Series 2000

AP-DOC-010 Application Note

Designing with the DiskOnChip® 2000

Yigal Ben-Zeev

Product Manager

Jul-97

91-SR-002-01-7L REV. 2.0

M-Systems

Flash Disk Pioneers

1. Preface

This application note describes how to integrate the DiskOnChip 2000 with PC compatible systems. The DiskOnChip 2000 is a single chip FlashDisk designed to plug into a standard 32-pin EEPROM socket. The DiskOnChip 2000 should be mapped into an 8KByte window in the BIOS expansion address space of the PC, which is usually located between address 0C0000H to 0EFFFFH.

The DiskOnChip 2000 contains a built-in copy of the M-Systems industry-standard TrueFFS software, which makes the DiskOnChip operate as a standard disk drive. The DiskOnChip 2000 can contain the operating system in it to allow systems to boot without a hard disk. The DiskOnChip 2000 can also be configured as the boot device in systems with a hard disk (see below "Configuring the DiskOnChip 2000 as the first drive").

The DiskOnChip is a self-contained device. The installation of the DiskOnChip does not require any software installation. The design of the DiskOnChip allows for full upward and downward compatibility. While available today in capacities of 2 to 72MBytes, future DiskOnChip devices with higher densities, will be fully compatible with standard DiskOnChip sockets. The basic design of the DiskOnChip actually supports an unlimited capacity.

2. Operating the DiskOnChip

2.1 Installing the DiskOnChip 2000

When installing or removing the DiskOnChip, be sure to first touch a grounded surface to discharge any static electricity from your body. Use the following procedure to install the DiskOnChip:

1. Align pin 1 on the DiskOnChip with pin 1 of socket.
2. Push the DiskOnChip into the socket carefully until it is fully seated.
3. Check to make sure the DiskOnChip is installed securely, and there are no bent pins.

   Caution: The DiskOnChip may be permanently damaged if installed incorrectly!

4. To install the DiskOnChip as drive C on a system without a hard disk, set the CMOS setup of drive C to "not installed" (indicating that no physical magnetic disk is installed), and reboot the computer. The DiskOnChip 2000 will install as drive C. The DiskOnChip needs to be formatted with the System files in order for it to be a bootable drive. See "Configuring the DiskOnChip as the BOOT device" below.
5. To install the DiskOnChip as drive D on a system with a hard disk, just reboot the system, and the DiskOnChip will install as drive D.
6. To install the DiskOnChip as Drive C on a system with a hard disk, see below "Configuring the DiskOnChip as the first drive".

2.2 Configuring the DiskOnChip 2000 as the Boot device

In order to configure the DiskOnChip as the boot device, the operating system files need to be copied into it. Copying the operating system files into DiskOnChip should be done like in any other hard disk. The following is an example of a typical initialization process:

1. Set the DiskOnChip as a regular drive in your system (not a boot drive).
2. Install a bootable floppy diskette in drive A and boot the system.
3. At the DOS prompt, type SYS C: to transfer the DOS system files to the DiskOnChip (assuming the DiskOnChip is installed as drive C).
4. Copy any files needed into the DiskOnChip.
5. Remove the floppy diskette and reboot the system. The system will boot from the DiskOnChip, and will allow you to run and access any files that have been copied into the DiskOnChip.

2.3 Configuring the DiskOnChip 2000 as the first drive

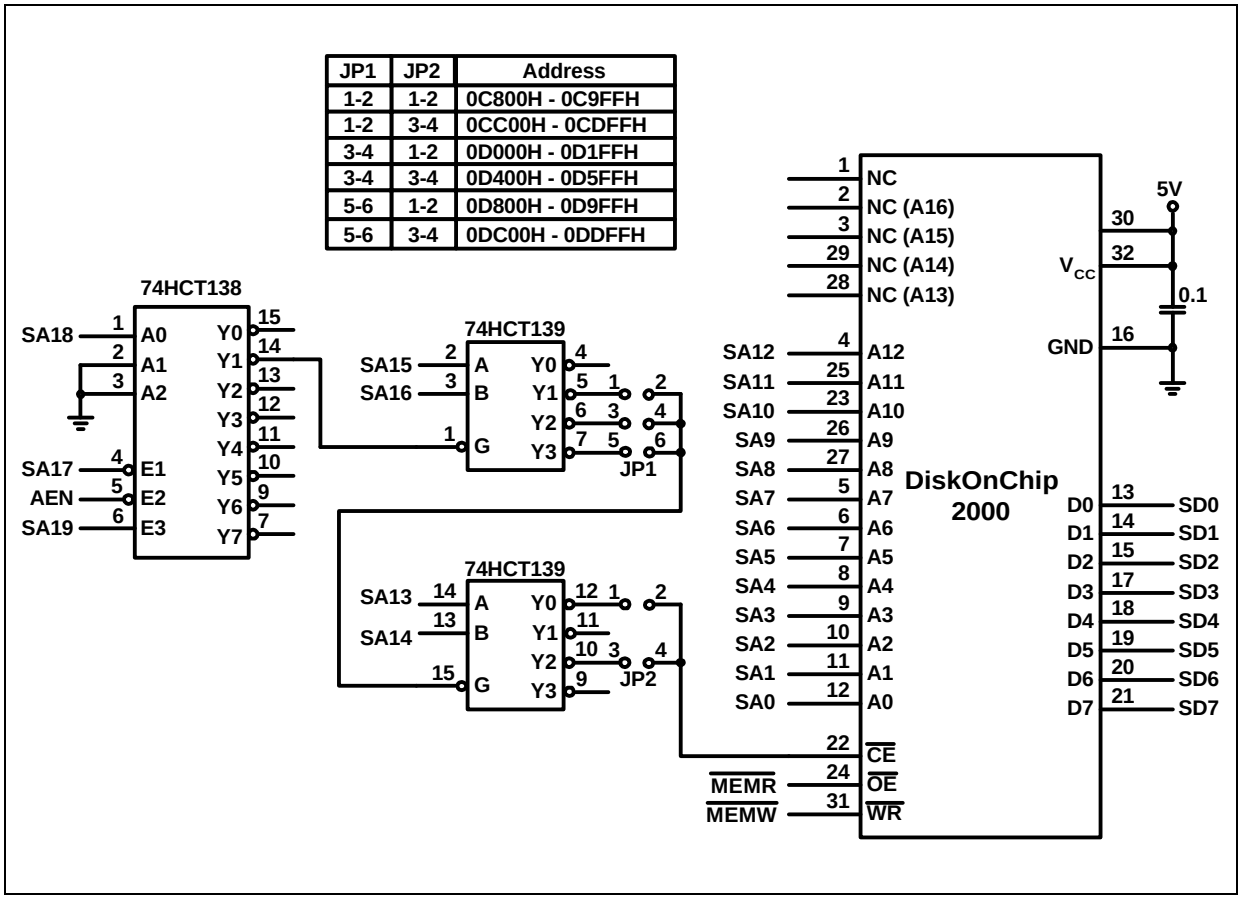
The DiskOnChip can be configured to be installed as the last drive (default), or as the first drive in the system. When configured as the last drive, the DiskOnChip is installed as disk D if there is another hard drive installed, and as drive C if no other hard disk is installed. When configured as the first drive, the DiskOnChip is always installed as drive C. The DiskOnChip is shipped from the factory, configured to install as the last drive. To configure the DiskOnChip to be installed as the first drive, proceed as follows:

1. Boot the system and make sure the DiskOnChip is installed correctly as drive D
2. At the DOS prompt type: DUPDATE D: /FIRST /S:DOC2000.EXB
3. After re-booting the system, the DiskOnChip will appear as drive C:

3. DiskOnChip EVB - An Evaluation Board for the DiskOnChip

The DiskOnChip 2000 Evaluation Board (EVB) is provided by M-Systems as an evaluation tool for the DiskOnChip. The package includes an ISA board with a DiskOnChip socket, software and detailed documentation. The DiskOnChip 2000 EVB enables the evaluation and testing of the DiskOnChip in a standard PC environment.

4. DiskOnChip socket design example



Notes:
1 The above design example shows a DiskOnChip 2000 mapped into an 8KByte window. The DiskOnChip 2000 is compatible with larger windows and will operate and BOOT properly with larger window sizes such as 32KByte and 64KByte windows.
2 Pin 30 connection to VCC is optional - to support 28 pin devices

5. Additional information and Tools

Document/ Tool	Description
DiskOnChip 2000 Data Sheet	DiskOnChip Data Sheet
DiskOnChip 2000 Utilities	DiskOnChip 2000 Utilities User Manual
DiskOnChip2000-EVB	DiskOnChip Evaluation Board
DiskOnChip2000-PIK	DiskOnChip Programmer and Integrators Kit
DiskOnChip-GANG	8 Socket Gang Programmer[1]

1 Contact M-Systems for availability

USA - M-Systems Inc., Phone: 510-413-5950, Fax: 510-413-5950, email: info@ccm.msyscal.com
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266, email: info@msys.nl
Israel - M-Systems LTD, Phone: 972-3-647-7776, Fax: 972-3-647-6668, email: info@m-sys.com
http://www.m-sys.com.

Series 2000

AP-DOC-017 Application Note

Using the DiskOnChip®2000 with Windows® CE

Yair Harel

Software Engineer

Apr-98

91-SR-005-07-7L REV. 1.0

M-Systems

Flash Disk Pioneers

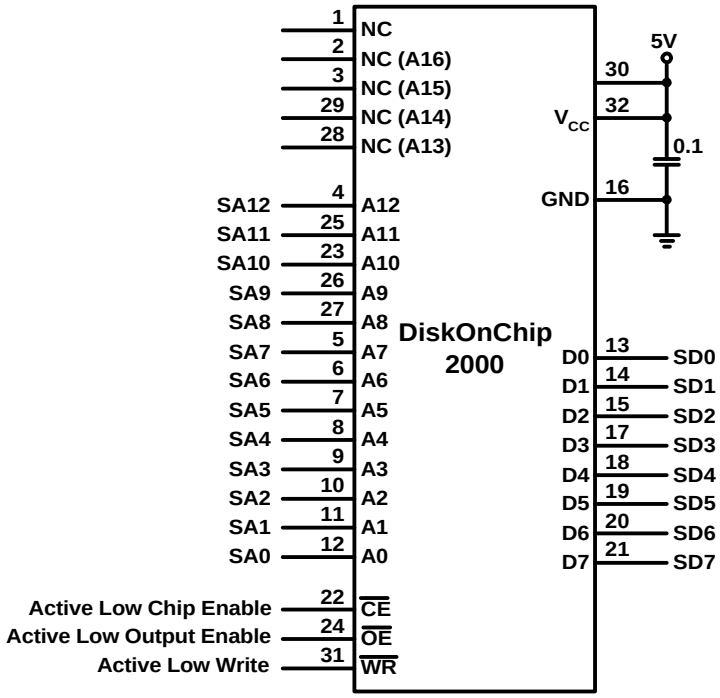
1. Introduction

Starting from version 2.1, Windows®CE supports the DiskOnChip®2000 through the TrueFFS driver. The driver is built into the operating system, allowing system integrators to take advantage of the DiskOnChip storage solution.

This application note is intended for system integrators and OEMs wishing to integrate the DiskOnChip into their design. It will briefly discuss the hardware requirements of the DiskOnChip. Next it will discuss the Windows CE environment and the DiskOnChip driver. It is assumed that the reader has basic familiarity with the Windows CE Embedded Tool Kit or OEM Adaptation Kit. A functional copy of either of these is required.

2. Hardware Requirements for the DiskOnChip

The DiskOnChip can be easily connected to any CPU BUS. The minimum requirements are a twelve-bit address bus, an eight-bit data bus, and three control signals. These signals are similar to SRAM signals for read, write and chip enable. They are typically found on every hardware platform and can be easily interfaced to. Following is a drawing of the DiskOnChip and its pins. For a detailed discussion of the DiskOnChip hardware environment, please refer to the DiskOnChip specifications or to Application Note AP-DOC-10, “Designing with the DiskOnChip 2000”.



3. Using the DiskOnChip in Windows CE

The DiskOnChip can be used for four main purposes in Windows CE:

- Storing user programs and data, thus extending the size of the Object Store. Programs stored on the DiskOnChip will not be fully loaded into RAM for execution, but rather be paged into it upon demand. In system configurations that have a shell, the DiskOnChip will show up as an icon, just like any other folder in the system.
- Storing Registry data. The system integrator can store and retrieve registry information on the DiskOnChip. Being a non-volatile storage media, the DiskOnChip is ideal for registry data storage.
- Storing Databases. This option is configurable by the system integrator.
- Booting up the system. See section 6 for further details.

If the DiskOnChip is not formatted, a dialog will pop up at startup. The user will be asked to confirm the formatting process. If confirmed, the DiskOnChip will be formatted and prepared for use. The boot sequence will continue seamlessly and re-boot will not be required.

4. Using the DiskOnChip from Application Programs

Files on the DiskOnChip are accessed via the Win32 file I/O APIs (Application Program Interface functions). The device name for that should be supplied to these functions can be *DSK1*, *DSK2* etc. The number that follows *DSK* depends on the order in which built-in storage devices are loaded. If the DiskOnChip is the only built-in storage device installed, which is the most common situation, then the number would be *1*.

Calling a *CreateFile()* on the appropriate device name will create a file on the DiskOnChip and return a file handle. Applications can use this handle to read or write data to the file. In order to write data to the file, the application calls the function *WriteFile()*. FAT (the Windows CE File System) translates the write request to logical FAT blocks. FAT searches the buffer cache for the requested block(s). If these are not present, FAT issues an *IOControl* request to read bytes from the DiskOnChip.

The TrueFFS driver, responsible for the DiskOnChip, receives the *IOControl* request and then tries to fulfill it by accessing the media.

5. The DiskOnChip Driver

Windows CE has a driver that controls the DiskOnChip. The driver, which is an integral part of the operating system, is a Dynamically Linked Library named TrueFFS.dll. It must be present in the Windows CE image in order for the DiskOnChip to be detected at startup. Several registry entries must also be present at that time.

The TrueFFS driver is capable of controlling Linear Flash cards in addition to the DiskOnChip.

5.1 Incorporating the Driver in the Windows CE Image

The next three subsections discuss the configuration files modifications that are required in order to include the driver and the required registry entries in the operating system image.

5.1.1 Updating the Sysgen File

CEsysgen.bat specifies the modules of the Windows CE operating system to be included in the target project. Once executed by *sysgen.bat*, this file sets the value of the *CE_MODULES* environment variables.

The copy of *CEsysgen.bat* in the *%_WINCEROOT%\Public\<Configuration>\Oak\Misc d*irectory should be modified to include the TrueFFS component, where *<Configuration>* stands for the name of the configuration being built. To add the TrueFFS component, append the following line to this file:

SET CE_MODULES=%CE_MODULES% TrueFFS

The *DEMO7* configuration, for example, already includes the TrueFFS driver. Search for "trueffs" in the file *%_WINCEROOT%\Public\demo7\Oak\Misc\CEsysgen.bat*

5.1.2 Updating the Binary Image Builder (.Bib) Files

Binary Image Builder files are used by *makeimg.exe* while building the Windows CE image. Operating system components can be included or excluded in the image by modifying these files. In order to include the DiskOnChip driver, the system integrator has to take care of two environment variables prior to invoking *makeimg*. This file is automatically executed as part of a complete build process (e.g. *blddemo)*, therefore these variables are best taken care of before starting this process.

The environment variable *IMGNODRIVERS* must be undefined in order to include any drivers in the image, including the DiskOnChip driver.

The environment variable *CE_MODULES_TRUEFFS* must be defined in order to specifically include the DiskOnChip driver.

5.1.3 Updating the Registry

The Windows CE Device Manager uses several registry sub-keys under the *HKEY_LOCAL_MACHINE\Drivers* key when loading the DiskOnChip driver. Built-in devices, such as the DiskOnChip, rely on the registry settings to be present at boot time, therefore these keys must be part of the default registry. This section describes the registry entries the Device Manager will use to install the DiskOnChip driver.

When building a CEPC image for x86 based PC platforms, the system integrator can define the environment variable CEPC_DISKONCHIP prior to building the image. This will ensure the proper registry entries are included. For other platforms, it is required that the following entries will be added to the appropriate *platform.reg* file. This file typically resides in *%_WINCEROOT%\Platform\<Platform name>\Files, where <Platform name>* stands for the relevant platform being used, e.g. *ODO*.

```
[HKEY_LOCAL_MACHINE\Drivers\BuiltIn\TrueFFS]
   "Index"       =  dword:1
   "Dll"         =  "TrueFFS.dll"
   "Prefix"      =  "DSK"
   "Order"       =  dword:1
   "Ioctl"       =  dword:4
   "FSD"         =  "FATFS.DLL"
   "WindowBase"  =  dword:D0000
   "Folder"      = "DiskOnChip"
```

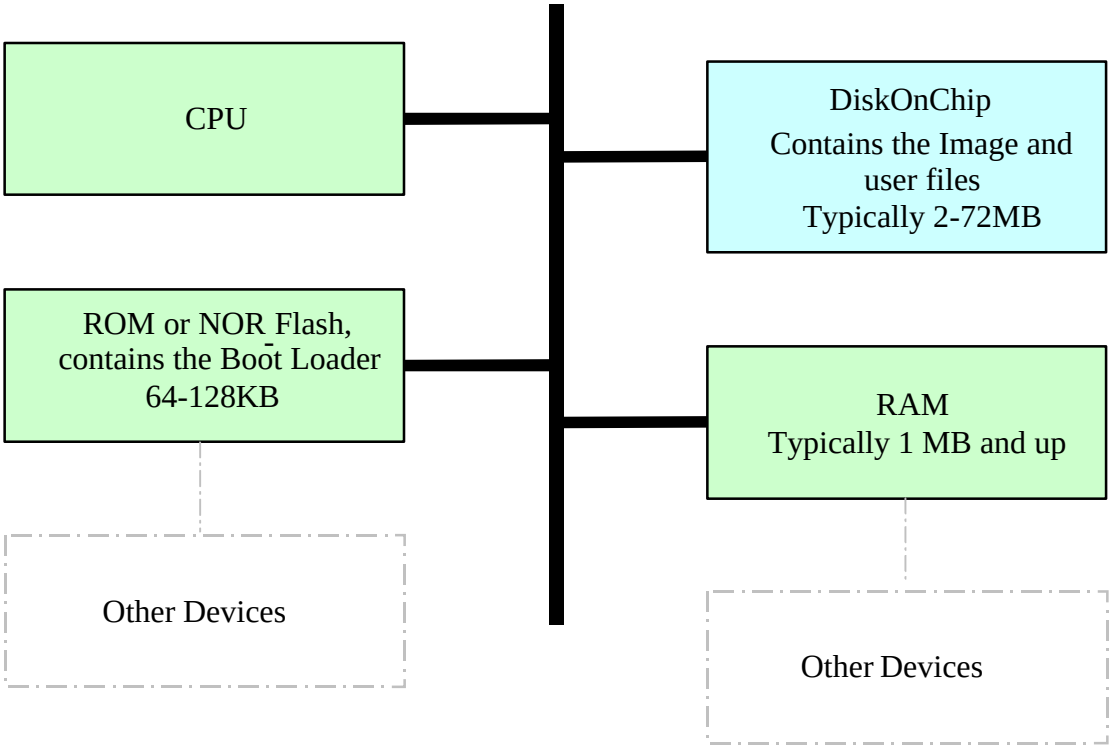
The "*WindowBase*" value determines the address of the DiskOnChip in system memory. The value should be within the 32 bit address range, and is determined during hardware design or system integration. The value shown here serves as an example only.

