

VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS Alpha and IA-64

Operating System and Version: VSI OpenVMS Alpha Version 8.4-2L1 or higher
VSI OpenVMS IA-64 Version 8.4-1H1 or higher

VSI LAN Devices, Counters, and Functions Reference Manual for OpenVMS Alpha and IA-64



VMS Software

Copyright © 2026 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

Legal Notice

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

All other trademarks and registered trademarks mentioned in this document are the property of their respective holders.

Table of Contents

Chapter 1. Introduction	1
Chapter 2. TGEC 10Mb Ethernet	3
2.1. TGEC Internal Counters	3
2.2. TGEC Device-Specific Functions	3
Chapter 3. LANCE 10Mb Ethernet	5
3.1. LANCE Internal Counters	5
3.2. LANCE Device-Specific Counters	5
Chapter 4. Tulip 10/100Mb Ethernet	7
4.1. Tulip Driver Internal Counters	7
4.2. Tulip Device-Specific Functions	14
Chapter 5. Intel 82599 10/100Mb Ethernet	17
5.1. 82599 Driver Internal Counters	17
5.2. 82599 Device-Specific Functions	24
Chapter 6. DEGPA 1Gb Ethernet	29
6.1. DEGPA Driver Internal Counters	29
6.2. DEGPA Device-Specific Functions	46
Chapter 7. Broadcom 1Gb Ethernet	49
7.1. Broadcom 5700 Driver Internal Counters	49
7.2. Broadcom 5700 Device-Specific Functions	63
Chapter 8. Intel 1Gb Ethernet	69
8.1. Intel 1Gb Internal Counters	70
8.2. Intel 1Gb Device-Specific Functions	80
Chapter 9. Neterion XFRAME 10Gb Ethernet	85
9.1. XFRAME 10Gb Internal Counters	85
9.2. XFRAME 10Gb Device-Specific Functions	96
Chapter 10. Broadcom BCM57711 10Gb Ethernet	101
10.1. Broadcom BCM57711 10Gb Internal Counters	101
10.2. Broadcom BCM57711 10Gb Device-Specific Functions	111
Chapter 11. Emulex BE3 10Gb Ethernet	115
11.1. Emulex BE3 10Gb Internal Counters	116
11.2. Emulex BE3 10Gb Device-Specific Functions	131

Chapter 1. Introduction

This document describes the driver-specific counters and functions maintained by the LAN drivers internally on VSI OpenVMS Alpha and IA-64 systems.

The LAN drivers maintain the following information that the **LAN Control Program (LANCP)** and the **LAN System Dump Analyzer (SDA)** extension can display:

- Device counters, packets, and bytes sent and received and error counters for transmit and receive.
- Driver-specific internal counters, device and driver settings, and counters maintained internal to the driver. This includes driver messages showing settings and link state changes.
- Error log information.
- Driver tracing (by default, mostly errors), state changes, and link events. Selecting which events to trace is done using LANCP commands and LAN_FLAGS system parameter settings.
- Bandwidth monitoring (x86-64 only at present). Adjusting bandwidth monitoring settings is done using LANCP commands and LAN_FLAGS settings.

The LAN drivers provide additional capabilities, mostly for diagnostic and test purposes. These capabilities are called **device-specific functions** and are initiated by LANCP commands. They can be used to reset a device, cause various errors, set promiscuous mode, and so on. The selection of functions is driver specific.

The devices covered in this document are as follows:

- TGEC 10Mb Ethernet
- LANCE 10Mb Ethernet
- Tulip 10/100Mb Ethernet
- Intel 82559 10/100Mb Ethernet
- DEGPA 1Gb Ethernet
- Broadcom 1Gb Ethernet
- Intel 1Gb Ethernet
- Neterion XFRAME 10Gb Ethernet
- Broadcom BCM57711 10Gb Ethernet
- Emulex BE3 10Gb Ethernet

Chapter 2. TGEC 10Mb Ethernet

The TGEC Ethernet devices are:

Device	Platform	Characteristics
TGEC	Alpha	10Base2 (thinwire)

The driver is SYS\$EZDRIVER.EXE (Alpha).

