NAME

bnxt - Broadcom NetXtreme-C/NetXtreme-E Family Ethernet driver

SYNOPSIS

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

```
device iflib
device bnxt
```

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

```
if_bnxt_load="YES"
```

DESCRIPTION

The **bnxt** driver provides support for various NICs based on the Broadcom BCM57301/2/4, and BCM57402/4/6 Ethernet controller chips.

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

HARDWARE

The **bnxt** driver provides support for various NICs based on the Broadcom NetXtreme-C and NetXtreme-E families of Gigabit Ethernet controller chips, including the following:

- Broadcom BCM57301 NetXtreme-C 10Gb Ethernet Controller
- Broadcom BCM57302 NetXtreme-C 10Gb/25Gb Ethernet Controller
- Broadcom BCM57304 NetXtreme-C 10Gb/25Gb/40Gb/50Gb Ethernet Controller
- Broadcom BCM57304 NetXtreme-C Ethernet Virtual Function
- Broadcom BCM57314 NetXtreme-C Ethernet Virtual Function
- Broadcom BCM57402 NetXtreme-E 10Gb Ethernet Controller
- Broadcom BCM57402 NetXtreme-E Ethernet Partition
- Broadcom BCM57404 NetXtreme-E 10Gb/25Gb Ethernet Controller
- Broadcom BCM57404 NetXtreme-E Ethernet Virtual Function
- Broadcom BCM57404 NetXtreme-E Partition
- Broadcom BCM57406 NetXtreme-E 10GBASE-T Ethernet Controller
- Broadcom BCM57406 NetXtreme-E Partition
- Broadcom BCM57407 NetXtreme-E 10GBase-T Ethernet Controller
- Broadcom BCM57407 NetXtreme-E 25Gb Ethernet Controller
- Broadcom BCM57407 NetXtreme-E Partition
- Broadcom BCM57412 NetXtreme-E Partition
- Broadcom BCM57414 NetXtreme-E Ethernet Virtual Function

- Broadcom BCM57414 NetXtreme-E Partition
- Broadcom BCM57416 NetXtreme-E Partition
- Broadcom BCM57417 NetXtreme-E Ethernet Partition
- Broadcom BCM57454 NetXtreme-E 10Gb/25Gb/40Gb/50Gb/100Gb Ethernet

SYSCTL VARIABLES

These variables must be set before loading the driver, either via loader.conf(5) or through the use of kenv(1). These are provided by the iflib(4) framework, and might be better documented there.

dev.bnxt.X.iflib.override_nrxds

Override the number of RX descriptors for each queue. The value is a comma separated list of three positive integers: the size of the completion ring, the size of the receive ring, and the size of the aggregation ring respectively. The completion ring should be at least the size of the aggregation ring plus four times the size of the receive ring. These numbers must be powers of two, and zero means to use the default. Defaults to 0,0,0.

dev.bnxt.X.iflib.override_ntxds

Override the number of TX descriptors for each queue. The value is a comma separated list of two positive integers: the size of the completion ring, and the size of the transmit ring respectively. The completion ring should be at least twice the size of the transmit ring. These numbers must be powers of two, and zero means to use the default. Defaults to 0,0.

dev.bnxt.X.iflib.override_qs_enable

When set, allows the number of transmit and receive queues to be different. If not set, the lower of the number of TX or RX queues will be used for both.

dev.bnxt.X.iflib.override_nrxqs

Set the number of RX queues. If zero, the number of RX queues is derived from the number of cores on the socket connected to the controller. Defaults to 0.

dev.bnxt.X.iflib.override_ntxqs

Set the number of TX queues. If zero, the number of TX queues is derived from the number of cores on the socket connected to the controller.

These sysctl(8) variables can be changed at any time:

dev.bnxt.X.vlan_only

Require that incoming frames must have a VLAN tag on them that matches one that is configured for the NIC. Normally, both frames that have a matching VLAN tag and frames that have no VLAN tag are accepted. Defaults to 0.

dev.bnxt.X.vlan_strip
When non-zero the NIC strips VLAN tags on receive. Defaults to 0.

dev.bnxt.X.rx_stall
Enable buffering rather than dropping frames when there are no available host RX buffers for DMA. Defaults to 0.

dev.bnxt.X.rss_type
Comma-separated list of RSS hash types to support. Default is all types. Defaults to ipv4,tcp_ipv4,udp_ipv4,ipv6,tcp_ipv6,udp_ipv6.

dev.bnxt.X.rss_key
Current RSS key. Defaults to a randomly generated value which is generated for each device during attach.

dev.bnxt.X.ver.hwrm_min_ver
Minimum HWRM (HardWare Resource Manager) firmware API to support. If the firmware implements an older version, a warning will be printed, and the firmware should be upgraded. Defaults to 1.2.2.

These sysctl(8) variables are read-only:

dev.bnxt.X.if_name
Current interface name of the device. This will normally be *bnxtX*, but this can be changed using **ifconfig name**. This sysctl allows correlating an interface with a child of *dev.bnxt*.

*dev.bnxt.X.nvram.**
Information about the NVRAM device which contains the device firmware.

*dev.bnxt.X.ver.**
Version-related information about the device and firmware:

dev.bnxt.X.ver.hwrm_if
Supported HWRM API version of the currently running firmware.

dev.bnxt.X.ver.driver_hwrm_if
HWRM API version the driver was built to support.

*dev.bnxt.X.hwstats.**
Per-queue statistics tracked by the hardware.

*dev.bnxt.X.hwstats.port_stats.**
Per-port statistics tracked by the hardware.

dev.bnxt.X.hwstats.rxq0.drop_pkts
Number of packets dropped by hardware on queue zero. This number might seem high, but the count includes packets dropped due to incorrect destination MAC, unsubscribed multicast address, and other normal reasons to ignore Ethernet frames.

*dev.bnxt.X.hwstats.rxq0.tpa_**
statistics related to HW LRO.

*dev.bnxt.X.hw_lro.**
Enable / Disable HW LRO feature. Defaults to disable. Enabling HW LRO could cause issues when forwarding is enabled on host.

dev.bnxt.X.fc
Enable / Disable Flow Control feature. Defaults to Enable

DIAGNOSTICS

bnxt%d: %s command returned %s error. Device firmware rejected a command from the driver. There might be a driver/firmware HWRM API mismatch.

bnxt%d: Timeout sending %s (timeout: %d) seq %d Device firmware unresponsive. A PCI device reset is likely needed.

bnxt%d: Timeout sending %s (timeout: %d) msg {0x%x 0x%x} len:%d v: %d Partial firmware response. A PCI device reset is likely needed.

As of this writing, the system must be rebooted to initiate a PCI device reset.

SEE ALSO

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

HISTORY

The **bnxt** device driver first appeared in FreeBSD 11.1.

AUTHORS

The **bnxt** driver was written by Jack Vogel <*jfvogel@gmail.com*> and Stephen Hurd <*shurd@freebsd.org*>, and is currently maintained by Broadcom Limited <*freebsd.pdl@broadcom.com*>.