The "*Folder*" string is optional. Its value determines the folder name associated with the DiskOnChip. If this string is absent, the default name will be "*Storage Card*".

If the "*Folder*" string is used to change the folder name to, say, "*DiskOnChip*", then this will be the actual name of the first device. The second will be called "*DiskOnChip2*", the third "*DiskOnChip3*" and so forth.

6. Booting Up the System through the DiskOnChip

Being able to boot up Windows CE from a DiskOnChip device is of high importance. It allows the system integrator to use a very small ROM in the system, storing the operating system itself on the DiskOnChip. The following figure shows a simplified structure of the hardware system:

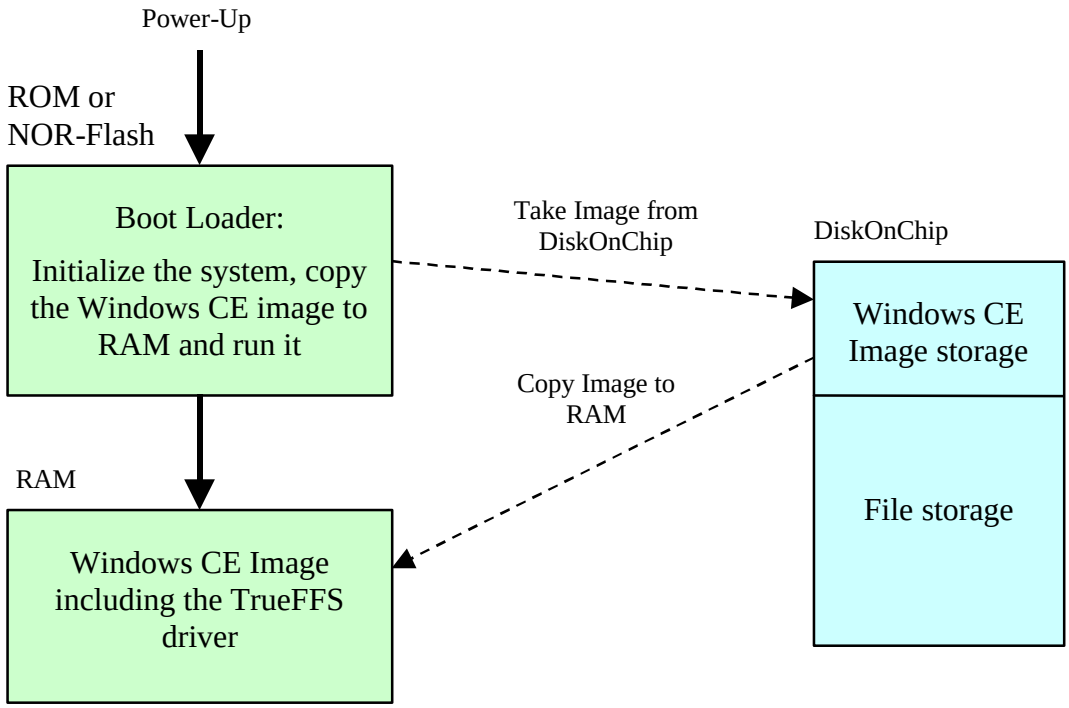


The idea behind booting is to separate the software that is used to boot up the system from the TrueFFS driver.

The system integrator will generate a Windows CE image that contain the TrueFFS driver. This image will then be placed on the DiskOnChip, either as a standard file in the FAT file system or in a designated area that is especially reserved for boot images. A special piece of software called the “Boot Loader”, will then load this image into memory.

The boot loader will be stored on ROM or NOR-Flash. It will be invoked at boot time and will do some basic initialization tasks. Next, it will take the Windows CE image (e.g. NK.BIN) that is stored on the DiskOnChip and copy it to RAM. Finally, it will and jump to the start address of the image, and Windows CE will start running. From this moment on Windows CE will take control over the system, invoking the TrueFFS driver among other drivers. This will allow the usage of the DiskOnChip for user files, databases and registry information. The amount of ROM required to implement this boot solution is very small, usually in the range of a few tens of Kilobytes.

The next drawing outlines the flow of control:



Annasoft Systems implemented a boot solution for systems based on x86 PC architecture. The idea is similar to the one described above, only that no boot loader has to be designed. The solution relies on the BIOS to bring up the system. Once the system is up, a special loader called “CE Launcher™”, copies the Windows CE image into RAM and execute it.

7. Additional information and Tools

Additional information about the DiskOnChip, including application notes, can be found at http://www.m-sys.com.

Information about Annasoft Systems can be found at http://www.annsoft.com.

More information about booting-up a Windows CE system using the DiskOnChip can be obtained from Bsquare at http://www.bsquare.com.

Additional tools and documents are listed in the following table:

Document/ Tool	**Description**
AP-DOC-10	Designing with the DiskOnChip 2000
AP-DOC-16	Using the DiskOnChip 2000 with QNX
AP-DOC-19	Using the DiskOnChip 2000 with Windows 95
AP-DOC-11	Write Protecting the DiskOnChip 2000
DiskOnChip 2000 Data Sheet	DiskOnChip Data Sheet
DiskOnChip 2000 Utilities	DiskOnChip 2000 Utilities User Manual
DiskOnChip2000-EVB	DiskOnChip Evaluation Board
DiskOnChip2000-PIK	DiskOnChip Programmer and Integrators Kit
DiskOnChip-GANG	8 Socket Gang Programmer

USA - M-Systems Inc., Phone: 510-413-5950, Fax: 510-413-5980, email: info@m-sys.com
Taiwan - M-Systems Asia, Phone: 886-2-25501741, Fax: 886-2-25501745
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266
Israel - M-Systems Ltd, Phone: 972-3-647-7776, Fax: 972-3-647-6668
http://www.m-sys.com

Series 2000

AP-DOC-016

Application Note

Using the DiskOnChip®2000 with QNX®

Andray Kaganovsky

Software Engineer

Apr-98

91-SR-005-06-7L REV. 1.0

M-Systems

Flash Disk Pioneers

1. Introduction

M-Systems' DiskOnChip® 2000 is a new generation of single-chip Flash disk. It contains built-in firmware that provides full hard disk emulation that allows the DiskOnChip to operate as a boot device.

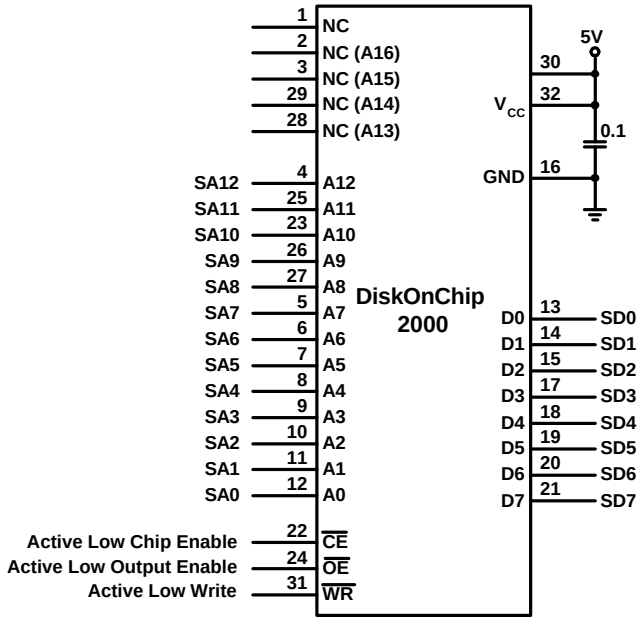
When used under QNX®, the DiskOnChip is managed by a TrueFFS® technology based device driver attached to the standard QNX file system (`Fsys`) or to a DOS® -compatible file system (`Dosfsys`).

This application note describes how to use the DiskOnChip as a bootable data storage device under the QNX Operating System, version 4.23 or higher.

We will briefly discuss the hardware requirements of the DiskOnChip. The main part of this application note will discuss software installation. This will include basic driver installation, boot issues, using the utilities and partitioning.
It is assumed that the reader is familiar with QNX system administration.

2. Hardware Requirements for the DiskOnChip

Originally designed for PC systems, the DiskOnChip can also be used in different hardware environments. The minimum requirements are a twelve-bit address bus, an eight-bit data bus, and three control signals. These signals are similar to SRAM signals for read, write and chip enable. They are typically found on every hardware platform and can be easily interfaced to. Following is a drawing of the DiskOnChip and its pins. For a detailed discussion of the DiskOnChip hardware environment, please refer to the DiskOnChip specifications or to Application Note AP-DOC-10, "Designing with the DiskOnChip 2000".



3. Distribution Diskette Content

The DiskOnChip distribution floppy holds a file named `doc2_tar.f`. This is a compressed archive that contains the following files:

`Fsys.diskonchip`	DiskOnChip device driver for QNX version 4.23 or higher
`dformat`	DiskOnChip formatting utility
`dupdate`	Utility for updating DiskOnChip firmware
`doc107.exb`	DiskOnChip firmware. "`107`" is the firmware version, the actual diskette might contain higher versions of the firmware, e.g. `doc108.exb`, `doc109.exb` etc.
`copy2doc`	Shell script containing a list of files that are essential for booting QNX from the hard disk. The files in this list have to be copied from the hard disk to the DiskOnChip before operation.
`doc2000`	Sample build-file for building a custom QNX boot-image that includes the DiskOnChip driver.

4. Installing the DiskOnChip and its Software

In order to perform the following steps you must be able to log-in as a superuser (root). This is required for some of the system configuration activities that have to be done. If you are unable to do so, contact the your on-site guru, or consult the QNX documentation. Some basic ability of hardware installation is also required. If you find it difficult to manage, consult the hardware or system-administration people at your organisation.

4.1 Verifying Floppy Driver Presence

In order to go forward with installation, the floppy driver must be active. To verify that this is actually the case, try the following procedure. "#" is the QNX prompt, as in the rest of this document.
Type:

```
# sin ver
```

If "Floppy" doesn't appear in the NAME column, type:

```
# Fsys.floppy
```

Insert the distribution diskette and run the `Dosfsys` command to access the DOS floppy drives:

```
# Dosfsys &
```

From this point on it is assumed that the distribution diskette is mounted at `/dos/a`. If your diskette turns out to be mounted somewhere else (use "`sin ver`" to verify this), change the respective instructions that follow to work with this alternate name.

4.2 Unpacking and Copying Required Files

This section will explain how to unpack the distribution archive and copy the files to their target directories.

Start by creating a sub-directory named `tffs` under `/usr/local` and make it your current directory. Type the following:

```
# mkdir /usr/local/tffs
# cd /usr/local/tffs
```

Copy the file `doc2_tar.f` from the distribution floppy to the newly created directory. Change its name to `doc2_tar.tar.F` while copying:

```
# cp /dos/a/doc2_tar.f doc2_tar.tar.F
```

Melt down the file and un-tar the resulting archive:

```
# freeze -d doc2_tar.tar.F
# tar -xf doc2_tar.tar
```

Next you should to create a link to the DiskOnChip driver:

```
# ln -f ./Fsys.diskonchip /bin/Fsys.diskonchip
```

The last step in this sequence is copying the file `doc2000` to directory `/boot/build`:

```
# cp doc2000 /boot/build
```

At this point the driver and all auxiliary files should have been already copied to the proper locations on your disk. Proceed to the next section for installing the DiskOnChip itself.

4.3 Installing and Formatting the DiskOnChip

This section will explain how to install the DiskOnChip in your system, how to format and how to mount it. Before proceeding, shut down your computer. This is required before any hardware device can be added or removed. When the power is turned off, the DiskOnChip can safely be installed on the board. Refer to the DiskOnChip product manual for hardware installation instructions.

Once the DiskOnChip is properly installed, proceed with software installation. A hard disk drive is necessary, as you will need to first boot QNX from a hard disk.

Login as superuser and follow the next sequence of commands:

Make `/usr/local/tffs` your current directory:

```
# cd /usr/local/tffs
```

Format the DiskOnChip using the `dformat` utility. Note: All information previously stored on the DiskOnChip will be erased. If you have any valuable information stored there, create a backup copy now.

Type the following command:

```
# ./dformat
```

Now run the DiskOnChip driver in order to create a raw device `/dev/tffs0`:

```
# Fsys.diskonchip
```

Make sure a device named `tffs` was successfully created using `ls`:

```
# ls /dev/tffs0
```

Next step would be to create a QNX partition on the DiskOnChip. You can do that using the QNX `fdisk` utility. Refer to the QNX Utilities Reference for details regarding this utility. The following command sequence will create a QNX partition that spans the entire DiskOnChip, mark it as bootable and write the QNX partition loader to it:

```
# fdisk /dev/tffs0 add -f 1 QNX ALL
# fdisk /dev/tffs0 boot QNX
# fdisk /dev/tffs0 loader
```

To verify the result of the above commands, use the following command to display the contents of the DiskOnChip partition table:

```
# fdisk /dev/tffs0 show
```

You should get a table similar to the following, assuming you are using an 8MB DiskOnChip. Other capacities will result in a slightly different output.

```
      _____OS_____  Start    End      ______Number_____  Size
Boot
      name    type  Cylinder Cylinder Cylinders  Blocks
  1.  QNX    ( 77)      0     1004      1005     16079   7 MB
*
  2.  -----  (---)   -----    -----     -----    ------  -----
  3.  -----  (---)   -----    -----     -----    ------  -----
  4.  -----  (---)   -----    -----     -----    ------  -----
```

Once the partition is correctly set up, it has to be mounted before it can be used. Use the following command to mount it:

```
# mount -p /dev/tffs0
```

In the typical situation, the mounted device will be named `/dev/ffs0t77`. You can verify this by using `ls`:

```
# ls /dev/tffs0t77
```

If you see an entry with that name in the output, the mount process was successful. Otherwise, repeat the last steps from the beginning of section 4.3.

Next step would be to initialise the file system on the QNX partition. Use the `dinit` command as following:

```
# dinit -h /dev/tffs0t77
# dinit -hb /dev/tffs0t77
```

Notice that `dinit` might issue a warning message saying that initialisation was attempted on a hard disk. This warning can be safely ignored.

Once the file system is initialised, it has to be mounted. Type in the following to mount the QNX partition as `/mnt/diskonchip`:

```
# mkdir /mnt/diskonchip
# mount /dev/tffs0 -p/dev/tffs0t77 /mnt/diskonchip
```

Finally you have to verify that the mount process was successful. Use `chkfsys` to check the file system on the DiskOnChip, and `ls` to see mounted DiskOnChip entry:

```
# chkfsys –u /mnt/diskonchip
# ls /mnt/diskonchip
```

At this stage, the DiskOnChip is installed and mounted in your system as `/mnt/diskonchip`.

4.4 Mounting the DiskOnChip Automatically

To automatically mount the DiskOnChip during boot, modify the system initialisation file `/etc/config/sysinit.NNN`.

"`NNN`" stands for your local node number. If you don't have a network connection, NNN equals 1. Add the next two lines to this file:

```
Fsys.diskonchip
```

```
mount /dev/tffs0 -p/dev/tffs0t77 /mnt/diskonchip
```

If you plan to boot QNX from the DiskOnChip, you can now proceed to section 5. Otherwise, to use the DiskOnChip as an additional disk in the system, continue to section 6.

5. Booting QNX from the DiskOnChip

Being able to boot up QNX from the DiskOnChip is of great importance. It will allow you to use the DiskOnChip as the only disk in the system, holding the operating system itself in addition to all other applications and files.
Follow the next steps in order to use the DiskOnChip as the boot device:

5.1 Updating the DiskOnChip Firmware

First you will need to update the DiskOnChip's firmware. Change the current directory to `/usr/local/tffs` and run the `dupdate` utility with the file `doc107.exb` as a source:

```
# cd /usr/local/tffs
# ./dupdate -s:doc107.exb -f
```

Note: the `-f` switch instructs the DiskOnChip to act as a primary boot device. This means that the BIOS will attempt to boot the OS from the DiskOnChip, even if there are other bootable disks in the system.

As an alternative, you might consider omitting the `-f` switch. In this case, you must disable all other disks using the BIOS set-up utility in order to boot from the DiskOnChip. Whenever you will need to boot from a different disk, simply re-enable it using the BIOS set-up utility.

The procedures described in sections 5.2 and 5.3 are typically performed done only once, on the very first DiskOnChip you use. Once you have successfully gone through on of these procedures, it is possible to copy the binary image of the DiskOnChip to others DiskOnChips (i.e. cloning the first DiskOnChip). The easiest way to clone a DiskOnChip is to use M-System's DiskOnChip Gang Programmer. This is a special tool that makes the cloning operation very fast and fully automatic. Please contact M-Systems for availability of this product.

If you use a DiskOnChip with capacity smaller than 16MB, go directly to section 5.3. Otherwise, please continue to the next section.

5.2 Building a Custom QNX Image for a DiskOnChip 16MB or Higher

If you use a DiskOnChip with a capacity of 16MB or higher, you can use the standard QNX installation procedure. To do this, you first need to go through a very simple procedure of making a custom QNX boot diskette that includes the DiskOnChip software support.

Start by making an exact copy of the standard QNX boot diskette in your PC, using the DOS utility `DISKCOPY`:

```
> DISKCOPY A: B:
```

Move this diskette to your QNX system and mount it at `/mnt/floppy`. We assume here that the `Fsys.floppy` driver has already been started.

```
# mount /dev/fd0 /mnt/floppy
```

Finally, copy the DiskOnChip software to `/mnt/floppy/bin`:

```
# cp /usr/local/tffs/Fsys.diskonchip /mnt/floppy
# cp /usr/local/tffs/dformat         /mnt/floppy
# cp /usr/local/tffs/doc107.exb      /mnt/floppy
```

Note. To make space for the DiskOnChip software, you might need to delete some unnecessary device drivers (e.g. all files named `Fsys.*` in `/mnt/floppy/bin`).

From this point on you can use this newly created diskette instead of the standard QNX boot diskette when installing QNX onto a DiskOnChip.

Start the standard QNX installation procedure by booting QNX from the newly created QNX boot diskette. When the operating system prompt appears, format the DiskOnChip using the following command :

```
# dformat -s:/bin/doc107.exb -f
```

Now start the installation process by typing the following command:

```
# install -d -p "-n:hd0"
```

When "`third party disk`" appears, press Enter. When presented with the list of device drivers to select from, pick `Fsys.diskonchip` and proceed with installation.

Note: Upon boot, the DiskOnChip will appear as `/dev/hd0`, and its QNX partition as `/dev/hd0t77` respectively. Other disks installed in your system might not be immediately accessible. If this is the case, simply add the following two lines to the file `etc/config/sysinit.NNN`. "`NNN`" stands for your local node number. If you don't have a network connection, NNN equals 1.

```
Fsys.ide &
mount -p /dev/hd1 /dev/hd1t77 /mnt/hard_disk
```

The first line starts the IDE hard disk device driver. If you have disk of some other type, start the appropriate driver, e.g. `Fsys.eide`, `Fsys.ata`, etc.

