

**NAME**

**alc** - Atheros AR813x/AR815x/AR816x/AR817x Gigabit/Fast Ethernet driver

**SYNOPSIS**

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

```
device miibus  
device alc
```

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

```
if_alc_load="YES"
```

**DESCRIPTION**

The **alc** device driver provides support for Atheros AR813x, AR815x, AR816x and AR817x PCI Express Gigabit/Fast Ethernet controllers.

All LOMs supported by the **alc** driver have TCP/UDP/IP checksum offload for transmit, TCP segmentation offload (TSO), hardware VLAN tag stripping/insertion features, Wake On Lan (WOL) and an interrupt moderation mechanism as well as a 64-bit multicast hash filter.

The AR813x, AR815x, AR816x and AR817x supports Jumbo Frames (up to 9216, 6144, 9216 and 9216 bytes, respectively), which can be configured via the interface MTU setting. Selecting an MTU larger than 1500 bytes with the ifconfig(8) utility configures the adapter to receive and transmit Jumbo Frames.

The **alc** 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.

**100baseTX** Set 100Mbps (Fast Ethernet) operation.

**1000baseTX** Set 1000baseTX operation over twisted pair.

The **alc** 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 `alc` device driver provides support for the following Ethernet controllers:

- ⊕ Atheros AR8131 PCI Express Gigabit Ethernet controller
- ⊕ Atheros AR8132 PCI Express Fast Ethernet controller
- ⊕ Atheros AR8151 v1.0 PCI Express Gigabit Ethernet controller
- ⊕ Atheros AR8151 v2.0 PCI Express Gigabit Ethernet controller
- ⊕ Atheros AR8152 v1.1 PCI Express Fast Ethernet controller
- ⊕ Atheros AR8152 v2.0 PCI Express Fast Ethernet controller
- ⊕ Atheros AR8161 PCI Express Gigabit Ethernet controller
- ⊕ Atheros AR8162 PCI Express Fast Ethernet controller
- ⊕ Atheros AR8171 PCI Express Gigabit Ethernet controller
- ⊕ Atheros AR8172 PCI Express Fast Ethernet controller
- ⊕ Killer E2200 Gigabit Ethernet controller
- ⊕ Killer E2400 Gigabit Ethernet controller
- ⊕ Killer E2500 Gigabit Ethernet controller

**LOADER TUNABLES**

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

*hw.alc.msi\_disable*

This tunable disables MSI support on the Ethernet hardware. The default value is 0.

*hw.alc.msix\_disable*

This tunable disables MSI-X support on the Ethernet hardware. The default value is 2, which means to enable or disable MSI-X based on the card type; for "Killer" cards (E2x00) MSI-X will be disabled, while on other cards it will be enabled. Set this to 0 to force MSI-X to be enabled, or 1 to force it to be disabled regardless of card type.

**SYSCTL VARIABLES**

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

*dev.alc.%d.int\_rx\_mod*

Maximum amount of time to delay receive interrupt processing in units of 1us. The accepted range is 0 to 130000, the default is 100(100us). Value 0 completely disables the interrupt

moderation.

*dev.alc.%d.int\_tx\_mod*

Maximum amount of time to delay transmit interrupt processing in units of 1us. The accepted range is 0 to 130000, the default is 1000(1ms). Value 0 completely disables the interrupt moderation.

*dev.alc.%d.process\_limit*

Maximum amount of Rx frames to be processed in the event loop before rescheduling a taskqueue. The accepted range is 32 to 255, the default value is 64 events. The interface does not need to be brought down and up again before a change takes effect.

### SEE ALSO

altq(4), arp(4), miibus(4), netintro(4), ng\_ether(4), vlan(4), ifconfig(8)

### HISTORY

The **alc** driver was written by Pyun YongHyeon <yongari@FreeBSD.org>. It first appeared in FreeBSD 8.0.