

## **MANUAL**

**- HG/GOUDA SCSI INTERFACE v1.1**

**- NOVAXIS ROM v1.5**

**- NFDISK v1.0**

Written by *Gert de Boom, Jurgen Kramer and Roderik Muit*

Corrected by *Jan Botman, John van Poelgeest and Erno Prosman*

Version 1.1, July 1995

Printed in Gouda - Holland

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission, in written, from the authors. For permission contact Gert de Boom, Gouderaksedijk 113, 2808 ND, Gouda, Holland.

Permission is granted for translations into French, German, Italian, Spanish and Japanese, but we do want to receive a copy of these translations.

**Copyright ©1995 by MSX Club Gouda.**

**All rights reserved.**

## **PREFACE**

Many Dutch buyers of the SCSI interface with NOVAXIS ROM are probably wondering why this manual is in English. Even I have had some trouble with it. However, in the present situation it was simply the best solution. Many, most is more honest, SCSI interfaces are sold outside Holland and sadly enough Dutch is not a world language, so we decided to write the manual in English. Maybe we will translate it into Dutch in the near future, but do not count on it.

By publishing all calls in the SCSI BIOS we hope to encourage programmers to write their own programs for the SCSI interface. Maybe this will result in faster programs and programs that are especially written for hard disk users, like a Norton Commander for MSX.

With the new FDISK program it is possible to create MS-DOS compatible partitions so that the connection with MS-DOS PCs will be much better. This is a very handy option for MSX emulator users and users of cross-compilers. Now do your development on the PC and immediately try the resulting code on your (real) MSX.

We wish you the best with our SCSI interface.

# CONTENTS

CONTENTS .....	- 1 -
CHAPTER 1 - GENERAL SCSI INFORMATION .....	- 3 -
1.1 - What is SCSI? .....	- 3 -
1.2 - What about SCSI on MSX?.....	- 3 -
1.2.1 - Hardware requirements.....	- 4 -
CHAPTER 2 - HARDWARE .....	- 5 -
2.1 - What kind of SCSI harddisk to buy.....	- 5 -
2.2 - Connecting hard disk and SCSI interface .....	- 6 -
CHAPTER 3 - NOVAXIS SCSI ROM .....	- 7 -
3.1 - What is new?.....	- 7 -
3.2 - Getting Started with NOVAXIS ROM .....	- 7 -
3.3 - Setup Utility .....	- 9 -
3.3.1 - Host ID .....	- 10 -
3.3.2 - Target ID .....	- 10 -
3.3.3 - Multiple HDD support.....	- 10 -
3.3.4 - Extended partitions .....	- 10 -
3.3.5 - Rescan .....	- 11 -
3.3.6 - Quit without save / Save and quit.....	- 11 -
3.3.7 - Units online.....	- 11 -
3.3.8 - Current Target.....	- 11 -
3.3.9 - Device type .....	- 11 -
3.4 - Extra programs built in.....	- 12 -
3.4.1 - CALL SLOW/MEDIUM/FAST.....	- 12 -
3.4.2 - CALL MAP/MAP2.....	- 12 -
3.4.3 - CALL HOST/TARGETID, SETHOST/TARGETID INQUIRY, FORMAT.....	- 13 -
3.5 - Errors .....	- 15 -
CHAPTER 4 - NFDISK.....	- 16 -
4.1 - What is NFDISK? .....	- 16 -
4.2 - Starting NFDISK.....	- 16 -
4.3 - The beginning .....	- 16 -
4.4 - The edit-screen.....	- 16 -
4.5 - The write-ID .....	- 17 -

4.6 - Editing the partition data.....	- 18 -
4.7 - Different partitionable-formats .....	- 18 -
4.8 - So now what ?.....	- 19 -
4.9 - Problems .....	- 20 -
4.10 - Remarks.....	- 20 -
 CHAPTER 5 - NETWORKING.....	 - 22 -
5.1 - Hardware.....	- 22 -
5.2 - Setting up hosts and targets.....	- 22 -
5.3 - Adding an MS-DOS PC to the network .....	- 22 -
 CHAPTER 6 - WRITING SOFTWARE.....	 - 24 -
6.1 - BIOS CALLS.....	- 24 -
6.2 - Additional NOVAXIS info.....	- 27 -
6.3 - Disk Parameter Block.....	- 27 -
6.4 - Layout of Work Area .....	- 28 -
6.5 - An example in BASIC .....	- 28 -
6.6 - An example in ML.....	- 29 -
6.7 - An example in ASCII-C (1.2).....	- 30 -
 CHAPTER 7 - UTILITIES.....	 - 31 -
7.1 - Low Level Format.....	- 31 -
7.2 - LOCK.....	- 31 -
7.3 - UNLOCK .....	- 32 -
7.4 - MAP .....	- 32 -
 APPENDICES.....	 - 33 -
A. PROBLEMS.....	- 33 -
B. SCSI Message codes.....	- 35 -
C. SCSI Target STATUS (HDD) .....	- 35 -
D. SCSI Sense Keys .....	- 35 -
E. SCSI-Sense-codes .....	- 35 -



## **CHAPTER 1 - GENERAL SCSI INFORMATION**

### **1.1 - What is SCSI?**

It is not very easy to give a clear explanation of SCSI. SCSI is an American standard from the middle eighties. It is short for **S**mall **C**omputers **S**ystem **I**nterface. During the first few years the standard was a terrible mess. In 1984 the standard was almost completed, but never published as the standard. So every producer made his own definition of the standard. Nothing was such incompatible as the SCSI standard then. In spite of this incompatibility SCSI had many advantages. In network systems (IBM compatible PCs, Apple) the SCSI was able to control hard disks with great capacities, like one gigabyte. Later the compatibility between SCSI improved.

SCSI was originally developed as the SASI-bus (Shugart Associated System Interface). It was meant to be the first standard with interface on both sides of the device. Earlier this kind of interfaces were too expensive. But with VLSI (very large scale integrated) techniques it was affordable for almost everyone. SASI was offered to the ANSI committee as a standard. This eventually resulted in the SCSI standard.

SCSI is a kind of bus where you could connect up to 7 devices. This could be all kinds of devices: CD-ROM, SCSI interface, hard disk, etc. SCSI has its own protection scheme to prevent data collisions on the bus. Thus, offering a simple network environment.

### **1.2 - What about SCSI on MSX?**

SCSI on an MSX computer is nothing more than a way to work with hard disks. Although some interfaces in Holland were also able to control floppy drives (the so-called ED-drives) this option is not used anymore. The ED-drives are too expensive.

New device types like tapestreamers, scanners, etc. are not really interesting for MSX users. Prices are high and there is almost no need for these devices. CD-ROM is an option that would be interesting on MSX, even though there are some problems. Most CD-ROMs use the ISO-9660 standard. This is a standard with a 16 bit FAT. Sadly enough the 12 bit FAT is used on MSX. Another problem is the partition size. With MSX-DOS 2.x, we are restricted to the 32MB partition size. On some CD-ROMs hundreds of megabytes are collected in just one partition.

What we needed was a new MSX-DOS. Yet with ASCII quitting development of MSX software this is not a real option. We will keep working on a way to use the CD-ROM even on MSX as a normal hard disk.

Although it seems that a CD-ROM on your MSX is useless, it is not. Henrik Gilvad wrote some pleasing utilities to copy and type files from CD-ROM. It is not quite the way we wanted it to be, but at least we now can use some of the numerous CD-ROMs available. The utilities are on the disk supplied with your interface.

With this new ROM we decided to publish full documentation. This would enable programmers to write their own software for the SCSI interface. This information is on the disk. We would like to receive your programs, for as much as they are public domain or freeware. We added some utilities ourselves, like the ScanID and FrmtDsk utilities. A small example is included in chapter 6 of this manual.

### **1.2.1 - Hardware requirements**

MSX2

- Dos 2 kernel
- One free cartridgeslot

Turbo R

- One free cartridgeslot

## CHAPTER 2 - HARDWARE

### 2.1 - What kind of SCSI harddisk to buy

Maybe you do not own a hard disk yet. Well, just read this chapter and you will get an idea what to keep in mind when buying one.

If you buy a hard disk it is not likely it is built in a case. The first thing you need is a good case, with a built in power supply or a separate one. Make sure the power supply is able to supply the power needed by the hard disk. On PC fairs in Holland great cases from IBM are available, all with sufficient power supply built in. These cases are about 25,= to 50,= Dutch guilders.