The second line mounts the QNX partition at `/mnt/hard_disk`.

Your DiskOnChip is now bootable, and installation is essentially complete. Section 5.3 can be skipped. Information about the DiskOnChip utilities can be found at section 6.

5.3 Building a Custom QNX Image for a DiskOnChip Smaller than 16MB

Relatively small capacity DiskOnChip requires the use of a special QNX image. This image must include the DiskOnChip driver. In order to build such an image follow the next steps.

First, change the current directory to `/boot`:

```
# cd /boot
```

Now run the command `make` to create a custom QNX boot-image named `doc2000` located at `/boot/images/`:

```
# make b=doc2000
```

Copy the custom QNX boot-image `doc2000` from directory `/boot/images/` to `.boot` at `/mnt/diskonchip/`. We assume that the DiskOnChip is mounted at `/mnt/diskonchip`, as explained in section Installing and Formatting the DiskOnChip.

```
# cp /boot/images/doc2000 /mnt/diskonchip/.boot
```

Next, copy all the required system and shell files to the appropriate directories, located under `/mnt/diskonchip`. If you are unclear about which files are needed at boot time, try using the shell script file `/usr/local/tffs/copy2doc` provided on the distribution diskette. Type in the following:

```
# cd /usr/local/tffs
# chmod a+x copy2doc
# ./copy2doc
```

Note: The shell script file `copy2doc` is essentially an edited output of the following command:

```
# ls -lR / >/usr/local/tffs/copy2doc
```

This file refers to configuration files, license files etc. These might have different names on your machine. It is advisable that you have a closer look at the contents of `copy2doc` to see if anything needs to be changed.

If you plan to use a 32-bit device driver (Applicable to QNX version 4.23 and above), perform the following commands to copy files from `/bin` directory to directory `/mnt/diskonchip/bin`:

```
# cp /bin/Dev32       /mnt/diskonchip/bin/Dev32
# cp /bin/Dev32.ansi  /mnt/diskonchip/bin/Dev32.ansi
# cp /bin/Dev32.par   /mnt/diskonchip/bin/Dev32.par
# cp /bin/Dev32.pty   /mnt/diskonchip/bin/Dev32.pty
# cp /bin/Dev32.ser   /mnt/diskonchip/bin/Dev32.ser
```

If you plan to use a 16-bit device driver, perform the following sequence:

```
# cp /bin/Dev16       /mnt/diskonchip/bin/Dev16
# cp /bin/Dev16.ansi  /mnt/diskonchip/bin/Dev16.ansi
# cp /bin/Dev16.par   /mnt/diskonchip/bin/Dev16.par
# cp /bin/Dev16.pty   /mnt/diskonchip/bin/Dev16.pty
# cp /bin/Dev16.ser   /mnt/diskonchip/bin/Dev16.ser
```

When in doubt about which drivers should be used, simply copy both 32 and 16-bit versions as shown above.

Next step would be to establish all the necessary symbolic links. We assume you use a 32-bit device driver, which is the QNX 4.23 default. If you use 16-bit device driver, change the names accordingly (i.e. change all appearances of `Dev32` to `Dev16`):

```
# cd /mnt/diskonchip/bin
# ln -s Dev32      Dev
# ln -s Dev32.ansi Dev.ansi
# ln -s Dev.ansi   Dev.con
# ln -s Dev.ansi   Dev.ditto
# ln -s Dev32.par  Dev.par
# ln -s Dev32.pty  Dev.pty
# ln -s Dev32.ser  Dev.ser
```

To finalise this step, it is required to reboot your machine. Enter the following command:

```
# shutdown
```

While the system reboots, you should see a DiskOnChip installation message, followed by messages from the QNX bootstrap loader.

Note: Right after booting from the DiskOnChip, other disks installed in your system might not be immediately accessible. If this is the case, simply add the following two lines to the file `etc/config/sysinit.NNN`. "`NNN`" stands for your local node number. If you don't have a network connection, NNN equals 1.

```
Fsys.ide &
mount -p /dev/hd1 /dev/hd1t77 /mnt/hard_disk
```

The first line starts the IDE hard disk device driver. If you have disk of some other type, start the appropriate driver, e.g. `Fsys.eide, Fsys.ata`, etc.

The second line mounts the QNX partition at `/mnt/hard_disk`.

Your DiskOnChip is now bootable, and installation is essentially complete. Information about the DiskOnChip utilities can be found at section DiskOnChip Driver Parameters.

6. DiskOnChip Driver Parameters

The DiskOnChip has a built-in firmware that is visible in the CPU address range, reserved for BIOS expansions. Being executed by the BIOS, the DiskOnChip firmware installs itself at INT 13h (BIOS disk services), thus providing full read/write emulation of a hard disk.

Since QNX runs entirely in 32-bit protected mode and does not rely on BIOS services, the `Fsys.diskonchip` device driver must be active at boot time to allow access to the DiskOnChip. The installation procedure described in the last sections guarantees this.

Normally, the driver is started without any parameters. Since the DiskOnChip is a memory-mapped device, the driver should locate it in memory. By default, the driver looks in the memory range 0C8000h to 0E0000h. If the memory window address is known, it is possible to pass it to the driver from the command line, thus eliminating the search. For example, if the DiskOnChip address is D4000h, the driver can be started as follows:

```
# Fsys.diskonchip -w:D4000 &
```

The DiskOnChip driver creates a raw DiskOnChip device, named `tffs0` by default. This default name can be overridden by specifying a different one at the command line:

```
# Fsys.diskonchip -n:MyName &
```

In this case the device will be named `MyName`.

7. DiskOnChip Utilities

Two utilities accompany the DiskOnChip driver. They allow the user to format the DiskOnChip and update its firmware.

7.1 DiskOnChip Formatting Utility (`dformat`)

The DiskOnChip arrives in a pre-formatted condition. It can be re-formatted many times though. Please note that just like a hard drive, all data stored on the media is forever lost after formatting. Exercise caution while formatting the DiskOnChip, or you might accidentally wipe out some valuable information that might have been stored there.

It is strongly suggested to re-boot the machine after you have formatted the DiskOnChip and before you run the its driver, `Fsys.diskonchip`.

The DiskOnChip must be un-mounted before re-formatting it. Assuming it has been mounted at `/mnt/diskonchip`, type the following command in order to un-mount it:

```
# umount /mnt/diskonchip
# rm /dev/tffs0
```

To format the DiskOnChip, type in the following commands:

```
# cd /usr/tffs/local
# ./dformat
```

When formatting, the DiskOnChip firmware is retained by default. It can be updated by using the `-s` option. As an example, the following commands can be typed in to update the DiskOnChip firmware using the file `/usr/local/tffs/doc107.exb` as a source:

```
# cd /usr/tffs/local
# ./dformat -s:doc107.exb
```

By default, the DiskOnChip firmware installs the DiskOnChip as an additional disk in the system. This default allows you to boot an operating system from the DiskOnChip on a diskless machine. In case your machine is equipped with other hard disk(s) and you still want to boot from the DiskOnChip, you would have to install the DiskOnChip as the first drive. To do so, add the `–f` option to the formatting command line:

```
# cd /usr/tffs/local
# ./dformat -s:doc107.exb -f
```

In some cases it is useful to prevent the DiskOnChip firmware from installing at boot time. In such a case you should use the `–s` option with an alternate firmware image file named `/usr/tffs/local/doc2.fff`:

```
# cd /usr/tffs/local
# ./dformat -s:doc2.fff
```

7.2 DiskOnChip Firmware Update Utility (dupdate)

The DiskOnChip firmware can be updated without affecting the QNX partition. This can be done using the `dupdate` utility. There is no need to un-mount DiskOnChip before using this utility. To update the firmware using the source file `doc107.exb`, type in the following commands:

```
# cd /usr/tffs/local
# ./dupdate -s:doc107.exb
```

By default, the firmware installs the DiskOnChip as an additional disk in the system. This will allow you to boot an operating system from the DiskOnChip on a diskless machine. In case your machine is equipped with a hard disk(s) and you still want to boot from the DiskOnChip, you would have to install the DiskOnChip as the first driver. To do so, use the `–f` option:

```
# cd /usr/tffs/local
# ./dupdate -s:doc107.exb -f
```

In some cases it is useful to prevent the DiskOnChip firmware from installing at boot time. In such a case you should use the `–s` option with an alternate firmware image file named `/usr/tffs/local/doc2.fff`:

```
# cd /usr/tffs/local
# ./dupdate -s:doc2.fff
```

8. Setting Up Multiple Disk Partitions on a DiskOnChip

The DiskOnChip can be installed in a QNX system either as an additional disk or as a boot device. Since the DiskOnChip supports multiple boot partitions, several operating systems can be installed and booted from it. The DiskOnChip fully emulates a hard drive, therefore it is simple to create multiple partitions on it. The command sequence shown below, when executed instead of the `fdisk` sequence in section 4.3, will create a bootable QNX partition along with a DOS partition:

```
# fdisk /dev/tffs0 add -f 1 QNX HALF
# fdisk /dev/tffs0 add -f 2 t4  HALF
# fdisk /dev/tffs0 boot QNX
# fdisk /dev/tffs0 loader
```

To verify the consequence of the above commands, use the following command to display the contents of the DiskOnChip partition table:

```
# fdisk /dev/tffs0 show
```

Assuming you use an 8MB DiskOnChip, you should get the following table:

```
      _____OS_____  Start    End      ______Number_____   Size
Boot
      name    type  Cylinder Cylinder Cylinders  Blocks
  1.  QNX    ( 77)       0      501       502      8039   7 MB    *
  2.  DOS    (  4)     502     1004       503      8040   7 MB
  3.  -----  (---)   -----    -----     -----    ------  -----
  4.  -----  (---)   -----    -----     -----    ------  -----
```

Note that the DOS partition has only been created, but not yet initialised. In order to initialise it you will have to boot DOS from a floppy and format this partition using DOS `format`. After you have done this, you will be able to mount the DOS partition of the DiskOnChip under QNX. Assuming the mounting point is `/dos/d`, type in the following command to mount this partition:

```
# Dosfsys d=/dev/tffs0t4 &
```

9. Additional information and Tools

Document/ Tool	Description
AP-DOC-10	Designing with the DiskOnChip 2000
AP-DOC-17	Using the DiskOnChip 2000 with Windows CE
AP-DOC-19	Using the DiskOnChip 2000 with Windows 95
AP-DOC-11	Write Protecting the DiskOnChip 2000
DiskOnChip 2000 Data Sheet	DiskOnChip Data Sheet
DiskOnChip 2000 Utilities	DiskOnChip 2000 Utilities User Manual
DiskOnChip2000-EVB	DiskOnChip Evaluation Board
DiskOnChip2000-PIK	DiskOnChip Programmer and Integrators Kit
DiskOnChip-GANG	8 Socket Gang Programmer

Series 2000

IM-DOC-021
Installation Manual

Using the DiskOnChip®2000 with Linux OS

Ron Dick, Esther Spanjer

August 98
91-SR-005-10-7L REV. 1.0

M-Systems
Flash Disk Pioneers

Limited Warranty

(a) M-Systems warrants that the Licensed Software — **prior to modification and adaptation by Licensee —** will conform to the documentation provided by M-Systems. M-Systems does **not** warrant that the Licensed Software will meet the needs of the Licensee or of any particular customer of Licensee, nor does it make any representations whatsoever about Licensed Software that has been modified or adapted by Licensee. .

(b) Subsection (a) above sets forth Licensee's sole and exclusive remedies with regard to the Licensed Software.

M-SYSTEMS MAKES NO OTHER WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THE LICENSED SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. THERE ARE NO OTHER WARRANTIES WITH RESPECT TO THE LICENSED SOFTWARE ARISING FROM ANY COURSE OF DEALING, USAGE OR TRADE OR OTHERWISE.

IN NO EVENT SHALL M-SYSTEMS BE LIABLE TO LICENSEE FOR LOST PROFITS OR OTHER INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES, WHETHER UNDER THIS AGREEMENT, IN TORT OR OTHERWISE.

(c) Licensee shall not make any promise, representation, warranty or guaranty on behalf of M-Systems with respect to the Licensed Software except as expressly set forth herein.

Please note: The Licensed Software is not warranted to operate without failure. Accordingly, in any use of the Licensed Software in life support systems or other applications where failure could cause injury or loss of life, the Licensed Software should only be incorporated in systems designed with appropriate and sufficient redundancy or back-up features.

1. Introduction

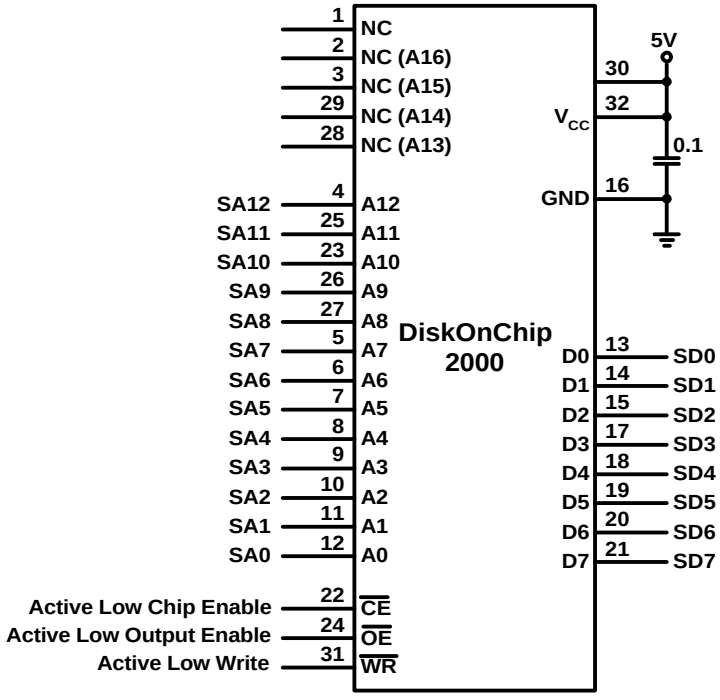
M-Systems' DiskOnChip ®2000 is a new generation of single-chip flash disk. It contains built-in firmware that provides full hard disk emulation and allows the DiskOnChip®2000 to operate as a boot device.
When used under Linux (kernel version 2.0.29 – 2.0.35), the DiskOnChip®2000 is managed by a TFFS® (True Flash File System) technology based device driver, attached to the standard Linux file system [`ext2`].

This installation manual describes how the DiskOnChip®2000 can be installed as an additional disk or as a boot device under Linux.

The hardware requirements of the DiskOnChip®2000 will be briefly discussed, the main part of this installation manual is related to software installation. This will include basic driver installation and boot issues.

2. Hardware requirements for the DiskOnChip®2000

Originally designed for PC environments, the DiskOnChip®2000 can also be used in different hardware environments. The minimum requirements are a twelve-bit address bus, an eight-bit databus, and three control signals (CE, OE, WR). Following is a drawing of the DiskOnChip®2000 and its pins. For a detailed explanation of the DiskOnChip®2000 hardware environment, please refer to the DiskOnChip®2000 specifications or to Application Note AP-DOC-010, "Designing with the DiskOnChip®2000".



3. Requirements for installation of the DiskOnChip into Linux

In order to prepare the DiskOnChip®2000 to boot Linux, the following software programs and tools are required:

- Linux (kernel version 2.0.29 – 2.0.35) should be installed on your HDD. It is possible to check the kernel version by typing the command `uname -r`.
- The Linux kernel sources should be installed in Linux (if you don't have them, refer to kernel-HOWTO at http://sunsite.unc.edu/LDP/).
- A DOS boot diskette or a HDD that boots into DOS.
- M-Systems' DiskOnChip®2000 DOS utilities diskette.
- M-Systems' DiskOnChip®2000 driver for Linux.

Note: If you are about to install Linux, make sure you also install the kernel sources package and that it is possible to pass a full compilation of your sources.

3.1 Utilities diskette content

The DiskOnChip®2000 utility diskette contains the following files:

`dformat`	DiskOnChip®2000 formatting utility
`dupdate`	Utility for updating DiskOnChip®2000 firmware
`docpmap`	Utility to retrieve information about the DiskOnChip®2000
`doc110.exb`	DiskOnChip®2000 firmware image. "110" is the firmware version, the actual diskette might contain higher versions of the firmware, e.g. doc111.exb, doc112.exb, etc.
`doc2.fff`	Alternative firmware image for the DiskOnChip®2000.

3.2 Linux driver for DiskOnChip®2000

The compressed file `driver.tgz` contains the following files:

/usr/src/linux/drivers/block/fl.##	DiskOnChip®2000 device driver for 2.0.## kernel version.
/tmp/doc-driver/doc-patch-2.0.##	Patch for kernel sources 2.0.##
/tmp/doc-driver/plilo	Linux loader, patched to use with the DiskOnChip®2000.
/tmp/doc-driver/lilo.conf	Lilo sample configuration file
/tmp/doc-driver/boot.b	Boot loader, udpated to use with

	DiskOnChip®2000
/tmp/doc-driver/copy2doc	Sample scripts, aid tool to create a root file system.
/tmp/doc-driver/pam.d/other	File meant only for Redhat & Caldera, defines permission access
/tmp/doc-driver/samplefs.txt	Sample root file system listing

4. Installing the DiskOnChip®2000 as an additional drive

Before the DiskOnChip®2000 can be used as the boot disk for Linux (see Chapter 5), it first needs to be installed as an additional disk in the system (Linux is booted from a HDD). This chapter describes how to prepare the DiskOnChip®2000 and Linux to configure the DiskOnChip®2000 as an additional disk in the system. In order to achieve this, the firmware on the DiskOnChip®2000 needs to be updated (par. 4.1) and then the DiskOnChip®2000 device driver needs to be integrated into Linux (par. 4.2). This is done in the following order:

1. Linux needs to be configured with the required devices (par. 4.2.1 Preparing Linux for integration).
2. The DiskOnChip®2000 driver needs to be added to the kernel and the kernel is recompiled (par. 4.2.2. Adding the driver to the kernel).
3. The compiled kernel is booted from HDD (par. 4.2.3. Booting the compiled kernel from HDD).
4. A Linux partition needs to be created on the DiskOnChip®2000 (par. 4.2.4. Creating a Linux partition on the DiskOnChip®2000)
5. A native Linux file system needs to created on the DiskOnChip®2000 (par. 4.2.5. Creating a native Linux file system on the DiskOnChip®2000)

4.1 Updating the firmware

Before the DiskOnChip®2000 can be used as the boot disk or as an additional disk in Linux, it needs to be formatted with the alternative firmware image. Please perform the following steps:

1. Plug the DiskOnChip®2000 into its socket and boot your system into DOS
2. Insert the DiskOnChip®2000 utility diskette into your floppy drive and type the following command to format the DiskOnChip®2000 with alternative firmware:

```
dformat /win:d000 /s:doc2.fff /y
```

Note: If you receive the error: “No DiskOnChip 2000 (R) was found at D000:0”, then run the DOS command `docpmap /i` to find out at which address the DiskOnChip®2000 is located.

