

The tmcsim driver

1. Purpose and history
2. Installation
3. Features
4. Configuration via /proc/scsi/tmcsim/?
5. Configuration via boot/module params
6. Potential improvements
7. Bug reports, debugging and updates
8. Acknowledgements
9. Copyright

1. Purpose and history

The tmcsim driver supports PCI SCSI Host Adapters based on the AM53C974 chip. AM53C974 based SCSI adapters include:

Tekram DC390, DC390T

Dawicontrol 2974

QLogic Fast! PCI Basic

some on-board adapters

(This is most probably not a complete list)

It has originally written by C.L. Huang from the Tekram corp. to support the Tekram DC390(T) adapter. This is where the name comes from: tm = Tekram scsi = SCSI driver, m = AMD (?) as opposed to w for the DC390W/U/F (NCR53c8X5, X=2/7) driver. Yes, there was also a driver for the latter, tmcsiw, which supported DC390W/U/F adapters. It's not maintained any more, as the ncr53c8xx is perfectly supporting these adapters since some time.

The driver first appeared in April 1996, exclusively supported the DC390 and has been enhanced since then in various steps. In May 1998 support for general AM53C974 based adapters and some possibilities to configure it were added. The non-DC390 support works by assuming some values for the data normally taken from the DC390 EEPROM. See below (chapter 5) for details.

When using the DC390, the configuration is still be done using the DC390 BIOS setup. The DC390 EEPROM is read and used by the driver, any boot or module parameters (chapter 5) are ignored! However, you can change settings dynamically, as described in chapter 4.

For a more detailed description of the driver's history, see the first lines of tmcsim.c.

The numbering scheme isn't consistent. The first versions went from 1.00 to 1.12, then 1.20a to 1.20t. Finally I decided to use the ncr53c8xx scheme. So the next revisions will be 2.0a to 2.0X (stable), 2.1a to 2.1X (experimental), 2.2a to 2.2X (stable, again) etc. (X = anything between a and z.) If I send fixes to people for testing, I create intermediate versions with a digit appended, e.g. 2.0c3.

2. Installation

If you got any recent kernel with this driver and document included in linux/drivers/scsi, you basically have to do nothing special to use this

tmscsim.txt

driver. Of course you have to choose to compile SCSI support and DC390(T) support into your kernel or as module when configuring your kernel for compiling.

NEW: You may as well compile this module outside your kernel, using the supplied Makefile.

If you got an old kernel (pre 2.1.127, pre 2.0.37p1) with an old version of this driver: Get dc390-21125-20b.diff.gz or dc390-2036p21-20b1.diff.gz from my web page and apply the patch. Apply further patches to upgrade to the latest version of the driver.

If you want to do it manually, you should copy the files (dc390.h, tmscsim.h, tmscsim.c, scsiiom.c and README.tmscsim) from this directory to linux/drivers/scsi. You have to recompile your kernel/module of course.

You should apply the three patches included in dc390-120-kernel.diff (Applying them: cd /usr/src; patch -p0 <~/dc390-120-kernel.diff)
The patches are against 2.1.125, so you might have to manually resolve rejections when applying to another kernel version.

The patches will update the kernel startup code to allow boot parameters to be passed to the driver, update the Documentation and finally offer you the possibility to omit the non-DC390 parts of the driver.

(By selecting "Omit support for non DC390" you basically disable the emulation of a DC390 EEPROM for non DC390 adapters. This saves a few bytes of memory.)

If you got a very old kernel without the tmscsim driver (pre 2.0.31) I recommend upgrading your kernel. However, if you don't want to, please contact me to get the appropriate patches.