When you have bought a case you have to find a hard disk. What capacity do you need ? Will it be 850 MB or 20 MB ? This depends on the way you are using your hard disk. Is it just for fun (games) or are you an enthusiastic collector of GIF and music files? When it is just for fun a 100 MB will be more than enough. In the other case you probably need a hard disk with the maximum our interface could access, 480 MB. This 480 MB is only available with the MAP-program. When not using the MAP-program the maximum access quantity is 192 MB. But even this will be sufficient for most users.

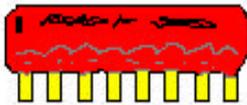
Another problem is the kind of SCSI hard disk. Do you need SCSI-1, SCSI-2 or wide SCSI ? It is easiest to say only SCSI-1 and SCSI-2 drives are to be connected to our interface. This not 100% true, but these drives will always work and for others nothing can be guaranteed. Please do not buy the 5.25" SCSI-1 hard disks. They are very old and very likely to give problems, like disk-errors, refusing to start on MSX2 computers upgraded to 7 Megahertz, etc.

Right now you have a case, a hard disk and an interface. It seems you are ready to connect all of them together. This is not true. You are missing one important part of your SCSI system: the cable to connect interface and hard disk. You can make it yourself , just buy 100 cm of 50-pin cable and two connectors, and place the connectors on both ends of the cable. Only do this if you know what you are doing, a cable with the connectors upside down, or whatever else can go wrong, may damage both your hard disk and interface.

From the above we extract the following: do not behave like a fool ! Just spend those few dollars more after you have spent a few hundred dollars for a hard disk and an interface. Every computershop will supply a SCSI cable if you ask for it.

## 2.2 - Connecting hard disk and SCSI interface

The SCSI cable is actually a SCSI bus connecting all devices on the cable. For a correct working both ends of the SCSI cable should be terminated. This is done with the so-called terminators. All SCSI devices have terminators on it. These look something like this:



Mostly they are removable. Our interface, which is usually not used in a network environment, does not have removable terminators. If you ask for it, we will make them removable before deliverance.

Remember, only the devices on the beginning and on the end of the cable should have these terminators. From all the other devices, if available, the terminators should be removed. Write down the original position before you remove them. Mark which terminator was mounted where and how. This will prevent you from troubles when relocating the terminators.

## **CHAPTER 3 - NOVAXIS SCSI ROM**

### **3.1 - What is new?**

Although the ROM has changed a lot internally, the outside has not undergone extreme changes. The only thing noticeable for the user is the new option: Extended Partitions.

The ROM will recognize (almost) all partitiontables used on MSX. The extended partition table is MS-DOS compatible. With the MAP-command, 480 MB per hard disk is made available to MSX. This feature will be explained later.

Inside, the ROM is made even more stable than its predecessor. CD-ROM devices are recognized more easily and a RESCAN option is available in the Setup-menu if a CD-ROM is not recognized on start-up time.

Further we deliver a few handy utilities on disk and some new calls in ROM.

If you have never used a NOVAXIS ROM before the changes are a lot more:

- Built-in BIOS Setup
- Software adjustable Host-ID
- Software adjustable Target-ID (hard disk from which to boot)
- Recognizing four different types of partitiontables, including the new extended partition table
- Breaking the 192 MB barrier with MAP-command and extended partitions
- Enabling use of multiple hard disks in an easy way

### **3.2 - Getting Started with NOVAXIS ROM**

To install the SCSI interface turn off the power of your system and external equipment. Now insert the SCSI interface cartridge in any unused slot. Connect the cable to the cartridge connector and the SCSI peripherals if you did not do this already.

The NOVAXIS SCSI BIOS supports SCSI hard disks (and some Magnetic Optical disks). Other devices should be managed by external programs (like we did for the CD-ROM) or drivers made especially for this device (none available at the moment).

Booting is only possible from a hard disk. Be sure the (first) hard disk's ID is between 0 and 3 if you want to boot from it.

After you have properly connected all devices you can turn on the power. The computer should be turned on first. Just after you turn on your computer you turn on the other SCSI

equipment. Some hard disks take some time to spin up. If you do have such a hard disk just reset the computer once to give the hard disk some more time.

The SCSI BIOS appears to be installed if you see this message appear on the screen:

```
NOVAXIS MSX2/Turbo-R SCSI BIOS version 1.51
(c) 1994-1995 KMcs / MSX Club Gouda
Written by Jurgen Kramer - July 3rd. 1995 -

Hardware by H.G. 1993. version 1.1
```

Then if your hardware is checked ok, the following text should be displayed:

```
Hit <DEL> to run SETUP
```

By pressing <DEL> you enter the SETUP menu, in which you can configure your system. The SETUP menu is described in paragraph 3.3. When everything works properly the following message should appear on the screen:

```
please wait ---' (Here the system waits for the Target which you selected in the
                  SETUP menu)
```

When the target is connected and ready a message similar to this one below appears on the screen:

```
Host ID: 7, Target ID: 0
```

Now the SCSI BIOS will search for a valid partition table on the first Target, and installs this Target if it contains a valid -MSX- partition. If the Multiple HDD switch in the SETUP menu is on it will continue to search for other Targets. After the first Target is checked, the other Targets at ID 0 up to 7 will be checked accordingly to the following sequence:

#### **Example 1 - Target ID is 0, Host ID is 7**

```
0 1 2 3 4 5 6 (The Host ID is skipped)
```

#### **Example 2 - Target ID is 2, Host ID is 6**

```
2 0 1 3 4 5 7 (The Host ID is skipped)
```

All installed Targets will be displayed on the screen, for example:

ID#0 - OLIVETTI	CP3200-200mb-3.5	Revision : 4041 3
ID#2 - IBM	7910A06	Revision : 01 1
ID#3 - QUANTUM	ELS42S	Revision : 4.07 N
ID#4 - IBM KZ-P	Revision : 1.60	E
ID#5 - CDROM	Compaq A1V23	Revision : 1.1c -

The number of partitions used on this device is shown as the last argument. At max. all partitions summed will be 6. The '-' sign means partitions on these devices are not supported by the SCSI ROM, 'E' means an Error in the partition table and 'N' means No(ne) partitions found.

The system will boot from the first Target and it tries to start MSX-DOS 2.x.

When your hard disk is not partitionized properly you should use the NFDISK program to partitionize the hard disk. Mostly you will run into errors like "Drive not ready", "Wrong driveletter" when a hard disk is not partitionized properly. For information on how to partitionize you hard disk see chapter 4.

Pressing <GRAPH> will prevent the NOVAXIS SCSI BIOS from being installed.

### 3.3 - Setup Utility

When starting with the interface connected to the computer and the hard disk to the interface the following message is printed on the screen:

```
NOVAXIS MSX2/Turbo-R SCSI BIOS version 1.51
(c) 1994-1995 KMcs / MSX Club Gouda
Written by Jurgen Kramer - July 3rd. 1995 -

Hardware by H.G. 1993. version 1.1

Hit <DEL> to run SETUP
```

By pressing the del-key you will get the following (PC BIOS SETUP-like) inputscreen.

NOVAXIS Setup version 1.06 (c) 1994-1995 KMcs  
Written by Jurgen Kramer - 14/05/95

```
[H] - Host ID           : 7           [T] - Target ID           : 0
[M] - Multiple HDD support : Disabled
[E] - Extended Partitions : Disabled
                                [R] - Rescan

[Q] - Quit without save
                                [S] - Save and quit

Units online : 1                               ID #: 0 1 2 3 4 5 6 7

Current Target: IBM           KZ-P           Device type: Direct Access Device

Input : _
```

### 3.3.1 - Host ID

The Host ID is the ID of your interface. Usually it will be 7, but if you are working in a network environment where all IDs have to be different it might be in the range from 4 to 7.

### 3.3.2 - Target ID

The Target ID is the ID of the hard disk from which to boot. Normally with one hard disk attached this will be 0, but if you are working in a multiple hard disk environment the Target ID might be in the range from 0 to 3.

There is one more thing. If you are working in a multiple hard disk environment the way your drive letters (A-F) are filled with partitions is a bit different. This is described in chapter 3.2 (getting started with NOVAXIS ROM).