4.2 Integrating the DiskOnChip®2000 driver into Linux

4.2.1 Preparing Linux for integration

In order to perform the following steps, you must be logged in as the superuser (root). To prepare Linux for integration of the driver, please perform the following steps:

1. The floppy drive must be active. To verify that this is actually the case, type:

   ```
   # mount
   ```

   If the `/dev/fd0` doesn't appear in the first column, type:

   ```
   # mount/dev/fd0 /mnt
   ```

   From this point it is assumed that the DiskOnChip®2000 utility diskette that contains the Linux driver is mounted at `/mnt`.

2. It is necessary that the kernel sources are installed. To check this, type:

   ```
   #ls /usr/src/linux
   ```

   If this directory exists, then the kernel sources are installed. If not, please refer to http://sunsite.unc.edu/LDP/ or type:

   ```
   zcat /usr/doc/HOWTO/Kernel-HOWTO.gz | more
   ```

3. In order to install the driver into Linux, unzip and untar the file `driver.tgz` as follows:

   ```
   # cd /tmp
   # mkdir temp
   # cd temp
   # tar -zxvf /mnt/driver.tgz
   # cp -rf . /
   # cd ..
   # rm -fr temp
   # cd /usr/src/linux/
   # patch -p0 < /tmp/doc-driver/doc-patch-2.0.##
   ```

   The last command will patch the current kernel sources in order to include the DiskOnChip®2000 driver for Linux (with ## = kernel version). Please notice that the patch for the kernel is working only on original kernel sources, and the patch is version specific. In case the utility `patch` is not available in your Linux environment, refer to chapter 6 "Troubleshooting".

4. Configure the kernel as follows:

   ```
   # cd /usr/src/linux
   # make menuconfig
   ```

   or `# make config` if the last command doesn't work.

Define your system by marking the correct devices. If you aren't sure what the purpose of the device is, then leave it as it is. Make sure that you mark the M-Systems DiskOnChip as 'Y'es under the 'Floppy, IDE, and other block devices'. After you are finished, exit and the new configuration will be saved.

Note: For further details about compiling the kernel or how to apply a patch, refer to http://sunsite.unc.edu/LDP/ or type:

```
# zcat /usr/doc/HOWTO/Kernel-HOWTO.gz | more
```

5. After finishing the configuration, it is necessary to remove the old object files. Type the following command:

```
# make clean
```

4.2.2 Adding the driver to the kernel

In order to make sure that the driver matches the kernel version, find out what the current version of the kernel is by typing the following command:

```
# uname -r
```

The driver can be compiled with kernel version 2.0.29 – 2.0.35.

Adding the driver to the kernel version is done by the following command:

```
# cp /usr/src/linux/drivers/block/flash_doc/fl.##
/usr/src/linux/drivers/block/flash_doc/fl.o
```

(with ## = kernel version)

Check dependencies:

```
# make dep
```

And compile the kernel (this may take up to 15 minutes):

```
# make zImage
```

If the kernel is compiled successfully, a similar message should be shown:

```
Root device is (3, 3)
Boot sector 512 bytes
Setup is 4332 bytes
System is 374 kB
Sync
Make[1]:Leaving directory '/usr/src/linux-2.0.32/arch/i386/boot'
```

If there are any compilation errors, refer to chapter 6 "Troubleshooting".

4.2.3 Booting the compiled kernel from HDD

Type the following commands:

```
# cp /usr/src/linux/arch/i386/boot/zImage  /doc2000
# vi /etc/lilo.conf
```

Please add the following lines at the bottom of the file (press 'INS' to edit the file and save and exit by typing <Esc>, ':', 'w', 'q'):

```
image = /doc2000
root = /dev/hda1
label = doc2000
read-only
```

Note: The device `/dev/hda1` points to HDD that Linux boots from. Hda1 is the *first* partition (1) on the *first* IDE hard disk (a), hdb2 is the *second* partition (2) on the *second* IDE hard disk (b), etc. In case it is not clear which device it is, look at the start of the file `lilo.conf` and search for the line `first root = ...`)

Run Lilo (Linux Loader) to create the map for the kernel and make sure that "doc2000' is listed (if not, please return to the beginning of paragraph 4.2.3):

```
# lilo
```

Make the inodes for the DiskOnChip®2000:

```
# cd /dev
# mknod fla b 62 0
# mknod fla1 b 62 1
# mknod fla2 b 62 2
# mknod fla3 b 62 3
# mknod fla4 b 62 4
```

Number 62 stands for major device number. Since it is hard coded into the driver, any other number wouldn't work.

To load the updated kernel with the driver for the DiskOnChip®2000, please perform the following steps:

1. Reboot the computer and load Linux.
2. When the Lilo prompt is displayed, press <Ctrl> or <Alt> or <Tab>. The screen will show :

   ```
   Lilo boot:
   ```

3. Type "doc2000" to load the recompiled kernel:

   ```
   Lilo boot:doc2000
   ```

4.2.4 Creating a Linux partition on the DiskOnChip®2000

In order to create a Linux partition on the DiskOnchip®2000, all the DOS partitions on the DiskOnChip®2000 need to be removed and a native Linux File system [`ext2`] needs to be created. This step is vital in order to make the DiskOnChip®2000 bootable.

Run the `fdisk` utility to create a Linux partition that spans the enitre DiskOnChip®2000. Mark the partition as bootable and write the partition table to the DiskOnChip®2000. Type the following command to run the `fdisk` utility:

```
# fdisk /dev/fla
```

Note: The current release of the driver does not support multiple partitions.

In order to delete the existing partition on the DiskOnChip®2000 and to create a new one, perform the following steps within the `fdisk` utility:

1. To display the contents of the partition table:

```
Command(m for help):p
```

2. To delete all existing partitions. Enter each partition number for deletion:

```
Command(m for help):d
```

3. Create a new Linux native partition:

```
Command(m for help): n
Command action e extended
               p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-XXX): 1
Last cylinder or +size or +sizeM or +sizeK ([1]-XXX):XXX
```

4. Change the type of the partition to `Linux native`:

```
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): 83
```

5. Make the partition bootable:

```
Command (m for help): a
Partition number (1-4): 1
```

6. Recheck the partition table:

```
Command (m for help): p

Disk /dev/fla: 16 heads, 9 sectors, 1002
Cylinder units = cylinders of 144 * 512 bytes

Device     boot  begin  Start  End   Blocks Id  System
/dev/fla1    *     1      1    1002  72139+ 83  Linux native
```

7. Save the new partition table (disregard any `fdisk` warnings):

```
Command (m for help): w
```

Note: For further information on the `fdisk` utility, refer to the man pages.

4.2.5 Creating a native Linux file system on the DiskOnChip®2000

In order to initialize the file system on the newly created Linux partition on the DiskOnChip®2000, type the following command:

```
# mke2fs /dev/fla1
```

Note: When a small capacity DiskOnChip®2000 is used (4MB or smaller), more space for inodes need to be allocated. Type the following command:

```
# mke2fs –i 2048 /dev/fla1
```

Mount the file system to a directory, in order to make it accessible:

```
# mkdir /diskonchip
# mount /dev/fla1 /diskonchip
```

From now onwards it is possible to use the DiskOnChip®2000 as an additional disk in your system.

5. Booting Linux from the DiskOnChip®2000

Being able to boot Linux from the DiskOnChip®2000 is of great importance. It makes it possible to use the DiskOnChip®2000 as the only disk in the system, holding the OS itself in addition to all other applications and files.

In order to make a block device bootable on Linux there are several steps that need to be taken. The kernel and the Lilo program should be copied to the block device, and a root file system needs to be created. Creating a root file system on Linux is necessary, as from this root file system the kernel is activating several programs, such as:

`Init`	Initialize all processes
`Swapon`	Activate swapping
`Mount`	Mount the root and proc filesystems
`Sh`	Shell

Note: For further details about, refer to http://sunsite.unc.edu/LDP/ or type the following command: `#zcat /usr/doc/HOWTO/Bootdisk-HOWTO.gz | more`

Please notice that the DiskOnChip®2000 firmware collides with Lilo. This means that it

is not possible to load Linux from the HDD after the original firmware is restored (see par. 5.1.4). This doesn't mean that the HDD is non-functional, it only means that the alternate firmware needs to be reloaded (see par. 4 .1).

If it is required to boot Linux from both the HDD and the DiskOnChip®2000, it is necessary to use both pLilo and `boot.b` (provided with the DiskOnChip®2000 driver).

For further details, refer to Appendix II.

5.1 Creating a root file system

5.1.1 Introduction

This section is based on Bootdisk-HOWTO and other experiments. Since each distribution has different file locations and different installations, it is possible that you won't succeed booting Linux the first time from the DiskOnChip®2000. When you receive errors, follow each error and try to fix things, but reload Linux from the HDD and remount the DiskOnChip®2000 every time you do.

Note: There are several programs that create a root file system. It is not possible to say that any of them is complete right now, but they are worth trying. The following program is recommended:

Yard:
Yard creates rescue disks (also called boot disks) for Linux. A rescue disk usually contains utilities for diagnosing and manipulating hard disks and file systems. It is used when it is not possible (or if it is not required) to boot from your HDD. This package contains mainly perl scripts. Refer to http://www.croftj.net/~fawcet/yard/

Note: For a sample root file system, please refer to the file `/tmp/doc-driver/samplefs.txt` which is part of the compressed file `driver.tgz`

5.1.2 Overview

A root file system must contain everything that is needed to support a full Linux system. To achieve this, the disk must include the minimum requirements for a Linux system:

- Basic file system structure
- Minimum set of directories: /dev, /proc, /bin, /etc, /lib, /usr, /tmp
- Basic set of commands: sh, ls, cp, mv, etc.
- Minimum set of config files: rc, inittab, fstab, etc.
- Devices: /dev/hd*, /dev/tty*, /dev/fd0, etc.
- Runtime libraries to provide basic functions used by utilities.

5.1.3 Populating the file system

From this point onwards it is assumed that the DiskOnChip®2000 is mounted to the `/diskonchip` directory.

The sample script `copy2doc` that is provided with the driver is located in directory `/tmp/doc-driver`. This sample script is created for RedHat 5, SuSE 5.2 and Caldera,

and is customized for RedHat 5. It might be necessary to change the script according to your distribution. Files that are specific to a distribution can be found in the script with the distribution name written after it. To exclude a file, make sure the '#' mark is present at the beginning of the line. To include a file, make sure the '#' mark is removed.

The sample script only copies the basic files that are needed for booting. For any other operation it is necessary to add more files.

To create a root file system based on the sample script `copy2doc`, perform the following steps:

1. Go to the `/diskonchip` directory:

   ```
   # cd /diskonchip
   ```

2. If necessary, customize the sample script (for other distributions):

   ```
   # vi /tmp/doc-driver/copy2doc
   ```

3. Run the script:

   ```
   # sh /tmp/doc-driver/copy2doc
   ```

4. Create the list of files to be mounted:

   ```
   # vi /diskonchip/etc/fstab
   ```

5. Press <INS> to start editing and insert the following lines:

   ```
   /dev/fla1   /       ext2  defaults 1 1
   /proc       /proc   proc  defaults 0 0
   ```

   Press <ESC>, ':', 'w', 'q' to save the file. It is possible to add more devices here (for more information, refer to the man pages).

6. For RedHat and Caldera it is also necessary to copy the configuration file for the pam library (responsible for making authentic users):

   ```
   # cp /tmp/doc-driver/pam.d/other
         /diskonchip/etc/pam.d/other
   ```

Note: Appendix I explains how to manually create your own root file system. Although this is more complicated than using the above mentioned sample script, it is highly recommended.

All modules should be placed in `/lib/modules/`. It is necessary to at least include the programs `insmod`, `rmmod` and `lsmod`. If it is required to load the modules automatically, then also include `modprobe`, `depmod` and `swapout`. When using `kerneld`, include it along with `/etc/conf.modules`.

Some system programs, such as `login`, complain when the file `/var/run/utmp` and

the directory `/var/log` do not exist. To solve this, type the following commands:

```
# mkdir -p /diskonchip/var/{log,run}
# touch /diskonchip/var/run/utmp
```

After all the needed libraries and programs are set up, run `ldconfig` to remake `/etc/ld.so.cache` on the root file system. The cache tells the loader where to find the libraries. To remake `ld.so.cache`, type the following command:

```
# cd /diskonchip

# chroot /diskonchip  /sbin/ldconfig
```

The command `chroot` is necessary, because `ldconfig` always remakes the cache for the root file system.

5.1.4 Copying the kernel, updating the boot sector and rebooting

To copy the kernel and to update the boot loader files, type the following commands:

```
# mkdir /diskonchip/boot
# cp /usr/src/linux/arch/i386/boot/zImage
    /diskonchip/boot/doc2000
# rdev /diskonchip/boot/doc2000 /dev/fla1
# cp /tmp/doc-driver/plilo /diskonchip/sbin
# cp /tmp/doc-driver/boot.b /diskonchip/boot
# cp /tmp/doc-driver/lilo.conf /diskonchip/etc
# /diskonchip/sbin/plilo -C /diskonchip/etc/lilo.conf -i
     /diskonchip/boot/boot.b –m /diskonchip/boot/map
```

Verify that after the last command the device `doc2000` is listed on the screen.

If it is required to load other partitions, then the file `/diskonchip/etc/lilo.conf` should be edited.

Note: pLilo is the patched Lilo, in order for the DiskOnChip®2000 firmware not to collide with the Linux bootloader.

The final steps in the process of making the DiskOnChip®2000 bootable for Linux are as follows:

1. Unmount the DiskOnchip®2000

```
# cd /
# unmount  /dev/fla1
```

2. Reboot and load DOS, and reinstall the original firmware:

```
A:> dupdate /win:D000 /s:DOC110.EXB
```

3. Reboot the machine and disable the HDD in the BIOS setup or make the DiskOnChip®2000 the first boot device in the system by using the following command :

```
A:> dupdate /win:D000 /s:DOC110.EXB /FIRST
```

4. . Linux will boot now from the DiskOnChip®2000.

6. Troubleshooting

1. Adding more programs to Linux root file system

If the DiskOnChip®2000 boots Linux without a problem and it is required to add more programs to the Linux root file system, then mount the HDD and copy the needed files.

2. DiskOnChip®2000 does not boot Linux

There are several errors that you can encounter during boot:

1. If the DiskOnChip®2000 does not boot at all, please follow all the instructions from the start of this Installation manual. Remember to also update the original firmware of the DiskOnChip®2000 (`doc110.exb` for example) with the alternate firmware (`doc2.fff`) in order to boot Linux from your HDD.
2. If the kernel boots, but it gets stuck on:

```
VFS: Unable to mount -.
```

   Most likely you forgot to do:

```
# rdev /diskonchip/boot/doc2000  /dev/fla1
```

3. If the DiskOnChip®2000 boots and the kernel is loading, but it gets stuck after:

```
VFS: Mounted root (ext 2 filesystem) readonly.
```

   Most likely the `init` program or some if its configuration files weren't copied.

3. Can't log in

If you can't login when booting Linux from the DiskOnChip®2000, make sure that:

- your default shell is installed;
- the pam libraries were placed as explained in par. 5.1.3 (only for RedHat and Caldera);

Refer to Appendix II in order to solve this problem.

4. Kernel doesn't compile correctly

If the kernel doesn't compile correctly and the problem is not caused by the DiskOnChip®2000 driver, please refer to http://sunsite.unc.edu/LDP/ or type:

```
#zcat /usr/doc/HOWTO/Kernel-HOWTO.gz | more
```

7. Additional information and Tools

Additional information about the DiskOnChip®2000, including application notes, can be found at http://www.m-sys.com.

Additional tools and documents are listed in the following table:

Document/Tool	**Description**
AP-DOC-10	Designing with the DiskOnChip®2000
AP-DOC-11	Write Protecting the DiskOnChip®2000
AP-DOC-16	Using the DiskOnChip®2000 with QNX
AP-DOC-17	Using the DiskOnChip®2000 with Windows CE
AP-DOC-19	Using the DiskOnChip®2000 with Windows 95
AP-DOC-20	DiskOnChip®2000 Boot Developer Kit
DiskOnChip2000 Data Sheet	DiskOnChip®2000 Data Sheet
DiskOnChip2000 Utilities	DiskOnChip®2000 Utilities User Manual
DiskOnChip2000-EVB	DiskOnChip®2000 Evaluation Board
DiskOnChip2000-PIK	DiskOnChip®2000 Programmer and Integrators Kit
DiskOnChip-GANG	8 Socket Gang Programmer

Appendix I : Making a root file system

In order to create your own root file system, please perform the following steps:

1. Make the following directories:

```
# cd /diskonchip
# mkdir bin dev etc lib mnt proc sbin tmp usr var
```

2. Create devices in the `/dev` directory. You can either do this manually or just copy the `/dev` directory from the HDD. If you wish to save space, it is possible to remove non-required devices, i.e. if you don't have a SCSI drive, then remove all the sd* devices.

```
# cp -dpR  /dev  /diskonchip
```

   This commands copies many unnecessary inodes to the DiskOnChip®2000. Removing them causes no problem, as long as you make sure that the ones listed in the sample file system are present.

3. Copy and configure the files in the `/etc` directory:

```
# cp -dr  /etc/rc.d  /diskonchip/etc  (for RedHat)
```

   or

```
# cp -dr /sbin/init.d  /diskonchip/sbin  (for SuSE)
# cp -d /etc/inittab  /diskonchip/etc
```

4. Copy the password file and make sure that each user has it's default shell installed:

```
# cp /etc/passwd  /diskonchip/etc
# cp /etc/shadow /diskonchip/etc (possible that you don't have this)
# cp /etc/group   /diskonchip/etc
```

5. Create the file `/etc/fsstab` that contains the list of files to be mounted:

```
# vi /diskonchip/etc/fstab
```

   Press <INS> to start editing and insert the following lines:

```
/dev/fla1 /         ext2      defaults 1 1
/proc     /proc     proc      defaults 0 0
```

   Press <ESC>, ':', 'w' and 'q' to save the file.

   It is possible to add more devices here. For more information, refer to the man pages.

6. There are several programs that need to be copied in order to have a functional environment. Other programs are not as important, although it would be rather difficult to work without them. All other programs that are not listed below are considered optional. Copy these programs to the directories `/bin` or `/sbin` as follows:

```
# cp /bin/{program_name}
     /diskonchip/bin/{program_name}
```

/bin directory:

cat	echo	mount
chmod	hostname	mv
chown	kill	ps
cp	ln	rm
cut	login	rmdir
dd	ls	sh
df	mkdir	su
dircolors	mke2fs	sync
du	mknod	umount
e2fsck	more	uname

/sbin directory:

mingetty	shutdown
halt	swapoff
init	swapon
ldconfig	telinit
mkswap	update
reboot	runlevel
rdev	

The filename of `mingetty` varies with the distribution, i.e. RedHat and SuSE use `mingetty`, Slackware uses `agetty`. To find out what the name of this file in your distribution is, perform a grep on "getty":

```
# grep getty /etc/inittab
```

7. The `/lib` directory contains all the shared libraries and loaders. Only the appropriate libraries need to be copied to the `/lib` directory. In order to check which libraries are needed, type the following command for each file in these 2 directories:

```
# ldd /sbin/{filename}
```
or
```
# ldd /bin/{filename}
```

For example:

```
# ldd /sbin/mke2fs
      libext2fs.so.2 → /lib/libext2fs.so.2
      libcom_err.so.2 → /lib/libcom_err.so.2
      libuuid.so.1 → lib/libuuid.so.1
      libc.so.5 → /lib/libc.so.5
```

This will show which libraries are needed for the program `mke2fs`. In this example, it would be necessary to copy the following 4 libraries.

```
# cp /lib/ext2fs.so.2  /diskonchip/lib
# cp /lib/libcom_err.so.2  /diskonchip/lib
```

```
# cp /lib/libuuid.so.1  /diskonchip/lib
# cp /lib/libc.so.5  /diskonchip/lib
```

In case you have a long list of files to be copied, it is also possible to run the following command:

```
# ldd /bin/* > lib_list
# more lib_list
```

7. Copy the library loaders as follows:

```
# cp lib/ld.so /diskonchip/lib          (a.out loader)
# cp /lib/ld_linux.so  /diskonchip/lib  (elf loader)
```

Note: It is possible to use objcopy to reduce the size of the libraries. For example:

```
# objcopy -strip-debug  /diskonchip/lib/lib.so.5
```

Appendix II: Booting from a HDD when DOC firmware is active

Since Lilo and the DiskOnChip®2000 firmware share the same memory area, the system will hang during boot when using an unpatched Lilo.

