

---

**MORLEY**  
ELECTRONICS LTD

**SCSI Interface**

---

## © Morley Electronics Limited 1992

Morley House, West Chirton, North Shields, Tyne & Wear, NE29 7TY  
Tel 091 257 6355, Fax 091 257 6373

**N.B. Technical support is available from 4:00 pm to 5:30 pm, Monday to Friday.** Please make sure you have read the appropriate parts of the manual before seeking assistance.

Product data and specifications contained in this document are subject to change without prior notice.

Morley Electronics Ltd do not assume any responsibility for damage or loss to the customer due to the use of, or the inability to use, this product.

The information supplied in this document is believed to be accurate and reliable. However, no responsibility is assumed for inaccuracies that may not have been detected prior to printing and for those which occur beyond our control.

This document may not, in whole or part, be copied, photocopied, reproduced, translated or converted to any machine readable form without prior written consent of Morley Electronics Ltd.

Acorn and Archimedes are registered trademarks of Acorn Computers Ltd.

---

# **Contents**

## **Chapter 1**

<b>Hardware Installation .....</b>	1-1
Introduction .....	1-1
ANSI SCSI standard.....	1-1
<b>A300/400/540 Installation.....</b>	1-2
Warning.....	1-2
Installation .....	1-2
Backplanes .....	1-3
Installing your SCSI interface.....	1-3
Fitting An Internal SCSI Device.....	1-5
<b>A3000 Installation.....</b>	1-9
Warning.....	1-9
<b>A5000 Installation.....</b>	1-11
Warning.....	1-11
Installation .....	1-11
Fitting An Internal SCSI Mechanism.....	1-12
Installing your SCSI interface card .....	1-15
<b>Termination - Multiple Initiators &amp; Targets .....</b>	1-18
<b>Cable Requirements.....</b>	1-19

---

## **Chapter 2**

<b>Software Installation .....</b>	<b>2-1</b>
Updates.....	2-1
Configuring SCSI Devices.....	2-1
Transfer Speed Test .....	2-1
SCSI Device Formatter.....	2-2
<b>Advanced Configuration Options.....</b>	<b>2-4</b>
<b>Speeding up your system.....</b>	<b>2-6</b>
<b>Networking Capabilities.....</b>	<b>2-7</b>
<b>Acorn PC Emulator.....</b>	<b>2-8</b>

## **Chapter 3**

<b>SCSI SWI Calls.....</b>	<b>3-1</b>
SCSI Version (&403C0) .....	3-3
SCSI_Initialise 0 (&403C1).....	3-5
SCSI_Initialise 1 (&403C1).....	3-6
SCSI_Initialise 2 (&403C1).....	3-7
SCSI_Initialise 3 (&403C1).....	3-9
SCSI_Control 0 (&403C2).....	3-10
SCSI_Control 1 (&403C2).....	3-11
SCSI_Control 2 (&403C2).....	3-12
SCSI_Control 3 (&403C2).....	3-13
SCSI_Control 4 (&403C2).....	3-14
SCSI Control 5 (&403C2) .....	3-15

---

SCSI_Control 6 (&403C2).....	3-17
SCSI_Control 7 (&403C2).....	3-19
SCSI_Op (&403C3).....	3-20
SCSI_Status 0 (&403C4).....	3-22
<b>SCSI Reserve.....</b>	<b>3-23</b>
SCSI_Reserve 0 (&403C7).....	3-24
SCSI_Reserve 1 (&403C7).....	3-25
SCSI_Reserve 2 (&403C7).....	3-26
SCSI_List (&403C8).....	3-27
SCSI_Target 0 (&403C9).....	3-28
SCSI_Target 1 (&403C9).....	3-29
<b>SCSIFS SWI calls.....</b>	<b>3-30</b>
SCSIFS_TestReady (&40986) .....	3-31
<b>SWI Error Messages.....</b>	<b>3-32</b>
* Commands.....	3-35
<b>Chapter 4</b>	
<b>SCSI Explained .....</b>	<b>4-1</b>
Introduction .....	4-1
SCSI Signals.....	4-2
SCSI Bus Phases.....	4-3
Bus Free Phase .....	4-4
Arbitration Phase.....	4-4
Selection Phase.....	4-4
Reselection Phase.....	4-5

---

Information Transfer Phases.....	4-5
Command Phase.....	4-6
Data Phase.....	4-6
Data In Phase .....	4-7
Data Out Phase.....	4-7
Status Phase.....	4-7
Message In Phase .....	4-7
Message Out Phase.....	4-7
Messages.....	4-8
Phase Sequences With Arbitration.....	4-10
SCSI Pointers.....	4-11
Asynchronous Data Transfer.....	4-11
Synchronous Data Transfer.....	4-12
Command Descriptor Block (CDB).....	4-13
Operation Code.....	4-15
Logical Unit Number (LUN).....	4-16
Logical Block Address.....	4-16
Relative Address Bit.....	4-16
Transfer Length.....	4-17
Control Byte.....	4-17
Command Example Including Disconnect.....	4-18

---

## **Appendix A**

<b>ANSI Statement of Conformance.....</b>	A-1
ANSI SCSI standard.....	A-1
ANSI Conformance Level.....	A-1
SCSI interface.....	A-1
Messages.....	A-2
Extended Messages.....	A-2

<b>A p p e n d i x   B   S p e c i f i c a t i o n s .....</b>	B-1
Hardware.....	B-1
Software.....	B-2

## **Appendix C**

<b>Jumper Settings .....</b>	C-1
Standard SCSI Card.....	C-1
Cached SCSI Card.....	C-2

# **Chapter 1**

# **Hardware Installation**

## **Introduction**

The first two chapters of this manual give details of both hardware and software installation for the Morley 16 bit SCSI interface and the Morley Cached 16 bit SCSI interface. Differences in operation between these two cards are hardware orientated and is not apparent in operation.

**You are advised to read chapters one and two before attempting to unpack your SCSI interface to avoid causing unnecessary damage to either the interface or your computer.**

## **ANSI SCSI standard**

The Morley SCSI interface interfaces conform to the ANSI Small Computer System Interface (SCSI): Document Number ANSI X3.131-1986 (X3T9/84-40 Rev. 1B) (X3T9.2/82-2 Rev. 17B).

# A300/400/540 Installation

## Warning

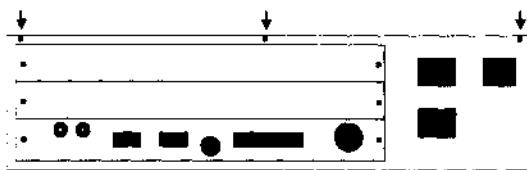
**Before installing your SCSI interface ensure that your machine is both switched off and unplugged from the mains to avoid risk of electric shock.**

Owners of Archimedes A305, A310 and A440 (not the newer 440/1) should note that this interface will not function at full speed without a MEMC1a installed, this is due to a bug in the MEMC1.

## Installation

Remove the five screws that hold the lid in place as shown (Figure 1.1).

**Figure 1.1**



Three on the rear panel



one either side of the front fascia

---

Then slide the lid towards rear of the machine. If the lid is stiff pull the lid alternatively at either side until the lid becomes free.

## **Backplanes**

A305 and A310 owners should ensure that the machine is fitted with both a backplane and a fan. If the backplane you have installed is not multi-layer this could result in unreliable data transfer on the SCSI bus. If you are in any doubt contact your supplier for further details.

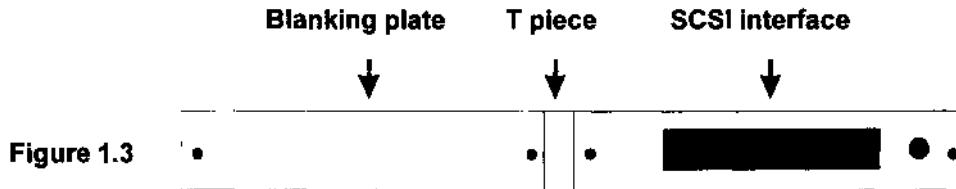
## **Installing your SCSI interface**

If you are fitting an internal SCSI disc drive it is easiest if your SCSI interface is located in the upper most interface slot beside the power supply. If no interface cards are fitted remove the uppermost blanking plate by removing the two fixing screws (Figure 1.2).

**Figure 1.2**



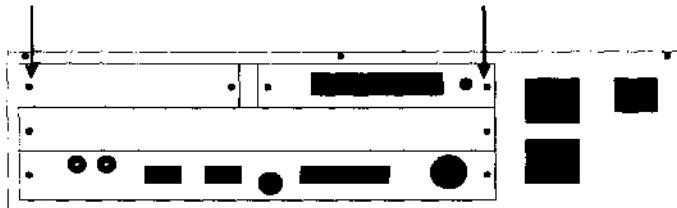
Once the blanking plate is removed it is no longer required and should be stored in a safe place. If you are not installing another interface card next to the SCSI interface then take the half slot blanking plate supplied in the kit and using the T-piece supplied screw the SCSI interface and the half slot spacer together as shown (Figure 1.3)



**Figure 1.3**

Taking the full assembly of interface card, T piece and bracket, install in your machine, making sure that the 64 way interface connector meets with the backplane correctly, and secure the assembly by replacing the two screws shown (Figure 1.4).

**Figure 1.4**



If you are not installing an internal disc drive at this point refit the lid to the machine and replace screws shown (Figure 1.1) .