Upgrading a SCSI driver is always a delicate thing to do. The 2.0 driver has proven stable on many systems, but it's still a good idea to take some precautions. In an ideal world you would have a full backup of your disks. The world isn't ideal and most people don't have full backups (me neither). So take at least the following measures:

- * make your kernel remount the FS read-only on detecting an error:
tune2fs -e remount-ro /dev/sd??
- * have copies of your SCSI disk's partition tables on some safe location:
dd if=/dev/sda of=/mnt/floppy/sda bs=512 count=1
or just print it with:
fdisk -l | lpr
- * make sure you are able to boot Linux (e.g. from floppy disk using InitRD) if your SCSI disk gets corrupted. You can use
ftp://student.physik.uni-dortmund.de/pub/linux/kernel/bootdisk.gz

One more warning: I used to overclock my PCI bus to 41.67 MHz. My Tekram DC390F (Sym53c875) accepted this as well as my Millenium. But the Am53C974 produced errors and started to corrupt my disks. So don't do that! A 37.50 MHz PCI bus works for me, though, but I don't recommend using higher clocks than the 33.33 MHz being in the PCI spec.

If you want to share the IRQ with another device and the driver refuses to do so, you might succeed with changing the DC390_IRQ type in tmscsim.c to IRQF_SHARED | IRQF_DISABLED.

3. Features

- SCSI
 - * Tagged command queueing
 - * Sync speed up to 10 MHz
 - * Disconnection
 - * Multiple LUNs
- General / Linux interface
 - * Support for up to 4 AM53C974 adapters.
 - * DC390 EEPROM usage or boot/module params
 - * Information via `cat /proc/scsi/tmcsim/?`
 - * Dynamically configurable by writing to `/proc/scsi/tmcsim/?`
 - * Dynamic allocation of resources
 - * SMP support: Locking on `io_request` lock (Linux 2.1/2.2) or adapter specific locks (Linux 2.5?)
 - * Uniform source code for Linux-2.x.y
 - * Support for dyn. addition/removal of devices via `add/remove-single-device` (Try: `echo "scsi add-single-device C B T U" >/proc/scsi/scsi`
C = Controller, B = Bus, T = Target SCSI ID, U = Unit SCSI LUN.)
Use with care!
 - * Try to use the partition table for the determination of the mapping

4. Configuration via `/proc/scsi/tmcsim/?`

First of all look at the output of `/proc/scsi/tmcsim/?` by typing

```
cat /proc/scsi/tmcsim/?
```

The "?" should be replaced by the SCSI host number. (The shell might do this for you.)

You will see some info regarding the adapter and, at the end, a listing of the attached devices and their settings.

Here's an example:

```
garloff@kurt:/home/garloff > cat /proc/scsi/tmcsim/0
Tekram DC390/AM53C974 PCI SCSI Host Adapter, Driver Version 2.0e7 2000-11-28
SCSI Host Nr 1, AM53C974 Adapter Nr 0
IOPortBase 0xb000, IRQ 10
MaxID 8, MaxLUN 8, AdapterID 6, SelTimeout 250 ms, DelayReset 1 s
TagMaxNum 16, Status 0x00, ACBFlag 0x00, GlitchEater 24 ns
Statistics: CmnDs 1470165, CmnDs not sent directly 0, Out of SRB conds 0
          Lost arbitrations 587, Sel. connected 0, Connected: No
Nr of attached devices: 4, Nr of DCBs: 4
Map of attached LUNs: 01 00 00 03 01 00 00 00
Idx ID LUN Prty Sync DsCn SndS TagQ NegoPeriod SyncSpeed SyncOffs MaxCmd
00 00 00 Yes Yes Yes Yes Yes 100 ns 10.0 M 15 16
01 03 00 Yes Yes Yes Yes No 100 ns 10.0 M 15 01
02 03 01 Yes Yes Yes Yes No 100 ns 10.0 M 15 01
03 04 00 Yes Yes Yes Yes No 100 ns 10.0 M 15 01
```

Note that the settings `MaxID` and `MaxLUN` are not zero- but one-based, which means that a setting `MaxLUN=4`, will result in the support of LUNs 0..3. This is somehow inconvenient, but the way the mid-level SCSI code expects it to be.

ACB and DCB are acronyms for Adapter Control Block and Device Control Block. These are data structures of the driver containing information about the adapter and the connected SCSI devices respectively.