2.1. TGEC Internal Counters

The TGEC driver includes driver-specific counters but does not support the LANCP command to display them. Access to these counters is through SDA with the command **SHOW LAN/INTERNAL**.

2.2. TGEC Device-Specific Functions

The TGEC driver provides no device-specific functions.

Chapter 3. LANCE 10Mb Ethernet

The LANCE Ethernet devices are:

Device	Platform	Characteristics
DE422	Alpha, EISA	10BaseT (UTP), 10Base2 (thinwire)
DE200	Alpha, ISA	10Base2 (thinwire), 10Base5 (thickwire)
DE201	Alpha, ISA	10BaseT (UTP)
DE202	Alpha, ISA	10Base2 (thinwire), 10BaseT (UTP)

The drivers are `SY$ESDRIVER.EXE`, `SY$ECDRIVER.EXE`, and `SY$ERDRIVER_NE2000.EXE` (Alpha).

3.1. LANCE Internal Counters

The LANCE driver includes driver-specific counters but does not support the `LANCP` command to display them. Access to these counters is through `SDA` with the command **SHOW LAN/INTERNAL**.

3.2. LANCE Device-Specific Counters

The LANCE driver provides no device-specific functions.

Chapter 4. Tulip 10/100Mb Ethernet

The Tulip Ethernet devices are:

Device	Bus	Characteristics
DE425	EISA	10Base2 (thinwire), 10Base5 (thickwire), 10BaseT (UTP)
DE434	PCI	10BaseT (UTP)
DE435	PCI	10Base2 (thinwire), 10Base5 (thickwire), 10BaseT (UTP)
DE436	PCI	Quad DE435
DE450	PCI	10Base2 (thinwire), 10Base5 (thickwire), 10BaseT (UTP)
DE500-XA	PCI	10BaseT (UTP), 100BaseTX (UTP), auto-negotiation not supported
DE500-AA	PCI	10BaseT (UTP), 100BaseTX (UTP), auto-negotiation supported
DE500-BA	PCI	10BaseT (UTP), 100BaseTX (UTP), auto-negotiation supported
DE500-FA	PCI	100BaseFX (multimode fiber), auto-negotiation not supported
DE504-BA	PCI	Quad DE500-BA
P2SE	PCI	Combo SCSI + DE434
P2SE+	PCI	Combo SCSI + DE500-XA
21142	PCI LOM	Digital Personal Workstation, all modes depending on MAU options, auto-negotiation supported
21143	PCI LOM	Alpha Professional Workstation XP900/XP1000, all modes depending on MAU options, auto-negotiation supported
A5230A	PCI	DE500-BA equivalent
A5506B	PCI	DE504-BA equivalent

The drivers are `SYSEWDRIVER.EXE`, `SYSERDRIVER.EXE` (Alpha, IA-64).

4.1. Tulip Driver Internal Counters

The LANCP command `SHOW DEVICE/INTERNAL_COUNTERS EWA` displays the entire set of internal counters maintained by the Tulip driver. Some counters are special debug counters. These are not displayed unless the additional qualifier `/DEBUG` is specified. Counters that are zero are not displayed unless the additional qualifier `/ZERO` is specified.

The LAN SDA extension also displays the complete set of internal counters with the command `LAN INTERNAL/DEVICE=EWA`.

Counters

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the `ANALYZE/IMAGE` command output for the driver image. It includes variant information, if

applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EWA** command in the quadword driver version field.

Driver flags

Driver flags including the following bits:

- Map_ALL – All transmit and receive buffers are described using map registers.
- CSR6 – CSR6 needs to be written.
- Setup – A setup buffer needs to be transmitted.
- Hash – Hash filtering is in use.
- Stop – The Tulip chip is stopping.
- Autosense – Autosense is selected.
- Auto-neg – Auto-negotiation is selected.
- PDet – Parallel Detection was used to determine speed.
- HdxOper – Tulip chip is in half-duplex mode.
- FdxOper – Tulip chip is in full-duplex mode.
- 10mbit – Tulip chip is running at 10 Mbit/second.
- 100mbit – Tulip chip is running at 100 Mbit/second.
- SoftError – Soft error handling is in progress (a reset of the chip without telling applications, as compared to a hard error for which applications are notified that error recovery is in progress).
- LWalign – Unaligned transmit buffers are being copied to longword aligned buffers.
- AUI – AUI mode is selected.
- BNC – BNC mode is selected.
- UTP – UTP mode is selected.
- Phy_Chip – An external PHY chip has been detected.
- SMC_BNC – The BNC autosense selection has been made for this SMC NIC.
- P2SE_BNC – The BNC autosense selection has been made for this P2SE combo card.
- More_Rcv – A larger quantity of receive buffers have been enabled.
- FailoverInProgress – A failover operation is in progress.
- FdxEna – Full-duplex mode is enabled.