At present the problem is solved by updating the Lilo. The existing boot loader needs to be updated as follows:

```
# /tmp/doc-driver/plilo -i  /tmp/doc-driver/boot.b
```

This command uses the patched Lilo supplied by the driver and patched `boot.b`, and your default /etc/lilo.conf. Make sure that you run this command in Linux that booted from HDD.

USA - M-Systems Inc., Phone: 510-413-5950, Fax: 510-413-5980, email: info@m-sys.com
Taiwan - M-Systems Asia, Phone: 886-2-25501741, Fax: 886-2-25501745
Japan – M-Systems Japan, Phone: 81-3-3445-9042, Fax: 81-3-3445-9045
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266
Israel - M-Systems Ltd, Phone: 972-3-647-7776, Fax: 972-3-647-6668
http://www.m-sys.com

Series 2000

AP-DOC-019
Application Note

Using the DiskOnChip® 2000 with Windows® 95

Dimitry Shmidt
Software Engineer

Apr-98
91-SR-005-09-7L REV. 1.0

M-Systems
Flash Disk Pioneers

1. Introduction

The DiskOnChip 2000 hooks interrupt INT13h and uses it to emulate a hard disk. It is possible to use the DiskOnChip 2000 under Windows 95, just like any other hard drive. However, the capacity of the DiskOnChip 2000 should be large enough to contain Windows 95 operating system files, user's applications and data.

Being a memory mapped device, the DiskOnChip 2000 requires special attention when installing Windows 95 on it. This is required in order to prevent Windows95 from using the memory window in which the DiskOnChip 2000 resides. This is especially important when using the DiskOnChip in a "Multi DiskOnChip" configuration (more than one DiskOnChip 2000 emulating a single disk drive), as each DiskOnChip 2000 has its own upper memory window. Please refer to AP-DOC-013 ("Designing a Multi DiskOnChip 2000 Disk") for further information.

Using a small capacity DiskOnChip with Windows 95 can be done by cutting the size of the Windows 95 image. Please refer to section 3 of this application note for more information.

The terms "DiskOnChip 2000" and "Multi DiskOnChip 2000" are used interchangeably in this application note.

2. Installation

Follow the instructions below for installing Windows 95 on a DiskOnChip 2000. Please make sure to follow the instructions with caution, otherwise you might face unpredictable consequences.

1. Preparing the DiskOnChip 2000

1.1 Performing Low-Level Format

If a single DiskOnChip 2000 is being used, nothing has to be done in this step.

When using the "Multi DiskOnChip 2000", formatting should be done according to the instructions in AP-DOC-013.

1.2. Making the DiskOnChip 2000 Bootable

Follow the next steps in order to format the DiskOnChip 2000 in DOS and make it bootable. You will have to use a bootable floppy that contains the DOS system files and the *FORMAT.EXE* utility.

1.2.1. Booting up the system

Place a bootable floppy in the drive and boot the machine. You will get a drive letter for the DiskOnChip 2000, typically "*C:*". If you have other hard drives hooked, the drive letter might be different.

1.2.2. Formatting the DiskOnChip 2000

Issue the DOS format command: *FORMAT C: /S*

The /S flag is required to make the DiskOnChip 2000 bootable.

Remove the bootable floppy from the driver and reboot the machine.

1.3. Copying CD-ROM Drivers (optional)

In case you want to install Windows 95 from a CD-ROM, make sure that the *autoexec.bat* and/or *config.sys* files on the DiskOnChip 2000 contain the DOS CD-ROM drivers. Follow the instructions in your CD-ROM manual for installation of these drivers. Once installation is complete, reboot the machine and proceed to the next step. This step is necessary only for the first Windows 95 installation you do on the target platform.

2. Installing Windows 95 on the DiskOnChip 2000

2.1. Running the Windows 95 setup program

Windows 95 setup can be found either on floppies or on a CD-ROM. (Note that only old versions of Windows 95 are available on floppy disks)

Run the Windows 95 setup program and follow the instructions. When asked to select the installation type, select “Compact”. This will copy the minimal subset of Windows 95 files to the DiskOnChip 2000.

Proceed with the setup process until it reaches its final step, offering you to reboot the system. Before you continue, carefully read the following instructions.

It is crucial that the first boot after the Windows 95 setup will be done from a bootable floppy and not from the DiskOnChip 2000. Before rebooting, Windows 95 checks to see that there is no floppy in the drive. You will have to keep the floppy half inserted, then immediately after pressing “Reboot”, fully insert it into the drive. The screen will vanish at this time as the reboot process begins.

2.2. Changing the Windows 95 Initialization File

Once the system has come up, you will get the “*A:*” prompt. Change the current directory to *C:\WINDOWS* and edit the file *SYSTEM.INI*. Add the following line to the *[386enh]* section:

emmexclude=D000-DFFF

This will prevent Windows 95 from accessing the DiskOnChip 2000 memory window.

Note: If your DiskOnChip 2000 is located in a different address range, change the above address range accordingly.

After you finished changing the initialization file, remove the bootable floppy and reboot the machine.

2.3. Finishing the Windows 95 Setup

After the system is up, Windows 95 will continue the setup process. Follow the instructions on the screen until it is finished. From this point on you can freely use Windows 95 installed on the DiskOnChip 2000.

3. Minimizing the Win95 Footprint for the DiskOnChip 2000

The core facilities of Windows 95 occupy just a fraction of its overall code. By stripping away facilities that are not required for the target system, it is possible to cut the size of the Windows 95 image in half, producing relatively small configurations.

For additional information please refer to the article “Minimizing the Windows 95 Footprint for Embedded Systems Applications” by Sean Liming, AnnaSoft Corp., Microsoft Embedded Review, September 16, 1996.

4. Additional information and Tools

Document/ Tool	**Description**
AP-DOC-10	Designing with the DiskOnChip 2000
AP-DOC-17	Using the DiskOnChip 2000 with Windows CE
AP-DOC-16	Using the DiskOnChip 2000 with QNX
AP-DOC-11	Write Protecting the DiskOnChip 2000
DiskOnChip 2000 Data Sheet	DiskOnChip Data Sheet
DiskOnChip 2000 Utilities	DiskOnChip 2000 Utilities User Manual
DiskOnChip2000-EVB	DiskOnChip Evaluation Board
DiskOnChip2000-PIK	DiskOnChip Programmer and Integrators Kit
DiskOnChip-GANG	8 Socket Gang Programmer

USA - M-Systems Inc, Phone: 510-413-5950, Fax: 510-413-5980, email: info@m-sys.com
Taiwan - M-Systems Asia, Phone: 886-2-25501741, Fax: 886-2-25501745
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266
Israel - M-Systems LTD, Phone: 972-3-647-7776, Fax: 972-3-647-6668
http://www.m-sys.com

M-Systems
Flash Disk Pioneers

DiskOnChip®2000
Quick Installation Guide

Dear Customer,

Thank you for purchasing the DiskOnChip 2000. This guide is designed to assist you in a quick and easy installation of the DiskOnChip 2000 in your target platform .

1. DiskOnChip 2000 Installation Instructions

1. **Make sure the target platform is powered OFF**
2. **Plug the DiskOnChip 2000 device into its socket. Verify the direction is correct (pin 1 of the DiskOnChip 2000 is aligned with pin 1 of the socket)**
3. **Power up the system**
4. **During power up you may observe the messages displayed by the DiskOnChip 2000 when its drivers are automatically loaded into system's memory**
5. **At this stage the DiskOnChip 2000 can be accessed as any disk in the system**
6. **If the DiskOnChip 2000 is the only disk in the system, it will appear as the first disk (drive C: in DOS)**
7. **If there are more disks besides the DiskOnChip 2000, the DiskOnChip 2000 will appear by default as the last drive, unless it was programmed as first drive. (please refer to the DiskOnChip 2000 utilities user manual)**
8. **If you want the DiskOnChip 2000 to be bootable:**

   a - copy the operating system files into the DiskOnChip by using the standard DOS command (for example: sys d:)

   b - The DiskOnChip should be the only disk in the systems or should be configured as the first disk in the system (c:) using the DUPDATE utility

2. Additional information and assistance

1. **Visit M-Systems Web site at www.m-sys.com where you can find Utilities Manual, Data Sheet and Application Notes. In addition, you can find the latest DiskOnChip 2000 S/W Utilities**
2. **Contact your dealer for technical support if you need additional assistance, and have the following information ready:**
   - **Product name and serial number**
   - **Description of your computer hardware (manufacturer, model, attached devices, etc.)**
   - **Description of your software (operating system, version, application software, etc.)**
   - **A complete description of the problem**
   - **The exact wording of any error messages**

USA - M-Systems Inc, Phone: 510-413-5950, Fax: 510-413-5980, email: info@ccm.msyscal.com
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266, email: info@msys.nl
Israel - M-Systems LTD, Phone: 972-3-647-7776, Fax: 972-3-647-6668, email: info@m-sys.com
http://www.m-sys.com
91-SR-002-06-7L 7/97

Series 2000

User Manual

DiskOnChip®2000

Utilities

July-97

91-SR-002-02-8L REV. 2.0

M-Systems

Flash Disk Pioneers

1. Introduction

M-Systems' *DiskOnChip2000* is a new generation of high performance single-chip Flash Disk. The DiskOnChip MD2000 provides a Flash Disk in a standard 32-pin DIP package.

This unique data storage solution offers a better, faster, and more cost-effective Flash Disk for Single Board embedded systems, Internet devices and portable applications with limited space and modest disk capacity requirements.

The new DiskOnChip 2000 provides a Flash Disk (as BIOS expansion) which does not require any bus, slot or connector. Simply insert the DiskOnChip2000 into a 32-pin socket on your CPU board, with a minimal installation cost, and you have a bootable Flash Disk.

Various Operating Systems are supported by DiskOnChip 2000 : DOS, Windows, Win95. Additional support offered: pSOS+, QNX, VxWorks and others.

DiskOnChip2000 is the optimal solution for Single Board Computers - it's a small, fully functional, easy to integrate, plug-and-play Flash Disk with a very low power consumption.

This manual describes the software utilities for the DiskOnChip 2000.

2. DFORMAT

Before TrueFFS can access a flash media, the media must be formatted, just as a floppy disk must be formatted. Formatting initializes the media and writes to it a new and empty DOS file system. When formatting is complete, the media contains only a root directory.

The DiskOnChip is fully tested and formatted before the product is shipped, but it can be formatted more than once. Each time it is formatted, naturally all data on the media is destroyed. When reformatting, the boot-image is **retained** by default.

The DFORMAT syntax is:

```
Usage: DFORMAT {drive-letter | /WIN:segment} [/SIZE:size]
[/USE:nnn]
[LABEL:label] [/DOSVER:n] [/SPARE:n] [/Y]
```

The DFORMAT options are:

drive-letter DOS drive letter of the TrueFFS drive.

/WIN:Segment	Memory address in which the DiskOnChip is located. Use either this flag or the ***drive-letter*** flag.
/LABEL:*label*	A string to be used as the DOS label of the formatted media.
/SIZE:*size*	The size of the flash media to be formatted (including the install partition). By default the entire media is formatted by DFORMAT. This option limits the formatted size.
/USE:*nnn*	Percentage of available space on the flash media to be used for file storage. ***nnn*** can be any number from 1 to 100. Default is 99 (99%). The value of this option may affect the write performance of TrueFFS.
/DOSVER:*dos-major-version*	Format for a target system running the specified DOS version. The default is the current DOS version (the one on which DFORMAT is executed). For example, /DOSVER:3 formats for DOS 3.x. Valid values are 1 to 6.
/SPARE:*n*	Number of spare units. Default is 1. A value 0 selects a WORM (Write Once Read Many)
/Y	Do not pause for confirmation before beginning to format.

Note: All sizes specified in DFORMAT options are in bytes if specified as simple numbers, in KBytes if specified with the suffix **K**, or in megabytes if specified with the suffix **M**.

Example 1: DFORMAT C:

Formats the DiskOnChip which is used as drive C.

Example 2: DFORMAT /WIN:D000

Formats the DiskOnChip which is located at memory address hex D000. If any other hard disk is present in the system, the DiskOnChip will be identified as drive D:

2.1 Configuring the DiskOnChip as a Bootable Disk

The DiskOnChip fully supports the BOOT capability. In order for the DiskOnChip to be bootable, it should be DOS formatted as bootable, like any floppy or hard disk that required to be bootable.

Example: SYS D:

Change the disk into bootable (assuming the DiskOnChip is disk D):

3. DUPDATE - Updating DiskOnChip 2000 Firmware

In case a firmware update will be required, M-Systems will deliver a new .EXB file which should be written into the firmware portion of the Flash media within the DiskOnChip, using the DUPDATE utility.

DUPDATE requires that the DiskOnChip will be already programmed with previous firmware file programmed into, which is the default Since the DiskOnChip is shipped fully tested and programmed.

The DUPDATE syntax is:

```
DUPDATE [drive-letter | /WIN:Segment] /S:BootImage /FIRST
```

drive-letter DOS drive letter of the TrueFFS drive.

/WIN:Segment Memory address in which the DiskOnChip is located. Use either this parameter or the drive-letter. The segment should be specified in Hex (e.g. /win:d000)

/S:BootImage The boot image file of the new firmware to be written to the DiskOnChip. Usually the file type is .EXB

/FIRST Use this flag to program the DiskOnChip to be the first disk if more disks are installed in the system. This flag has no effect if the DiskOnChip is the only disk in the system. The /S parameter must be supplied when /FIRST flag is used.

Example 1: DUPDATE C: /S:DOC2000.EXB

Program the firmware which is supplied in DOC2000.EXB file into the DiskOnChip located as drive C:

Example 2: DUPDATE /WIN:D000 /S:DOC2000.EXB

Program the firmware which is supplied in DOC2000.EXB file into the DiskOnChip which is located at memory address hex D000.

Example 3: DUPDATE /WIN:D000 /S:DOC2000.EXB /FIRST

Program the firmware which is supplied in DOC2000.EXB file into the DiskOnChip which is located at memory address hex D000. The DiskOnChip will be the first drive (C:) in case a hard disk is available in the system.

Example 4: DUPDATE /WIN:D000 /S:DOC2000.EXB

Program the firmware which is supplied in DOC2000.EXB file into the DiskOnChip which is located at memory address hex D000. The DiskOnChip will be the last drive in the system (e.g. D: if one magnetic hard drive is already configured).

4. DINFO

The DINFO Information utility provides background information regarding the DiskOnChip 2000, and the environment in which it is working. DINFO reports:

- TrueFFS drive letters
- Installed software and its version compliance.
- The size of the Flash media.

The DINFO syntax is:

```
DINFO
```

Example:

DINFO Search the system for DiskOnChip.

Following is the report that was generated in a specific system.

```
DINFO Version 3.3.3 for DiskOnChip 2000 (V1.00)
Copyright (C) M-Systems, 1992-1997
DiskOnChip 2000(R) found at D000:0000
---------------------------------------
Disk statistics:
  Software version: 3.3.03
  Drive letter : D
  Disk size    : 1,992 Kbytes
  Boot size    : 44 Kbytes

Flash media statistics :
  Chip size    : 2,048 Kbytes
  No Of Chips  : 1
  Chip type    : Toshiba TC5816FT
  Total units  : 512
  Free  units  : 494
  Unit size    : 4,096 bytes
  Interleaving : 1
```

These DINFO results show the following:

- A 2MB DiskOnChip
- Programmed with firmware version 3.3.03
- It was assigned a drive letter D:
- The disk size after format is 1,992 KB
- Space allocated for Boot is 44KB
- The Flash media is composed of one Flash device, manufactured by Toshiba
- The Flash media is composed of 512 units
- 494 units are free.
- Each unit is 4.096 bytes

5. Duplicating DiskOnChip 2000

Copying DiskOnChip device is the procedure of copying a “source” DiskOnChip contents into an “image file”, then copying the “image file” contents into as many target DiskOnChip devices as required. All target DiskOnChip devices will have exactly the same contents as the source DiskOnChip, which means they will have exactly the same functionality when plugged into target platform. The only limitation for this process is that all target DiskOnChip devices must have the same capacity of the “source” DiskOnChip. For example: if the “source” DiskOnChip has a 12MB capacity then the “target” DiskOnChip should have 12MB capacity as well.

The duplicating process includes 3 stages:

1. Prepare “source” DiskOnChip.
2. Copy “source” DiskOnChip into an image file.
3. Copy the image file into as many as required “target” DiskOnChip devices.

5.1 Stage 1: Creating the “source” DiskOnChip

The source DiskOnChip includes all target application files. Usually, it will be bootable (see chapter 4.1). The following commands are usually used in order to prepare the “source” DiskOnChip:

1. Format DiskOnChip with DFORMAT utility in target platform, using version 1.04 or above.
2. Copy all target application files onto the DiskOnChip.
3. If required, make the DiskOnChip bootable (this is not a must - but mostly it is required. refer to chapter2.1) - this operation must be done in the target platform (not in PIK).

After the source device was properly prepared, follow the guidelines described below in order to duplicate it as many times as required.

5.2 Stage 2: Copy the “source” DiskOnChip into image file

At this stage, the source DiskOnChip includes all target application files, and it is ready to be duplicated as many times as required. Each duplicated copy will function on the target platform, as the “source” DiskOnChip.

Use GETMIMG utility to copy the “source” DiskOnChip contents into an image file on disk, to be used later as source file for duplications.

1. Run GETMIMG image_file_name.
   (for example: GETMIMG MYDOC.SRC)

5.3 Stage 3: Copy the image file onto “target” DiskOnChip devices

At this stage, the contents of the “source” DiskOnChip is stored in the disk in what we call “image file”. Copying this image file into target DiskOnChip, will result with identical DiskOnChip target device to the one that used as “source”. Use the PUTMIMG utility to perform this task:

1. Power off the system.
2. Insert a target DiskOnChip with the same capacity as the source DiskOnChip into it’s socket.
3. Power on the system
4. Run: PUTMIMG image_file_name.
   (for example: PUTMIMG MYDOC.SRC)
5. The target DiskOnChip will have the exact contents and functionality as the source DiskOnChip when this operation is done. Repeat steps 1 to 3 for each additional target DiskOnChip.