## Fitting An Internal SCSI Device

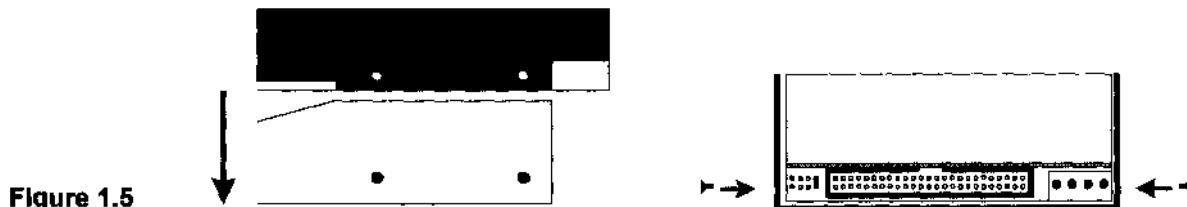
When handling the hard disc drive use extreme caution as hard discs are fragile items and easily damaged. Remove the disc drive from the packaging, but before installing the disc drive into the cradle take the following into consideration:

1. When installing an internal SCSI device it must reside at the end of the SCSI bus and must be terminated. Therefore no SCSI bus may have more than two machines fitted with internal SCSI device(s) that are terminated.

- 
2. Set SCSI id on the device.\*
  3. Ensure that parity is enabled on the device.\*

\* Refer to documentation supplied with device if not supplied by Morley Electronics Ltd.

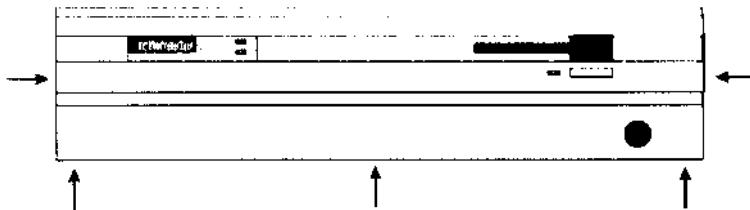
After setting the required device configuration fit the disc unit into the cradle using the four screws supplied (Figure 1.5).



**Figure 1.5**

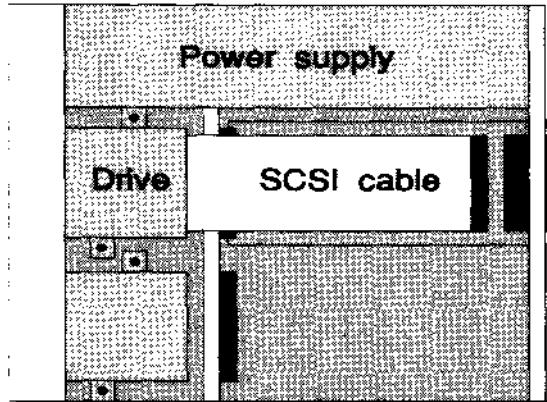
Remove the front facia from the machine by removing the five retaining screws show (Figure 1.6).

**Figure 1.6**



Using the two self tapping screws in the kit secure the cradle to the machine (Figure 1.7).

**Figure 1.7**



Taking a 50 way SCSI cable (provided, if card is purchased with hard disc from Morley Electronics Ltd), plug one IDC connector into the internal SCSI header on the SCSI interface and plug the other IDC connector into the rear of the disc drive. Now plug the cable from the power supply unit into the power socket on the disc drive. This is a keyed four way connector. Replace the front facia to the machine by replacing the five screws shown (Figure 1.6), Replace lid and refit screws shown (Figure 1.1).

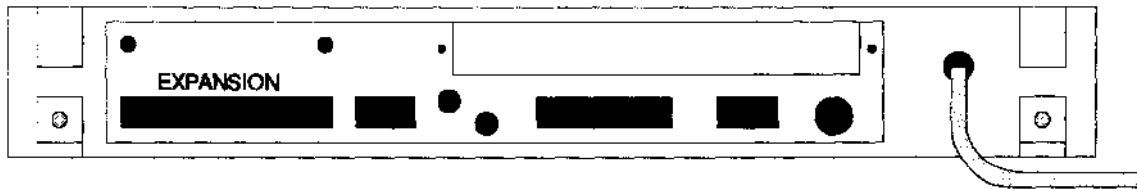
# A3000 Installation

## Warning

**Before installing your SCSI interface ensure that your machine is both switched off and unplugged from the mains to avoid risk of electric shock.**

Take the external interface case from the box and locate the tongue of the interface box into the groove on the underside of the machine below the expansion slot (Figure 1.8).

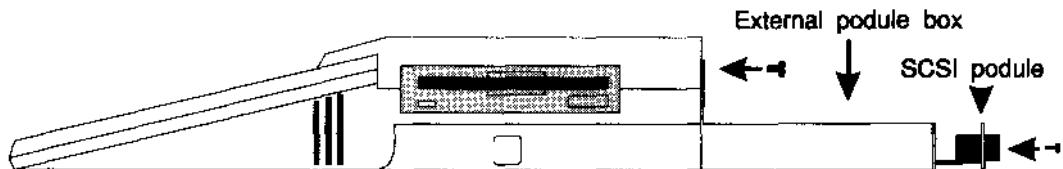
**Figure 1.8**



---

Push the interface box home so the front flange meets the backplate of the A3000. Using the two large screws provided firmly screw the flange of the interface box to the rear of the A3000. After securing the interface box insert the SCSI interface into the interface box and secure with the two screws provided (Figure 1.9).

**Figure 1.9**



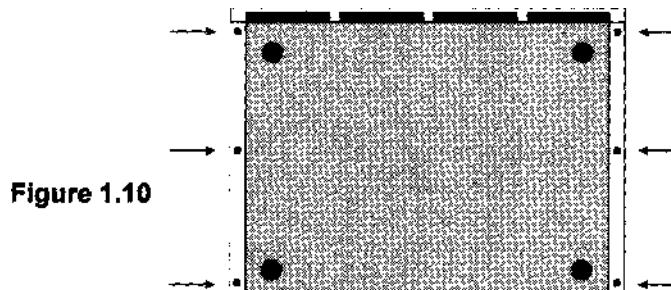
# A5000 Installation

## Warning

Before installing your SCSI Interface ensure that your machine is both switched off and unplugged from the mains to avoid risk of electric shock.

## Installation

Remove monitor and disconnect keyboard and all cables. Remove the six cover fixing screws on the underside of the A5000 (Figure 1.10).



**Figure 1.10**

---

Slide the top cover towards the rear of the machine and remove.

Some models of the A5000 have an EMC case shield (a metal plate) which fits over the case. To remove this, unscrew the five fixing screws on the righthand side of the computer, as seen from the rear. The shield has interleaved tabs which fit into the edges of the case. Slide the shield off the computer carefully; there is a hole on the top to help you slide the shield off by inserting a screwdriver.

## Fitting An Internal SCSI Mechanism

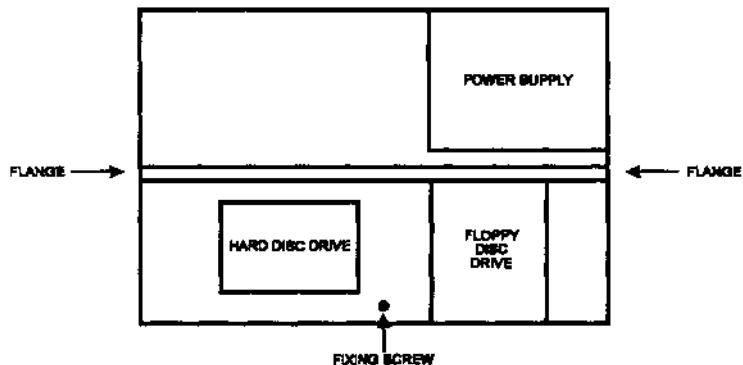
If you have any doubts about performing this part of the upgrade, please have it done by an authorised Acorn dealer as any damage caused will not be covered by your warranty.

When handling a hard disc drive use extreme caution as hard discs are fragile items and easily damaged. Remove the disc drive from the packaging, but before installing the disc drive take the following into consideration:

1. When installing an internal SCSI device it must reside at the end of the SCSI bus and must be terminated. Therefore no SCSI bus may have more than two machines fitted with internal SCSI device(s) that are terminated.
2. Set SCSI id on the device.\*
3. Ensure that parity is enabled on the device.\* \* Refer to documentation supplied with device if not supplied by Morley Electronics Ltd.

---

Remove any internal interfaces fitted to the backplane. Locate and remove the fixing screw from the drive chassis and backplane as shown in figure 1.11.



**Figure 1.11**

Unplug the power and data cables from both the floppy disc drive and the hard disc drive (if fitted).

Lift the chassis and backplane out of the two metal flanges at either side of the A5000 (figure 1.11) and carefully remove the complete backplane and chassis. No undue force should be necessary.

Attach the power splitter cable (supplied, if SCSI mechanism and SCSI interface purchased together from Morley Electronics Ltd) to the four way power cable removed from the internal hard disc drive.

---

To fit the SCSI mechanism beneath the existing floppy disc drive follow steps 1 to 3. To replace the current IDE hard disc follow steps 4 to 6.

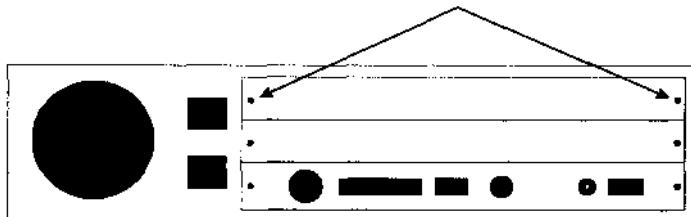
1. With the four M3 screws provided, securely fit the SCSI mechanism into the bottom of the A5000 (the mounting holes are already drilled) with the 50way SCSI connector towards the power supply of the A5000. On early A5000s it may be necessary to make the holes slightly larger with something like a needle file. This is not a problem on later A5000s as the holes are the correct size.
2. Connect the 50way SCSI cable to the rear of the SCSI mechanism. One of the plugs from the 4 way power splitter cable should be attached to the corresponding socket on the SCSI mechanism. (If your SCSI mechanism, such as a Floptical drive, has the smaller type of power connector, use the second plug on the floppy disc drive power connector). The remaining power plug should be connected to the IDE hard disc drive when it has been replaced in the A5000.
3. Insert the backplane and chassis making sure that it lines up correctly with the two flanges and the plug on the main circuit board and that the floppy disc drive passes through the cutout in the front metal panel. Once again, no undue force should be necessary. Reconnect the data and power cables to the floppy disc drive and the IDE hard disc drive. Replace the chassis screw (figure 1.11). Now proceed to the next section, **Installing your SCSI interface card**.
4. Remove the four screws that hold the IDE hard disc to the chassis and store the hard disc in a safe place. Remove the small data cable that was connected to the IDE hard disc from the main circuit board and store that with the IDE hard disc.