Device type

Device type, as follows:

- DE435_PASS1 – DE425, DE434, DE435, or DE436 NIC.

- DE435_PASS2 – DE425, DE434, DE435, or DE436 NIC.
- DE435_PASS3 – DE425, DE434, DE435, or DE436 NIC.
- DC21040 – Embedded Tulip chip (LOM) on the AlphaServer 2100 system.
- DC21041 – DE450 NIC.
- DC21140 – DE500-XA NIC.
- DC21140A – DE500-AA or P2SE+ NIC.
- DE500BA – DE500-BA NIC.
- DE500FA – DE500-FA NIC.
- MIATA – Embedded chip on the Digital Personal Workstation (PWS).
- MAU – A 10Base2 (thinwire), 10Base5 (thickwire), or 10BaseT (UTP) option has been selected (as compared to the default UTP connection on the back of the system).
- DC21142 – Some Miata chips.
- DC21143 – DE500-BA or some Miata chips.
- TSUNAMI – Embedded 21143 chip (LOM) on the XP900/1000.
- HP-SINGLE – A5230A NIC.

Link status

Current link state, as follows:

- Undefined – Link state not determined yet or not applicable to this Tulip chip.
- LinkDown – Link is down, disconnected from the network.
- LinkWait – Link is down, and we are waiting a few seconds before doing link down handling in case it comes back up.
- LinkUp – Link is up, connected to the network, ready to transmit and receive.

Link down timer

Number of seconds before transitioning from LinkWait to LinkDown.

Link transition count

Total number of link state up and link state down events.

Tulip reset count

Number of times the driver has reset the Tulip chip.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the Tulip chip, and completed outstanding I/O requests with error status.

Transmit underflows

Number of transmits that failed because the Tulip chip stopped transmitting in the middle of a packet due to running out of data (Direct Memory Access (DMA) from the transmit buffer in host memory took too long).

Transmit length errors

Number of transmits that exceeded the 1518-byte limit and were therefore truncated by the Tulip chip.

Receive overflows

Number of receives that failed because the Tulip chip could not DMA the received packet to host memory before its internal First-In First-Out (FIFO) was overrun.

Receive collisions seen

Number of receives that failed because the packet was damaged by a collision that occurred after the first 64 bytes of the packet (a late collision).

Calls to TU_START_TULIP

Number of times the Tulip chip was started.

Setup buffers issued

Number of setup buffers sent to the Tulip chip for initialization purposes.

CSR6 changes

Number of times CSR6 (Command Register) was written for initialization purposes.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid Page Table Entry (PTE) to a valid Page Frame Number (PFN).

Original CSR13 contents

Original contents of CSR13 (SIA Connectivity Register); shows how the console or boot driver had set up the Tulip chip.

Transmit copies (too many segments)

Number of buffers copied because the transmit request contained too many segments.

Transmit failures (too few segments)

Number of transmits that failed because the transmit request did not include enough data.

Longword align workarounds (one segment)

Number of transmits copied to a longword-aligned buffer to work around a Tulip chip hardware bug where there was only one transmit segment.

Loss of carrier workaround

Number of times the Tulip chip erroneously indicated that a loss of carrier error occurred on a transmit.

Transmit multiple workarounds

Number of multi-segment transmits that were done to a workaround buffer, including transmits with too many segments or multi-segment longword-aligned workarounds.

Transmit multiple addresses

Number of transmits descriptors which required multiple buffer addresses to describe. For example, a transmit segment crossed a page boundary and was not physically contiguous.

Device interrupts

Number of times the driver interrupt service routine was called.

