**NAME**

    **axp** - Advanced Micro Devices 10G Ethernet driver

**SYNOPSIS**

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

        **device iflib**
        **device axp**

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

        if_axp_load="YES"

**DESCRIPTION**

    The **axp** driver enables PCI-E based 10G Ethernet controller inbuilt in the AMD EPYC processors.

    The following features are supported.

- 1G/10G SFP+ Link
- Jumbo frames (9000 Bytes)
- Transmit and Receive checksum offload
- TCP segmentation offload (TSO)
- VLAN tag insertion/extraction
- VLAN checksum offload
- VLAN TSO
- Receive side steering (RSS)
- IPV4 and IPV6 capable
- MSI-X interrupts
- Split header

    All the above mentioned features are enabled by default.

    For hardware related questions, please refer the documentation supplied along with AMD EPYC processors.

**SYSCTL VARIABLES**

    The following variables are available as sysctl(8) variables:

    *dev.ax.X.mac_stats*

        Dumps the transmit and receive statistics counter values for the controller.  This includes

statistics specific to each transmit and receive queue.

*dev.ax.X.channels_info*
> Dumps the permissible and default configured transmit and receive channel information.

*dev.ax.X.ringparam_info*
> Dumps the permissible and default configured descriptor information for the transmit and receive queue.

*dev.ax.X.link_ksettings_info*
> Dumps the current link setting like link mode, speed, duplex settings.

*dev.ax.X.pauseparam_info*
> Dumps the current flow-control settings.

*dev.ax.X.coalesce_info*
> Dumps the current interrupt coalescing settings.

*dev.ax.X.link_info*
> Dumps the current state of the Link.

*dev.ax.X.drv_info*
> Dumps the driver and controller firmware version information.

*dev.ax.X.YYYY_register*

*dev.ax.X.YYYY_register_values*
> Sysctl to dump a specific register from a specific block of the controller.  YYYY specifies the block.  The following blocks are supported.
> - xpcs
> - xgmac
> - xprop
> - xi2c
>
> Set the offset of the register to the first variable, and then read the value of the register by reading the second variable.

*dev.ax.X.axgbe_debug_level*
> Configure the log-level for the driver.  Default is 0.  Supports 0-3.

*dev.ax.X.link_workaround*

> This variable enables the workaround for an intermittent link issue.  When link does not come up for long time, this variable can be set to 1 to reset the phy and bring up the link.

## LOADER TUNABLES

The following variable is available as loader.conf(5) tunable.

*dev.ax.X.sph_enable*

> This variable controls split header feature for the interface.  Default is 1, meaning the split header support is enabled.

> This variable must be set before loading the driver, either via loader.conf(5) or through kenv(1). This cannot be modified when driver is loaded.

> Setting this variable in loader.conf(5) needs the system to be restarted to take effect.  When using kenv(1), use the wrapper variable *dev.ax.sph_enable*, which will configure(enable/disable) split header support for all **axp** interfaces.

> To use netmap with this device, split header support must be disabled (set this variable to 0).

## SEE ALSO

arp(4), iflib(4), netmap(4), vlan(4), ifconfig(8)

## HISTORY

The **axp** device driver first appeared in FreeBSD 13.0.

Another version of the driver is already present in FreeBSD.  This driver was named as "axgbe" earlier, which is renamed as "axa" now.  This driver is for the ACPI based Ethernet controllers in the previous/older version of the hardware.  This driver is authored by *<andrew@FreeBSD.org>*.

## AUTHORS

The **axp** device driver was written by Advanced Micro Devices Inc.

For any issues and support requirements, email the details to *<rajesh1.kumar@amd.com>*.