- 
5. With the four screws that you have removed from the IDE Harddisc position and fit the SCSI mechanism in the same position on the chassis with the power and data connectors towards the floppy disc drive. Connect the 50way SCSI cable (supplied if mechanism and interface purchased together from Morley Electronics Ltd) to the SCSI mechanism.
  6. Insert the backplane and chassis making sure that it lines up correctly with the two flanges and the plug on the main circuit board and that the floppy disc drive passes through the cutout in the front metal panel. Once again, no undue force should be necessary. Reconnect the data and power cables to the floppy disc drive and the power cable to the new SCSI mechanism. Replace the chassis screw (figure 1.11).

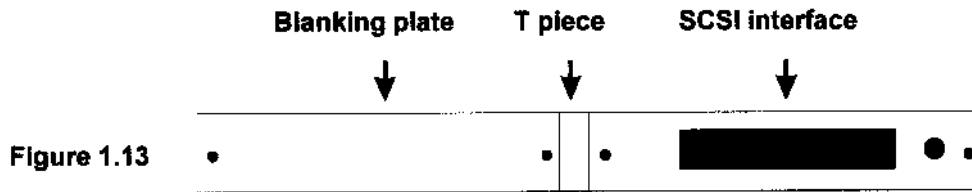
## Installing your SCSI interface card

If you are fitting an internal SCSI disc drive it is easiest if your SCSI interface is located in one of the uppermost interface slots. If no interface cards are fitted remove the uppermost blanking plate by removing the two fixing screws (Figure 1.12).

**Figure 1.12**



Once the blanking plate is removed it is no longer required and should be stored in a safe place. If you are not installing another interface card next to the SCSI interface then take the half slot blanking plate supplied in the kit and using the T-piece supplied screw the SCSI interface and the half slot spacer together as shown (Figure 1.13).

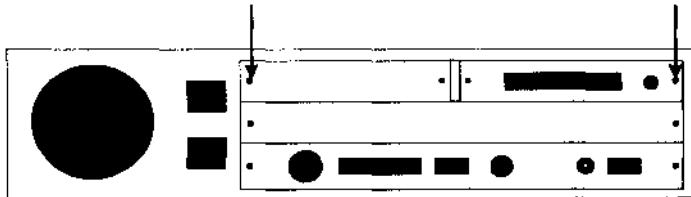


**Figure 1.13**

Taking the full assembly of interface card, T piece and bracket, install in your machine, making sure that the 64 way interface connector meets with the backplane correctly, and secure the assembly by replacing the two screws shown (Figure 1.14).

---

**Figure 1.14**



If you have installed an internal SCSI mechanism, connect the SCSI cable from the mechanism to the socket on the interface card, making sure you have the plug the correct way round. Refit any internal interface cards you may have removed.

Refit the EMC case shield if this was originally present and replace the five screws.

Refit the lid to the machine and replace screws shown (Figure 1.10).

---

## **Termination - Multiple Initiators & Targets**

Your SCSI interface may be daisy-chained together with other SCSI devices using a common cable. The SCSI devices that reside at either end of the daisy-chain must be terminated, the devices which reside between the two terminated devices must not be terminated as this may cause damage to the SCSI interface and or the devices on the SCSI bus. Your SCSI interface may be daisy-chained only with other SCSI devices with single ended drivers. A maximum of 8 devices may reside on the SCSI bus at any one time. SCSI terminating resistors can be found on the SCSI interface behind the external SCSI bus connector, these three resistor packs are marked RPACK 1 TO 3. The SCSI firmware will, under most circumstances, warn you if the SCSI bus is not terminated correctly during its power up diagnostics. If the SCSI bus is terminated using a free-standing terminator that takes power from the SCSI bus, this power can be provided from the SCSI interface. Terminator power supplied from the SCSI bus is current limited via a fuse and diode.

---

## Cable Requirements

If you are installing an internal disc drive it is recommended that you use a 50 conductor flat cable or 25 twisted pair cable with an impedance characteristic of 100 ohms +1-10 %. For external disc drives, only shielded twisted pair cable should be used. When choosing a cable be sure that the cable contains 50 conductors as some cables claiming to be 'SCSI' contain only one earth conductor (poor quality cables can lead to transfer problems). You should also avoid using cables of different impedances on the same SCSI bus as this can cause signal reflection problems.

---

# **Chapter 2**

# **Software Installation**

## **Updates**

Any updates to the software since the printing of this manual are detailed in the file 'ReadMe' located on the SCSI Support Disc.

## **Configuring SCSI Devices**

Once you have completed your hardware installation, turn on the computer and any external SCSI devices and allow the devices to initialise (hard discs typically take about 30 seconds). Insert the SCSI utilities disc into the floppy drive and double click on the application called ISCSI\_Conf. This will automatically assign SCSI device links by assigning drive numbers to devices and store them in the machines CMOS RAM. It will also setup other configurable SCSI parameters which are stored in CMOS RAM, searching for drives/partitions, and allocating up to eight drives or drive partitions to the logical SCSIFS drive numbers 0-7. Once this program has been run, a hard reset is necessary. This is done by pressing <ctrl><reset> simultaneously to restart the system. Your machine should now be ready for use.

## **Transfer Speed Test**

The application on the utilities disc called !SCSI\_Spd' will save a file once or more times to the disc on which it resides. To test the speed of a SCSI device, copy the application '!SCSI\_Spd' to the SCSI device and then double click on the icon to run it. The results for read and write operations are shown, plus an overall average.

---

This could be used to compare the Morley SCSI system with other 3rd party systems. Please take into consideration that the slowest part of the system is generally the drive itself. A standard 20/40 MB drive normally gives about 500-600 KB/s, whereas some high speed drives can give speeds in excess of 4 MB/s.

## SCSI Device Formatter

NB. Morley Electronics supplies drives that are pre-formatted and initialised. The SCSI formatter application supplied on the utilities disc is called '! SCSI\_Form'. This utility should be used with great caution, as ***once a device is formatted all data on the formatted device will be permanently destroyed.***

To run the 'SCSI\_Form' application insert the utilities disc into the floppy drive and double click on the icon. When the formatter is first run, it will show a list of currently configured drives and their device names. Using the mouse, click the select (left) button over the drive you wish to format. The current size of the drive will now be shown, as well as the block size. The block size is generally set 512 bytes, as most hard disc drives give their maximum capacity with a logical block size of 512 bytes. If a block size other than 512 bytes is shown, we recommend that 512 bytes is selected. After selecting the logical block size, click on 'Continue'. If you have selected a block size other than the one displayed for the existing size, then a low level format is automatically implied, if no change has been made to the logical block size you will be prompted whether or not you require a low level format. If 'Yes' is selected, you are prompted to continue with the format. The interleave should be set at '1' as the SCSI podule can sustain a 1:1 interleave on most hard disc drives. If left at '0' the drives default interleave is used.

The low level format procedure may take some time. Once the low level format is complete, or when no low level format was requested, the partition configuration is given. If the disc already has legal partition information

---

on it, then it is shown and can be modified. Normally the first partition gains all the disc space up to 512 MB, with any extra space allocated to the next partition etc. Four partitions are allowed. To change the partition size, click on the partition required, and change the number to the size required (in Kilobytes). If this makes the partition smaller, the remainder is placed in the 'spare' box, or if the partition is made larger the extra space is taken from the 'spare' box. Once the partitions are satisfactory, click on 'initialise disc'. This will allow the user to enter or delete any defects. This should not normally be necessary, but if it is, click over the white defect box and enter the disc address of the defect in hex and then click on 'Add defect'. If a defect is to be removed, the same procedure is used, but click on the 'Remove Defect' instead. The allocation unit shown may be altered, but is not recommended, as the best value is shown by default. This will alter the way filecore allocates disc space to files. The larger it is, the faster filecore will operate but it will be less efficient on disc management. Click on 'Configure Partition' to acknowledge that the partition is okay. Once all partitions are acknowledged, a message will appear informing you the that formatting has been completed. The drive may now be used, although it may be wise to run '!SCSI\_Conf', as partition information may have changed during the format.

# Advanced Configuration Options

To configure the SCSI system, a utility called '!SCSI\_Conf' is supplied on the utilities disc. If further customising is required, then this section should be read.

## \*Configure SCSIFSLink <drv> <id>

This sets up a logical path between a FileCore logical drive (0-7) and a physical SCSI drive ID. The drive (drv) number is a value between 0 and 7 inclusive, and the SCSI ID number is a value between 0 and 32 inclusive. The IDs 0 to 30 represent a device on one of the 4 possible SCSI buses, the bottom 3 bits representing the drive ID and the next 2 bits for the bus ID. Any value above 30 will render that logical drive with no SCSI device, or in effect that drive will not exist to FileCore. N.B. SCSIFS will not recognise a drive ID of 7 on the last bus (ID=31) due to a CMOS RAM restriction.

## \*Configure SCSIFSDirCache <size>

This sets up a directory cache size for FileCore. When FileCore reads a directory, it is cached in memory for speed. The size specified is in Kilobytes and is rounded down to the nearest 8 Kilobytes. The default value is 16 Kilobytes with a maximum of 56 Kilobytes.

