NAME

isp – ISP SCSI Host Bus Adapter Driver

SYNOPSIS

isp@*sbus-slot*,0x10000

DESCRIPTION

The **isp** Host Bus Adapter is a SCSA compliant nexus driver that supports the Qlogic ISP1000 SCSI chip. The ISP1000 SCSI is an intelligent SCSI Host Bus Adapter chip that reduces the amount of CPU overhead used in a SCSI transfer.

The **isp** driver supports the standard functions provided by the SCSA interface. The driver supports tagged and untagged queuing, fast and wide SCSI, auto request sense but does not support linked commands.

CONFIGURATION

The **isp** driver can be configured by defining properties in **isp.conf** which override the global SCSI settings. Supported properties are **scsi-options, target<***n***>-scsi-options, scsi-reset-delay, scsi-watchdog-tick, scsi-tag-age-limit, scsi-initiator-id**.

target<*n***>-scsi-options** overrides the **scsi-options** property value for **target<***n***>**. **<***n***>** can vary from 0 to 15.

Refer to **scsi_hba_attach**(9F) for details.

EXAMPLES

Create a file **/kernel/drv/isp.conf** and add this line:

scsi-options=0x78;

This will disable tagged queuing, fast SCSI, and Wide mode for all **isp** instances. To disable an option for one specific **isp** (refer to **driver.conf**(4)):

```
name="isp"  parent="/iommu@f,e0000000/sbus@f,e0001000"
    reg=1,0x10000,450
    target1-scsi-options=0x58
    scsi-options = 0x178 scsi-initiator-id = 15;
```

Note that the default initiator ID in OBP is **7** and that the change to ID **15** will occur at attach time. It may be preferable to change the initiator ID in OBP.

The above would set **scsi-options** for target 1 to **0x58** and for all other targets on this SCSI bus to **0x178**.

The physical pathname of the parent can be determined using the **/devices** tree or following the link of the logical device name:

```
example# ls -l /dev/rdsk/c2t0d0s0
lrwxrwxrwx   1 root   root  76 Aug 22 13:29 /dev/rdsk/c2t0d0s0 ->
../../devices/iommu@f,e0000000/sbus@f,e0001000/QLGC,isp@1,10000/sd@0,0:a,raw
```

The register property values can be determined from **prtconf**(1M) output (**–v** option):

```
    QLGC,isp, instance #0
        ...
      Register Specifications:
        Bus Type=0x1, Address=0x10000, Size=450
```

FILES

/kernel/drv/isp	ELF Kernel Module
/kernel/drv/isp.conf	Configuration file

SEE ALSO

prtconf(1M), **driver.conf**(4), **scsi_abort**(9F), **scsi_hba_attach**(9F), **scsi_ifgetcap**(9F), **scsi_reset**(9F), **scsi_transport**(9F), **scsi_device**(9S), **scsi_extended_sense**(9S), **scsi_inquiry**(9S), **scsi_pkt**(9S)

Writing Device Drivers

ANSI Small Computer System Interface-2 (SCSI-2)

ISP1000 Firmware Interface Specification, QLogic Corp.

ISP1000 Technical Manual, QLogic Corp.

DIAGNOSTICS

The messages described below are some that may appear on the system console, as well as being logged.

This first set of messages may be displayed while the **isp** driver is first trying to attach. All of these messages mean that the **isp** driver was unable to attach. These messages are preceded by "isp%d", where "%d" is the instance number of the isp controller.

Device in slave-only slot, unused
The SBus device has been placed in a slave-only slot and will not be accessible; move to non-slave-only SBus slot.

Device is using a hilevel intr, unused
The device was configured with a interrupt level that cannot be used with this **isp** driver, check the SBus device.

Failed to alloc soft state
Driver was unable to allocate space for the internal state structure. Driver did not attach to device, SCSI devices will be inaccessible.

Bad soft state
Driver requested an invalid internal state structure. Driver did not attach to device, SCSI devices will be inaccessible.