All DiskOnChip devices programmed according to the above procedure are ready to be plugged into the target platforms, and will function exactly the same as the source DiskOnChip.

M-Systems
Flash Disk Pioneers

Series 2000

M-Systems Confidential
Advance Information

DiskOnChip®2000
MD2200, MD2201 Data Sheet

Features

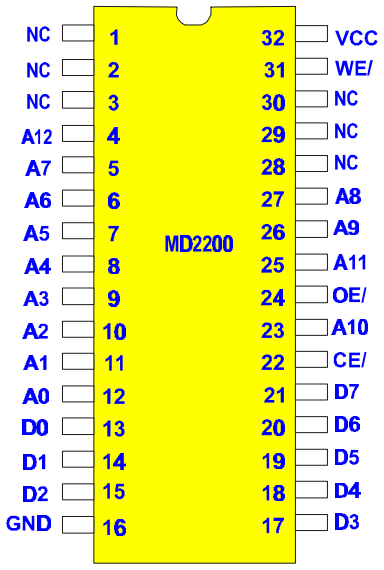
- **Single chip plug-and-play Flash Disk**
- **2 - 144MB capacity (1GB in 2H00)**
- **Simple, easy to integrate interface**
- **32-pin DIP JEDEC standard EEPROM compatible pin-out**
- **Memory window size - 8KB**
- **Embedded *TrueFFS*® software provides:**
  - **Full hard disk read/write capability**
  - **Third generation wear leveling**
  - **Automatic management of bad blocks**
- **Operates with DiskOnChip-OSAK in O/S-less environments**
- **EDC/ECC for high data reliability**
- **Full boot capability**
- **Low power consumption**
- **Broad O/S support: DOS, Windows, Windows 95, Windows NT4.0/5.0, Windows CE; Additional support offered: pSOS+, QNX, VxWorks and others**
- **DiskOnChip-OSAK, ANSI-C source code kit, supports O/S-less environments**
- **Single 3.3V or 5V supply**

1. General Description

The DiskOnChip 2000 product line provides a single chip, solid-state flash disk in a standard 32-pin DIP package. The DiskOnChip is intended for use in embedded and portable computers that have limited space and minimal power consumption requirements. By placing the DiskOnChip in a standard socket, physical space requirements are reduced. Unlike standard IDE drives, no cables or extra space is required. The DiskOnChip is a solid-state disk with no moving parts, resulting in a significant reduction in power consumption and an increase in reliability. The DiskOnChip is a small, plug-and-play Flash disk. It is easy to use and reduces integration overhead.

The DiskOnChip family of products is available in capacities ranging from 2MB up to 144MB, unformatted. In future versions the capacity will be dramatically increased (up to 1GB in 2nd half of the year 2000), yet the same pin-out will be retained. Therefore the socket on the target platform does not have to be changed to accommodate larger capacities. In order to manage the disk, the DiskOnChip includes TrueFFS®, M-Systems' Flash File System. The DiskOnChip package is pin-to-pin compatible with standard 32-pin EEPROM devices.

The DiskOnChip is shipped as a plug-and-play device, i.e. it is programmed, formatted and ready to be used. Future software upgrades and formatting can be done on the target platform. There is no need to remove the DiskOnChip from its socket in order to modify its contents or to reformat it.



Pin Name	Description	Pin Number	Direction
A0 - A12	Address bus	4-12,23,25-27	Input
D0 - D7	Data bus	13-15,17-21	I/O
CE/	Chip Enable	22	Input
OE/	Output Enable	24	Input
WE/	Write Enable	31	Input
NC	Not Connected	1,2,3,28,29,30	
VCC	Power	32	
GND	Ground	16	

Figure 1 - MD2200, MD2201 Pin Out

2. Operation In a PC Compatible Platform

2.1 Functional

In PC compatible platform, the DiskOnChip is mounted in a 32-pin DIP socket, compatible with a standard EEPROM. The DiskOnChip should be mapped into the expansion BIOS memory space of the PC. During the boot process, the DiskOnChip loads its software into the PC's memory and installs itself as an additional disk drive in the system. When the PC's operating system is loaded, the DiskOnChip is recognized as a standard disk. No external software is required. The DiskOnChip can be used as the only disk in the system, allowing the system to boot from it. In addition, the DiskOnChip can also work with other hard disks or floppies as the boot device or as a secondary disk.

Figure 3 shows the DiskOnChip memory location in relation to the PC memory map.

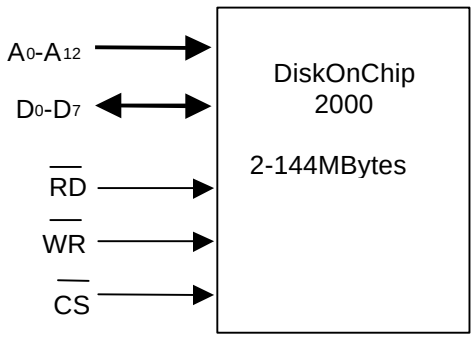


Figure 2 - DiskOnChip Interface Diagram

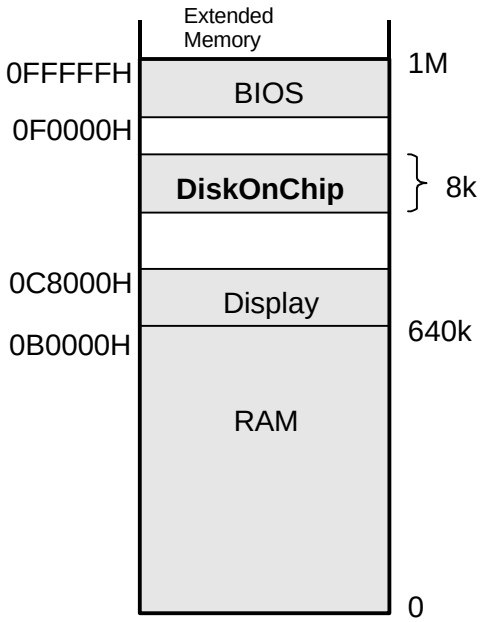


Figure 3 - PC memory

The DiskOnChip should be mapped to an 8 KB window in the BIOS expansion range, which is usually located between 0C0000H to 0EFFFFH. Note that the 32 KB of BIOS expansion range from 0C0000H to 0C8000H is typically reserved for the video BIOS expansion ROM.

After reset, the BIOS first executes the POST (Power on Self-Test). Then the BIOS searches for all expansion ROM devices. When the DiskOnChip is found, the BIOS executes the initialization code ("IPL") located on the DiskOnChip. This code loads the TrueFFS driver into system memory, installs the DiskOnChip as a disk in the system, and then returns control back to the BIOS code. The operating system subsequently attempts to identify the disks that are available and the DiskOnChip software (i.e. the TrueFFS) responds by emulating a hard disk.

From this point on the DiskOnChip appears as a standard disk drive, i.e. it is assigned a drive letter and it can be used by any software application. No BIOS set-up modifications or autoexec.bat/config.sys modifications are required.

The flash memory within the DiskOnChip is accessed by the TrueFFS software through an 8KB window in the PC's upper memory area. TrueFFS handles the paging of this window in the flash array, as well as providing Flash Disk emulation, which includes flash table management, wear leveling and background space reclamation on unused flash blocks. The same window will be used with larger capacities in future versions of the DiskOnChip . No redesign of the socket or larger memory resources will be required.

The DiskOnChip in combination with TrueFFS provides a true sector-based disk emulation that is compatible with conventional hard disks. Furthermore, it provides a bootable drive that is compatible with standard disk utilities.

2.2 System Requirements

In a PC compatible platform there are only two requirements for the socket into which the DiskOnChip is inserted:

1. The socket should be mapped into the BIOS expansion space of the PC (typically 0C8000H to 0E0000H).
2. The socket memory window must occupy at least an 8KB-memory space.

The first requirement allows the BIOS to load the DiskOnChip firmware into the memory automatically during the boot process. The second requirement defines the minimum window size required by the DiskOnChip. The window size can be larger than 8KB, due to the internal anti-alias algorithm of the DiskOnChip.

2.3 TrueFFS

TrueFFS is M-Systems' FTL Flash File System management technology that allows flash components to fully emulate a hard disk. This capability simplifies the usage of flash, as no special or complicated algorithms are needed to work with it - just read and write to it like any other disk drive.

Working under the native O/S and file system, TrueFFS acts as a block device driver. In this way it maintains full compatibility with other disks in the system and all disk utilities and drivers that accompany them. Because TrueFFS is loaded at start-up, it allows the flash disk to function as the boot device. No other system disk is required.

TrueFFS enables the Flash Disk to nearly match the maximum read/write speed boundaries inherent to the flash components. Background erasure of used flash blocks eliminates the long delays commonly encountered when writing to flash, and read speeds are exceptionally fast due to direct access to the flash. Additionally, there are no access, seek or spin-up delays that are typically encountered when using mechanical disk drives.

2.4 Boot

As described above, the DiskOnChip is recognized by the system like any hard disk. The DiskOnChip can be used as the only disk in the system, in which case it will be accessed as drive C:. The DiskOnChip can work together with or without a floppy drive, or with another hard disk(s). When working with another hard disk, the DiskOnChip can be configured as the last drive (default). In this case the hard disk will be C: and the DiskOnChip will be D:. It can also be configured as the first drive. In this case the hard disk will be D: and the DiskOnChip will be C:.

The DiskOnChip can be used as the boot device when configured as drive C:. In this configuration, the user is required to format the DiskOnChip as a bootable device, i.e. copy the OS files onto the disk. When running DOS, this can be done by using the SYS command.

2.5 Integrating the DiskOnChip 2000

The DiskOnChip is plug-and-play, making it very easy to integrate:

1. Plug the DiskOnChip into the DIP socket
2. Power up the system
3. If the DiskOnChip is to be the bootable drive, format it as a bootable disk and re-boot the system
4. Begin accessing the DiskOnChip as a regular drive, and copy your application files onto the DiskOnChip.

The DiskOnChip is shipped formatted and pre-programmed to be plug-and-play. In addition, M-Systems provides utilities for testing and formatting the DiskOnChip . Utilities are also provided for updating the DiskOnChip's internal firmware (TrueFFS) in the system. This eliminates the need to remove the DiskOnChip from the socket.

3. Operation in non-PC Architectures

In addition to DOS or Windows, the DiskOnChip can work with any other CPU or operating system. The following sections describe how to use the DiskOnChip in these environments.

3.1 Operation with Standard Operating Systems

The DiskOnChip is compatible with many operating systems in addition to DOS, including QNX, pSOS+, VxWorks, WindowsCE, Windows NT4.0, Windows NT5.0 and many others.

In order to use the DiskOnChip with these operating systems, the appropriate DiskOnChip device driver should be installed. The DiskOnChip device driver can be part of the operating system or it can be supplied by M-Systems.

When operating under one of these Operating Systems in a PC compatible host system (i.e. x86 CPU and standard BIOS), the boot process is performed in the same way as described above for DOS.

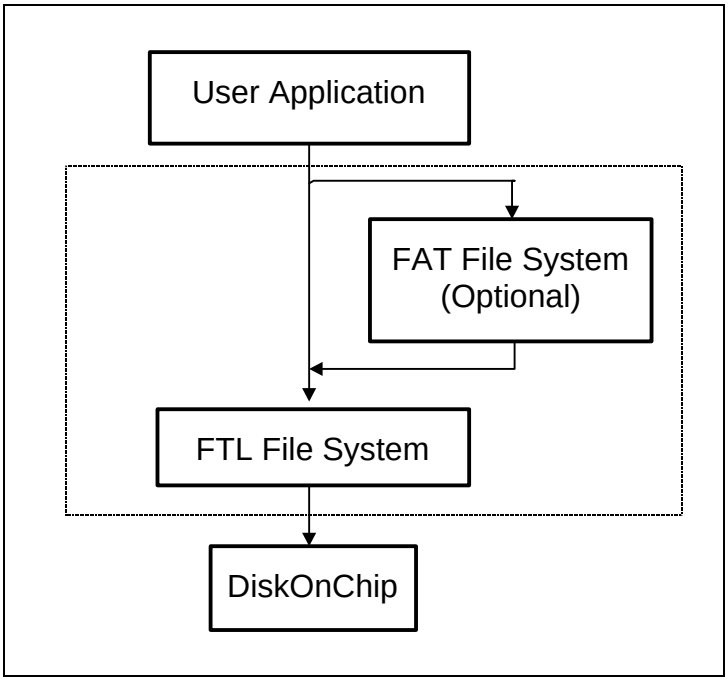


Figure 4 - Software Block Diagram

4. Theory of Operation

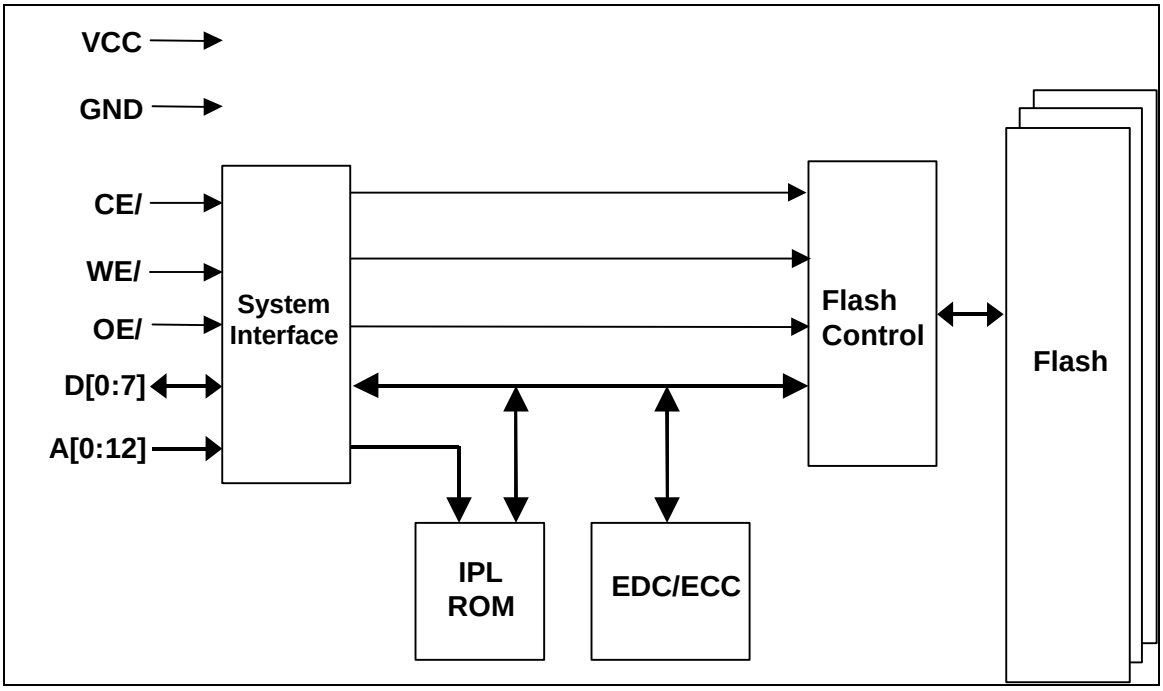


Figure 5 - DiskOnChip Block Diagram

The DiskOnChip is integrated into the system as a standard EEPROM. The system interface is controlled by the host bus signals (read, write, chip select, addresses and data), which generate the appropriate control signals to the internal blocks.

The internal ROM provides the IPL (Initial Program Loader) code that loads the TrueFFS software from the Flash into the PC memory during boot (after power up or reset). This code is necessary since NAND-type flash devices have no linear address space and therefore cannot be used as boot code storage devices.

The Flash Control block interfaces with the NAND flash devices. The Flash block is comprised of the Flash devices, which are mounted inside the DiskOnChip. There may be more than one Flash device. The software automatically detects the number of Flash devices and their capacity by reading their ID code and calculates the total formatted capacity of the DiskOnChip.

5. Disk Capacities

Each DiskOnChip device is fully tested and formatted when shipped from M-Systems. The exact capacity of each model is detailed in the following table:

Model	Formatted Capacity (bytes)	Sectors	Formatted Capacity under DOS 6.22 (bytes)	Sectors under DOS 6.22
MD2200-D02MB	1,998,848	3904	1,986,560	3880
MD2200-D04MB	4,038,656	7888	4,022,272	7856
MD2200-D08MB	8,151,040	15920	8,128,512	15876
MD2200-D12MB	12,263,424	23952	12,228,608	23884
MD2200-D16MB				
MD2200-D32MB				
MD2200-D48MB				
MD2200-D24MB	24,592,384	48032	24,516,608	47884
MD2201-D72MB	73,891,840	144,320	73,789,440	144,120
MD2201-D80MB				
MD2201-D112MB	114,999,296	224,608	114,786,304	224,192
MD2201-D144MB	147,881,984	288,832	147,578,880	288,240

6. Electrical Specifications

6.1 Absolute Maximum Ratings

Parameter	Symbol	3.3V Model Rating[1]	5V Model Rating[1]	Units	Notes
DC supply voltage	V_{CCS}	-0.5 to 4.6	-0.3 to 6.0	V	
Input pin voltage[2]	V_{IN}	-0.5 to V_{CC} + 0.3	-0.3 to V_{CC} + 0.3	V	
Input pin current	I_{IN}	Not Specified	-10 to 10	mA	25°C
Storage temperature	T_{STG}	-45 to 85	-45 to 85	°C	

Notes:

1 Permanent device damage may occur if Absolute Maximum Ratings are exceeded. Exposure to Absolute Maximum Rating conditions for extended periods may affect device reliability.

2 The voltage on any pin may undershoot to -2.0V or overshoot to Vcc+2.0V for periods <20ns.

6.2 Capacitance

Symbol	Parameter	Conditions	3.3V Model Rating[1]	5V Model Rating[1]	Unit
$C_{I/O}$	Input/Output Capacitance	MD2200, V_{IN} =0V	12	15	pF
		MD2201, V_{IN} =0V	36	45	pF

Note: Capacitance is not 100% tested.

6.3 Operating Temperature Ranges

Commercial operating temperature.....0°C to +70°C

Enhanced operating temperature .....-25°C to +75°C

Extended operating temperature .....-40°C to +85°C

6.4 Humidity

10% - 90% relative, non-condensing.

6.5 EDC/ECC

Enhanced Reed-Solomon ECC:

- Corrects up to two 10-bit symbols, including two random bit errors.
- Corrects single bursts up to 11 bits.
- Detects single bursts up to 31 bits and double bursts up to 11 bits.
- Detects up to 4 random bit errors.