## \*Configure SCSIFSBuffers <number>

This sets up a cache in memory for background data transfer. The number given sets up x times 1072 bytes of memory (1024 bytes plus workspace) and must be set to zero at present. It should be noted that future Morley SCSI filing systems may utilise this feature.

---

**\*Configure SCSIFSDrive <drv>**

Sets up the default drive number. It is a value between 4 and 7 inclusive and defaults to 4.

**\*Configure SCSIFSHost <Id>**

Sets up the default Host ID for all the cards within the host computer. The value may be in the range of 0 to 7 inclusive and defaults to 7.

**\*Configure FileSystem SCSI**

Sets the default Filing System to SCSI. This normally defaults to ADFS.

**\*Configure SCSIFSNetwork <0|1>**

This enables file interlocking on multiple initiator systems when set to '1'. See section on 'Networking Capabilities'.

---

# Speeding up your system

Many modern drives 'Disconnect' during operation (see Chapter 4 SCSI Explained for more details) as this releases the SCSI bus for other devices. On most systems, there will only be one initiator so the Disconnect feature is of no advantage. As Disconnect can slow down the system, especially on high speed drives, speed increases in data transfer can be obtained by disabling the Disconnect feature. This is done by issuing SWI SCSI\_Control. Here is an example of turning off Disconnect on device 0.

**SYS"SCSI\_Control",6,0,1**

Refer to the section on SWIs for more information on this SWI. The CLI command 'SCSIDisconnection <ID> <TYPE>' also carries out this command.

---

# Networking Capabilities

The ANSI SCSI standard allows for a system called 'Multiple Initiator'. This is when more than one machine can act as a bus master on the same bus and accessing the same device(s). The SCSIFS software contains all the necessary interlocks for safe operation.

N.B. Due to limitations of RISC OS 2.00/2.01, full interlocking is not possible on write access, but RISC OS 3 has this implemented. To enable network interlocking, use "Configure SCSIFSNetwork 1' then perform a hard reset.

---

# Acorn PC Emulator

## Installation

When setting up a partition on a hard disc, remove the ADFS prefix on the path name and substitute SCSI instead.  
All other operations are as specified in the Acorn PC Emulator manual.

This chapter gives details of SWI operating system calls contained in the Relocatable Module 'SCSIDriver'.

## **SWI Calls**

The following SWIs are included in the SCSIDriver module:

- SCSI\_Version
- SCSI\_Initialise
- SCSI\_Control
- SCSI\_Op
- SCSI\_Status
- SCSI\_Reserve
- SCSI\_List
- SCSI\_Target

## **\*\* Commands**

The following '\*' commands are included in the SCSIDriver module:

- \*Devices
- \*SCSIBlock
- \*SCSICard
- \*SCSITest
- \*SCSI Disconnection
- \*SCSITransferType

---

All SWIs conform to the RISC OS standard:  
Any undefined registers are preserved.  
Processor flags N,Z and C are preserved.  
On error the returning call will set the V flag and R0 will point to an error block (If the bit 17 is clear version is called, control passes to the error handler.)

---

## **SCSI\_Version (&403C0)**

<b>Function</b>	Returns SCSI Driver Revision
<b>On Entry</b>	No parameters
<b>On Exit</b>	<p>R0 bits 0-7 software minor version number bits 8-31 software major version number (eg 100 for 1.00)</p> <p>R1 bitset of software features</p> <p>R2 bits 0-7 hardware minor version number bits 8-31 hardware major version number (eg 100 for 1.00)</p> <p>R3 bitset of hardware features</p> <p>R4 pointer to vendor string</p>
<b>Use</b>	Returns software and hardware supported features

---

Bit set of software features:

Bit	Effect
31	Supports Target mode
30	Supports configuration
29	Supports reading from control lines
28	Supports device reservation
27	Supports multiple SCSI cards
26	Supports *SCSIBlock
25..16	Reserved, must be 0
15..7	Reserved, must be 1
6	Supports *Devices
5	Supports scatter list for SWI SCSI_Op
4	Supports messaging
3	Supports automatic Request sense
2	Supports command queuing
1	Supports background data transfer
0	Supports target disconnect/reconnect

Bit set of hardware features:

31	Supports configuration
30..16	Reserved must be 0
15..1	Reserved must be 1
0	Supports target mode

---

## SCSI\_Initialise 0 (&403C1)

<b>Function</b>	Reset SCSI bus
<b>On Entry</b>	R0 0 R1 host
<b>On Exit</b>	No results
<b>Errors</b>	&20100 - No room for SCSI driver workspace &20103 - SCSI bus reset failed &20104 - Invalid SCSI host id
<b>Use</b>	Resets all SCSI_Control options to their default values and drives SCSI bus RESET to an active state.  Reservation                none Timeout                  250ms (ANSI recommended timeout) Error response           report check condition as error Queue behaviour          wait until slot becomes free Disconnection            allow disconnect/reselect Transfer Type           set to async

---

# **SCSI\_Initialise 1 (&403C1)**

<b>Function</b>	Reset SCSI device
<b>On Entry</b>	R0 1 R1 device id R8 access key
<b>On Exit</b>	No results
<b>Use</b>	Resets a SCSI device. Sends device reset message to specified device and returns.
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error

---

## SCSI\_Initialise 2 (&403C1)

<b>Function</b>	Performs an Inquiry command and Read-Capacity		
<b>On entry</b>	<p>R0 2</p> <p>R1 device id</p> <p>R2 pointer to buffer for description (16 bytes, word aligned)</p>		
<b>On exit</b>	<p>Buffer filled</p> <p>Offset R2</p> <p>0 type =      &amp;00      Direct-access      (eg Hard Disc)</p> <p>                  &amp;01      Sequential-access</p> <p>                  &amp;02      Printer device</p> <p>                  &amp;03      Processor device</p> <p>                  &amp;04      Write once read many (eg WORM drive)</p> <p>                  &amp;05      Read-only      (eg CD ROM)</p> <p>                  &amp;06-7D      Reserved</p> <p>                  &amp;7E      Device present but does not conform to CCS 4.B</p> <p>                  &amp;7F      Logical unit not</p> <p>                  present &amp;80-FF      Vendor unique</p> <p>1      Removable medium (bit7)/Device-Type Qualifier (bits 0 to 6)</p> <p>2      ISO (b7 to 6)/ECMA (b5 to 3)/ANSI(b2 to 0) version</p> <p>3      Reserved(reserved field returned by Inquiry command)</p>		

- 
- |       |  |                      |
|-------|--|----------------------|
| 4     | Additional length, extra data that could be obtained by an Inquiry command |                      |
| 5-7   | Reserved   |                      |
| 8-11  | Word number of logical blocks  | (converted to normal |
| 12-15 | Word block length  | ARM byte sex)        |
- Bytes 0 to 4 are the first five bytes returned by an inquiry command. Byte 5 is provided to indicate how much additional data (vendor id, product id, etc.) that could be obtained by an Inquiry command. Bytes 5 to 7 are reserved by the device driver.
- Bytes 8 to 11 (word 2) and 12 to 15 (word 3) are the block count and block size returned by a Read Capacity command but they have the byte sex reversed to suit ARM's LDR instruction.

- Errors**
- &20105 - Invalid SCSI device id
  - &2010A - Timeout

---

## SCSI\_Initialise 3 (&403C1)

<b>Function</b>	Enumerate device. Performs an Inquiry command to see what type the device is then performs the appropriate Read-Capacity command. Data is returned as formatted text.
<b>On entry</b>	R0 3 R1 device id R2 pointer to buffer for description (word aligned) R3 buffer size
<b>On exit</b>	The data returned is a zero terminated string of the format as that printed by *Devices, i.e.:  ddd ttttttttttttttt cccc bytes vvvwww ppppppppppppppppp rrrr tttt  Where: ddd device number tt..tt device type cccc device capacity w..w vendor pp..pp product rrrr revision number ttttt transfer type

If the device does not respond a null string is returned. If the device responds but does not conform to CCS 4.B a string containing "Device does not conform to CCS 4.B" is returned. If the device responds but the Read-Capacity call fails, the capacity field will be set to unknown.

---

## **SCSI\_Control 0 (&403C2)**

<b>Function</b>	Abort device
<b>On entry</b>	R0 0 R1 device id
<b>Use</b>	Abort device. Cancels any outstanding operation on a device. 'Abort op' should be used instead wherever possible
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error

---

## **SCSI\_Control 1 (&403C2)**

<b>Function</b>	Abort op
<b>On entry</b>	R0 1 R1 device id R2 operation id as returned by SCSI_Op R8 access key
<b>On exit</b>	No results
<b>Use</b>	Abort op. Cancels a particular operation on a device.
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error

---

## **SCSI\_Control 2 (&403C2)**

<b>Function</b>	Set LED status
<b>On entry</b>	R0 2 R1 0=off Non 0=on
<b>On exit</b>	No results
<b>Use</b>	Sets/resets LED status if hardware option fitted to SCSI card. Used by SCSIFS when accessing default drive.

---

## **SCSI\_Control 3 (&403C2)**

<b>Function</b>	Set timeout
<b>On entry</b>	R0 3 R1 device id R2 timeout value in centi-seconds, -1 returns original value R8 access key
<b>On exit</b>	R0 preserved R1 preserved R2 previous timeout setting
<b>Use</b>	Set timeout. Sets default timeout for a device.
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error
	N.B. The longest timeout period generated by the NCR53C94 is 42 centiseconds.

