

NAME

bce - QLogic NetXtreme II (BCM5706/5708/5709/5716) PCI/PCIe Gigabit Ethernet adapter driver

SYNOPSIS

To compile this driver into the kernel, place the following lines in your kernel configuration file:

```
device miibus
device bce
```

Alternatively, to load the driver as a module at boot time, place the following line in loader.conf(5):

```
if_bce_load="YES"
```

DESCRIPTION

The **bce** driver supports QLogic's NetXtreme II product family, including the BCM5706, BCM5708, BCM5709 and BCM5716 Ethernet controllers.

The NetXtreme II product family is composed of various Converged NIC (or CNIC) Ethernet controllers which support a TCP Offload Engine (TOE), Remote DMA (RDMA), and iSCSI acceleration, in addition to standard L2 Ethernet traffic, all on the same controller.

The following features are supported in the **bce** driver under FreeBSD:

- IP/TCP/UDP checksum offload
- Jumbo frames (up to 9022 bytes)
- VLAN tag stripping
- Interrupt coalescing
- 10/100/1000Mbps operation in full-duplex mode
- 10/100Mbps operation in half-duplex mode

The **bce** driver supports the following media types:

autoselect Enable autoselection of the media type and options. The user can manually override the autoselected mode by adding media options to rc.conf(5).

10baseT/UTP Set 10Mbps operation. The ifconfig(8) **mediaopt** option can also be used to select either **full-duplex** or **half-duplex** modes.

100baseTX Set 100Mbps (Fast Ethernet) operation. The ifconfig(8) **mediaopt** option can also be used to select either **full-duplex** or **half-duplex** modes.

1000baseSX Sets 1000Mbps operation. Only **full-duplex** mode is supported at this speed.

1000baseT Set 1000baseT operation over twisted pair. Only **full-duplex** mode is supported.

2500BaseSX Set 2500Mbps operation. Only **full-duplex** mode is supported.

The **bce** driver supports the following media options:

full-duplex Force full duplex operation.

half-duplex

Force half duplex operation.

For more information on configuring this device, see `ifconfig(8)`.

HARDWARE

The **bce** driver provides support for various NICs based on the QLogic NetXtreme II family of Gigabit Ethernet controllers, including the following:

- ⌘ QLogic NetXtreme II BCM5706 1000Base-SX
- ⌘ QLogic NetXtreme II BCM5706 1000Base-T
- ⌘ QLogic NetXtreme II BCM5708 1000Base-SX
- ⌘ QLogic NetXtreme II BCM5708 1000Base-T
- ⌘ QLogic NetXtreme II BCM5709 1000Base-SX
- ⌘ QLogic NetXtreme II BCM5709 1000Base-T
- ⌘ QLogic NetXtreme II BCM5716 1000Base-T
- ⌘ Dell PowerEdge 1950 integrated BCM5708 NIC
- ⌘ Dell PowerEdge 2950 integrated BCM5708 NIC
- ⌘ Dell PowerEdge R710 integrated BCM5709 NIC
- ⌘ HP NC370F Multifunction Gigabit Server Adapter
- ⌘ HP NC370T Multifunction Gigabit Server Adapter
- ⌘ HP NC370i Multifunction Gigabit Server Adapter
- ⌘ HP NC371i Multifunction Gigabit Server Adapter
- ⌘ HP NC373F PCIe Multifunc Giga Server Adapter
- ⌘ HP NC373T PCIe Multifunction Gig Server Adapter
- ⌘ HP NC373i Multifunction Gigabit Server Adapter
- ⌘ HP NC373m Multifunction Gigabit Server Adapter
- ⌘ HP NC374m PCIe Multifunction Adapter
- ⌘ HP NC380T PCIe DP Multifunc Gig Server Adapter
- ⌘ HP NC382T PCIe DP Multifunction Gigabit Server Adapter

- HP NC382i DP Multifunction Gigabit Server Adapter
- HP NC382m DP 1GbE Multifunction BL-c Adapter

SYSCTL VARIABLES

The following variables are available as both sysctl(8) variables and loader(8) tunables:

hw.bce.verbose

Enable/Disable verbose logging and output to the console. Useful for debugging (default 0).

hw.bce.msi_enable

Enable/Disable MSI support (default 1).

hw.bce.tso_enable

Enable/Disable TSO support (default 1).

hw.bce.strict_rx_mtu

Enable/Disable strict RX frame size checking (default 0).