### 3.3.3 - Multiple HDD support

Usually, this option is disabled, meaning just one hard disk (or other SCSI device) is supported. Yet with prices of hard disk dropping almost every day, it is not very rare to have more than one hard disk. For example one solely on MSX and one shared by the MSX and a PC. This option enables the support for more than one hard disk (or other SCSI device) on MSX.

### 3.3.4 - Extended partitions

The restricted number of only 6 partitions was a great restriction on hard disk size. With only 32 MB per partition it gives a maximum hard disk size of 192 MB.

Even on hard disks with hundreds of MBs more.

Extended partitions are the answer to this problem. With extended partitions you can make up to 15 partitions per hard disk. Thus, making it possible to use hard disks of 480 MB. You will need one external program, MAP, to use that many partitions and megabytes. MAP is described in chapter 7.

### **3.3.5 - Rescan**

Sometimes a hard disk or CD-ROM is not recognized immediately by the ROM, for example when you turn on the CD-ROM right after you turn on your computer. The rescan option makes the interface search again for all connected devices, and if connected it will be shown on the screen.

### **3.3.6 - Quit without save / Save and quit**

This BIOS setup program stores its information in the clockchip of your MSX. With the save and quit option you save the information to the clockchip. Thus, enabling to read it the next time on startup. Meaning you set up your BIOS once and then just keep using the installed settings. However, sometimes the bits the BIOS uses are also used by other programs. This will result in a mess. Keep this in mind when you have trouble booting.

Save and quit will save the changes you made in the clockchip. Quit without save does not.

### **3.3.7 - Units online**

The number of devices attached to your interface. The IDs these devices have will be shown in another color in the 'ID # : 0 1 2 3 4 5 6 7' message. The devices attached are colored yellow.

### **3.3.8 - Current Target**

This shows the status of the Target-ID. This could be online and offline. By changing the Target-ID, you have a way to check the status of all attached devices.

### **3.3.9 - Device type**

There are a few different SCSI device types. This option shows what kind of device the current Target-ID is. Hard disks will be of type "direct access device". When you get the message "unknown device type" and you are certain you have a hard disk currently installed on this Target-ID, you should check the terminators.

### 3.4 - Extra programs built in

The NOVAXIS ROM has some nice little options built in. You will get a list of the build in programs by typing CALL INFO in basic. A list like the one below will appear on screen.

CALL statements available :

CALL INFO	Show this text
CALL SLOW	Z80 mode
CALL MEDIUM	R800 ROM mode
CALL FAST	R800 RAM mode
CALL MAP	Fix DOS2 error
CALL MAP2	ditto - version 2
CALL HOSTID(variable)	Get SCSI HOST ID
CALL TARGETID(variable)	Get SCSI TARGET ID
CALL SETHOSTID(var.)	Set SCSI HOST ID
CALL GETTARGETID(var.)	Set SCSI TARGET ID
CALL INQUIRY(var, str, str, str)	Get INQUIRY of specified Target

#### 3.4.1 - CALL SLOW/MEDIUM/FAST

These calls only work on Turbo R.

CALL SLOW	-	Activates Z80 mode.
CALL MEDIUM	-	Activates R800 ROM mode.
CALL FAST	-	Activates R800 DRAM mode.

#### 3.4.2 - CALL MAP/MAP2

CALL MAP	-	CALL MAP is a utility which alters the working of memory mapper routines under DOS 2.x. Altering these will make some programs, which normally do not work on DOS 2.x, running.
CALL MAP2	-	CALL MAP2 is a newer version of CALL MAP and should be used first. If CALL MAP2 is not working, try CALL MAP. Ease from Philips is one the programs which will work fine with CALL MAP2 being executed first.

### 3.4.3 - CALL HOST/TARGETID, SETHOST/TARGETID, INQUIRY, FORMAT

CALL HOSTID (variable)

After executing this call the variable is filled with the host ID number. The variable is allowed to be one of the following types: integer, single precision and double precision

Examples: 10 CALL HOSTID(C): PRINT C  
11 CALL HOSTID(D!): PRINT D!  
12 CALL HOSTID(ID%): PRINT ID%

CALL TARGETID (variable)

After executing this call the variable is filled with the target ID number. The variable is allowed to be one of the following types: integer, single precision and double precision

Examples: 10 CALL TARGETID(C): PRINT C  
11 CALL TARGETID(D!): PRINT D!  
12 CALL TARGETID(ID%): PRINT ID%

CALL SETHOSTID(x)

This call is used to set the host ID. The x is to be replaced with one of these:

- a numeric value (4-7)
- integer, single precision, double precision
- expression

Examples: 10 CALL SETHOSTID(4)  
11 I=4: CALL SETHOSTID(I)  
12 V=3: CALL SETHOSTID(7-V)

CALL SETTARGETID(x)

This call is used to set the target ID. The x is to be replaced with one of these:

- a numeric value (4-7)
- integer, single precision, double precision
- expression

Examples: 10 CALL SETTARGETID(4)  
11 I=4: CALL SETTARGETID(I)  
12 V=3: CALL SETTARGETID(7-V)

CALL INQUIRY

This call enables the BASIC programmer to get some basic information from the harddisk, like code of make and revision number.

Examples: 10 CALL INQUIRY(4,A\$, B\$, C\$)  
11 I=4: CALL INQUIRY(I,A\$, C\$, D\$)  
12 V=3: CALL INQUIRY(7-V, F\$, K\$, G\$)

A result could be: A\$ = "IBM", B\$= "KZ-P" and C\$="R160".

Meaning **IBM** is the maker of hard disk **KZ-P** and the revision is **R160**.

CALL FORMAT

Protects or enables writing to current partition.

Example: call format  
Drive name? (A,B,C,D,E,F) A  
  
1 - Write protect partition  
2 - Write enable partition  
  
? 1  
Strike a key when ready  
Not a DOS disk  
Ok

Now you are no longer able to write on drive A. Only resetting the computer or write enable the partition using call format will make it possible to write on that particular partition again. If a partition is write protected because of the owner ID of the partition is different from the current host ID, this call format function will not remove the write protection.

### **3.5 - Errors**

These are the messages generated by the ROM when something is wrong:

#### **Controller test failed**

Your eprom is unable to control the SCSI controller of your interface. Mostly indicating a defective controller. Sometimes the crystal is broken. This is a fatal error, and you should contact your supplier for repairing the interface.

#### **ROM checksum error**

A few bits in the ROM have fallen from 0 to 1 or vice versa and now your ROM is creating the wrong checksum. It is a fatal error and means you have to replace your ROM with a new one.

#### **SCSI bus is down**

This is a general error. It is not a fatal one. Mostly it indicates something is wrong with your connections. Is the SCSI cable ok ? What about the terminators, are they all in the right place ? Is your harddisk turned on yet ? Is it already at speed ?

## **CHAPTER 4 - NFDISK**

### **4.1 - What is NFDISK?**

NFDISK is a program that enables you to initialize your hard disk. It is different from formatting a disk, but you need to initialize your hard disk before it can be used. You can create different partitions on your hard disk. (In general, each partition is treated as a different drive by the computer.) Splitting the harddisk into partitions is mostly necessary, because the MSX cannot handle drives larger than 32 MB.

### **4.2 - Starting NFDISK**

NFDISK has to be started from MSX-DOS 2.x. If your harddisk is not initialized yet, start the computer while pressing the <GRAPH> key, so the harddisk is not seen by the computer on initialization. Then start MSX-DOS 2.x and NFDISK from a diskette.

If your hard disk has already been initialized but you want to take a look at NFDISK anyway, this can be done without any problem. Nothing on your hard disk will be changed - if you do not press the <W> key during execution of the program.

The program is easy to use, so you can start it without reading this manual first. Still, a reasonably complete description follows below.

### **4.3 - The beginning**

NFDISK starts with displaying a copyright-window and waits for a key to be pressed. What happens next depends on the fact of having one or more hard disks connected to your system. In the last case, you first get a window where you have to choose one of the hard disks you want to work with.

After that, the information of the hard disk is read and the edit-screen is displayed. If you have started the computer with the <GRAPH> key pressed, reading the hard disk may take some seconds.