---

## SCSI\_Control 4 (&403C2)

<b>Function</b>	Set error response
<b>On entry</b>	R0 4 R1 device id R2 0 If device returns check condition, return that as an error message. This is for badly behaved devices with non-standard request sense return blocks 1 If a device returns check condition do the request sense automatically, report the error 2 If the device returns check condition, do request sense automatically, if unit attention, ignore it and reissue the command, else report the error. -1 No action , return current setting. R8 access key
<b>On exit</b>	R2 Previous setting
<b>Use</b>	Set Error Response. Controls the behaviour of the driver on receipt of a check condition status.
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error

---

## SCSI\_Control 5 (&403C2)

<b>Function</b>	Set queue behaviour
<b>On entry</b>	
R0	5
R1	device ID
R2	0     If queue is full, loop until a slot becomes free. If a command operates totally in the background, control returns to the caller as soon as the command is queued. If the command operates in the foreground, control returns when the command completes. 1     Reject if queue is full. The command is accepted only if it can be queued/ run without waiting. If a command operates totally in the background, control returns to the caller as soon as the command is queued. If the command operates in the foreground, control returns when the command completes. 2     Reject if the device is busy (or queue is full). The command is accepted only if there are no other queued/running for this device. 3     Reject if expansion card is busy. Accept command only if it will run immediately. -1    No action, return current setting.
R8	access key

---

<b>On exit</b>	R2 Previous setting The driver supports disconnection/reselection and queuing of commands. This mean: that when a command is issued the expansion card hardware may be in use servicing another command, a command may be queued/running for the target device or the target device or the queue may be full.
<b>Errors</b>	&20105 - Invalid SCSI device id. &2010E - SCSI driver reservation error.

---

## SCSI\_Control 6 (&403C2)

<b>Function</b>	Disconnect
<b>On entry</b>	R0 6 R1 device Id R2 0     disconnect/reselect (i.e., when the bus selection phase is complete, the driver will attempt to send an IDENTIFY message with bit 6 set. This tells the target that it may disconnect if it wants to). 1 Inhibit disconnect/reselect. Once started, a command will proceed to completion without allowing any other command to start/reconnect (i.e., when the bus selection is complete, the driver will attempt to send an IDENTIFY message with bit 6 clear). 2 Inhibit disconnect/reselect by not sending an IDENTIFY message -1 No action, return current setting

N.B. Codes 0/1 cause SWI SCSI\_Op to attempt to send an Identify message once the bus selection phase is complete. The target may choose not to take the message (not an error), in which case the driver proceeds to the command phase. The target may respond to the message with a message reject. These two cases do not produce an error from SCSI\_Op, as the command may still complete successfully. It will however remain connected all the time.

---

		Applications/filing systems are not expected to issue this call, as all software/hardware should work happily regardless of disconnections that may occur. It should only be used by:
	1	Users, in their boot files, if their particular devices misbehave when sent messages or for increasing device speed performance.
	2	Application and filing systems software specific to a SCSI device that is known to misbehave when sent messages.
<b>On exit</b>	R2	Previous setting
	R8	access key
<b>Use</b>		Disconnect. Allow/Inhibit disconnect/reselect. Device must be idle.
<b>Errors</b>		&20105 - Invalid SCSI device id. &20109 - Device not idle &2010E - SCSI driver reservation error.

---

## **SCSI\_Control 7 (&403C2)**

<b>Function</b>	Transfer type
<b>On entry</b>	R0 7 R1 device id R2 transfer type 0=Async Non 0=Sync R8 access key
<b>On exit</b>	No results
<b>Use</b>	If Transfer Type is set to synchronous and a device does not support this then it remains in asynchronous mode or in the worst case ceases to operate if not fully ANSI X3.1311986 Level 2 compliant.  This SWI is added to allow the user to control which devices may run in synchronous mode. Some drives that do not comply to ANSI X3.131-1986 Level 2 do not like this so this option is set to zero by default.
<b>Errors</b>	&20105 - Invalid SCSI device id &2010E - SCSI driver reservation error

---

## **SCSI\_Op (&403C3)**

<b>Function</b>	Issues a SCSI command to a device and can be called in the background
<b>On entry</b>	<p>R0 b0-b7 device id b25-b24 00 = no data transfer, 01 = read, 10 = write, 11 reserved b26 Scatter bit if set, R3 is the pointer to scatter list b27 If clear poll escape during transfer and abort if escape pressed b28 If set, repeatedly retry on timeout b29 Set if background transfer (possible zero length).</p> <p>R1 Length of SCSI control block R2 Pointer to SCSI control block R3 Memory pointer for start of transfer, or pointer to scatter list of address length pairs if bit 26 of R0 is set. After each chunk in the scatter list has been transferred the address will be incremented and the length set to zero. R4 Length of foreground transfer (in bytes). R5 Timeout in centi-seconds or 0 for default timeout R6 Address to call back when transfer complete if background bit set. The call is made with IRQ's disabled. The routine addressed by R6 may enable IRQs if it wishes to do so. If an error has occurred, V is set, R0 points to an error block, R1 indicates the cause of the error, R2 should be the logical address on the device where the error occurred (if request sense has been issued).</p>

- 
- R7    Workspace pointer to **be** returned in R12 for background call back.  
R8    Access key

**On exit V clear**

- R0    Returns a 32 bit id incremented for each operation. Used to cancel op.  
R3    Updated to indicate how much of the transfer was successful. (If scatter list, pointer to first unused/partially used entry.)  
R4    Updated to the amount not transferred.

**V set**

- R0    Points to an error block.  
R1    Indicates the cause of an error.  
R2    Holds the logical block address of the device on which the error occurs.  
R3    Updated to indicate how much of the transfer was successful. (If scatter list, points to first unused entry. Entry is updated to indicate amount transferred.)  
R4    Updated to the amount not transferred.

**Errors**

See the following section on SWI error messages for more information.

---

## **SCSI\_Status 0 (&403C4)**

<b>Function</b>	Check device status
<b>On entry</b>	R0 0 R1 device id
<b>On exit</b>	R0 status 1 - idle 2 - busy
<b>Use</b>	Check device status.Returns the status of the selected device.
<b>Errors</b>	&20105 - Invalid SCSI device id.

# SCSI Reserve

**Device reservation** The following SWIs use R8 as an access key:

- SCSI\_Initialise
- SCSI\_Control
- SCSI\_Op

Any of these SWIs can return &0002010E (reservation error) if an attempt is made to access a reserved device without quoting its key.

The SWI SCSI\_Reserve is used to control this mechanism. This SWI allows a caller to claim exclusive use of a device and prevent data corruption problems that may arise if a program (by accident or design) issues a Write/Format/Mode Select command to a device currently mounted by SCSIFS. It also allows for a clean change of device ownership from one user to another. Because !SCSI\_Form modifies direct access devices (e.g. hard discs) you must first dismount a device before using !SCSI\_Form.

**Passing access key** The access key (typically a workspace pointer) should be passed in R8 and is only checked if the device specified is claimed and the call/reason code is dangerous.

N.B. BASIC (before RISC OS 3) only passes R0-R7 on a SYS call. If a BASIC program wishes to reserve a device a small piece of ARM assembler code that loads the access key into R8 and then calls the required SWI will have to be included.

---

## **SCSI\_Reserve 0 (&403C7)**

<b>Function</b>	Claim. Claim exclusive use of a device
<b>On entry</b>	R0 0 R1 device id R2 release address or zero if no release address R3 workspace pointer, passed in R12 when release address is called R8 access key
<b>On exit</b>	VC VS and RO = error
<b>Use</b>	This call attempts to claim exclusive use of the given device. If the call is successful, the caller will be granted full use of the dangerous SCSI driver calls for that device, and may use them itself only by supplying the access key registered at claim time. Other users may only use safe calls, such as SCSI_Initialise-Determine Device.  If the device has already been claimed, an error message Reservation error will be returned.
<b>Errors</b>	&20105 - Invalid SCSI device id. &2010E - SCSI driver reservation error.

---

## SCSI\_Reserve 1 (&403C7)

<b>Function</b>	Force claim
<b>On entry</b>	R0 1 R1 device id R2 release address or zero if no release address R3 workspace pointer, passed in R12 when release address is called R8 access key
<b>On exit</b>	VC VS and R0 = error This call is similar to 'claim' but if the device has already been claimed, it will ask the current claimant to release it by calling the release address registered for that device. The current owner should try to tidy up but may refuse by returning VS and R0 pointing to an error message.
<b>Errors</b>	&20105 - Invalid SCSI device id. &2010E - SCSI driver reservation error or an error message passed back by the current claimant. When the current claimant is called, R0 will hold a reason code indicating: R0 = 1      called from ForceClaim

---

## **SCSI\_Reserve 2 (&403C7)**

<b>Function</b>	Release. Will remove the reservation of a device
<b>On entry</b>	R0 2 R1 device id R8 access key (to prove you actually owned the device)
<b>On exit</b>	VC VS = error
<b>Errors</b>	&20105 - Invalid SCSI device id. &2010E - SCSI driver reservation error.

---

## **SCSI\_List (&403C8)**

**On exit** Null terminated list of expansion card addresses.

**Use** This call returns a pointer to a null terminated list of SCSI expansion cards available to SCSIDriver. The addresses are those of the slow address space for the expansion card, i.e. &03240000, &03244000, &03248000, &0324C000, for expansion card slots 0,1,2 and 3 respectively.

**N.B. This table is used by SCSIDriver and is therefore READ ONLY.**

---

# **SCSI\_Target 0 (&403C9)**

<b>Function</b>	Configures the selected SCSI card to be a SCSI target.
<b>On entry</b>	<p>R0 0</p> <p>R1 bit 0-2 SCSI id bit 3-4 SCSI card id</p> <p>R2 base address of target buffer</p> <p>R3 logical block length, multiple of 2"</p> <p>R4 number of block starting at R2</p>
<b>On exit</b>	On return the selected SCSI card is placed in target mode and will service all SCSI bus requests in the background.
<b>Use</b>	This command puts the selected SCSI card into target mode, acting as a direct access device.

N. B. Registers R2 to R4 are not range checked so exercise extreme caution. The SCSI card can only be a target or an initiator at any one time.