Unable to map registers;
Driver was unable to map device registers; check for bad hardware. Driver did not attach to device, SCSI devices will be inaccessible.

Cannot add intr
Driver was not able to add the interrupt routine to the kernel. Driver did not attach to device, SCSI devices will be inaccessible.

Unable to attach
Driver was unable to attach to the hardware for some reason that may be printed. Driver did not attach to device, SCSI devices will be inaccessible.

This next set of messages can be displayed at any time, they will be printed with the full device pathname followed by the shorter form described above.

Firmware should be < 0x%x bytes
Firmware size exceeded allocated space; will not download firmware. This could

mean that the firmware was corrupted somehow; check the **isp** driver.

Firmware checksum incorrect
Firmware has an invalid checksum and will not be downloaded.

Chip reset timeout
ISP1000 failed to reset in the time allocated, may be bad hardware.

Stop firmware failed
Stopping the firmware failed, may be bad hardware.

Load ram failed
Unable to download new firmware into the ISP1000 chip

DMA setup failed
The DMA setup failed in the host adapter driver on a **scsi_pkt**; this will return **TRAN_BADPKT** to a SCSA target driver.

Bad request pkt
The ISP Firmware rejected the packet as being setup incorrectly. This will cause the **isp** driver to call the target completion routine with the reason of **CMD_TRAN_ERR** set in the **scsi_pkt**. Check the target driver for correctly setting up the packet.

Bad request pkt header
The ISP Firmware rejected the packet as being setup incorrectly. This will cause the **isp** driver to call the target completion routine with the reason of **CMD_TRAN_ERR** set in the **scsi_pkt**. Check the target driver for correctly setting up the packet.

Polled command timeout on %d.%d
A polled command experienced a timeout; the target device, as noted by the target lun (%d.%d) info, may not be responding correctly to the command, or the ISP1000 chip may be hung. This will cause an error recovery to be initiated in the **isp** driver. This could mean a bad device or cabling.

Firmware error
The ISP1000 Chip encountered a firmware error of some kind. This error will cause the **isp** driver to do error recovery by resetting the chip.

Received unexpected SCSI Reset
The ISP1000 chip received an unexpected SCSI Reset and has initiated its own internal error recovery, which will return all the **scsi_pkt** with reason set to **CMD_RESET.**

Fatal timeout on target %d.%d
The **isp** driver found a command that had not completed in the correct amount of time; this will cause error recovery by the **isp** driver. The device that experienced the timeout was at target lun (%d.%d).

Fatal error, resetting interface

This is an indication that the **isp** driver is doing error recovery. This will cause all outstanding commands that have been transported to the **isp** driver to be completed via the **scsi_pkt** completion routine in the target driver with reason of **CMD_RESET** and status of **STAT_BUS_RESET** set in the **scsi_pkt**.

NOTES

The **isp** driver exports properties indicating per target the negotiated transfer speed **(target<n>-sync-speed),** whether tagged queuing has been enabled **(target<n>-TQ),** and whether the wide data transfer has been negotiated **(target<n>-wide).** The sync-speed property value is the data transfer rate in KB/sec. The target-TQ and target-wide properties have no value. The existence of these properties indicate that tagged queuing or wide transfer has been enabled. Refer to **prtconf**(1M) (verbose option) for viewing the **isp** properties.

```
QLGC,isp, instance #2
  Driver software properties:
    name <target0-TQ> length <0> -- <no value>.
    name <target0-wide> length <0> -- <no value>.
    name <target0-sync-speed> length <4>
      value <0x000028f5>.
    name <scsi-options> length <4>
      value <0x000003f8>.
    name <scsi-watchdog-tick> length <4>
      value <0x0000000a>.
    name <scsi-tag-age-limit> length <4>
      value <0x00000008>.
    name <scsi-reset-delay> length <4>
      value <0x00000bb8>.
```