### **4.4 - The edit-screen**

At this point, three windows are on the screen. Below, an overview of the keys and their functions is displayed. Above, on the left you can see some general information about the hard disk and on the right you can see the free space on the hard disk. Three values are displayed in megabytes.

The top value represents the space on the harddisk that is free for use by the partitions. The middle value represents the current space that is used by the partitions. On the bottom the unused space is displayed, which is the subtraction of the upper two values.

**Note:** the upper value is not equal to the total space on the harddisk! Some of the harddisk space is used for storing the partition-table. This space is not counted with the total free space.

The middle window contains the data of the partitions on the hard disk. There are six columns:

- Nr : The number of the partition.
- WID(Write-ID): The SCSI-ID number that can write on this partition. See paragraph 4.5
- Type : The partition type. Normally this will be 'FAT12 (MSX)'. This is the only type of partition that NFDISK can generate. If there are already other types of partitions on the hard disk, NFDISK will recognize them. These partitions cannot be edited, however.
- Sectors : De partition size in sectors
- MB : De partition size in megabytes
- Volume name : The volume name. This name is displayed when you give a DIR command in MSX-DOS 2.x.

The fields 'Nr' and 'Type' are only on the screen for your information. The other data can be edited.

The fields 'Sectors' and 'MB' are two different representations of the same value. If you change one of the values, the other value will change too.

#### 4.5 - The write-ID

The Write-ID is only important if there are multiple computers connected to a hard disk. If more computers can write on the same partition, things could go wrong.

Sometimes, a computer stores part of the information that is on disk (especially the FAT and/or directory) in memory so it does not have to read this from disk all the time. If one computer keeps information about a partition in its memory and another computer writes something on the same partition, the information in the computer's memory is not correct anymore. When this computer writes something to that partition, and uses, the (incorrect) information from its memory, data on the disk can be damaged.

The SCSI ROM offers a solution by only having one ID reading and writing a certain partition. Other IDs only have read rights. Therefore, in this version of NFDISK it is only

possible to give one specific SCSI-ID write-access to a partition. This 'write-ID' should be the same as the host-ID of the computer from which you want to write on the partition.

The write-ID is only supported by two of the four different partitionable-formats. If you have selected a format where the write-ID is not supported, the write-ID cannot be set and dashes are displayed in this column.

#### 4.6 - Editing the partition data

You can change the current values by selecting these with the cursor keys and space bar, and entering a new value. While entering values you can always restore the old values by pressing <ESC>. If in a 'Sectors' or 'MB' field nothing is entered, but <RETURN> is pressed immediately, NFDISK will give a value that is equal to the remaining unused space on the hard disk. The maximum size of a partition is 65535 sectors, which is 32 MB. If the unused space is larger than 32MB, NFDISK will enter 65535 sectors. If the unused space is negative, NFDISK will restore the old value.

You can delete partitions from the table with <DEL>. The computer will first ask for a confirmation. You can insert new partitions with <INS>. The position where the partition is inserted depends on the position of the cursorbar. On inserting a partition, the size of the partition (in MB) has to be entered. Here, a value is also inserted automatically when <RETURN> is pressed immediately.

During the editing of the partition data, the changed data are not stored on the harddisk yet. The partition data are written when you press <W>. This version of NFDISK will initialise all partitions when writing the data. On one hand, this is convenient because the whole harddisk is ready for use with one keystroke. On the other hand, **you cannot change partition information afterwards because all data on harddisk will be lost while writing the partition data !**

#### 4.7 - Different partitionable-formats

Over the past years, different ways have been used to store the partition data on harddisk. NFDISK can read different types of partitionables, just like the new SCSI ROM, and can also write data in these different formats. By pressing <P> a window is displayed where you can choose one of the different table-formats.

An overview:

Type	no. of partitions	Used by interfacetype
(old) PC	1 - 4	MS-DOS PC interfaces, MK 1.x
(old) MSX	1 - 6	HSH, MAK 2.0, MK 2.x, Gouda/Henrik Gilvad with other ROM than Novaxis 1.5, HandlerGemeinschaft
MAK 3.0	1 - 8	MAK 3.0
Extended	1 - 32 1 - 15 (Novaxis)	MS-DOS PC interfaces, HPN/B.E.R.T., Gouda/Henrik Gilvad with Novaxis 1.5 ROM

**Note** : Another typical thing of the extended format is that for each partition one sector of the harddisk (of 512 bytes) is used to store the partition data.

The other formats use one sector, regardless the number of partitions. This means that the free space on the harddisk, as it is displayed on the top right, can change when you change the partitionable-format!

#### 4.8 - So now what ?

When you have changed and written the partition data, the harddisk is fully initialised and ready for use. You have to restart the computer at this point. Now suppose you have made three partitions on the harddisk, then the drives A: to C: are most probably used by the harddisk and drive D: will be the diskette drive. The computer will start MSX-BASIC, as it does not find anything on the A: drive.

Now you will probably want to put MSX-DOS on the A: drive, so the computer starts MSX-DOS at startup. This can be done from BASIC, or you can start MSX-DOS from disk by giving the right CALL CHDRV ("x:") and a CALL SYSTEM.

Be sure to copy both MSXDOS2.SYS and COMMAND2.COM onto drive A:! When MSXDOS2.SYS is present and COMMAND2.COM is absent or corrupted, this is fatal and you have to re-initialise your harddisk! (Or find a good program(mer) to resolve this situation.)

## 4.9 - Problems

- PCs do not fully recognise extended partitions. Depending on the operating system, only two or three partitions are recognised.
- Some PC programs may have problems with the bootsector that is generated by NFDISK v1.0. To fix this, the number on position 015h of the bootsector (which is the Media ID) should be changed from 0F0h to 0F8h. This does not work with NOVAXIS SCSI ROM v1.0. Only higher versions work with Media ID 0F8h.
- Although under normal circumstances this will never happen, it is possible that the partition table or a bootsector on the hard disk contains invalid values. This is not checked by NFDISK v1.0, which can have two effects:
  - At the beginning, NFDISK tries to read a sector that does not exist. This causes a 'Read Error' which cannot be fixed in NFDISK.
  - At writing the partition, a far too large number of FAT or directory sectors is initialised. The computer will be busy writing very long and will probably give a 'Write Error' in the end. This can be fixed by deleting the partition with the invalid bootsector values and creating the same one again.

## 4.10 - Remarks

NFDISK will be under development for a long time, but even this 1.0 version is a lot better than previous FDISK programs.

NFDISK v1.0 is made by: Roderik Muijt and Arjan Mels

Thanks go to :

- Jurgen Kramer (maker of NOVAXIS SCSI ROM) for cooperation and suggestions
- Gert de Boom, Jan Botman and Arjan Prosman for support and supplying hardware
- Jeffrey Timmer for making the window routines that are used in NFDISK

NFDISK v1.0 is copyrighted by MSX Club Gouda. NFDISK v1.0 is made only for spreading with the NOVAXIS SCSI ROMs and only works with these ROMs. For future versions, other copyrights and spreading rules may apply.

When new versions of NFDISK are finished, they will probably be spread freely. For questions about this, contact Roderik Muijt. This can be done best by writing e-mail to <roderik@ripe.net>. Unfortunately it is impossible at this moment to supply addresses or phone numbers that will be valid for a longer period. If you do not have e-mail access, then contact MSX Club Gouda for the right address.

**INTENTIONALLY LEFT BLANK**

## CHAPTER 5 - NETWORKING

### 5.1 - Hardware

Hardware connections in a network are not very different from the standard situation with one hard disk and one interface. At first you have to make sure that all hard disks and interfaces have different IDs. Interfaces are usually numbered from 7 down to 4 and hard disks are numbered from 0 up to 3.

In a network you have to remove the terminators from some of the devices. Only the device on the beginning of the cable and the device on the end of the cable must have terminators installed.

If your cable does not have enough connectors, just put on a few more. Look at the lips on the connectors. The new ones should have the same direction as the old ones. Further you have to keep in mind the maximum length of a SCSI cable, which is 6 meters.

Not keeping to the rules above means trouble. Network errors, read and write errors, hardware errors, etc.

### 5.2 - Setting up hosts and targets

Make sure all SCSI devices have different IDs. Interface have IDs from 7 to 4, hard disks have IDs from 0 to 3.