---

## SCSI\_Target 1 (&403C9)

<b>Function</b>	Configures the selected SCSI card to be an initiator.
<b>On entry</b>	R0 1 R1 bit0-2SCSI id bit 3-4 SCSI card id
<b>On exit</b>	Return to initiator mode.
<b>Use</b>	Returns a previously configured SCSI target to initiator mode.

---

# SCSIFS SWI calls

**SWI Calls**

The following SWIs are provided by SCSIFS:

SCSIFS_DiscOp	(&40980)
SCSIFS_Drives	(&40982)
SCSIFS_FreeSpace	(&40983)
SCSIFS_DescribeDisc	(&40985)
SCSIFS_MiscOp	(&4098C)

For entry and exit conditions, see FileCore\_DiscOp, FileCore\_Drives, FileCore\_FreeSpace, FileCore\_DescribeDisc and FileCore\_MiscOp in the RISC OS Programmer's Reference Manual.

The SWI SCSIFS\_TestReady (&40986) is described overleaf.

---

## SCSIFS\_TestReady (&40986)

On entry            R1     Drive number (0...7)

**On exit**           R0     0     drive not present  
                      1     not present or present but not ready  
                      2     drive present and ready

# SWI Error Messages

## Driver Errors

Errors generated by SCSI driver:

- &20100 No room for SCSI driver workspace
- &20101 Unknown SCSI SWI number
- &20102 Unknown reason code for SCSI SWI
- &20103 SCSI bus failed to reset
- &20104 Invalid SCSI host id
- &20105 Invalid SCSI device id
- &20109 Device not idle
- &2010A Timeout during selection phase
- &2010B Timeout during any other phase
- &2010C Command queue not empty
- &2010D Command queue full
- &2010E Device reservation error
- &2010F Invalid parameters
- &20110 Parameter error
- &20111 Not callable from IRQ routine (reported by expansion card loader)
- &20112 Operation aborted
- &20113 Unexpected disconnection
- &20114 May only be initiator or target
- &20115 Unexpected interrupt
- &20120 Gross error
- &20121 Illegal command
- &20122 SCSI bus reset
- &20123 Illegal selection/reselection
- &20124 Illegal bus phase

---

<b>SCSI status byte errors</b>	On completion of a SCSI command the target device returns a command status byte to indicate completion status. If this byte is non-zero the following errors are generated: &20180 Target status-Check condition &20181 Target status-Busy &20182 Target status-Unknown &20183 Target status-Intrmd/goodstatus &20184 Target status-Reservation conflict  If error response level is set to 1 or 2, then on a completion status of check condition, a Request Sense command is issued to the target device automatically.
<b>Sense data</b>	<p>Sense data is interpreted in the following manner, bit 7 of byte 0 is masked to leave:</p> <p>bits 6 - 4      Error class bits 3 - 0      Extended sense data</p> <p>These are reported in the following manner: Target error (00/xx) where (xx) reflects additional sense error code.</p> <p>&amp;201C0 Target error - (00/xx) No sense &amp;201C1 Target error - (01/xx) Recovered error &amp;201C2 Target error - (02/xx) Not ready &amp;201C3 Target error - (03/xx) Medium error &amp;201C4 Target error - (04/xx) Hardware error &amp;20105 Target error - (05/xx) Illegal request &amp;20106 Target error - (06/xx) Unit attention &amp;201C7 Target error - (07/xx) Data protect &amp;201C8 Target error - (08/xx) Blank check</p>

---

&201C9 Target error - (09/xx) Vendor unique  
&201CA Target error - (0A/xx) Copy aborted  
&201CB Target error - (0B/xx) Aborted command  
&201CC Target error - (0C/xx) Equal  
&201CD Target error - (0D/xx) Volume overflow  
&201CE Target error - (0E/xx) Miscompare  
&201CF Target error - (0F/xx) Reserved sense key  
&201D0 Target error - (xx/xx) Unknown

<b>SCSI_Op errors</b>	Errors produced by calling SWI SCSI_Op
R0	= error message. Pointer to a standard RISC OS error block, consisting of one word error number followed by an error message.
R1	= error indication. This is a composite of the LSB of the above error number and the status, sense-errorclass and sense key that produced the error.
R2	= Logical address on the device where the error occurred if possible.

---

# \* Commands

Six '\*' commands are supported by SCSI Driver.

## \*Devices

This lists all the devices found on the SCSI bus(es) with their name, manufacturer, capacity (if known), version number and device type. An example of the \*Devices command :

	Device	Type	Capacity	Vendor	Product	Rev	Trans
0	Direct access	169 Mbytes	CDC	ST1201NS		9204	Sync
4	Direct access	31 Mbytes	SEAGATE	ST138N		4	Async
7	Host (Cached)		MORLEY	SCSI CARD		1.10	
8	Sequential	Unknown	WANGTEK	5150ES SCSI ES41 C230			Async
15	Host (Uncached)		MORLEY	SCSI CARD		1.10	

This system has two SCSI buses, with the tape streamer residing on the second bus, with the SCSI ID of 0 and a bus ID of 8.

---

<b>*SCSICard</b>	This command lists all the SCSI podules in the host computer, along with the Host ID for each card and the onboard cache size, e.g. Card :0 host id :7 cache size :8 Kbytes Card :1 host id :7 cache size :16 bytes
<b>*SCSIBlock &lt;addr&gt; &lt;accesskey&gt; &lt;id&gt; &lt;cdb 0&gt; ... &lt;cdb n&gt;</b>	This command allows direct access to the command interface of a SCSI device. It is recommended that software should use the SWI SCSI_Op as this "*" command gives no access to the data transfer size ( <b>R4</b> on SCSI_Op) and is set to &7FFFFFFF by default. This command is supported for software which was written for other 3rd party SCSI systems, but we strongly recommend that all new software should not use it.
<b>*SCSITest</b>	This command tests each of the SCSI cards in a computer. A check is done on the DMA counter, FIFO and cache (if present). An error is given is a fault is found.
<b>*SCSIDisconnection &lt;DevID&gt; &lt;DisconType&gt;</b>	<DevID> is in hex and <DisconType> is zero/non-zero. This command calls up SWI SCSI_Control 6, with R1 set to <DevID> and R2 set to <DisconType>. This sets the disconnection type of a device.
<b>*SCSITransferType &lt;DevID&gt; &lt;TxType&gt;</b>	<DevID> is in hex and <TxType> is zero/non-zero. This command calls up SWI SCSI_Control 7 with R1 set to <DevID> and R2 set to <TxType>. This sets the transfer type of a device.

## Introduction

The SCSI standard was developed to meet the need in the small computer market for a generic interface, that is an interface that was inexpensive to implement and was not device dependent. The command structure used by the SCSI protocol is defined in such a way that the structure of the device being accessed is totally transparent to the user. On a hard disc for example the number of heads and cylinders is totally irrelevant as data is accessed as logical blocks. A logical block consists of a predefined number of bytes (e.g. 256 bytes) with each logical block being assigned a logical block address in the range of 0 to the total number of blocks - 1 available on that device. In this way the SCSI bus allows for devices like CD ROMs, hard discs, tape streamers, scanners and printers to be connected to one device interface, with out the need for individual device specific interfaces.

The foundation for the SCSI bus was taken from a commercial small system parallel bus, the Shugart Associates System Interface (SASI) and formalized by the American National Standards Institute. The SASI interface allowed for up to eight devices to be controlled from one computer with data rates in the region of 1-2 MB/s and had already enjoyed large market success. This was due to it being manufacturer independent and its device independence. This meant that the interface became quite attractive to computer manufacturers as they did not have to design complex controllers for their machines.

The SASI interface had the following extensions

1. A differential driver/receiver option to allow the use of cables up to 25 meters.
2. A synchronous transfer option was added to allow transfer rates in region of 3-4 MB/s

- 
3. An optional 'Extended' command set was added to allow the use of larger capacity devices.
  4. Command sets for magnetic tape, printers, processors, optical devices, and read only optical discs were added.

## SCSI Signals

Signals on the SCSI bus are divided into two groups - data and control. The data signals comprise of eight data bits DB0 to DB7 (DB7 being the most significant bit) and one parity bit DBP making up the nine data lines. The nine control signals are as follows:

- BSY (BUSY) An 'OR tied'\* signal to indicate the bus is being used.
- SEL (SELECT) A signal driven by the initiator to select a target, or by a reselecting target.
- C/D (CONTROL/DATA) A signal driven by the target to indicate that the current data on the bus is either control or data.
- I/O (INPUT/OUTPUT) A signal driven by the target to indicate the direction of data on the bus in relation to the initiator, if asserted initiator reads data.
- MSG (MESSAGE) A signal driven by the target during the message phase.
- REQ (REQUEST) A signal driven by a target to indicate a request for a REQ/ACK data transfer handshake.
- ACK (ACKNOWLEDGE) A signal driven by the initiator to indicate an acknowledgement for a REQ/ACK data transfer.

---

ATN (ATTENTION) A signal driven by the initiator to indicate the attention condition

RST (RESET) An 'OR tied'\* signal to indicate a reset condition.

\*An 'OR tied' signal is one that can be driven from several sources at the same time.

## **SCSI Bus Phases**

Communication across the SCSI bus is carried out by going through what are called phases. The SCSI architecture contains eight such phases:

1. Bus Free Phase
2. Arbitration Phase
3. Selection Phase
4. Reselection Phase
5. Command Phase
6. Data Phase (in and out)
7. Status Phase (in only)
8. Message Phase (in and out)

---

## **Bus Free Phase**

The bus free phase indicates that there is no device currently using the SCSI bus. The bus free phase is said to be true when no signals on the SCSI bus are being driven (except RST).