hw.bce.hdr_split

Enable/Disable frame header/payload splitting (default 1).

hw.bce.rx_pages

Set the number of memory pages assigned to receive packets by the driver. Due to alignment issues, this value can only be of the set 1, 2, 4 or 8 (default 2).

hw.bce.tx_pages

Set the number of memory pages assigned to transmit packets by the driver. Due to alignment issues, this value can only be of the set 1, 2, 4 or 8 (default 2).

hw.bce.rx_ticks

Time in microsecond ticks to wait before generating a status block updates due to RX processing activity. Values from 0-100 are valid. A value of 0 disables this status block update. Cannot be set to 0 if *hw.bce.rx_quick_cons_trip* is also 0 (default 18).

hw.bce.rx_ticks_int

Time in microsecond ticks to wait during RX interrupt processing before generating a status block update. Values from 0-100 are valid. Valid values are in the range from 0-100. A value of 0 disables this status block update (default 18).

hw.bce.rx_quick_cons_trip

Number of RX Quick BD Chain entries that must be completed before a status block is generated. Values from 0-256 are valid. A value of 0 disables this status block update. Cannot be set to 0 if `hw.bce.rx_ticks` is also 0 (default 6).

hw.bce.rx_quick_cons_trip_int

Number of RX quick BD entries that must be completed before a status block is generated during interrupt processing. Values from 0-256 are valid. A value of 0 disables this status block update (default 6).

hw.bce.tx_ticks

Time in microsecond ticks to wait before a status block update is generated due to TX activity. Values from 0-100 are valid. A value of 0 disables this status block update. Cannot be set to 0 if `hw.bce.tx_quick_cons_trip` is also 0 (default 80).

hw.bce.tx_ticks_int

Time in microsecond ticks to wait in interrupt processing before a status block update is generated due to TX activity. Values from 0-100 are valid. A value of 0 disables this status block update (default 80).

hw.bce.tx_cons_trip

How many TX Quick BD Chain entries that must be completed before a status block is generated. Values from 0-100 are valid. A value of 0 disables this status block update. Cannot be set to 0 if `hw.bce.tx_ticks` is also 0 (default 20).

hw.bce.tx_cons_trip_int

How many TX Quick BD Chain entries that must be completed before a status block is generated during an interrupt. Values from 0-100 are valid. A value of 0 disables this status block update (default 20).

DIAGNOSTICS

bce%d: PCI memory allocation failed! The driver has encountered a fatal initialization error.

bce%d: PCI map interrupt failed! The driver has encountered a fatal initialization error.

bce%d: Unsupported controller revision (%c%d) The driver does not support the controller revision in use.

bce%d: Controller initialization failed! The driver has encountered a fatal initialization error.

bce%d: NVRAM test failed! The driver could not access the controller NVRAM correctly.

bce%d: DMA resource allocation failed! The driver could not allocate DMA memory to setup the controllers host memory data structures.

bce%d: Interface allocation failed! The driver could not create a network interface for the controller.

bce%d: PHY probe failed! The driver could not access the PHY used by the controller.

bce%d: Failed to setup IRQ! The driver could not initialize the IRQ handler.

bce%d: Error: PHY read timeout! The driver could not read a PHY register before the timeout period expired.

bce%d: PHY write timeout! The driver could not write to the PHY register because a timeout occurred.

bce%d: Timeout error reading NVRAM at offset 0x%08X! The driver could not write to NVRAM because a timeout occurred.

bce%d: Unknown Flash NVRAM found! The driver does not recognize the NVRAM device being used and therefore cannot access it correctly.

bce%d: Invalid NVRAM magic value! The driver cannot read NVRAM or the NVRAM is corrupt.

bce%d: Invalid Manufacturing Information NVRAM CRC! The driver cannot read NVRAM or the NVRAM is corrupt.

bce%d: Invalid Feature Configuration Information NVRAM CRC! The driver cannot read NVRAM or the NVRAM is corrupt.

bce%d: DMA mapping error! The driver was unable to map memory into DMA addressable space required by the controller.

bce%d: Could not allocate parent DMA tag! The driver could not allocate a PCI compatible DMA tag.

bce%d: Could not allocate status block DMA tag! The driver could not allocate a DMA tag for the controller's status block.

bce%d: Could not allocate status block DMA memory! The driver could not allocate DMA addressable memory for the controller's status block.

bce%d: Could not map status block DMA memory! The driver could not map the status block memory

into the controller's DMA address space.