### 5.3 - Adding an MS-DOS PC to the network

With NOVAXIS' 'network' facilities it is possible to connect your MSX to an IBM compatible PC sharing SCSI devices. This PC should of course be equipped with a SCSI HOST adapter (e.g. from Adaptec). Most PC host adapters have an external SCSI connector, which can be used to connect the shared hard disk to it. Now you can use NFDISK (on the MSX) to create an extended or (old) PC partition on the hard disk.

The (old) MSX partitionable can also be used, but keep in mind these things:

- it might be possible that the PC's bootprogram is destroyed
- only partitions 3 to 6 are seen by the PC

The PC mostly uses a 16-bit FAT, and your MSX only uses a 12-bit FAT, so the PC will be able to use your MSX partition, but you can not use PC partitions.

When you share a PC drive with a MSX, you should **disable** all diskcaching on that

particular drive because your PC will not know that you have modified some data with your MSX and still displays that directory you have just deleted. Do not use compressing software ! Tools like neverending disk, doublespace, drvspace, stacker, lzexe and all these kind of compressing software are not allowed if you want your data to be consistent.

## CHAPTER 6 - WRITING SOFTWARE

### 6.1 - BIOS CALLS

For the programmers we have published all the BIOS calls in the NOVAXIS SCSI BIOS. This is the complete list.

Register A is mostly used as follows:

```
A = LLL00III
   |||  +++----- Target ID
   +++----- Logical Unit Number (LUN) usually 0
```

New routines, which are new regarding to the first SCSI BIOS made by Uwe Schroeder, are marked with '1.00' or '1.50', indicating the first NOVAXIS ROM in which they were implemented.

Sometimes these calls do not give enough information. The following documents, which are on the supplied disk, should be examined for more sophisticated information about how some SCSI commands should be used, ANSI document X3.131-199X (SCSI-2) or X3-131-1986 (SCSI-1) .

At address 07f80h the SCSI BIOS ROM ID is found: HD# (in ASCII). Checking this ID is not enough for finding out if the BIOS is made by NOVAXIS. For that you need a find-routine which is on the disk supplied with your interface.

And here is the list with calls:

Name : **SetWD3393**  
Address : 07f83h  
Descript. : Initialise SCSI controller  
Input : -  
Output : -  
Modify : All  
Note : -

Name : **TermAct**  
Address : 07f86h  
Descript. : Terminate HDD actions  
Input : -  
Output : A = SCSI STATUS  
D = Target Status  
E = Message  
Modify : AF, D, E  
Note : -

Name : **RdLogBlk**  
Address : 07f89h  
Descript. : READ logical block(s) from HDD  
Input : A = LUN + SCSI Target ID  
B = # of blocks to read  
C D E = 21 bit logical block #  
HL = transfer address  
Output : A = SCSI STATUS  
D = Target Status  
E = Message  
Modify : AF, DE, IX, IY, HL  
Note : See ANSI 8.2.5

**Name** : **WrLogBlk**  
**Address** : (07f8ch)  
**Descript.** : WRITE logical block(s) to HDD  
**Input** : A = LUN + SCSI Target ID  
           B = number of blocks to write  
           C D E = start logical block #  
           HL = transfer address  
**Output** : A = SCSI STATUS  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,DE,HL,IX  
**Note** : See ANSI 8.2.5

**Name** : **ReqSense**  
**Address** : 07f8fh  
**Descript.** : REQUEST SENSE  
**Input** : A = LUN + SCSI Target ID  
**Output** : A = DOS error code  
           IX = address of sectorbuffer filled  
               with Sense Data  
**Modify** : AF,BC,DE,  
**Note** : Sense data will be delivered in  
        Extended Sense Format (12h bytes) .  
        See ANSI 7.2.14

**Name** : **Inquiry**  
**Address** : 07f92h  
**Descript.** : INQUIRY  
**Input** : A = LUN + SCSI Target ID  
           HL = transfer address  
**Output** : A = SCSI STATUS  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,DE,HL  
**Note** : Returns 24h bytes of information, see  
        ANSI 7.2.5

**Name** : **ReadCap**  
**Address** : 07f95h  
**Descript.** : READ CAPACITY  
**Input** : A = LUN + SCSI Target ID  
           HL = transfer address  
**Output** : A = SCSI STATUS  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,DE,HL  
**Note** : Returns 8 bytes, see ANSI 8.2.7

**Name** : **ModeSense**  
**Address** : 07f98h  
**Descript.** : MODE SENSE  
**Input** : A = LUN + SCSI Target ID  
           HL = transfer address  
           B = PPCCCCC  
               ||+++++-- Page Code  
               +----- Page Control Field  
**Output** : A = SCSI STATUS  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,DE,HL  
**Note** : See ANSI 7.2.10

**Name** : **ModeSel**  
**Address** : 07f9bh  
**Descript.** : MODE SELECT  
**Input** : A = LUN + SCSI Target ID  
           B = Parameter List length  
           HL = transfer address  
**Output** : A = SCSI Target ID  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,HL,IX  
**Note** : See ANSI 7.2.8

**Name** : **FmtUnit**  
**Address** : 07f9eh  
**Descript.** : FORMAT UNIT  
**Input** : A = LUN + SCSI Target ID  
           B = 000FCDDD  
               ||+++---- Defectlist format  
               |+----- Complete  
               +----- Format data  
               DE = interleave (msb - lsb)  
               HL = data address  
**Output** : A = SCSI Status  
           D = Target Status  
           E = Message  
**Modify** : AF,BC,DE,HL  
**Note** : See ANSI 8.2.1

**Name** : **TstUnitRdy**  
**Address** : 07fa1h  
**Descript.** : TEST UNIT READY  
**Input** : A = LUN + SCSI Target ID  
**Output** : A = SCSI STATUS  
           D = Target Status  
           E = Message  
**Modify** : AF,DE  
**Note** : See ANSI 7.2.16

**Name** : **Initialise**  
**Address** : 07fa4h  
**Descript.** : Print Init Text  
**Input** : ?  
**Output** : ?  
**Modify** : ?  
**Note** : Do not use this call

**Name** : **InsWork**  
**Address** : 07fa7h  
**Descript.** : Install Workspace  
**Input** : ?  
**Output** : ?  
**Modify** : ?  
**Note** : Do not use this call

**Name** : **ClrEndLn**  
**Address** : 07faah  
**Descript.** : Clear To End Of Line (print escape

sequence)  
 Input : ?  
 Output : ?  
 Name : **Verify**  
 Address : 07fadh  
 Descript : VERIFY  
 Input : A = LUN + ID  
         B = Verification Length (Blocks)  
         C D E = 21 bit Logical Block #  
         HL = data address  
 Output : A = SCSI Status  
         D = Target Status  
         E = Message  
 Modify : AF,BC,HL,IX  
 Note : -

Name : **StrtStpUn**  
 Address : 07fb0h  
 Descript. : START STOP UNIT  
 Input : A = LUN + ID  
         B = 0 > Stop Unit  
         B = 1 > Start Unit  
 Output : A = SCSI Status  
         D = Target Status  
         E = Message  
 Modify : ?  
 Note : See ANSI 8.2.17

Name : **SndDiag**  
 Address : 07fb3h  
 Descript. : SEND DIAGNOSTIC  
 Input : A = LUN + ID  
 Output : A = SCSI status  
         D = Target Status  
         E = Message  
 Modify : ?  
 Note : On MSX this function always performs a Self Test, see ANSI 7.2.1

Name : **Reserved**  
 Address : (07fb6h)

Name : **Reserved**  
 Address : (07fb9h)

Name : **Copy (1.03)**  
 Address : 07fbch  
 Descript. : Copy  
 Input : A = LUN + ID  
         DE = Parameter List Length  
         HL = Data addresss  
 Output : A = SCSI Status  
         D = Target Status  
         E = Message  
 Modify : ?  
 Note : See ANSI 7.2.3

Modify : ?  
 Note : Do not use this call

Name : **RdDefect (1.00)**  
 Address : 07fbfh  
 Descript. : Read Defect Data  
 Input : A = LUN + ID  
         B = 000PGDLF  
               | | +++- Defect List Format  
               | +----- Glist  
               +----- Plist  
         DE = Allocation length  
         HL = Data addresss  
 Output : A = SCSI Status  
         D = Target Status  
         E = Message  
 Modify : AF,BC,DE,HL,IX,IY  
 Note : See ANSI 8.2.8