## **Arbitration Phase**

The arbitration phase allows a device to gain control of the SCSI bus for the purpose selecting a target or reselecting an initiator. To gain control of the SCSI bus a device must proceed as follows:

1. The device must first wait for the Bus Free phase.
2. After detecting the Bus Free Phase the device will assert the BSY line and place its own SCSI id bit on the data bus.
3. After waiting an arbitration delay the device will examine the SCSI bus. If a higher id bit than its own (DB7 being the highest) is present on the bus, the device has lost the arbitration and that device must release all bus signals and return to step 1. If no higher id is present on the bus the device has won the arbitration and will assert SEL.

## **Selection Phase**

After the Arbitration Phase, the selecting device has both BSY and SEL asserted and will then assert the data bus with its own SCSI id and the id of the target. The initiator will then release the BSY signal and wait for a response from the target.

---

## **Reselection Phase**

The reselection phase allows a target to reconnect to an initiator for the purpose of continuing a previous operation originated by the initiator but suspended by the target (i.e. the target disconnected to carry out some lengthy task thus freeing the bus). After the Arbitration Phase the selecting device has both BSY and SEL asserted and will then assert the data bus with its own SCSI id and the id of the initiator. The target will then release the BSY signal and wait for a response from the initiator.

The initiator detects that it is being reselected when SEL, I/O and its own SCSI id are true and BSY is false. At this point the Reselected Initiator may examine the bus to determine the reselecting target's id.

The initiator will then drive BSY true. As soon as the target detects BSY it will also assert BSY and the Release SEL. The target may then afterthe state of the I/O signal afterthe reselected initiator detects that SEL has gone false.

## **Information Transfer Phases**

The following bus phases are grouped together as information transfer phases as they all transfer either data or control information across the SCSI bus. The CO, I/O, and MSG signals are use to distinguish between the different phases as follows:

## Signals

MSG	C/D	I/O	Phase Name	Direction of transfer
0	0	0	DATA OUT	INITIATOR TO TARGET
0	0	1	DATA IN	INITIATOR FROM TARGET
0	1	0	COMMAND	INITIATOR TO TARGET
0	1	1	STATUS	INITIATOR FROM TARGET
1	0	0	RESERVED	
1	0	1	RESERVED	
1	1	0	MESSAGE OUT	INITIATOR TO TARGET
1	1	1	MESSAGE IN	INITIATOR FROM TARGET

## Command Phase

The command Phase allows the target to request command data from the initiator.

## Data Phase

The data phase is a term used to describe both data in and data out phases, i.e. transferring data from the target to the initiator and from the initiator to the target.

---

## **Data In Phase**

During the data in phase, data is transferred from the target to the initiator.

## **Data Out Phase**

During the data out phase data is transferred from the initiator to the target.

## **Status Phase**

During the status phase the target sends command completion status information to the initiator

## **Message In Phase**

The message phase allows for the target to send messages to the initiator during a transaction.

## **Message Out Phase**

The message out phase is requested by the initiator asserting the attention signal. The target will respond with a message out phase at its own convenience. In response to the message out phase the initiator will respond with the appropriate message.

---

A description of all messages in the SCSI protocol is beyond the scope of this manual and can be found in the ANSI SCSI specification manual however a small selection of messages are explained below.

## **Messages**

### **Command Complete (&00)**

This message is sent to the initiator to indicate completion of a command and that a valid status has been sent.

### **Save Data Pointers (&02)**

This message is sent by the target to the initiator. On receiving this message the initiator should save all its current pointers for the currently connected LUN.

### **Restore Data Pointers (&03)**

This message is sent by the target to the initiator. On receiving this message the initiator should restore the most recently saved pointers for the connected LUN. If a disconnect occurs during a data in or data out phase the restored data pointer allows the transfer of data to continue where it left off. This message can also be used to correct errors during transfers.

---

### **Disconnect (&04)**

This message is sent by the target to inform the initiator that the current physical path is about to be severed but the target will reconnect at a later stage to continue the operation.

### **Bus Device Reset (&0C)**

This message is used by the initiator to direct a target to clear all current commands. The target should also enter the Bus Free phase on reception of this message.

### **Identify (&80 - &FF)**

These messages are sent from either target or initiator to set up a physical path from an initiator to a target or vice versa.

---

## Phase Sequences With Arbitration

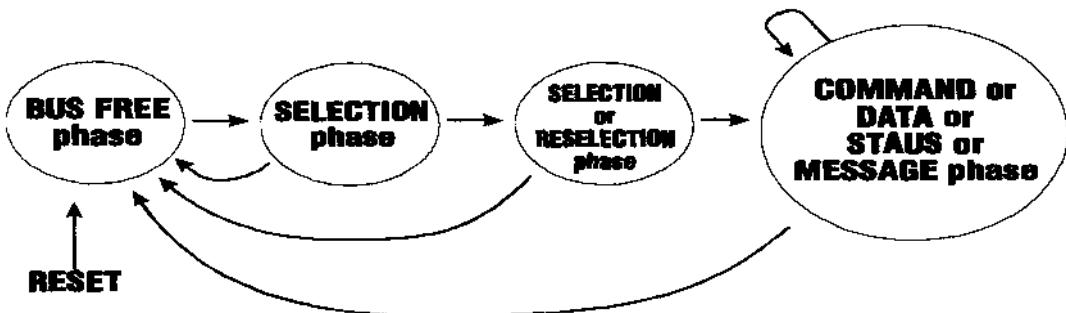


Figure 1.4 shows all allowable sequences for a device from the bus free phase to selection or reselection to command.

## **SCSI Pointers**

The SCSI standard allows for two sets of three pointers. Each set of pointers contains the following:

1. Data pointer (pointer to data source or destination address).
2. Command descriptor block pointer (pointer to command descriptor block in memory).
3. Status pointer (pointer to memory area where status is stored).

The first set of pointers are known as the current pointers. These pointers contain the current status of the initiator and point to the next command, data, or status byte to be transferred between the initiator and the target.

The second set of pointers, known as the saved pointers, represent the state of all active commands regardless of connection state. The initiator will store the contents of its current data pointer into the saved data pointer in response to a save data pointer message from the target. The saved data pointer will be restored to the current data pointer on reception of a restore data pointer message from the target. The target can also modify the initiator's current pointer by the use of the extended message Modify Pointer. This message instructs the initiator to add a four byte two's complement offset to the current data pointer.

## **Asynchronous Information transfer**

Asynchronous data transfers are carried out using a simple handshaking method. The target controls the direction of data transfer using the I/O line as follows; when the I/O line is driven true, data is transferred from the target to the initiator, when false, from the initiator to the target.

---

When data is being transferred from the target, the target will drive DB(0-7,P) to their desired values, then drive REQ true. When the initiator has recognised REQ asserted it will read the data on the bus and then assert the ACK signal indicating to the target that data has been read from the bus. In response to the ACK signal the target will release REQ and then the initiator will release ACK

When data is being transferred from the initiator, the initiator will drive DB(0-7,P) to their desired values then drive ACK true. When the target has recognised ACK asserted it will read the data on the bus and then assert the REQ signal indicating to the initiator that data has been read from the bus. In response to the REQ signal the initiator will release ACK and then the target will in turn release the REQ signal.

In this way every byte is transferred with an accompanying handshake.

## Synchronous Data Transfer

The synchronous data transfer is an enhancement to the asynchronous transfer method but is only used during Data out or Data in phases and only by prior arrangement between the initiator and the target using messages. The messages establish a REQ/ACK offset.

The REQ/ACK offset specifies the number of REQ pulses that can be sent by the target in advance of the number of ACK pulses sent by the initiator. In this way the target and the initiator can set a pace at which data are transferred with only two constraints:

1. The REQ/ACK handshakes should never be out of sync by more than the REQ/ACK offset.
2. In order for a successful termination of a data phase the number of REQ pulses should equal the number of ACK pulses.

---

## Command Descriptor Block (CDB)

During the command phase the target requests command data from the initiator. In response the initiator then sends a command descriptor block. The command data is sent in the form of a predefined number of bytes. The first byte of the command descriptor block is always a operation code and is followed by the logical unit number.

	7	6	5	4	3	2	1	0						
0	Operation Code													
1	Logical Unit No		Logical Block Address (MSB)											
2	Logical Block Address													
3	Logical Block Address (LSB)													
4	Transfer Length													
5	Control Byte													

Figure 4.2

---

	7	6	5	4	3	2	1	0				
0	Operation Code											
1	Logical Unit No			Reserved				Rel Ad				
2	Logical Block Address (MSB)											
3	Logical Block Address											
4	Logical Block Address											
5	Logical Block Address (LSB)											
6	Reserved											
7	Transfer Length (MSB)											
8	Transfer Length (LSB)											
9	Control Byte											

Figure 4.3

---

## Operation Code

The operation code of the command descriptor block has a group code field and a command code field.

The three bit group code field provides for eight groups of commands. The five bit command code field provides for thirty two command codes in each group giving a possible total of two hundred and fifty six codes.

The group code specifies one of the following groups:

	7	6	5	4	3	2	1	0
0	<b>Group Code</b>			<b>Command Code</b>				

Figure 4.4

Group 0 - Six byte commands.

Group 1 - Ten byte commands.

---

## **Logical Unit Number (LUN)**

The logical unit number (LUN) addresses up to eight physical devices attached to a target. A target device may be a floppy disc controller with two or more floppy disc units connected, in this case floppy drive 0 would be LUN 0, and floppy drive 1 would be LUN 1.

## **Logical Block Address**

Logical block addresses on SCSI devices begin with block 0 and are contiguous up to the last logical block on that logical unit.

Group 0 command descriptor blocks contain a 21 bit logical address where group 1 command descriptor blocks contain a 32 bit logical block address.

The logical block contains a number of bytes which can be established by issuing a read capacity command or a mode sense command or by prior arrangement. The maximum logic block address accessible by the initiator can be found by the use of a read capacity command.

## **Relative Address Bit**

The relative address bit when set in group 1 and group 5 commands indicates that the logical block address held within the command descriptor block is a two's complement displacement to be added to the last logical block accessed on that device.