6.6 DC Electrical Characteristics Over Operating Range

6.6.1 Vcc = 5V Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CCS}	System Supply Voltage		4.5	5.0	5.5	V
V_{IH}	High Level Input Voltage		2.0			V
V_{IL}	Low Level Input Voltage				0.8	V
V_{OH}	High Level Output Voltage	I_{OH} = -16 mA	2.4			V
V_{OL}	Low Level Output Voltage	I_{OL} = 16 mA			0.4	V
I_{IL}	Input Leakage Current	MD-2200			±1	µA
		MD-2201			±3	µA
I_{OZ}	Output Leakage Current	MD-2200			±10	µA
		MD-2201			±30	µA
I_{VCC}	Supply Current	Cycle Time = 200 ns, Outputs open		40	50	mA
I_{STDBY}	Standby Current	MD-2200		60	400	µA
		MD-2201		240	1200	µA

6.6.2 Vcc = 3.3V Characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{CCS}	System Supply Voltage		3.0	3.3	3.6	V
V_{IH}	High Level Input Voltage		2.7			V
V_{IL}	Low Level Input Voltage				0.6	V
V_{HYS}	Input Voltage Hysteresis		1.1		1.5	V
V_{OH}	High Level Output Voltage	I_{OH} = -18 mA	2.4		1.5	V
		I_{OH} = 0 mA	Vcc-0.1		1.5	V
V_{OL}	Low Level Output Voltage	I_{OL} = 18 mA			0.4	V
		I_{OL} = 0 mA			0.1	V
I_{IL}	Input Leakage Current	MD-2200			±10	µA
		MD-2201			±30	µA
I_{OZ}	Output Leakage Current	MD-2200			±10	µA
		MD-2201			±30	µA
I_{VCC}	Supply Current	Cycle Time = 150 ns, Outputs open		30	50	mA
I_{STDBY}	Standby Current	MD-2200		40	500	µA
		MD-2201		120	1500	µA

6.7 AC Operating Conditions

Timing specifications are based on the following conditions:

Parameter	3.3V Model	5V Model
Supply Voltage	V_{CC} = 3.3V ± 0.3V	V_{CC} = 5V ± 0.5V
Input Pulse Levels	0.2V to 2.9V	0.4V to 2.6V
Input Rise and Fall Times	1 ns	5 ns
Input and Output Timing Levels	1.5V	0.8V and 2.0V
Output Load	100 pF	50 pF

6.8 Timing Specifications

6.8.1 Read Cycle Timing

Symbol	Description	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Notes
		3.3V		5V		
$T_{SU}(A)$	Address to OE# ↓ setup	0		10		
$T_{HO}(A)$	OE#↓ to Address hold	35		56		
$T_{SU}(CE0)$	CE# ↓ to OE# ↓ setup	0		0		1
$T_{HO}(CE0)$	OE#↑ to CE#=0 hold	0		0		2
$T_{HO}(CE1)$	OE# or WE#↑ to CE#=1 hold	8		42		
$T_{SU}(CE1)$	CE#↑ to WE# or OE#↓ setup time	8		42		
T_{REC}	OE#↑ to start of next cycle	20		59		
T_{ACC}	Read access time		100		130	
$T_{EN}(D)$	OE#↓ to D active delay	15	75	7	91	
$T_{DIS}(D)$	OE#↑ to D Hi-Z delay		13		44	

Notes:

1 CE# may be asserted any time before or after OE# is asserted. If CE# is asserted after OE#, all timing relative to OE# asserted will be referenced instead to the time of CE# asserted.

2 CE# may be negated any time before or after OE# is negated. If CE# is negated before OE#, all timing relative to OE# negated will be referenced instead to the time of CE# negated.

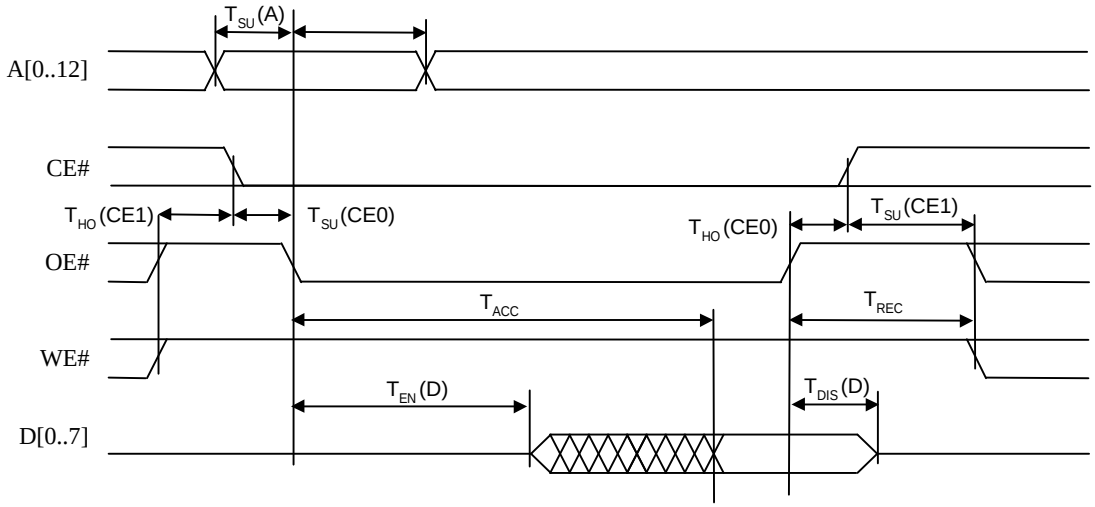


Figure 6 – Read Cycle

6.8.2 Write Cycle Timing

Symbol	Description	Min (ns)	Max (ns)	Min (ns)	Max (ns)	Notes
		3.3V		5V		
t_{SU}(A)	Address to WE#↓ setup time	0		10		
t_{HO}(A)	WE#↓ to Address hold time	35		56		
t_{W}(WE)	WE# asserted width	62		98		
t_{SU}(CE0)	CE#↓ to WE#↓ setup time	0		0		1
t_{HO}(CE0)	WE#↑ to CE#=0 hold time	0		0		2
t_{HO}(CE1)	OE# or WE#↑ to CE#=1 hold time	8		42		
t_{SU}(CE1)	CE#↑ to WE# or OE#↓ setup time	8		42		
t_{REC}	WE#↑ to start of next cycle	22		59		
t_{SU}(D)	D to WE#↑ setup time	50		48		
t_{HO}(D)	WE#↑ to D hold time	0		40		

Notes:

1 CE# may be asserted any time before or after WE# is asserted. If CE# is asserted after WE#, all timing relative to WE# asserted will be referenced instead to the time of CE# asserted.

2 CE# may be negated any time before or after WE# is negated. If CE# is negated before WE#, all timing relative to WE# negated will be referenced instead to the time of CE# negated.

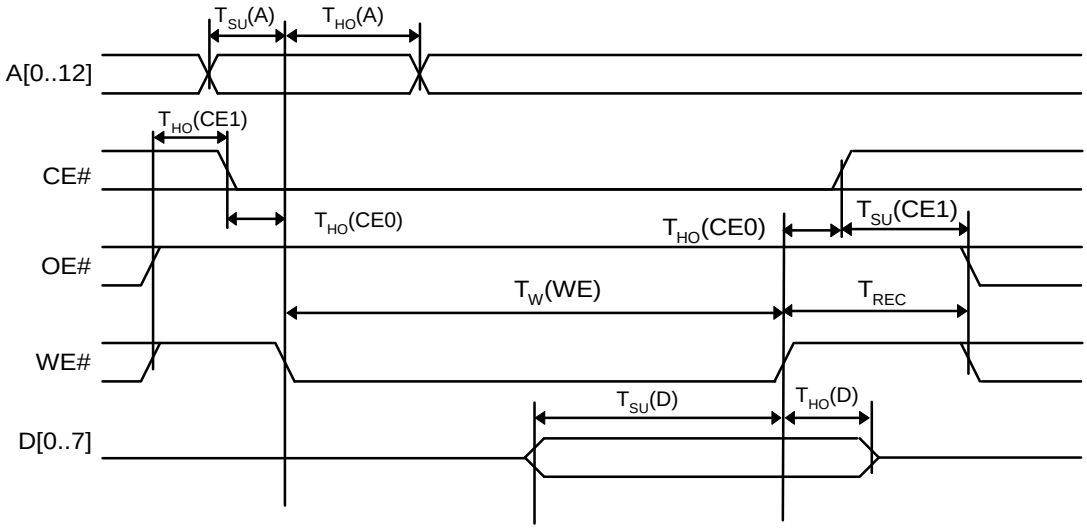


Figure 7 - Write Cycle

7. Mechanical Dimensions

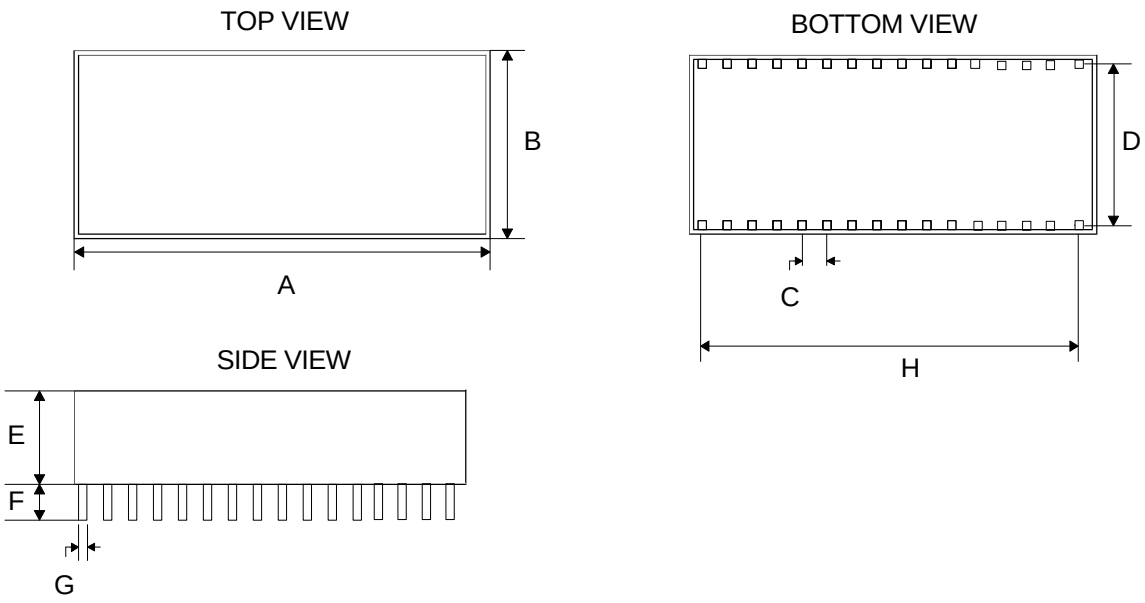


Figure 8 - MD2200 Mechanical Dimensions

	MD-2200-Dxx (Low)		MD-2201-Dxx (High)	
	Millimeters	**Inches**	**Millimeters**	**Inches**
A[1]	41.65 ± 0.10	1.64 ± 0.004	41.8 + 0.20	1.650 + 0.008
B	17.90 ± 0.10	0.704 ± 0.004	18.1+ 0.20	0.713 + 0.008
C	2.54	0.100	2.54	0.100
D	15.24	0.600	15.24	0.600
E	5.50 ± 0.20	0.216 ± 0.008	11.8 ± 0.20	0.464 ± 0.008
F	3.70 ± 0.20	0.145 ± 0.008	3.7 ± 0.10	0.153 ± 0.004
G	0.46 + 0.04	0.018 + 0.001	0.51 - 0.01	0.020-0.0004
H	38.10 ± 0.10	1.5 ± 0.004	38.10 ± 0.10	1.5 ± 0.004

Notes:

1 In future models of the DiskOnChip A=43.7± 0.10, in order to accommodate new and improved Flash technologies M-Systems recommends to use this as a reference when designing new socket for the DiskOnChip

8. Shocks and Vibrations

P/N : MD2200 – DXX, MD2201-DXX (including extended temperature models)

Test Item	Test Condition	Referred Standard
Vibration	100~2000Hz, 15 G* peak, 3 cycles per axis (1hr), 3 axes	STD-202F Method 204D
Mechanical Shock	Half sine shock 50G, 11 msec, +/-3 shock per axis, 3 axes	STD-202F Method 213B

9. Ordering Information

MD-YYYY-DCC-V-T-PX

Y:	Package Dimensions	2200	Low
		2201	High
CC:	Capacity (MB)	02, 04, 08, 12, 16, 24, 32, 48, 72, 80, 112, 144	
V:	Supply Voltage		Blank 5V
		V3	3.3V
T:	Temperature Range	Blank	Commercial 0°C to +70°C
		N	Enhanced -25°C to +75°C
		X	Extended -40°C to +85°C
PX:	Packaging	PB	Bulk
		PI	Individual (incl. manual and utilities diskette)

10. Additional Information

Document / Tool	Description
DiskOnChip 2000 Data Sheet	DiskOnChip 2000 Data Sheet
DiskOnChip 2000 Utilities	DiskOnChip 2000 Utilities User Manual
DiskOnChip 2000 Quick Installation Guide	DiskOnChip 2000 Quick Installation Guide (provided with the individual package)
AP-DOC-010	Application note - Designing with DiskOnChip 2000
AP-DOC-017	Application note - Using the DiskOnChip 2000 with Win CE
AP-DOC-016	Application note - Using the DiskOnChip 2000 with QNX
AP-DOC-019	Application note - Using the DiskOnChip 2000 with Win 95
AP-DOC-011	Application note - Write protecting the DiskOnChip 2000
DiskOnChip2000-EVB	DiskOnChip Evaluation Board
DiskOnChip2000-PIK	DiskOnChip Programmer and Integrator's Kit
DiskOnChip2000-GANG	8 Socket Gang Programmer

M-Systems' DiskOnChip® 2000 Evaluation Board

DiskOnChip ®2000-EVB

- **ISA card for DiskOnChip 2000 evaluation**
- **Convenient tool for programming the DiskOnChip2000**
- **Enables duplication of DiskOnChip 2000**
- **Shortens integration time**
- **Enables booting from DiskOnChip 2000**
- **Broad O/S support[1]: DOS, Windows, Windows 95, Windows CE, Windows NT**
- **Additional O/S support[1]: pSOS+, QNX, VxWorks and others**
- **Configurable memory window addresses (C8000, D0000, D8000)**
- **Uses 8 KB memory window**

Overview

DiskOnChip 2000 EVB - an ISA card with a socket for DiskOnChip 2000 is a useful tool for designers who need to evaluate, program and test the DiskOnChip 2000, even before the target platform is available.

DiskOnChip2000 overview

M-Systems' *DiskOnChip2000* is a new generation of high performance single-chip Flash Disk. The DiskOnChip MD2000 provides a Flash Disk in a standard 32-pin DIP package.

This unique data storage solution offers a better, faster, and more cost-effective Flash Disk for applications with limited space and modest disk capacity requirements.

DiskOnChip 2000 applications

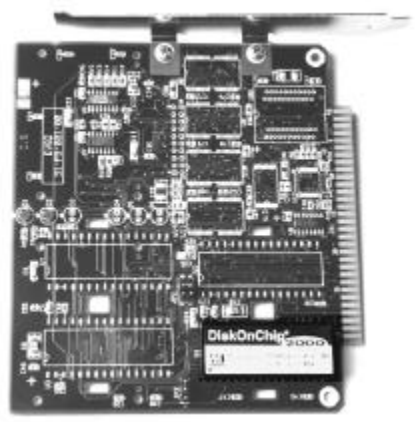
The DiskOnChip2000 has become the standard Flash Disk module for Embedded Single Board Computers. It is the optimal solution for mother boards used to control a variety of applications: Set-top boxes, Diskless Network Computers, Internet appliances, DVD and CD-Video players and recorders. It can also provide a hard disk O/S back-up for Network Servers.

DiskOnChip2000-EVB

The DiskOnChip 2000 Evaluation Board was developed in M-Systems to provide our customers with a simple, straightforward tool to evaluate, test and program the DiskOnChip 2000 even before the target platform is actually available, since it provides DiskOnChip functionality using standard ISA bus interface.

The DiskOnChip 2000 requires a memory window of only 8KB. It can be plugged into the customers' target board or the EVB , and with a minimal cost can be upgraded to a higher capacity device.

When connecting the DiskOnChip 2000 EVB to a PC, the system works automatically, without need for any software integration. The DiskOnChip is automatically configured as a system disk drive with a full hard disk functionality.



The DiskOnChip 2000 and the EVB enable the system to boot from the DiskOnChip 2000 when the DiskOnChip functions as the primary system disk. The DiskOnChip EVB can also be used as a convenient tool to duplicate, update and test the DiskOnChip 2000, using simple utilities provided with the EVB.

M-Systems
Flash Disk Pioneers

1 Please contact M-Systems for availability

91-SR-003-02-8L 7/97

USA - M-Systems Inc., Phone: 510-413-5950, Fax: 510-413-5980, email: info@ccm.msyscal.com
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266, email: info@msys.nl
Israel - M-Systems Ltd, Phone: 972-3-647-7776, Fax: 972-3-647-6668, email: info@m-sys.com
http://www.m-sys.com.

Series 2000

User Manual

DiskOnChip®2000 Stand Alone GANG Programmer

Nov-97

91-SR-004-03-7L Rev. 2.1

M-Systems
Flash Disk Pioneers

1. Overview

1.1 GANG Programmer Overview

DiskOnChip GANG is a standalone gang programmer which can duplicate DiskOnChip 2000 devices very quickly. It is intended for mass production quantities of DiskOnChip-based products.

DiskOnChip 2000-GANG includes 9 ZIF sockets for DiskOnChip devices: one for the "source" DiskOnChip device, and 8 for the "target" DiskOnChip devices. The DiskOnChip GANG can duplicate up to 8 DiskOnChip devices simultaneously, in high speed.

The "source" DiskOnChip is copied into an internal "image file", which then can be used as a source for duplication without the need for the "source" DiskOnChip device to be in the source socket. The duplication can be repeated as many times as required.

The Stand alone DiskOnChip GANG is shipped as a ready to work system: plug in the power and start working.

1.2 DiskOnChip2000 Overview

M-Systems' *DiskOnChip2000* is a new generation of high performance single-chip Flash Disk. The DiskOnChip MD2000 provides a Flash Disk in a standard 32-pin DIP package.

This unique data storage solution offers a better, faster, and more cost-effective Flash Disk for applications with limited space and modest disk capacity requirements.

The DiskOnChip2000 has become the standard Flash Disk module for Embedded Single Board Computers. It is the optimal solution for mother boards used to control a variety of applications: Set-top boxes, Diskless Network Computers, Internet appliances, DVD and CD-Video players and recorders. It can also provide a hard disk O/S back-up for Network Servers.

1.3 Package Contents

The GANG programmer is shipped with the following items. In case something is missing or damaged, please contact your M-Systems representative.

1. GANG programmer
2. Power cord
3. Firmware diskette
4. Bootable diskette

2. Quick Installation Guide

The gang is very easy to install:

- Select the power supply voltage according to required voltage (110V or 220V). PLEASE MAKE SURE THAT THE VOLTAGE IS SELECTED CORRECTLY!
- Connect Power Cord to power outlet.

The GANG is ready to work.

3. GANG – Description

The gang programmer, as displayed in figure 1, is composed of one source socket (labeled as "source", at the upper left corner, and 8 target sockets (labeled as 1 to 8), on the right hand side of the source socket. Each socket has an indication led. The led can be inactive (nothing to signal), blinking (active now), red (last operation failed) or green (last operation OK).

An LCD display resides below target sockets 6-8 and is used to display status messages and error codes. 7 push buttons are used to activate the gang programmer. Two are used during standard operation (S1 is labeled "Read" and S7 is labeled "Duplicate"),the other are used for technical support or reserved for future use.