Name : **GetWrk (1.00)**  
 Address : 07fc2h  
 Descript. : Get Work Area  
 Input : -  
 Output : HL = IX = Start of work area  
 Modify : AF,BC,HL,IX  
 Note : -

Name : **PartInfo (1.00)**  
 Address : 07fc5h  
 Descript. : Get Partition Info  
 Input : A = drive #  
 Output : HL = IX = start of work area desired drive  
 Modify : AF,BC,DE,HL,IX  
 Note : -

Name : **GetUnitsOn (1.00)**  
 Address : 07fc8h  
 Descript : Get # of Units Online  
 Input : -  
 Output : A = # of Units online  
         C = ID vector  
         D = HOST ID  
 Modify : AF,BC,DE  
 Note : -

Name : **SetHOST\_ID (1.00)**  
 Address : 07fcbh  
 Descript : Set HOST ID (4-7)  
 Input : A = HOST ID (4-7)  
 Output : Carry is set when error occurred  
 Modify : AF,D  
 Note : -

Name : **SetTargetID (1.00)**  
 Address : 07fceb  
 Descript. : Set Target ID (0-3)  
 Input : A = Target ID (0-3)  
 Output : Carry is set when error occurred

Name : **GetTargetID (1.00)**  
 Address : 07fd1h  
 Descript. : Get Target ID  
 Input : -  
 Output : A = Target ID  
 Modify : AF  
 Note : -

Modify : AF,B,D  
 Note : -

Name : **GetHOST\_ID (1.00)**  
 Address : 07fd4h  
 Descript. : Get HOST ID  
 Input : -  
 Output : A = HOST ID  
 Modify : AF  
 Note : -

Name : **GetSense (1.00)**  
 Address : 07fd7h  
 Descript. : Get Sense data  
 Input : A = LUN + ID  
 Output : A = Sense Key  
           B = Sense Code  
           C = Additional Sense Code  
           D = Target status  
           IX = Sense data address

Modify : AF,BC,DE  
 Note : This does the same as the ReqSense call except all necessary information is loaded into registers.

Name : **MedRemoval (1.09.03)**  
 Address : 07fdah  
 Descript. : Prevent Allow Medium Removal  
 Input : A = LUN + ID  
           B = 0, allow removal  
           B = 1, prevent removal

Output : A = SCSI Status  
           D = Target status  
           E = Target message

Modify : ?  
 Note : See ANSI 8.2.4

## 6.2 - Additional NOVAXIS info

Address	Description
07ff0h	NOVAXIS ROM ID (ASCII) - 'KMcs'
07ff4h	ROM type (byte) 0 = MSX2, 1 = Turbo-R, 2 = MSX2/Turbo-R
07ff5h	Version # High (BCD) - currently 1
07ff6h	Version # Low (BCD) - currently 51
07ff7h	Default HOST ID (byte) b0h = 7, b1h = 6, b2h = 5, b3h = 4
07ff8h	Base I/O (byte) - 1Ch, 24h, 34h, etc.
07ff9h - 07fffh	Reserved for future use

## 6.3 - Disk Parameter Block

+00h	- Drive #
+01h	- Media ID
+02h / 03h	- Bytes / sector
+04h	- Directory mask
+05h	- Directory shift
+06h	- Cluster mask
+07h	- Cluster shift
+08h / 09h	- First FAT sector
+0ah	- Number of FATs

+0bh            - Number of directory entries  
+0ch / 0dh     - First data sector  
+0eh / 0fh     - Number of clusters + 1  
+10h           - Sectors / FAT  
+11h / 12h     - First directory sector  
+13h / 14h     - FAT-address (not used)

## 6.4 - Layout of Work Area

The work area consists of 6\*8 bytes, which means 8 bytes for each logical drive.

Entry:

```
+00    P00FXTAR
      |  ||+++----- Target ID
      |  |+----- normaly 1, can be reset after DSKIO
      |  +----- Formatted, 1 = partition not (MSX) formatted
      +----- Partition Enable, 1 = enabled

+01-03 First sector in partition
+04-05 Number of sectors in partition

+06    W0000NNN
      |    +++----- Network Write ID
      +----- Write Protected, 0 = write protected

+07    F00000HM
      |    |+----- MSX type, 0 = MSX2/2+, 1 = Turbo-R
      |    +----- Multiple HDD support, 1 = on
      +----- Fast RAM transfer (currently not used)
```

## 6.5 - An example in BASIC

```
10 ' Only for use with Uwe Schroeder like BIOS
20 '                               and SCSI interface being the first diskdriver
30 ' The routine checks for "D" at #7F81 only
40 '
50 ' It will stop and start the motor of your harddisk
60 ' This stop/start sequence is not working on all harddisks
70 '
80 ' Henrik Gilvad
90 '
100 CLEAR200,&HD000
110 DEFINT A-Z
120 A=&HD000:DEFUSR=A
130 READA$:IFA$<>"$"THENPOKEA,VAL("&H"+A$):A=A+1:GOTO130
140 PRINT"Now turning the motor OFF"
150 A=USR(0)
160 PRINT"Press any key to turn it ON again."
170 Z$=INPUT$(1)
180 A=USR(1)
190 PRINT"That was all.":END
200 DATA 3A,22,FB,21,81,7f,CD,0C,00,FE,44,c0
210 DATA 3a,22,fb,26,40,CD,24,00,3A,F8,F7,47
220 DATA 3E,08,CD,B0,7F,AF,26,40,C3,24,00,$
```

## 6.6 - An example in ML

This example reads a sector. It does this only when the interface is in slot 1-0. If you want to have this demo program runned on all computers, you need to add a search-routine like the one on the disk supplied with your interface

```
; EXAMPLE.GEN - Example of programming NOVAXIS SCSI BIOS
; Written by Jurgen Kramer - 05/06/95 -
; This program is merely a example how to use the NOVAXIS SCSI calls

ENASLT    equ    0024h
EXPSLT    equ    0fcc1h

        org 0c000h

; use SRCHROM.GEN and CHKVER.GEN to get the right slot ID and NOVAXIS
; version number or see NOVAUTIL.MAC for an complete example.
; Assume NOVAXIS SCSI BIOS is installed in SLOT 1-0

begin:   ld      a,1                ; Slot ID
         ld      h,40h             ; ROM starts at 4000h
         call    ENASLT           ; Enable ROM

; This example reads logical blocks from SCSI device with ID 0
         ld      a,0                ; SCSI ID 0, LUN 0
         ld      c,0
         ld      de,0              ; Logical block # 0
         ld      b,1              ; Read 1 block
         ld      hl,0c100h        ; Transfer address
         call    07f89h           ; RdLogBlk
         inc     d
         dec     d                ; Status GOOD ?
         jr     nz,RdError

Exit:   ld      a,(EXPSLT)         ; Get slot ID of BASIC
         ld      h,40h
         call    ENASLT
         ret

; An error occurred, find out what's wrong
RdError:
         ld      a,0                ; SCSI ID 0, LUN 0
         call    07fd7h           ; Request Sense
         inc     d
         dec     d                ; Status GOOD ?
         jr     nz,Exit           ; Device is not accessable

; Now register A contains SCSI Sense Key
         and     a                ; No Sense?
         jr     z,Exit
```

```

cp 2 ; Not Ready?
jr z,Exit
cp 6 ; Unit Attention?
jr nz,Exit

; Register B contains the Sense Code
ld a,b ; Power on?
jr nz,Exit

; Register C contains the Additional Sense Code
ld a,c
and a
jr nz,Exit

; Now you could retry the read operation
jr Exit ; In this example just exit

```

## 6.7 - An example in ASCII-C (version 1.2)

Because ASCII-C is not used by many people we have decided to put this example only on the disk. You can find it in the directory \HDDINFO.

## CHAPTER 7 - UTILITIES

### 7.1 - Low Level Format

One utility on the disk is the Low Level Formatter (NFRMDSK). It will look like this:

```
NOVAXIS SCSI HDD Low-Level Formatter version 1.04
(c) 1994-1995 KMcs
Written by Jurgen Kramer - 06/04/95 -

Number of Units online : 1          HOST ID : 7          Taget ID : 0
-----

SCSI ID #0 - IBM

                !! please wait !! please wait !!
                Depending on your disk capacity,
                Formatting may take one minute
                to several hours
                !! please wait !! please wait !!

-----

[ESC] - Quit to DOS    [ENTER] - format a drive

Busy formatting...
```

This utility is to be used to format hard disks. Usually you will never need it, but sometimes it might be usefull and even necessary to get a hard disk working.

