## General Description

This driver supports the 53c700 and 53c700-66 chips. It also supports the 53c710 but only in 53c700 emulation mode. It is full featured and does sync (-66 and 710 only), disconnects and tag command queueing.

Since the 53c700 must be interfaced to a bus, you need to wrapper the card detector around this driver. For an example, see the NCR_D700.[ch] or lasi700.[ch] files.

The comments in the 53c700.[ch] files tell you which parts you need to fill in to get the driver working.

## Compile Time Flags

The driver may be either io mapped or memory mapped. This is selectable by configuration flags:

CONFIG_53C700_MEM_MAPPED

define if the driver is memory mapped.

CONFIG_53C700_IO_MAPPED

define if the driver is to be io mapped.

One or other of the above flags *must* be defined.

Other flags are:

CONFIG_53C700_LE_ON_BE

define if the chipset must be supported in little endian mode on a big endian architecture (used for the 700 on parisc).

CONFIG_53C700_USE_CONSISTENT

allocate consistent memory (should only be used if your architecture has a mixture of consistent and inconsistent memory). Fully consistent or fully inconsistent architectures should not define this.

## Using the Chip Core Driver

In order to plumb the 53c700 chip core driver into a working SCSI driver, you need to know three things about the way the chip is wired into your system (or expansion card).

1. The clock speed of the SCSI core
2. The interrupt line used
3. The memory (or io space) location of the 53c700 registers.

Optionally, you may also need to know other things, like how to read the SCSI Id from the card bios or whether the chip is wired for differential operation.

Usually you can find items 2. and 3. from general spec. documents or even by examining the configuration of a working driver under another operating system.

The clock speed is usually buried deep in the technical literature. It is required because it is used to set up both the synchronous and asynchronous dividers for the chip. As a general rule of thumb, manufacturers set the clock speed at the lowest possible setting consistent with the best operation of the chip (although some choose to drive it off the CPU or bus clock rather than going to the expense of an extra clock chip). The best operation clock speeds are:

53c700 - 25MHz
53c700-66 - 50MHz
53c710 - 40Mhz

Writing Your Glue Driver
========================

This will be a standard SCSI driver (I don't know of a good document describing this, just copy from some other driver) with at least a detect and release entry.

In the detect routine, you need to allocate a struct NCR_700_Host_Parameters sized memory area and clear it (so that the default values for everything are 0). Then you must fill in the parameters that matter to you (see below), plumb the NCR_700_intr routine into the interrupt line and call NCR_700_detect with the host template and the new parameters as arguments. You should also call the relevant request_*_region function and place the register base address into the `base' pointer of the host parameters.

In the release routine, you must free the NCR_700_Host_Parameters that you allocated, call the corresponding release_*_region and free the interrupt.

Handling Interrupts
-------------------

In general, you should just plumb the card's interrupt line in with

request_irq(irq, NCR_700_intr, <irq flags>, <driver name>, host);

where host is the return from the relevant NCR_700_detect() routine.

You may also write your own interrupt handling routine which calls NCR_700_intr() directly. However, you should only really do this if you have a card with more than one chip on it and you can read a register to tell which set of chips wants the interrupt.

Settable NCR_700_Host_Parameters
--------------------------------

The following are a list of the user settable parameters:

clock: (MANDATORY)

Set to the clock speed of the chip in MHz.

base: (MANDATORY)

set to the base of the io or mem region for the register set. On 64 bit architectures this is only 32 bits wide, so the registers must be mapped into the low 32 bits of memory.

pci_dev: (OPTIONAL)

set to the PCI board device. Leave NULL for a non-pci board. This is used for the pci_alloc_consistent() and pci_map_*() functions.

dmode_extra: (OPTIONAL, 53c710 only)

extra flags for the DMODE register. These are used to control bus output pins on the 710. The settings should be a combination of DMODE_FC1 and DMODE_FC2. What these pins actually do is entirely up to the board designer. Usually it is safe to ignore this setting.

differential: (OPTIONAL)

set to 1 if the chip drives a differential bus.

force_le_on_be: (OPTIONAL, only if CONFIG_53C700_LE_ON_BE is set)

set to 1 if the chip is operating in little endian mode on a big endian architecture.

chip710: (OPTIONAL)

set to 1 if the chip is a 53c710.

burst_disable: (OPTIONAL, 53c710 only)

disable 8 byte bursting for DMA transfers.