BNC selections done

Number of times a 10Base5 to 10Base2 autosense selection (applies to the DE450 device only).

FDX selection changes

Number of duplex selection changes done (only applies if autosense mode is selected).

Inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

Transmits issued (using map registers)

Number of transmit buffers that were described to the Tulip chip using map registers because part of the request existed in memory outside the DMA window.

Receives issued (using map registers)

Number of receive buffers that were described to the Tulip chip using map registers because part of the request existed in memory outside the DMA window.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the Tulip chip without notifying user applications.

Duplex mode mismatches

Number of times a potential duplex mode mismatch was detected by checking for transmit and receive errors.

Rescheduled forks (fork time limit)

Number of times that transmit and receive processing was suspended and the fork process rescheduled to allow fair usage of the CPU by other drivers.

Auto-negotiations done

Number of auto-negotiation sequences done.

Auto-negotiation enable

Auto-negotiation enable flag (1 if enabled).

Auto-negotiation state

Current auto-negotiation state.

Auto-negotiation link state counter

Counts half-second increments during the auto-negotiation process.

Autosense state

Current autosense state.

Saved speed setting

Speed setting saved before auto-negotiation is done.

Saved duplex setting

Duplex setting saved before auto-negotiation is done.

Link partner ability

Link partner ability mask recorded the last time auto-negotiation was done.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Transmit time limit

Transmit time limit (in seconds) after which a timeout is declared.

Timer routine interval

Resolution of the transmit timer (the real timeout is limit + interval).

Autosense failure switches

Number of times remaining to do an autosense failover (starts at 1).

Time autoselection timer started

Time that the last autosense was initiated.

Current time (EXE\$GL_ABSTIM_TICS)

Current time (at the time which the counters were read).

Last duplex mode mismatch reported

Time of the last duplex mode mismatch reported.

Previous CRC or late collision error

Time of the last Cyclic Redundancy Check (CRC) or late collision error recorded.

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the Tulip chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the Tulip chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Lost Packets after Fork Delay

The Tulip chip records how many packets are lost after it runs out of receive buffers. When the driver processes each receive packet, it also reads this "missed packet" counter to identify the fork delay value resulting in the most packets being lost.

Minimum buffers requested

Minimum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current minimum limit".

Maximum buffers requested

Maximum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current maximum limit".

Current minimum limit

Minimum number of receive buffers as determined by the driver and by management request, "Minimum buffers requested". The driver will not allow the number of receive buffers in its receive queues to drop below this value. This does not include receive buffers transferred to applications (and not yet returned).

Current maximum limit

Maximum number of receive buffers as determined by the driver and by management request, "Maximum buffers requested". The driver will deallocate receive buffers until the number allocated does not exceed this maximum. When applications return receive buffers, the number may exceed this maximum, at which point they are deallocated. Note that the driver allows this maximum to be exceeded slightly to limit allocation and deallocation activity. See "[Target number of buffers maximum](#)" below.

Current number of buffers

Current number of receive buffers owned by the driver.

Target number of buffers

Desired number of receive buffers. As receive activity does not result in lost packets, this number is gradually decreased toward the "Current minimum limit". If packets are lost, this number is dramatically increased toward the "Current maximum limit". When the driver allocates receive buffers, it allocates them until reaching this target number.

Target number of buffers maximum

Maximum desired number of receive buffers. When an application returns a buffer to the driver, if the current number does not exceed this target maximum, the buffer is kept by the driver. Otherwise, it is deallocated.

Driver Messages

Console messages issued by the driver.

