

**NAME**

**cxgbev** - Chelsio T4-, T5-, and T6-based 100Gb, 40Gb, 25Gb, 10Gb, and 1Gb Ethernet VF driver

**SYNOPSIS**

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

```
device cxgbe
device cxgbev
```

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

```
if_cxgbev_load="YES"
```

**DESCRIPTION**

The **cxgbev** driver provides support for Virtual Functions on PCI Express Ethernet adapters based on the Chelsio Terminator 4, Terminator 5, and Terminator 6 ASICs (T4, T5, and T6). The driver supports Jumbo Frames, Transmit/Receive checksum offload, TCP segmentation offload (TSO), Large Receive Offload (LRO), VLAN tag insertion/extraction, VLAN checksum offload, VLAN TSO, and Receive Side Steering (RSS). For further hardware information and questions related to hardware requirements, see <http://www.chelsio.com/>.

The **cxgbev** driver uses different names for devices based on the associated ASIC:

| <b>ASIC</b> | <b>Port Name</b> | <b>Parent Device</b> |
|-------------|------------------|----------------------|
| T4          | cxgbev           | t4vf                 |
| T5          | cxlv             | t5vf                 |
| T6          | ccv              | t6vf                 |

Loader tunables with the `hw.cxgbe` prefix apply to VFs from all cards. The Physical Function driver for Chelsio Terminator adapters shares these tunables. The driver provides sysctl MIBs for both ports and parent devices using the names above. For example, a T5 VF provides port MIBs under `dev.cxlv` and parent device MIBs under `dev.t5vf`. References to sysctl MIBs in the remainder of this page use `dev.<port>` for port MIBs and `dev.<nexus>` for parent device MIBs.

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

**HARDWARE**

The **cxgbev** driver supports Virtual Functions on 100Gb and 25Gb Ethernet adapters based on the T6 ASIC:

- ⊕ Chelsio T6225-CR
- ⊕ Chelsio T6225-SO-CR
- ⊕ Chelsio T62100-LP-CR
- ⊕ Chelsio T62100-SO-CR
- ⊕ Chelsio T62100-CR

The **cxgbev** driver supports Virtual Functions on 40Gb, 10Gb and 1Gb Ethernet adapters based on the T5 ASIC:

- ⊕ Chelsio T580-CR
- ⊕ Chelsio T580-LP-CR
- ⊕ Chelsio T580-LP-SO-CR
- ⊕ Chelsio T560-CR
- ⊕ Chelsio T540-CR
- ⊕ Chelsio T540-LP-CR
- ⊕ Chelsio T522-CR
- ⊕ Chelsio T520-LL-CR
- ⊕ Chelsio T520-CR
- ⊕ Chelsio T520-SO
- ⊕ Chelsio T520-BT
- ⊕ Chelsio T504-BT

The **cxgbev** driver supports Virtual Functions on 10Gb and 1Gb Ethernet adapters based on the T4 ASIC:

- ⊕ Chelsio T420-CR
- ⊕ Chelsio T422-CR
- ⊕ Chelsio T440-CR
- ⊕ Chelsio T420-BCH
- ⊕ Chelsio T440-BCH
- ⊕ Chelsio T440-CH
- ⊕ Chelsio T420-SO
- ⊕ Chelsio T420-CX
- ⊕ Chelsio T420-BT
- ⊕ Chelsio T404-BT

## LOADER TUNABLES

Tunables can be set at the loader(8) prompt before booting the kernel or stored in loader.conf(5).

*hw.cxgbe.nxq*

Number of tx queues used for a port. The default is 16 or the number of CPU cores in the system, whichever is less.

*hw.cxgbe.nrxq*

Number of rx queues used for a port. The default is 8 or the number of CPU cores in the system, whichever is less.

*hw.cxgbe.holdoff\_timer\_idx*

Timer index value used to delay interrupts. The holdoff timer list has the values 1, 5, 10, 50, 100, and 200 by default (all values are in microseconds) and the index selects a value from this list. The default value is 1 which means the timer value is 5us. Different interfaces can be assigned different values at any time via the `dev.<port>.X.holdoff_tmr_idx sysctl`.