Idx is the device index (just a consecutive number for the driver), ID and LUN are the SCSI ID and LUN, Prty means Parity checking, Sync synchronous negotiation, DsCn Disconnection, SndS Send Start command on startup (not used by the driver) and TagQ Tagged Command Queueing. NegoPeriod and SyncSpeed are somehow redundant, because they are reciprocal values ($1 / 112 \text{ ns} = 8.9 \text{ MHz}$). At least in theory. The driver is able to adjust the NegoPeriod more accurate (4ns) than the SyncSpeed ($1 / 25\text{ns}$). I don't know if certain devices will have problems with this discrepancy. Max. speed is 10 MHz corresp. to a min. NegoPeriod of 100 ns.

(The driver allows slightly higher speeds if the devices (Ultra SCSI) accept it, but that's out of adapter spec, on your own risk and unlikely to improve performance. You're likely to crash your disks.)

SyncOffs is the offset used for synchronous negotiations; max. is 15.

The last values are only shown, if Sync is enabled. (NegoPeriod is still displayed in brackets to show the values which will be used after enabling Sync.)

MaxCmd ist the number of commands (=tags) which can be processed at the same time by the device.

If you want to change a setting, you can do that by writing to `/proc/scsi/tmscsim/?.` Basically you have to imitate the output of driver. (Don't use the brackets for NegoPeriod on Sync disabled devices.)

You don't have to care about capitalisation. The driver will accept space, tab, comma, = and : as separators.

There are three kinds of changes:

(1) Change driver settings:

You type the names of the parameters and the params following it.

Example:

```
echo "MaxLUN=8 seltimeout 200" >/proc/scsi/tmscsim/0
```

Note that you can only change MaxID, MaxLUN, AdapterID, SelTimeOut, TagMaxNum, ACBFlag, GlitchEater and DelayReset. Don't change ACBFlag unless you want to see what happens, if the driver hangs.

(2) Change device settings: You write a config line to the driver. The Nr must match the ID and LUN given. If you give "-" as parameter, it is ignored and the corresponding setting won't be changed.

You can use "y" or "n" instead of "Yes" and "No" if you want to.

You don't need to specify a full line. The driver automatically performs an INQUIRY on the device if necessary to check if it is capable to operate with the given settings (Sync, TagQ).

Examples:

```
echo "0 0 0 y y y - y - 10 " >/proc/scsi/tmscsim/0
```

```
echo "3 5 0 y n y " >/proc/scsi/tmscsim/0
```

To give a short explanation of the first example:

The first three numbers, "0 0 0" (Device index 0, SCSI ID 0, SCSI LUN 0), select the device to which the following parameters apply. Note that it would be sufficient to use the index or both SCSI ID and LUN, but I chose to require all three to have a syntax similar to the output.

tmcsim.txt

The following "y y y - y" enables Parity checking, enables Synchronous transfers, Disconnection, leaves Send Start (not used) untouched and enables Tagged Command Queueing for the selected device. The "-" skips the Negotiation Period setting but the "10" sets the max sync. speed to 10 MHz. It's useless to specify both NegoPeriod and SyncSpeed as discussed above. The values used in this example will result in maximum performance.

- (3) Special commands: You can force a SCSI bus reset, an INQUIRY command, the removal or the addition of a device's DCB and a SCSI register dump. This is only used for debugging when you meet problems. The parameter of the INQUIRY and REMOVE commands is the device index as shown by the output of /proc/scsi/tmcsim/? in the device listing in the first column (Idx). ADD takes the SCSI ID and LUN.

Examples:

```
echo "reset" >/proc/scsi/tmcsim/0
echo "inquiry 1" >/proc/scsi/tmcsim/0
echo "remove 2" >/proc/scsi/tmcsim/1
echo "add 2 3" >/proc/scsi/tmcsim/?
echo "dump" >/proc/scsi/tmcsim/0
```

Note that you will meet problems when you REMOVE a device's DCB with the remove command if it contains partitions which are mounted. Only use it after unmounting its partitions, telling the SCSI mid-level code to remove it (scsi remove-single-device) and you really need a few bytes of memory.

The ADD command allows you to configure a device before you tell the mid-level code to try detection.

I'd suggest reviewing the output of /proc/scsi/tmcsim/? after changing settings to see if everything changed as requested.

5. Configuration via boot/module parameters

With the DC390, the driver reads its EEPROM settings and tries to use them. But you may want to override the settings prior to being able to change the driver configuration via /proc/scsi/tmcsim/?.