4.2. Tulip Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, . . .))** LANCP command provides a mechanism to issue device-specific functions to the Tulip driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A-L* [<https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590>] or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the Tulip chip using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The Tulip chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(32bits,16bits))**DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(32bits,16bits))****DEVICE_SPECIFIC=(FUNCTION="FIL3",VALUE=(32bits,16bits))****DEVICE_SPECIFIC=(FUNCTION="FIL4",VALUE=(32bits,16bits))**

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the Tulip chip, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="FXMT",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is limited to the range 2–40. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

Chapter 5. Intel 82599 10/100Mb Ethernet

The 82559 Ethernet devices are:

Device	Characteristics
DE600-AA	10BaseT (UTP), 100BaseTX (UTP)
DE602-AA	Dual DE600-AA
DE602-BA	Dual DE600-AA
DE602-BB	Dual DE600-AA
DE602-TA	Dual DE600-AA daughter card for DE602
DE602-FA	Dual 100BaseFX (multimode fiber) daughter card for DE602
Trifecta	Combo SCSI + DE600
82559ER	LOM 10BaseT (UTP), 100BaseTX (UTP)
82559	LOM 10BaseT (UTP), 100BaseTX (UTP)

The driver is SYS\$EIDRIVER.EXE (Alpha, IA-64).

5.1. 82559 Driver Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS EIA** displays the entire set of internal counters maintained by the 82559 driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EIA**.

Counters

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EIA** command in the quadword driver version field.

Driver flags

Driver flags including the following bits:

- Auto – Auto-negotiation is enabled.
- PDet – Parallel Detection was used to select speed.
- 10mbit – 82559 chip is running at 10 Mbit/second.

- 100mbit – 82559 chip is running at 100 Mbit/second.
- Hdx – 82559 chip is in half-duplex mode.
- Fdx – 82559 chip is in full-duplex mode.
- Flow – Flow control is enabled for auto-negotiation.
- Prm – 82559 chip is in promiscuous mode.
- AMC – 82559 chip is in all-multicast mode.
- Map_All – All transmit and receive buffers are described using map registers.
- CSR_IO – 82559 Command Status Registers (CSRs) are mapped in I/O space.
- CSR_MEM – 82559 CSRs are mapped in memory space.
- 100TX – The media type is UTP.
- 100FX – The media type is multimode fiber.
- CFG_Pend – A configuration buffer request is pending.
- RFlow – The link partner supports flow control.
- TXFlow – Transmit flow control is enabled.
- RXFlow – Receive flow control is enabled.

Device ID (subvendor ID <31:16>)

Device type, as follows:

- NC3121 (Curveball)
- NC3123 (Hascal)
- NC3131 (Tornado)
- NC3132 (Thunder)
- NC3133 (El Nino)
- NC3134
- NC3135
- Embedded 82559ER
- Unknown
- NC3120
- NC3122 Dual
- Embedded 82558 (HP Integrity rx2600 Server)

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the 82559 chip without notifying user applications.

Hard errors

Number of errors that resulted in notifying user applications.

Duplex mode mismatches

Number of times a potential duplex mode mismatch was detected by checking for transmit and receive errors.

Link state

Current link state, as follows:

- Undefined – Link state not determined yet or not applicable to this 82559 chip.
- LinkDown – Link is down, disconnected from the network.
- LinkWait – Link is down, and we are waiting a few seconds before doing link down handling in case it comes back up.
- LinkUp – Link is up, connected to the network, ready to transmit and receive.

Link transitions

Total number of link state up and link state down events.

Auto-negotiation resets

Number of times auto-negotiation was restarted.

Auto-negotiation state

Current auto-negotiation state.

Rescheduled forks (fork time limit)

Number of times transmit and receive processing was suspended and the fork process rescheduled to allow fair usage of the CPU by other drivers.

Device interrupts

Number of times the driver interrupt service routine was called.

Device resets done

Number of times the driver has reset the 82559 chip.

CXB transmits

Number of transmits issued via the Queue I/O (QIO) interface.

Inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

Receive buffers mapped

Number of receive buffers that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Transmit segments mapped

Number of transmit buffer segments that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Page boundary transmits

Number of transmit buffer segments that crossed a page boundary.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

PTE to PFN translations (second page)

Number of PTE to PFN translations where the segment crossed a page boundary.

Chained VCRP transmits

Number of transmit requests consisting of more than one buffer segment.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Transmit zero length DCBEs (Data Chain Block Extensions)

Number of chained transmit request buffer segments that contained no data.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the 82559 chip, and completed outstanding I/O requests with error status.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Control commands done

Number of control commands issued to the 82559 chip (for example, configuration and address filtering commands).

Receive lockups

Number of times the driver issued a start receiver command to the 82559 chip and it did not start after 100 milliseconds, resulting in the need to reset and restart the chip.