*hw.cxgbe.holdoff\_pktc\_idx*

Packet-count index value used to delay interrupts. The packet-count list has the values 1, 8, 16, and 32 by default, and the index selects a value from this list. The default value is -1 which means packet counting is disabled and interrupts are generated based solely on the holdoff timer value. Different interfaces can be assigned different values via the `dev.<port>.X.holdoff_pktc_idx sysctl`. This sysctl works only when the interface has never been marked up (as done by `ifconfig up`).

*hw.cxgbe.qsize\_txq*

Number of entries in a transmit queue's descriptor ring. A `buf_ring` of the same size is also allocated for additional software queuing. See `ifnet(9)`. The default value is 1024. Different interfaces can be assigned different values via the `dev.<port>.X.qsize_txq sysctl`. This sysctl works only when the interface has never been marked up (as done by `ifconfig up`).

*hw.cxgbe.qsize\_rxq*

Number of entries in a receive queue's descriptor ring. The default value is 1024. Different interfaces can be assigned different values via the `dev.<port>.X.qsize_rxq sysctl`. This sysctl works only when the interface has never been marked up (as done by `ifconfig up`).

*hw.cxgbe.interrupt\_types*

Permitted interrupt types. Bit 0 represents INTx (line interrupts), bit 1 MSI, and bit 2 MSI-X. The default is 7 (all allowed). The driver selects the best possible type out of the allowed types. Note that Virtual Functions do not support INTx interrupts and fail to attach if neither MSI nor MSI-X are enabled.

*hw.cxgbe.fl\_pktshift*

Number of padding bytes inserted before the beginning of an Ethernet frame in the receive

buffer. The default value of 2 ensures that the Ethernet payload (usually the IP header) is at a 4 byte aligned address. 0-7 are all valid values.

*hw.cxgbe.fl\_pad*

A non-zero value ensures that writes from the hardware to a receive buffer are padded up to the specified boundary. The default is -1 which lets the driver pick a pad boundary. 0 disables trailer padding completely.

*hw.cxgbe.buffer\_packing*

Allow the hardware to deliver multiple frames in the same receive buffer opportunistically. The default is -1 which lets the driver decide. 0 or 1 explicitly disable or enable this feature.

*hw.cxgbe.allow\_mbufs\_in\_cluster*

1 allows the driver to lay down one or more mbufs within the receive buffer opportunistically. This is the default. 0 prohibits the driver from doing so.

*hw.cxgbe.largest\_rx\_cluster*

*hw.cxgbe.safest\_rx\_cluster*

Sizes of rx clusters. Each of these must be set to one of the sizes available (usually 2048, 4096, 9216, and 16384) and *largest\_rx\_cluster* must be greater than or equal to *safest\_rx\_cluster*. The defaults are 16384 and 4096 respectively. The driver never attempts to allocate a receive buffer larger than *largest\_rx\_cluster* and falls back to allocating buffers of *safest\_rx\_cluster* size if an allocation larger than *safest\_rx\_cluster* fails. Note that *largest\_rx\_cluster* merely establishes a ceiling -- the driver is allowed to allocate buffers of smaller sizes.

Certain settings and resources for Virtual Functions are dictated by the parent Physical Function driver. For example, the Physical Function driver limits the number of queues available to a Virtual Function. Some of these limits can be adjusted in the firmware configuration file used with the Physical Function driver.

The PAUSE settings on the port of a Virtual Function are inherited from the settings of the same port on the Physical Function. Virtual Functions cannot modify the setting and track changes made to the associated port's setting by the Physical Function driver.

Receive queues on a Virtual Function always drop packets in response to congestion (equivalent to setting *hw.cxgbe.cong\_drop* to 1).

The VF driver currently depends on the PF driver. As a result, loading the VF driver also loads the PF driver as a dependency.

**SUPPORT**

For general information and support, go to the Chelsio support website at: <http://www.chelsio.com/>.

If an issue is identified with this driver with a supported adapter, email all the specific information related to the issue to [<support@chelsio.com>](mailto:support@chelsio.com).

**SEE ALSO**

[arp\(4\)](#), [cxgbe\(4\)](#), [netintro\(4\)](#), [ng\\_ether\(4\)](#), [ifconfig\(8\)](#)

**HISTORY**

The **cxgbev** device driver first appeared in FreeBSD 11.1 and FreeBSD 11.1.

**AUTHORS**

The **cxgbev** driver was written by Navdeep Parhar [<np@FreeBSD.org>](mailto:np@FreeBSD.org) and John Baldwin [<jhb@FreeBSD.org>](mailto:jhb@FreeBSD.org).