Not all hard disks will support the format-routine this program calls.

### 7.2 - LOCK

A utility designed to prevent ejecting a disc form a CD-ROM or other removable-media drive by pushing the drive's front panel button. **Note** : not all removable-media drives do support this option.

Example :

```
A:\LOCK 3
```

This will lock the device with ID 3.

### 7.3 - UNLOCK

A utility which is the opposite of LOCK; it “unlocks” a CD-ROM or other removable-media drive, permitting the disc to eject by pushing the drive’s front panel button.

Example :

```
A:\UNLOCK 3
```

This will unlock the device with ID 3.

### 7.4 - MAP

The utility to map a certain partition on a certain hard disk to a certain logical drive making it possible to break the 192 MB border.

Example:

```
A:\MAP A 1 4
```

This will MAP partition 4 on the hard disk with ID1 to drive A. Drive can not be one of the floppydrives.

A list of all partitions on a certain hard disk can be obtained as follows:

```
A:\MAP -L 2
```

This will give a list of all partitions on the hard disk with ID2. In case you do not have a hard disk with ID2 the utility will quit with the message “Invalid target ID”.

Typing MAP without argument gives some help information.

## **APPENDICES**

### **A. PROBLEMS**

#### **Problems with MSX2 machines running at 7 MHz.**

Some MSX2 computers which have been upgraded to 7 Megahertz will give problems if the 7 Megahertz is enabled and the SCSI interface is inserted. All kind of problems may appear: no boot, many errors, etc. This problem is likely to occur only when you have an old hard disk, an interface with NOVAXIS 1.5 ROM and 7 Megahertz enabled.

The only solution to the problem is to get a step back with one of the components. Not using 7 Megahertz is the easiest solution. Another solution is to buy another hard disk (newer) and the final solution is to keep working with 7 Megahertz and the old hard disk but now use NOVAXIS 1.0 ROM. This ROM is less sophisticated, but still very useful.

These problems mostly appear on Philips computers. The computers from Sony seem to have less problems.

#### **Problems with an RS232C interface**

MSX was never developed to work with speeds you are forcing it to work now. The SCSI interface is almost operating at your MSX top-speed. Connecting an RS232c interface which is mostly connected to some high-speed modem is asking too much from the computer. It will give transmission errors (either the SCSI interface, the RS232c interface or both) and sometimes will not operate at all.

Some people are lucky. They are working with this combination of interfaces and do not have problems at all. You cannot damage your computer trying it, but we can not give any guarantee it will work correctly.

**Hard disk starts normally and is initialized ok, but never goes to load MSX DOS 2.x. This mostly will happen after the drive is partitioned again. The answer is:**

Start with the hard disk attached and search which driveletter is the floppydrive. If you have made two partitions the driveletter will be "c:", if you made three partitions it will be "d:", etc. Type the command CALL CHDRV("floppydrive:") <return> in BASIC and copy the files COMMAND2.COM and MSXDOS2.SYS from the MSX DOS 2 bootfloppy to the A-partition of your hard disk. Now start again and you will see MSX DOS 2.x booting up..

**Even after trying the above solution my problems are not solved.**

This is a tough one. It could be a broken hard disk, but also a jumper problem. Both are equally difficult. A broken hard disk is mostly not repairable. Jumper problems could be fixed if you have the documentation by your hard disk.

**My set keeps showing the message "waiting.....`" and eventually is replaced by the message 'host interface selftest failed, scsi bus is down"**

This means one of the parts is not connected in the right way. Check everything, and try to start again.

## B. SCSI Message codes

00h	- COMMAND COMPLETE
01h,xx,00h	- MODIFY DATA POINTERS
* 01h,xx,01h	- SYNCHRONOUS DATA TRANSFER REQUEST
* 01h,xx,03h	- WIDE DATA TRANSFER REQUEST
02h	- SAVE DATA POINTERS
03h	- RESTORE POINTERS
04h	- DISCONNECT
05h	- INITIATOR DETECTED ERROR
06h	- ABORT
07h	- MESSAGE REJECT
08h	- NO OPERATION
09h	- MESSAGE PARITY ERROR
* 0ah	- LINKED COMMAND COMPLETE
* 0bh	- LINKED COMMAND COMPLETE (WITH FLAG)
0ch	- BUS DEVICE RESET
* 0dh	- ABORT TAG
* 0eh	- CLEAR QUEUE
* 0fh	- INITIATE RECOVERY
* 10h	- RELEASE RECOVERY
* 11h	- TERMINATE I/O PROCES
* 20h	- SIMPLE QUEUE TAG
* 21h	- HEAD OF QUEUE TAG
* 22h	- ORDERED QUEUE TAG
* 23h	- IGNORE WIDE RESIDUE
80h - 0ffh	- IDENTIFY

## C. SCSI Target STATUS (HDD)

00h	- GOOD
02h	- CHECK CONDITION
* 04h	- CONDITION MET
08h	- BUSY
* 10h	- INTERMEDIATE
* 14h	- INTERMEDIATE CONDITION MET
* 18h	- RESERVATION CONFLICT
* 22h	- COMMAND TERMINATED
* 28h	- QUEUE FULL

## D. SCSI Sense Keys

0h - NO SENSE  
 1h - RECOVERED ERROR  
 2h - NOT READY  
 3h - MEDIUM ERROR  
 4h - HARDWARE ERROR  
 5h - ILLEGAL REQUEST  
 6h - UNIT ATTENTION  
 7h - DATA PROTECT  
 8h - BLANK CHECK  
 Ah - COPY ABORTED  
 Bh - ABORTED COMMAND  
 Ch - EQUAL  
 Dh - VOLUME OVERFLOW  
 Eh - MISCOMPARE

## E. SCSI-Sense-codes

The first column shows the sensecode, the second the additional sensecode and the third a description by these sensecodes.

00	00	NO ADDITIONAL SENSE INFORMATION
00	01	FILEMARK DETECTED
00	02	END-OF-PARTITION / MEDIUM DETECTED
00	03	SETMARK DETECTED
00	04	BEGINNING-OF-PARTITION / MEDIUM DETECTED
00	05	END-OF-DATA DETECTED
00	06	I/O PROCESS TERMINATED
00	11	AUDIO PLAY OPERATION IN PROGRESS
00	12	AUDIO PLAY OPERATION PAUSED
00	13	AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED
00	14	AUDIO PLAY OPERATION STOPPED DUE TO ERROR
00	15	NO CURRENT AUDIO STATUS TO RETURN
01	00	INDEX / SECTOR SIGNAL

02	00	SEEK COMPLETE
03	00	PERIPHERAL DEVICE WRITE FAULT
03	01	NO WRITE CURRENT
03	02	EXCESSIVE WRITE ERRORS
04	00	LOGICAL UNIT NOT READY
04	01	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
04	02	LOGICAL UNIT NOT READY
04	03	LOGICAL UNIT NOT READY
04	04	LOGICAL UNIT NOT READY
05	00	LOGICAL UNIT DOES NOT RESPOND TO SELECTION
06	00	REFERENCE POSITION FOUND
07	00	MULTIPLE PERIPHERAL DEVICES SELECTED
08	00	LOGICAL UNIT COMMUNICATION FAILURE
08	02	LOGICAL UNIT COMMUNICATION PARITY ERROR

08	01	LOGICAL UNIT COMMUNICATION TIME-OUT
09	00	TRACK FOLLOWING ERROR
09	01	TRACKING SERVO FAILURE
09	02	FOCUS SERVO FAILURE
09	03	SPINDLE SERVO FAILURE
0A	00	ERROR LOG OVERFLOW
0C	00	WRITE ERROR
0C	01	WRITE ERROR RECOVERED WITH AUTO REALLOCATION
0C	02	WRITE ERROR - AUTO ALLOCATION FAILED
10	00	CRC OR ECC ERROR
11	00	UNRECOVERED READ ERROR
11	01	READ RETRIES EXHAUSTED
11	02	ERROR TOO LONG TO CORRECT
11	03	MULTIPLE READ ERRORS
11	04	UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED
11	05	L-EC UNCORRECTABLE ERROR
11	06	CIRC UNRECOVERED ERROR
11	07	DATA RESYNCHRONISATION ERROR
11	08	INCOMPLETE BLOCK READ
11	09	NO GAP FOUND
11	0A	MISCORRECTED ERROR
11	0B	UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT
11	0C	UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA
12	00	ADDRESS MARK NOT FOUND FOR ID FIELD
13	00	ADDRESS MARK NOT FOUND FOR DATA FIELD
14	00	RECORDED ENTITY NOT FOUND
14	01	RECORD NOT FOUND
14	02	FILEMARK OR SETMARK NOT FOUND