Last duplex mode mismatch reported

Time of the last duplex mode mismatch reported.

Previous CRC or late collision error

Time of the last CRC or late collision error recorded.

Transmit time limit

Transmit time limit (in seconds) after which a timeout is declared.

Timer routine interval

Resolution of the transmit timer (the real timeout is limit + interval).

Good transmit frames

Number of packets transmitted successfully, tallied by the 82559 chip.

Transmit maximum collisions

Number of transmits that failed due to excessive collisions, tallied by the 82559 chip. These occur in half-duplex mode only.

Transmit late collisions

Number of transmits that failed because a collision was detected later than 512 bit-times into the transmission of the packet, tallied by the 82559 chip. These occur in half-duplex mode only.

Transmit underruns

Number of transmits that failed because the 82559 chip could not DMA data from the transmit buffers fast enough, tallied by the 82559 chip.

Transmit lost carrier

Number of transmits that failed because the 82559 chip detected loss of carrier during the transmission, tallied by the 82559 chip.

Transmit deferred

Number of successful transmits that were delayed because the medium was busy, tallied by the 82559 chip. These occur in half-duplex mode only.

Transmit single collisions

Number of successfully transmitted packets that encountered exactly one collision during transmission (successful after retransmit), tallied by the 82559 chip. These occur in half-duplex mode only.

Transmit multiple collisions

Number of successfully transmitted packets that encountered more than one collision during transmission (successful after multiple retransmits), tallied by the 82559 chip. These occur in half-duplex mode only.

Transmit total collisions

Total number of collisions detected on transmit, tallied by the 82559 chip.

Transmit flow control pause

Number of flow control pause frames transmitted by the 82559 chip, tallied by the 82559 chip.

Good receive frames

Number of receive packets delivered to the LAN driver successfully, tallied by the 82559 chip.

Receive crc errors

Number of packets received with CRC errors that are an integral number of bytes long, tallied by the 82559 chip. These packets are discarded.

Receive alignment errors

Number of packets received with CRC errors that are not an integral number of bytes long, tallied by the 82559 chip. These packets are discarded.

Receive resource errors

Number of receive packets discarded due to lack of receive buffers, tallied by the 82559 chip.

Receive overrun errors

Number of receive packets discarded because the internal 82559 receive FIFO overflowed (for example, the 82559 chip was not able to DMA receive data to host memory fast enough), tallied by the 82559 chip.

Receive collision detect errors

Number of receive packets discarded because a collision occurred during reception, tallied by the 82559 chip.

Receive short frame errors

Number of receive packets that were shorter than 64 bytes (and therefore discarded), tallied by the 82559 chip.

Receive flow control pause

Number of flow control pause frames received by the 82559 chip, tallied by the 82559 chip.

Receive flow control unsupported

Number of flow control frames received with an unsupported type, tallied by the 82559 chip.

Number of RUS (Receive Unit State) state updates

Number of times the 82559 driver checked and updated the 82559 chip receive unit state.

Number of RUS idle state

Number of times the 82559 chip receive unit was in the idle state.

Number of RUS ready state

Number of times the 82559 chip receive unit was in the ready state.

Number of RUS ready state with no RBDs

Number of times the 82559 chip receive unit was in the ready state with no remaining buffer descriptors (RBDs).

Number of RUS no resources

Number of times the 82559 chip receive unit was in the no resources state discarding incoming receive packets.

Number of RUS fork instead of restart

Number of times the driver delayed an 82559 chip restart to do another receive completion check via a fork.

RX FIFO limit

Bytes remaining in the receive (RX) FIFO before the 82559 chip will request the bus. This is calculated as $128 \text{ bytes} - 8n$, such that $1 = 120 \text{ bytes}$, $2 = 112 \text{ bytes}$, and so on. The default value is 8 (64 bytes).

Diagnostic self-test status

Results of the self-test done by the 82559 chip during driver unit initialization.

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the 82559 chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the 82559 chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and 82559 chip, the driver records the time that each transmit buffer is given to the 82559 chip. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Minimum buffers requested

Minimum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current minimum limit".

Maximum buffers requested

Maximum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current maximum limit".