---

## Transfer Length

The transfers length normally defines the number of logical blocks to be transferred.

## Control Byte

The control byte is the last byte of every command descriptor block and is described as followed:

- Bits 7-2      These bits must be 0.
- Bit 1           Flag bit. If the link bit is 0, then the flag bit must be set to 0. If the link bit is 1 and if the command terminates successfully, the drive will send linked command complete with flag message if the flag bit is 1.
- Bit 0           Link bit. This bit is set to 1 to indicate that the initiator desires an automatic link to the next command.

Upon successful termination of a command the drive will return an intermediate status and will then send one of two messages defined by the flag bit.

---

## Command Example Including Disconnect

A typical operation to a hard disc would be a Read command (&08). The following example will describe a read operation in detail:

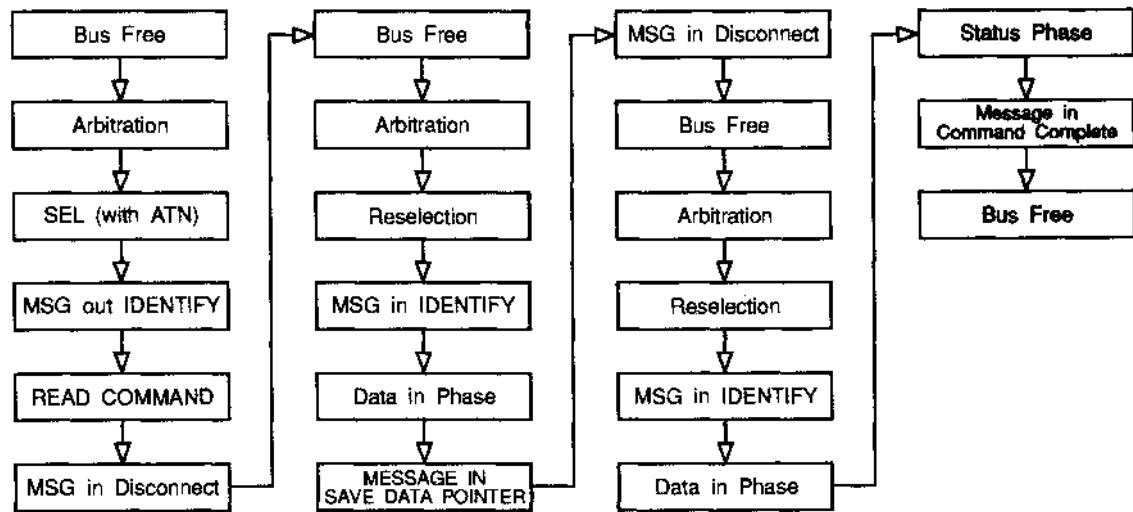
1. The initiator will first set up its active pointers for this command and wait until it detects the Bus Free phase.
2. The initiator will then arbitrate for the SCSI bus and select the hard disc with ATN true. The target will acknowledge the Attention condition by issuing a message out phase. The initiator will send an Identify message to set up a physical path between the initiator and the target.
3. The target will obtain the command from the initiator by issuing a command phase, in response to which the initiator will send the command descriptor block for a read command.

Once the drive has interpreted the read command it may decide that a long physical seek is necessary. In this case it will disconnect to free the bus allowing other operations to take place on the SCSI bus. The drive will do this by issuing a disconnect message and then releasing BSY.

4. When the drive is ready to send the data the drive will Reselect the initiator. This will cause the initiator to restore its saved pointers (implied by Reselect) to its current pointers. The initiator will continue with the command as normal until the command terminates with the drive sending the command completion status.

---

The drive may also wish to disconnect at some point during the data transfer to carry out some lengthy task. The drive will do this by issuing a Save Data Pointers message and then a Disconnect message. When the drive is ready to continue sending data to the initiator it will reselect and the command will continue as normal.



---

# **Appendix A    ANSI Statement of Conformance**

## **ANSI SCSI standard**

The Morley SCSI interfaces conform to the ANSI Small Computer System Interface (SCSI): Document Number **ANSI X3.131-1986 (X3T9/84-40 Rev. 1B) (X3T9.2/82-2 Rev. 17B)**.

## **ANSI Conformance Level**

The Morley SCSI podules provide level 2 conformance (highest level.) with the following options:

### **SCSI interface**

- (1) Single ended drivers/receivers.
- (2) Termination power supplied by host adapter.
- (3) Parity implemented.
- (4) Hard Reset option implemented.
- (5) Reservation implemented.
- (6) Synchronous transfer implemented.

## Messages

(&00)	Command Complete	Accepted/Generated	(target mode)
(&01)	Extended Message	Accepted	
(&02)	Save Data Pointer	Accepted	
(&03)	Restore pointers	Accepted	
(&4)	Disconnect	Accepted	
(&5)	Initiator Detected Error	Generated	
(&6)	Abort	Generated	
(&7)	Message Reject	Accepted/Generated	
(&8)	No Operation	Accepted/Generated	
(&9)	Message Parity Error	Generated	
(&0A)	Link Command Complete	Accepted	
(&0B)	Link Command Complete with flag	Accepted	
(&0C)	Bus Device Reset	Generated	
(&80-&FF)	Identify	Accepted/Generated	

## Extended Messages

(&00)	Modify Data Pointer	Accepted
(&1)	Synchronous Negotiation	Accepted/Generated
(&2)	Extended Identify	Accepted

The firmware allows for direct connection of devices conforming to Common Command Set (CCS) document Rev 4.B for (direct access devices). A generic interface is provided for devices not conforming to the CCS-4B.

# Appendix B

# Specifications

## Hardware

	<b>Standard SCSI Controller</b>	<b>Cached SCSI Controller</b>
SCSI controller	NCR 53C94 or Emulux ESP 216 or Emulux FAS 216	NCR 53C94 or Emulux ESP 216 or Emulux FAS 216
Internal transfer size	16 bits	16 bits
Internal transfer type	Pseudo DMA block mode	Demand mode DMA
Maximum transfer rate	6 MB/s asynchronous 5 MB/s synchronous	6 MB/s asynchronous 5 MB/s synchronous
Sustained data transfer	Machine dependent. (Sustained at 2.1 MB/s with ARM3)	Improved data transfer due to data decoupling. Machine dependent. (Sustained at 4.0 MB/s with ARM3)
Maximum devices	224 (4 cards, 7 devices and 8 LUNs)	224 (4 cards, 7 devices and 8 LUNs)
Cache size	8*18 bits including parity	18*512 bits including parity (standard). Expandable

---

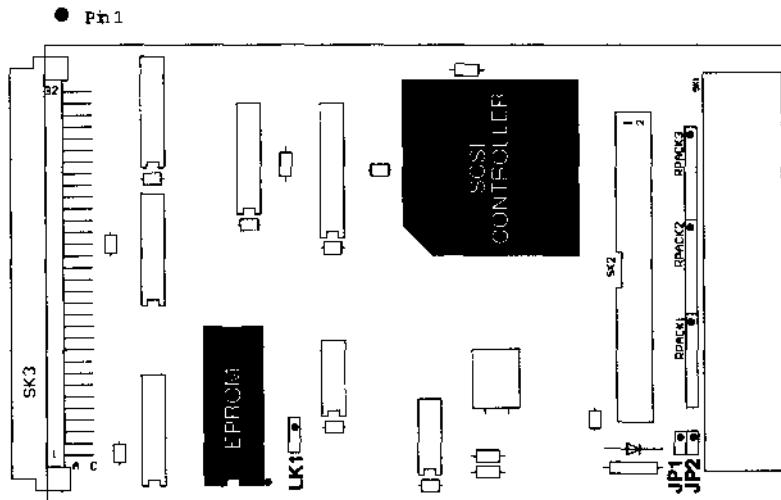
## **Software**

Filing system	Fully FileCore compatible.
Desktop	Custom Filer.
SCSI Networking	Multiple initiator with interlocks.
Target mode	Acts as direct access device.
Partitions	Up to 4 logical partitions per device.
Maximum filing system capacity	2 Gigabytes + 4 floppy discs.
Third party support	Full generic interface supplied.

## Appendix C

## Jumper Settings

Standard SCSI Card



JP1 JP2

Onboard Terminators take power from SCSI card



SCSI bus terminator power taken from SCSI card

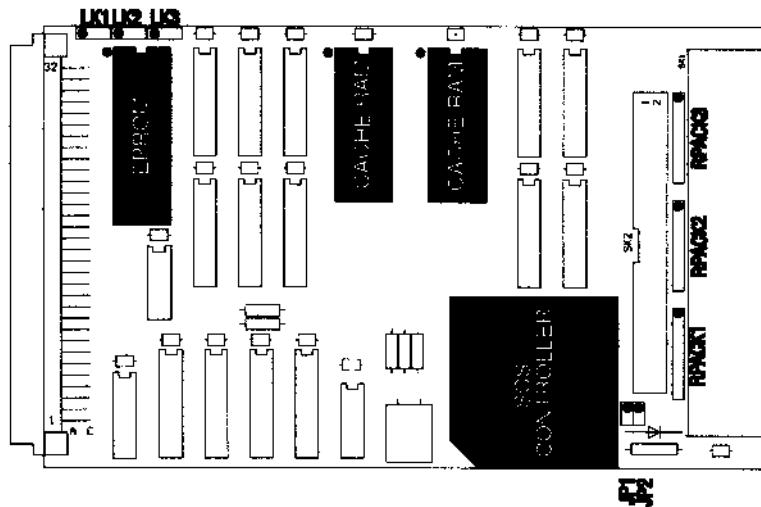


Onboard Terminators take power from SCSI bus



## Cached SCSI Card

• Pin 1



Onboard Terminators take power from SCSI card

SCSI bus terminator power taken from SCSI card

Onboard Terminators take power from SCSI bus