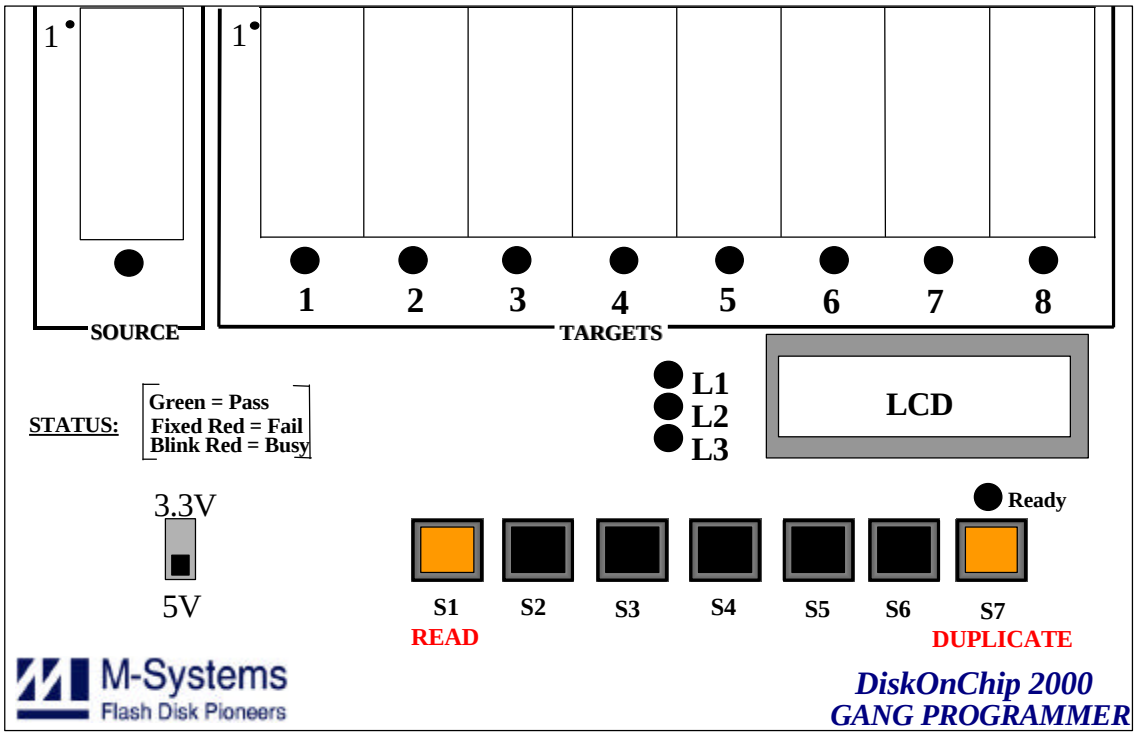


Figure 1 - GANG Programmer Layout

3.1 Push Buttons Description

The gang has 7 push buttons, numbered from S1 to S7. The following guide lines define the buttons behavior:

- The buttons are sampled by the software only when the GANG is not busy with previous command.
- If a button is pressed during the execution of a command, nothing will happened.

The following table defines the buttons behavior:

Table 1 - GANG buttons description

S1	READ. Press this button to copy the source DiskOnChip placed in the MASTER socket into an IMAGE file on the internal hard disk.
S2	Pressing S2 will cause the software version to be displayed on the LCD display (for example: “Gang ver 1.00”).
S3	Pressing S3 will start a verification process between the image file on the disk and all DiskOnChip devices plugged in the Target sockets.
S4	Pressing this button will show the image file description on the LCD display. If there is no image file available, a “Disk Image file not exists” message will appear on the LCD display. If the image file exists, the LCD will display “Disk Image File xxxMB”, in which the xxx represents the size of the image file.
S5	Reserved for future use.
S6	Reserved for future use.
S7	DUPLICATE – press this button to copy the image file from the hard disk to all the DiskOnChip devices plugged in the Target sockets.

3.2 Rules Of Operation

The following rules will guarantee a faultless operation of the GANG programmer product:

1. Place the GANG programmer in a stable location, not in direct sunlight.
2. Make sure that no DiskOnChip devices are placed in the Gang’s sockets when power is turned ON or OFF.
3. Don’t touch the DiskOnChip devices while the Ready LED is not green (indicating the GANG is busy).
4. Visit M-Systems’ web site once in a while – and look for new features or upgrades.

4. Duplicating DiskOnChip 2000 Devices

4.1 Required Stages

The DiskOnChip GANG is a tool used mainly to duplicate DiskOnChip 2000 devices. It takes one pre-prepared DiskOnChip 2000 device ("Master Device") and copies its image ("Source Image") to the internal hard disk. Then, it duplicates this "Source Image" into maximum of 8 target DiskOnChip 2000 devices, resulting in 8 DiskOnChip 2000 devices identical to the source DiskOnChip 2000 device .

Here is a summary of the required stages:

1. The customer receives a new DiskOnChip 2000 from M-Systems. The DiskOnChip is a formatted, empty disk.
2. The DiskOnChip 2000 device is placed in a target platform, and all application files are copied into the device. Usually it will be made as a bootable disk (remember to re-boot after the SYS or the FORMAT /S commands).
3. The DiskOnChip is tested in the target platform to make sure the integration into the platform is working well.
4. In the mass production stage, the GANG programmer enables the duplication of many units in a short time. In order to program many similar systems, all that is needed is one source DiskOnChip device ("Master Device") containing the data which all the other systems require. Instead of repeating the copying operation as many times as the number of systems, the GANG programmer duplicates 8 target DiskOnChip devices simultaneously, in a very short time.
5. Plug each of the target DiskOnChip devices into a target system and power up. The operation of each of those systems should be the same as the source system.
6. For control purposes, DiskOnChip devices can be verified, by placing them in any target socket, and pressing button S3.

4.2 Step by Step

Let's go step by step through DiskOnChip 2000 duplicating stages:

1. Make sure power is OFF, and no DiskOnChip devices are in the Gang's sockets.
2. Power up the gang programmer. You should get the following message on the LCD:

DiskOnChip 2000
GANG Ready

   The GANG is now ready to accept commands via the press buttons.

3. Take the pre-prepared source DiskOnChip 20000 device and place it in the "source" socket, on the top left hand side of the GANG. Verify the polarity is correct.
4. Press the "Read" button. The LED under the source socket should be blinking red, which means that the gang is busy. During this activity, the following message should appear on the LCD display:

Read from master
W A I T

5. When the "Read From Source" activity is successfully finished, the socket's LED should turn green, means the socket operation was successful, and the READY led should also turn green. The following message should appear on the LCD display:

Read from master
Size:xxxMB OK

   The capacity of the DiskOnChip device (in Mbytes) will appear on the LCD display .

   If the operation failed, the LED will turn red, and the LCD will display an error message, as described in chapter

6. The "source" DiskOnChip device can be taken out of the source socket and kept for future use (usually for verification). Note that the image of the source DiskOnChip is kept on the hard disk of the gang programmer until a new source DiskOnChip is read.
7. Up to 8 target DiskOnChip devices, same capacity as the source, should be placed in the target sockets. Please verify that the polarity is correct.
8. Press the "Duplicate" button. The GANG will start copying the "Source Image" into each DiskOnChip device. During operation, the LED of each socket will indicate the status by blinking red- in work, green- finished successfully, red- operation failed.

   The LCD will display the following message:

Copy to targets
W A I T

9. When the duplicate operation is finished, the ready LED will turn green, each socket will display it's own status (green for OK, blinking red for failed), and the following message will appear on the LCD:

1 2 3 4 5 6 7 8
X X X X X X X X

The numbers 1 to 8 represent the relevant target socket. The “x’ under each socket represent the return code (0: OK, other: error code). The codes are described in chapter

10. All the DiskOnChip devices that were successfully duplicated (LED is green) have now the exact same disk contents as the source DiskOnChip 20000 device.
11. All the DiskOnChip devices that had a red LED signal should be checked. Please refer to chapter 6.

5. Maintenance

5.1 Internal maintenance

The GANG is testing the internal disk integrity on every boot, to make sure the files are not damaged.

Software Upgrade

From time to time M-Systems releases new software versions. In that case you will get a floppy diskette which contains the software needed to update your GANG programmer. Please perform the following steps to update your software:

1. Power off the gang programmer.
2. Insert the upgrade diskette into the floppy drive.
3. Power on the GANG programmer. The gang will boot from the floppy and start to update itself automatically. The following message will be seen on the LCD display:

GANG Update VX.XX Starts

4. When the updating process is finished, the following message is displayed:

GANG Update VX.XX Finished

5. Power the GANG OFF.
6. Take the floppy diskette out of the floppy drive and keep it in safe place.
7. The gang is ready to work.

6. Error Codes

Several error codes can appear on the LCD display. The codes are described in this chapter, according to the operation that was performed when the message appeared.

6.1 Source DiskOnChip device errors

The following error messages can appear while executing the “Read from Master” operation, because of a fault device.

Error	Description
“DiskOnChip Not found in Socket”	No DiskOnChip device is available in the socket, the DiskOnChip device is damaged, or inserted wrongly in the socket.
“DiskOnChip error in Master socket”	The DiskOnChip device placed in the Master socket was recognized, but is a faulty device.
“General failure”	General system failure occurred.

6.2 “Read Source DiskOnChip” errors

The following error messages can appear while executing the “Read from Master” operation, because of bad format of the DiskOnChip device.

Error	Description
“DiskOnChip has bad Format”	The file format of the source DiskOnChip device is bad.
“Error writing DiskOnChip image”	The system failed writing the DiskOnChip image onto the hard disk.

6.3 Errors generated during “Copy to target DiskOnChip devices” or Verify

The following error messages can appear while executing the “Copy to targets” operation. Each error is displayed on the LCD display as an error code.

Code	Description
0	OK
1	General failure
2	No DiskOnChip in socket or illegal insertion.
3	Bad DiskOnChip device.
4	Capacity of target DiskOnChip doesn’t mach the source.
5	Error reading image file from hard disk.
6	DiskOnChip format failed, or Verify failed.

7. Troubleshooting

In case of a problem, please try to solve it according to the following guide lines. In case the problem is not solved, please contact the nearest M-Systems' representative for technical support.

Problem :	I pressed the power button, but nothing happened.
Solution:	Make sure the power cord is connected to the outlet, and the power supply switch select the same voltage as the outlet.
Problem :	The gang boots up, but no message is seen on the LCD display, and the buttons do not respond.
Solution:	Make sure no floppy diskette was left in the floppy drive.
Problem :	The duplication operation started, and all the devices were reported as failed, although I know they are good devices.
Solution:	Make sure that the voltage selected by the voltage select switch (5V/3.3V) is the same as required according to your DiskOnChip devices.
Problem :	One target DiskOnChip device failed. All others passed.
Solution:	First, try the last operation again. If it fails again, look at the DiskOnChip device leads. Are they clean? Was any pin broken? Try another device in the same socket, and try this device in another socket.
Problem :	One target DiskOnChip device failed. All other passed. I put it in a different target socket and it kept failing.
Solution:	If other DiskOnChip devices pass in the same activity, you might have a faulty device. Contact the nearest dealer.

8. Q&A

Following is a list of questions that M-Systems' technical support team is often asked. We recommend to read this chapter before contacting technical support.

Q:	Should I leave the "Master DiskOnChip" in the Master Socket after it was successfully read to image file ?
A:	You don't have to. After the file was read, it is kept on the internal hard disk, until a new source is being copied. However, it is recommended to take out the master DiskOnChip and keep it in safe place for backup.
Q:	I accidentally pushed the READ button, while no DiskOnChip was in the Master socket. Was the image file that was in the hard disk destroyed ? (the one that was copied last)
A:	No. The image file will be erased/replaced only when the READ button is pressed and there is a DiskOnChip in the master socket.
Q:	Will the GANG support future DiskOnChip 2000 products ?
A:	Yes. M-Systems will release new firmware version to the GANG any time new members of the DiskOnChip 2000 family products will be released.
Q:	Must I put 8 DiskOnChip devices in the Target sockets every time I duplicate ?
A:	No. You can duplicate up to 8 devices at a time. The software will automatically detect the sockets in which a DiskOnChip device was placed. All empty sockets will be ignored.
Q:	I accidentally programmed twice the same 8 DiskOnChip devices. Was any damage done?
A:	No. You can re-program the same devices as many times as you want. No damage can be done.

USA - M-Systems Inc, Phone: 510-413-5950, Fax: 510-413-5980, email: info@m-sys.com
Taiwan - M-Systems Asia, Phone: 886-3-5723438, Fax: 886-3-5723438
Europe - M-Systems BV, Phone: 31-20-69-69-586, Fax: 31-20-69-61-266
Israel - M-Systems LTD, Phone: 972-3-647-7776, Fax: 972-3-647-6668
http://www.m-sys.com

M-Systems
Flash Disk Pioneers

Series 2000

Questions and Answers about DiskOnChip2000

1. I am designing my next board right now, and am considering the DiskOnChip2000 as the storage media. What should I do?

   M-Systems has a full set of application notes to describe the necessary details. The main requirement is to add a 32 pin DIP socket to your board, connect it as a standard EEPROM, and map it into an 8KByte window in the Expansion BIOS area of the PC. The DiskOnChip is a self-contained, bootable disk drive, and it doesn't require any software integration. Please refer to AP-DOC-010 "Designing with the DiskOnChip2000" for further information

2. My systems require flexible storage solutions; sometimes I need a hard disk and sometimes a Flash Disk. Can DiskOnChip work in this type of environment?

   For applications requiring disk storage flexibility, DiskOnChip is a perfect solution. For flash disk-only applications, DiskOnChip offers a wide range of capacities to tailor the solution to the lowest price point. If no flash disk is needed, the overhead cost is about 5 cents for the (unused) DiskOnChip socket. For mixed solutions, DiskOnChip can work in conjuction with a hard disk and can be configured as the boot disk or next available drive.

3. What are the advantages of DiskOnChip2000 vs. a Resident Flash Array?

   A Resident Flash Array (RFA) is comprised of flash components soldered to a main or daughter board and integrated with interface logic and a flash file driver such as TrueFFS, to make them work as a flash disk. RFAs may appear to offer slightly lower component cost but they suffer from many restrictions compared to DiskOnChip2000:

   - Board space usage - up to six time reduction in space with DiskOnChip
   - Flexibility - easy to plug-in different capacities
   - Migrate to new technologies - no board redesign required or software updates
   - Ease of integration
   - Can be pre-programmed - lowers manufacturing cost

4. What modifications will I have to make to the AUTOEXEC.BAT and the CONFIG.SYS files in order to use the DiskOnChip2000?

   None.

5. Will I need to place any special files on the DiskOnChip2000 in order for it to work?

   No. The only files required are the customer's application files. The DiskOnChip driver is kept on the flash media, in a safe place, protected from and invisible to the user.

6. How will I upgrade the DiskOnChip firmware when M-Systems releases new software versions? Will I have to open my products and remove the DiskOnChip2000 from its socket?

   The DiskOnChip does not need to be removed from the socket. Running a simple utility program on the target system can perform the upgrade quickly and easily.

7. Is the DiskOnChip2000 Plug & Play?

   Yes, DiskOnChip is self-contained and needs no external drivers or jumper settings. Just plug the DiskOnChip2000 into the socket and power up the system.

8. Is the DiskOnChip2000 suitable for use in ultra-small portable products, such as Personal Digital Assistants running Microsoft CE?

   DiskOnChip is fully compatible with Windows CE. M-Systems has developed TrueFFS drivers for Windows CE although they have not yet been made available for the DiskOnChip. Because DiskOnChip is an ultra small, high capacity drive, it is ideal for storing large databases or applications, which will make it an attractive solution to many OEM's and vertical integrators of such PDAs. Alternatively, for this handheld market, the Series2000 is also available from M-Systems in the thin Type I PC Card (PCMCIA) format, with capacities as high as128 MBytes. These cards include the necessary Windows CE drivers.

9. I have the DiskOnChip 1000 programmer. Can I still use it with the DiskOnChip2000?

   The external programmer is composed of an ISA card and a socket with a flat cable. Both the ISA card and the socket should be upgraded to support the DiskOnChip2000 PIK. M-Systems will replace your old 32 pin socket and ISA Card with a new one free of charge.

10. What are the main improvements of DiskOnChip2000 Series over the previous DiskOnChip 1000 Series?

    The key improvements with DiskOnChip2000 are:
    - The ability to store up to 6 times more data, eventually to become a 36x improvement!
    - More than 5 times improvement in write performance
    - Up to 50% lower in price

    These improvements will become even more significant later this year when the maximum capacity increases to 72MB.

The DiskOnChip is a paradigm shift for the design engineer, who can consider it as a true alternative to hard disk drives.

For DiskOnChip Series 1000 customers, the new Series 2000 can be a drop-in replacement, providing the socket has a write signal, when working in Flash Disk Mode. No changes need to be made in design or configuration. All M-Systems' products provide such an easy migration path. They will grow with the your application, providing forward as well as backward compatibility.

Here are the main differences between DiskOnChip Series 2000 and 1000:

- Increased capacities - 2, 4, 8 or 12MB vs. 1 or 2MB
- Lower cost
- Better performance - write speeds of 250KB/sec vs. 50KB/sec.
- Lower power consumption
- Reduced memory requirement - 8KB memory window vs. 32KB.

11. Is the DiskOnChip dependent on certain Flash technologies or Flash chip vendors?

No. DiskOnChip is implemented with M-Systems' industry standard technology - TrueFFS and FTL, which stands for "Flash Translation Layer," works with any flash technology. Therefore, it can be implemented with any of the common flash technologies available in the market - NOR (Intel, Sharp, AMD, or Fujitsu), NAND (Toshiba or Samsung), and others. M-Systems' future road map will include other additional flash technologies.

- M-Systems' designers can therefore take advantage of the best technologies available to give DiskOnChip the best performance, capacity and price.
- The DiskOnChip is self contained, i.e. once plugged in the socket, it works as a Flash Disk. This is regardless of the flash technology being used inside of the product. The result is more flexibility, broader choices, and easier integration.
- All of M-Systems' Flash data storage solutions guarantee forward and backward compatibility. The underlying flash technology is transparent to the designer and quick migration to versions utilizing new, leading edge components is possible.

12. Is NAND technology mature and is there only a single source for this type of Flash memory component?

- NAND technology is mature. The third generation of product (64Mbit) is now appearing, while current versions have been shipping in volume for several years.
- There is more than one source for NAND: Samsung, and Toshiba Corporation, . Toshiba was the original developer of Flash memory back in the mid-eighties and they invented the NAND flash technology as well. DiskOnChip2000 can use NAND flash from either of these sources!

13. How easily can I upgrade from Series 1000 to DiskOnChip2000?

    DiskOnChip2000 is a drop-in replacement for the DiskOnChip1000 when operating in Flash Disk Mode, providing the write signal is connected to the socket.

    The DiskOnChip2000 does not support BIOS replacement mode.

14. In earlier stages of using the DiskOnChip 1000, I had to use an external programmer in order to modify the firmware or to test the DiskOnChip. Do I have to use the programmer with the DiskOnChip2000 for this purpose?

    No. All the tasks done by the programmer can be done while the DiskOnChip is plugged into your target platform. However, the PIK (Programmer and Integrator Kit) can be a useful tool during the manufacturing stages.