If you do have another AM53C974 based adapter, that's even the only possibility to adjust settings before you are able to write to the /proc/scsi/tmcsim/? pseudo-file, e.g. if you want to use another adapter ID than 7.

(BTW, the log message "DC390: No EEPROM found!" is normal without a DC390.)

For this purpose, you can pass options to the driver before it is initialised by using kernel or module parameters. See lilo(8) or modprobe(1) manual pages on how to pass params to the kernel or a module.

[NOTE: Formerly, it was not possible to override the EEPROM supplied settings of the DC390 with cmd line parameters. This has changed since 2.0e7]

The syntax of the params is much shorter than the syntax of the /proc/... interface. This makes it a little bit more difficult to use. However, long parameter lines have the risk to be misinterpreted and the length of kernel parameters is limited.

tmscsim.txt

As the support for non-DC390 adapters works by simulating the values of the DC390 EEPROM, the settings are given in a DC390 BIOS' way.

Here's the syntax:

tmscsim=AdaptID, SpdIdx, DevMode, AdaptMode, TaggedCmds, DelayReset

Each of the parameters is a number, containing the described information:

- * AdaptID: The SCSI ID of the host adapter. Must be in the range 0..7
Default is 7.
- * SpdIdx: The index of the maximum speed as in the DC390 BIOS. The values 0..7 mean 10, 8.0, 6.7, 5.7, 5.0, 4.0, 3.1 and 2 MHz resp. Default is 0 (10.0 MHz).
- * DevMode is a bit mapped value describing the per-device features. It applies to all devices. (Sync, Disc and TagQ will only apply, if the device supports it.) The meaning of the bits (* = default):

Bit	Val(hex)	Val(dec)	Meaning
*0	0x01	1	Parity check
*1	0x02	2	Synchronous Negotiation
*2	0x04	4	Disconnection
*3	0x08	8	Send Start command on startup. (Not used)
*4	0x10	16	Tagged Command Queueing

As usual, the desired value is obtained by adding the wanted values. If you want to enable all values, e.g., you would use 31(0x1f). Default is 31.

- * AdaptMode is a bit mapped value describing the enabled adapter features.

Bit	Val(hex)	Val(dec)	Meaning
*0	0x01	1	Support more than two drives. (Not used)
*1	0x02	2	Use DOS compatible mapping for HDs greater than 1GB.
*2	0x04	4	Reset SCSI Bus on startup.
*3	0x08	8	Active Negation: Improves SCSI Bus noise immunity.
4	0x10	16	Immediate return on BIOS seek command. (Not used)
(*)5	0x20	32	Check for LUNs >= 1.

The default for LUN Check depends on CONFIG SCSI_MULTI_LUN.

- * TaggedCmds is a number indicating the maximum number of Tagged Commands. It is the binary logarithm - 1 of the actual number. Max is 4 (32).

Value Number of Tagged Commands

0	2
1	4
2	8
*3	16
4	32

- * DelayReset is the time in seconds (minus 0.5s), the adapter waits, after a bus reset. Default is 1 (corresp. to 1.5s).

Example:

modprobe tmscsim tmscsim=6,2,31
would set the adapter ID to 6, max. speed to 6.7 MHz, enable all device

tmcsim.txt

features and leave the adapter features, the number of Tagged Commands and the Delay after a reset to the defaults.

As you can see, you don't need to specify all of the six params. If you want values to be ignored (i.e. the EEprom settings or the defaults will be used), you may pass -2 (not 0!) at the corresponding position.

The defaults (7,0,31,15,3,1) are aggressive to allow good performance. You can use tmcsim=7,0,31,63,4,0 for maximum performance, if your SCSI chain allows it. If you meet problems, you can use tmcsim=-1 which is a shortcut for tmcsim=7,4,9,15,2,10.