bce%d: Could not allocate statistics block DMA tag! The driver could not allocate a DMA tag for the controller's statistics block.

bce%d: Could not allocate statistics block DMA memory! The driver could not allocate DMA addressable memory for the controller's statistics block.

bce%d: Could not map statistics block DMA memory! The driver could not map the statistics block memory into the controller's DMA address space.

bce%d: Could not allocate TX descriptor chain DMA tag! The driver could not allocate a DMA tag for the controller's TX chain.

bce%d: Could not allocate TX descriptor chain DMA memory! The driver could not allocate DMA addressable memory for the controller's TX chain.

bce%d: Could not map TX descriptor chain DMA memory! The driver could not map the TX descriptor chain memory into the controller's DMA address space.

bce%d: Could not allocate TX mbuf DMA tag! The driver could not allocate a DMA tag for the controller's TX mbuf memory.

bce%d: Unable to create TX mbuf DMA map! The driver could not map the TX mbuf memory into the controller's DMA address space.

bce%d: Could not allocate RX descriptor chain DMA tag! The driver could not allocate a DMA tag for the controller's RX chain.

bce%d: Could not allocate RX descriptor chain The driver could not allocate DMA addressable memory for the controller's RX chain.

bce%d: Could not map RX descriptor chain DMA memory! The driver could not map the RX descriptor chain memory into the controller's DMA address space.

bce%d: Could not allocate RX mbuf DMA tag! The driver could not allocate a DMA tag for the controller's RX mbuf memory.

bce%d: Unable to create RX mbuf DMA map! The driver could not map the RX mbuf memory into the controller's DMA address space.

bce%d: Firmware synchronization timeout! The driver was not able to synchronize with the firmware running on the controller. The firmware may be stopped or hung.

bce%d: Invalid Ethernet address! The driver was not able to read a valid Ethernet MAC address from NVRAM.

bce%d: Reset failed! The driver has encountered a fatal initialization error.

bce%d: Byte swap is incorrect! The driver has encountered a fatal initialization error. Contact the author with details of the CPU architecture and system chipset in use.

bce%d: Firmware did not complete initialization! The driver has encountered a fatal initialization error.

bce%d: Bootcode not running! The driver has encountered a fatal initialization error.

bce%d: Error mapping mbuf into RX chain! The driver could not map a RX mbuf into DMA addressable memory.

bce%d: Error filling RX chain: rx_bd[0x%04X]! The driver was unable to allocate enough mbufs to fill the RX chain during initialization. Try increasing the number of mbufs available in the system, increase system memory, or if using jumbo frames, make sure enough 9KB mbufs are available.

bce%d: Failed to allocate new mbuf, incoming frame dropped! The driver was unable to allocate a new mbuf for the RX chain and reused the mbuf for the received frame, dropping the incoming frame in the process. Try increasing the number of mbufs available in the system or increase system memory.

bce%d: Controller reset failed! A fatal initialization error has occurred.

bce%d: Controller initialization failed! A fatal initialization error has occurred.

bce%d: Block initialization failed! A fatal initialization error has occurred.

bce%d: Error mapping mbuf into TX chain! The driver could not map a TX mbuf into DMA addressable memory.

bce%d: Error registering poll function! The driver received an error while attempting to register the poll function.

bce%d: Changing VLAN_MTU not supported. Changing the VLAN MTU is not currently supported by the driver.

bce%d: Cannot change VLAN_HWTAGGING while management firmware (ASF/IPMI/UMP) is running! Management firmware to support ASF/IPMI/UMP requires that VLAN tag stripping be enabled in the controller.

bce%d: Changing VLAN_HWTAGGING not supported! Disabling VLAN tag stripping is not currently supported by the driver.

bce%d: Watchdog timeout occurred, resetting! The device has stopped responding to the network, there is a problem with the cable connection, or a driver logic problem has occurred..

bce%d: Fatal attention detected: 0x%08X! A controller hardware failure has occurred. If the problem continues replace the controller.

SUPPORT

For support questions please contact your QLogic approved reseller or QLogic Technical Support at <http://support.qlogic.com>, or by E-mail at <support@qlogic.com>.

SEE ALSO

altq(4), arp(4), miibus(4), netintro(4), ng_ether(4), vlan(4), ifconfig(8)

HISTORY

The **bce** device driver first appeared in FreeBSD 6.1.

AUTHORS

The **bce** driver was written by David Christensen <davidch@broadcom.com>.