuming the library directory has been set up with the *LIB command.

If a ROM number is used which does not contain RAM then the error 'RAM not found' is displayed. If the switch on the cartridge is set to 2*8K and the program to be loaded is more than 8K then the error 'Wrong size' is displayed.

Saving ROMs onto Tape or Disc

Supplied on the cassette is a program called ROMSAVE. This program can be used to save programs in ROM onto Tape or Disc.

To run ROMSAVE from tape type:

*/ROMSAVE <fsp> <rom>

where <fsp> is the name to give the program on the tape and <rom> is the ROM number to be saved. If you do not want to save the program immediately after ROMSAVE on the tape then insert another cassette after ROMSAVE has been loaded.

To run ROMSAVE from disc type:

*ROMSAVE <fsp> <rom>

The parameters have the same meaning as above. If ROMSAVE is put into the library directory then a *ROMSAVE command can be used from any directory, assuming the library directory has been set up with the *LIB command.

The Advanced Printer Buffer

The Advanced Printer Buffer APB, is a program which runs in ASR and uses the RAM as a large printer buffer, instead of the 63 byte buffer normally used by the computer.

If ASR is configured as two 8K ROMs then 6K will be available as a printer buffer. If ASR is configured as one 16K ROM then 14K will be available as a printer buffer.

To load APB into ASR from tape type:

*/ASR APB <rom>

where <rom> is the ROM number containing ASR.

Type *HELP APB to list the commands in APB.

*APB ON/OFF

This command is used to turn the Advanced Printer Buffer on or off. If APB is off then the computer's own 63 byte buffer is used.

*ESCAPE ON/OFF

This command determines whether the buffer is cleared after pressing ESCAPE, or left intact. Normally pressing ESCAPE will clear the buffer. Typing *ESCAPE OFF will not clear the buffer if ESCAPE is pressed. This is useful with programs like View, so that ESCAPE can be used while the printer is still printing.

*PURGE

This command can be used to clear the contents of the buffer.

Note

To find the number of characters free in the printer buffer from BASIC type:

PRINT ADVAL(-4)

MakeROM

This is a program supplied on the cassette which can be used to create ROM files of your own programs, so that they can be used in the ROM filing system. The ROM filing system is very similar to the TAPE filing system except programs can only be read from ROM and not written back. The advantage is that programs can be loaded much faster from ROM than TAPE.

For example, instead of typing:

```
*TAPE
CHAIN "program"
```

you could type:

```
*ROM
CHAIN "program"
```

and the program would be loaded into memory from ROM much faster.

To run MakeROM type:

```
CHAIN "MakeROM"
```

Instructions are given in the program.

Transferring Tape Programs to Disc

To transfer the programs supplied on cassette to disc, load the programs into memory and save them back onto disc one at a time, making sure the programs addresses are the same when saving the program. For example, to transfer one program to disc, type:

*TAPE	Selects the Tape filing system
*OPT 1,2	Displays the programs addresses after loading
*LOAD "" 1D00	Loads the next program on the tape to &1D00 and prints its file information as: <fsp> <llll> <bbbb> <eeee>.
*ADFS	Selects the ADFS filing system
*SAVE <fsp> 1D00 + <bbbb> <eeee> <llll>	Saves the program onto the disc

where <fsp> is the file's name, <bbbb> is the file's length, <eeee> is the file's execution address and <llll> is the file's load address.

Alternatively, if you have the ADT ROM simply type:

```
*XFER TAPE ADFS
```

This will take care of all the file's addresses automatically.

Using ASR from Machine Code

Reading bytes from ASR

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

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

```
.program

    LDX #address MOD256
    LDY #address DIV256
    JSR read_ASR                byte read from ASR returned in A
    :
    :
    rest of program

.read_ASR   subroutine to read a byte from ASR

    STX &F6                     initialise low byte address
    STY &F7                     initialise high byte address
    LDY #asrrom                 Y=ROM number of ASR
    JMP OSRDRM                  call MOS to read a byte from ASR
```

Writing bytes to ASR

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 ASR.

```
.program

    LDX #address MOD256
    LDY #address DIV256
    LDA byte
    JSR write_ASR     write byte in A to address X,Y in ASR
    :
    :
    rest of program

.write_ASR   subroutine to write a byte to ASR

    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 &FF05                   must first deselect BASIC
    LDA #ASR_rom
    STA &F4
    STA &FE05                   now page in ASR ROM
    TXA                         restore data in A
    LDY #0
    STA (&F6),Y                 store byte to ASR
    PLA                         get old paged ROM
```

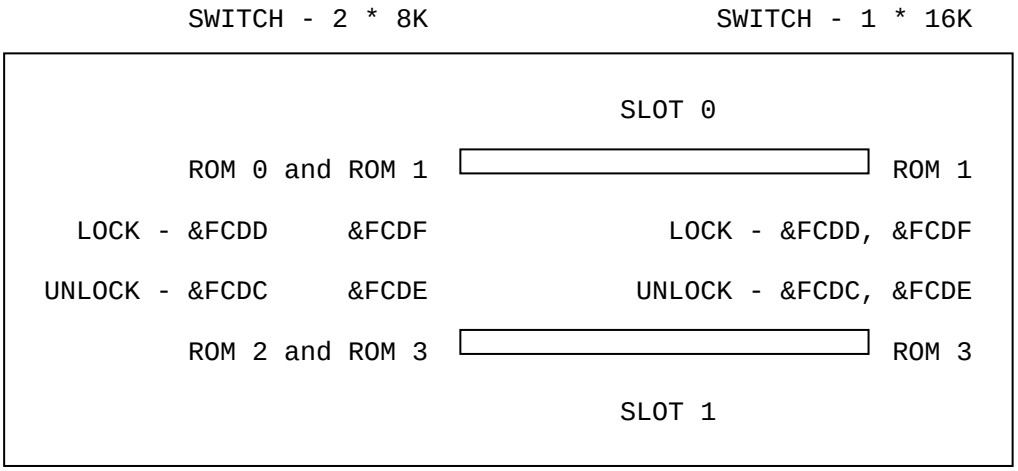
```
STA &F4
STA &FE05                    select current ROM
RTS
```

Note

Writing to ASR can only be done after the RAM has been unlocked. See below.

Write Protecting ASR

The RAM in ASR 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. Two addresses in memory are used to lock the RAM and two addresses are used to unlock the RAM.



Plus 1 showing lock and unlock addresses for different ROM sockets

For example, to lock the RAM which is configured as 1*16K ROM in ROM 1 or 3, write to addresses &FCDD and &FCDF e.g.

```
?&FCDD=0 : ?&FCDF=0
```

To lock the RAM which is configured as 2*8K ROMs in ROM 0 or ROM 2 write to address &FCDD e.g.

```
?&FCDD=0
```