6. Potential improvements

Most of the intended work on the driver has been done. Here are a few ideas to further improve its usability:

- * Cleanly separate per-Target and per-LUN properties (DCB)
- * More intelligent abort() routine
- * Use new_eh code (Linux-2.1+)
- * Have the mid-level (ML) code (and not the driver) handle more of the various conditions.
- * Command queueing in the driver: Eliminate Query list and use ML instead.
- * More user friendly boot/module param syntax

Further investigation on these problems:

- * Driver hangs with sync readcdda (xcdroast) (most probably VIA PCI error)

Known problems:

Please see <http://www.garloff.de/kurt/linux/dc390/problems.html>

- * Changing the parameters of multi-lun by the tmcsim/? interface will cause problems, cause these settings are mostly per Target and not per LUN and should be updated accordingly. To be fixed for 2.0d24.
- * CDRs (eg Yam CRW4416) not recognized, because some buggy devices don't recover from a SCSI reset in time. Use a higher delay or don't issue a SCSI bus reset on driver initialization. See problems page.
For the CRW4416S, this seems to be solved with firmware 1.0g (reported by Jean-Yves Barbier).
- * TEAC CD-532S not being recognized. (Works with 1.11).
- * Scanners (eg. Astra UMAX 1220S) don't work: Disable Sync Negotiation.
If this does not help, try echo "INQUIRY t" >/proc/scsi/tmcsim/? (t replaced by the dev index of your scanner). You may try to reset your SCSI bus afterwards (echo "RESET" >/proc/scsi/tmcsim/?).
The problem seems to be solved as of 2.0d18, thanks to Andreas Rick.
- * If there is a valid partition table, the driver will use it for determining the mapping. If there's none, a reasonable mapping (Symbios-like) will be assumed. Other operating systems may not like this mapping, though it's consistent with the BIOS' behaviour. Old DC390 drivers ignored the partition table and used a H/S = 64/32 or 255/63 translation. So if you want to be compatible to those, use this old mapping when creating partition tables. Even worse, on bootup the DC390 might complain if other mappings are found, so auto rebooting may fail.
- * In some situations, the driver will get stuck in an abort loop. This is a

tmcsim.txt

bad interaction between the Mid-Layer of Linux' SCSI code and the driver. Try to disable DsCn, if you meet this problem. Please contact me for further debugging.

7. Bug reports, debugging and updates

Whenever you have problems with the driver, you are invited to ask the author for help. However, I'd suggest reading the docs and trying to solve the problem yourself, first.

If you find something, which you believe to be a bug, please report it to me. Please append the output of /proc/scsi/scsi, /proc/scsi/tmcsim/? and maybe the DC390 log messages to the report.

Bug reports should be send to me (Kurt Garloff <dc390@garloff.de>) as well as to the linux-scsi list (<linux-scsi@vger.kernel.org>), as sometimes bugs are caused by the SCSI mid-level code.

I will ask you for some more details and probably I will also ask you to enable some of the DEBUG options in the driver (tmcsim.c:DC390_DEBUGXXX defines). The driver will produce some data for the syslog facility then. Beware: If your syslog gets written to a SCSI disk connected to your AM53C974, the logging might produce log output again, and you might end having your box spending most of its time doing the logging.

The latest version of the driver can be found at:

<http://www.garloff.de/kurt/linux/dc390/>

<ftp://ftp.suse.com/pub/people/garloff/linux/dc390/>

8. Acknowledgements

Thanks to Linus Torvalds, Alan Cox, the FSF people, the XFree86 team and all the others for the wonderful OS and software.

Thanks to C.L. Huang and Philip Giang (Tekram) for the initial driver release and support.

Thanks to Doug Ledford, Gérard Roudier for support with SCSI coding.

Thanks to a lot of people (espec. Chiaki Ishikawa, Andreas Haumer, Hubert Tonneau) for intensively testing the driver (and even risking data loss doing this during early revisions).

Recently, SuSE GmbH, Nuernberg, FRG, has been paying me for the driver development and maintenance. Special thanks!

9. Copyright

This driver is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; version 2 of the License.

If you want to use any later version of the GNU GPL, you will probably be allowed to, but you have to ask me and Tekram <erich@tekram.com.tw> before.

Written by Kurt Garloff <kurt@garloff.de> 1998/06/11

Last updated 2000/11/28, driver revision 2.0e7

tmcsim.txt

\$Id: README.tmcsim,v 2.25.2.7 2000/12/20 01:07:12 garloff Exp \$