14	03	END-OF-DATA NOT FOUND
14	04	BLOCK SEQUENCE ERROR
15	00	RANDOM POSITIONING ERROR
15	01	MECHANICAL POSITIONING ERROR
15	02	POSITIONING ERROR DETECTED BY READ OF MEDIUM
16	00	DATA SYNCHRONISATION MARK ERROR
17	00	RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17	01	RECOVERED DATA WITH RETRIES
17	02	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17	03	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
17	04	RECOVERED DATA WITH RETRIES AND / OR CIRC APPLIED
17	05	RECOVERED DATA USING PREVIOUS SECTOR ID
17	06	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED
17	07	RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT
18	00	RECOVERED DATA WITH ERPOR CORRECTION APPLIED
18	01	RECOVERED DATA WITH ERROR CORRECTION AND RETRIES APPLIED
18	02	RECOVERED DATA - DATA AUTO-REALLOCATED
18	03	RECOVERED DATA WITH CIRC
18	04	RECOVERED DATA WITH LEC
18	05	RECOVERED DATA - RECOMMEND REASSIGNMENT
19	00	DEFECT LIST ERROR
19	01	DEFECT LIST NOT AVAILABLE
19	02	DEFECT LIST ERROR IN PRIMARY LIST
19	03	DEFECT LIST ERROR IN GROWN LIST

1A	00	PARAMETER LIST LENGTH ERROR
1B	00	SYNCHRONOUS DATA TRANSFER ERROR
1C	00	DEFECT LIST IS NOT FOUND
1C	01	PRIMARY DEFECT LIST IS NOT FOUND
1C	02	GROWN DEFECT LIST IS NOT FOUND
1D	00	MISCOMPARE DURING VERIFY OPERATION
1E	00	RECOVERED ID WITH ECC CORRECTION
20	00	INVALID COMMAND OPERATION CODE
21	00	LOGICAL BLOCK ADDRESS OUT OF RANGE
21	01	INVALID ELEMENT ADDRESS
22	00	ILLEGAL FUNCTION (SHOULD USE 20 00)
24	00	INVALID FIELD IN CDB
25	00	LOGICAL UNIT NOT SUPPORTED
26	00	INVALID FIELD IN PARAMETER LIST
26	01	PARAMETER NOT SUPPORTED
26	02	PARAMETER VALUE INVALID
26	03	THRESHOLD PARAMETER NOT SUPPORTED
27	00	WRITE PROTECTED
28	00	NOT READY TO READY TRANSITION (MEDIUM MAY HAVE CHANGED)
28	01	IMPORT OR EXPORT ELEMENT ACCESSED
29	00	POWER ON
2A	00	PARAMETERS CHANGED
2A	01	MODE PARAMETERS CHANGED
2A	02	LOG PARAMETERS CHANGED
2B	00	COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT
2C	00	COMMAND SEQUENCE ERROR
2C	01	TOO MANY WINDOWS SPECIFIED

2C	02	INVALID COMBINATION OF WINDOWS SPECIFIED
2D	00	OVERWRITE ERROR ON UPDATE IN PLACE
2F	00	COMMANDS CLEARED BY ANOTHER INITIATOR
30	00	INCOMPATIBLE MEDIUM INSTALLED
30	01	CANNOT READ MEDIUM - UNKNOWN FORMAT
30	02	CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
30	03	CLEANING CARTRIDGE INSTALLED
31	00	MEDIUM FORMAT CORRUPTED
31	01	FORMAT COMMAND FAILED
32	00	NO DEFECT SPARE LOCATION AVAILABLE
32	01	DEFECT LIST UPDATE FAILURE
33	00	TAPE LENGTH ERROR
36	00	RIBBON
37	00	ROUNDED PARAMETER
39	00	SAVEING PARAMETERS NOT SUPPORTED
3A	00	MEDIUM NOT PRESENT
3B	00	SEQUENTIAL POSITIONING ERROR
3B	01	TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM
3B	02	TAPE POSITION ERROR AT END-OF-MEDIUM
3B	03	TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY
3B	04	SLEW FAILURE
3B	05	PAPER JAM
3B	06	FAILED TO SENSE TOP-OF-FORM
3B	07	FAILED TO SENSE BOTTOM-OF-FORM
3B	08	REPOSITION ERROR
3B	09	READ PAST END OF MEDIUM
3B	0A	READ PAST BEGINNING OF MEDIUM

3B	0B	POSITION PAST END OF MEDIUM
3B	0C	POSITION PAST BEGINNING OF MEDIUM
3B	0D	MEDIUM DESTINATION ELEMENT FULL
3B	0E	MEDIUM SOURCE ELEMENT EMPTY
3D	00	INVALID BITS IN IDENTIFY MESSAGE
3E	00	LOGICAL UNIT HAS NOT SELF-CONFIGURED YET
3F	00	TARGET OPERATING CONDITIONS HAVE CHANGED
3F	01	MICROCODE HAS BEEN CHANGED
3F	02	CHANGED OPERATING DEFINITION
3F	03	INQUIRY DATA HAS CHANGED
40	00	RAM FAILURE (SHOULD USE 40 NN)
40	NN	DIAGNOSTIC FAILURE ON COMPONENT NN (80H - FFH)
41	00	DATA PATH FAILURE (SHOULD USE 40 NN)
42	00	POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)
43	00	MESSAGE ERROR
44	00	INTERNAL TARGET FAILURE
45	00	SELECT OR RESELECT FAILURE
46	00	UNSUCCESSFUL SOFT RESET
47	00	SCSI PARITY ERROR
48	00	INITIATOR DETECTED ERROR MESSAGE RECEIVED
49	00	INVALID MESSAGE ERROR
4A	00	COMMAND PHASE ERROR
4B	00	DATA PHASE ERROR
4C	00	LOGICAL UNIT FAILED SELF-CONFIGURATION
4E	00	OVERLAPPED COMMANDS ATTEMPTED
50	00	WRITE APPEND ERROR

50	01	WRITE APPEND POSITION ERROR
50	02	POSITION ERROR RELATED TO TIMING
51	00	ERASE FAILURE
52	00	CARTRIDGE FAULT
53	00	MEDIA LOAD OR EJECT FAILED
53	01	UNLOAD TAPE FAILURE
53	02	MEDIUM REMOVAL PREVENTED
54	00	SCSI TO HOST SYSTEM INTERFACE FAILURE
55	00	SYSTEM RESOURCE FAILURE
57	00	UNABLE TO RECOVER TABLE-OF-CONTENTS
58	00	GENERATION DOES NOT EXIST
59	00	UPDATED BLOCK READ
5A	00	OPERATOR REQUEST OR STATE CHANGE INPUT (UNSPECIFIED)
5A	01	OPERATOR MEDIUM REMOVAL REQUEST
5A	02	OPERATOR SELECTED WRITE PROTECT
5A	03	OPERATOR SELECTED WRITE PERMIT
5B	00	LOG EXCEPTION
5B	01	THRESHOLD CONDITION MET
5B	02	LOG COUNTER AT MAXIMUM
5B	03	LOG LIST CODES EXHAUSTED
5C	00	RPL STATUS CHANGE
5C	01	SPLINDLES SYNCHRONIZED
5C	02	SPLINDLES NOT SYNCHRONIZED
60	00	LAMP FAILURE
61	00	VIDEO ACQUISITION ERROR
61	01	UNABLE TO ACQUIRE VIDEO
61	02	OUT OF FOCUS
62	00	SCAN HEAD POSITIONING ERROR

# NOVAXIS

63	00	END OF USER AREA ENCOUNTERED ON THIS TRACK
64	00	ILLEGAL MODE FOR THIS TRAC