Current minimum limit

Minimum number of receive buffers as determined by the driver and by management request, "Minimum buffers requested". The driver will not allow the number of receive buffers in its receive queues to drop below this value. This does not include receive buffers transferred to applications (and not yet returned).

Current maximum limit

Maximum number of receive buffers as determined by the driver and by management request, "Maximum buffers requested". The driver will deallocate receive buffers until the number allocated does not exceed this maximum. When applications return receive buffers, the number may exceed this maximum, at which point they are deallocated. Note that the driver allows this maximum to be exceeded slightly to limit allocation and deallocation activity. See "[Target number of buffers maximum](#)" below.

Current number of buffers

Current number of receive buffers owned by the driver.

Target number of buffers

Desired number of receive buffers. As receive activity does not result in lost packets, this number is gradually decreased toward the "Current minimum limit". If packets are lost, this number is dramatically increased toward the "Current maximum limit". When the driver allocates receive buffers, it allocates them until reaching this target number.

Target number of buffers maximum

Maximum desired number of receive buffers. When an application returns a buffer to the driver, if the current number does not exceed this target maximum, the buffer is kept by the driver. Otherwise, it is deallocated.

Driver Messages

Console messages issued by the driver.

5.2. 82559 Device-Specific Functions

The `SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, . . .))` LANCP command provides a mechanism to issue device-specific functions to the 82559 driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the [VSI OpenVMS System Management Utilities Reference Manual, Volume I: A-L](#) [<https://>

docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590] or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the 82559 chip using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The 82559 chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FILA",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILB",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILC",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILD",VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EIA/DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EIA/DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the 82559 chip, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=(FUNCTION="FXMT",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is limited to the range 2–40. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="RFIF",VALUE=*fifovalue*)

For receive DMA, override the number of bytes in the FIFO remaining before requesting the bus, to DMA the data in the FIFO to host memory. The default is 8.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="RSTA",VALUE=*timevalue*)

Avoid giving receive buffers to the 82559 chip for a portion of each 2.55-second time period. The time period is defined as the low byte of the current time (EXE\$GL_ABSTIM_TICS) minus the starvation time value specified. EXE\$GL_ABSTIM_TICS is the current system uptime in 10-millisecond units. For example, setting a starvation value of 50 avoids giving buffers to the 82559 chip for the first 500 milliseconds of each 2.55-second period. A value of 200 starves the 82559 chip for 2 seconds of every 2.55-second period.

This function is useful when investigating application or device behavior under receive buffer starvation conditions.

DEVICE_SPECIFIC=(FUNCTION="XSTA",VALUE=*timevalue*)

Delay completion of the next transmit request for the time specified in 10-millisecond units. Since transmits are completed in order, this delays all transmit completion until after the delay. When the delay is completed, the starvation time value is cleared. If the value is set too high, another device-specific function can be issued to change it.

This function is useful for investigating application behavior when transmit completion is delayed.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=FUNCTION="AUTO"

Restarts auto-negotiation.

DEVICE_SPECIFIC=(FUNCTION="MWIN",VALUE=*flagvalue*)

Set the PCI memory write and invalidate feature of the 82559 chip. Only the low bit of the value is used; enable (1) or disable (0) this function.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="RALI",VALUE=*flagvalue*)

Set the DMA read alignment feature of the 82559 chip. Only the low bit of the value is used; enable (1) or disable (0) this function. If enabled, the 82559 chip attempts to align read DMA on cache block boundaries.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="WALI",VALUE=*flagvalue*)

Set the DMA write alignment feature of the 82559 chip. Only the low bit of the value is used; enable (1) or disable (0) this function. If enabled, the 82559 chip attempts to align write DMA on cache block boundaries.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="OMSG"

Console messages are displayed by the driver during initialization, auto-negotiation, and link handling. However, the driver limits the total number of messages displayed. This function resets the count, allowing additional messages to be displayed.

DEVICE_SPECIFIC=FUNCTION="RMDI"

Read Medium Dependent Interface (MDI) registers. The registers are displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="RCOR",VALUE=*pty*)**DEVICE_SPECIFIC=(FUNCTION="XCOR",VALUE=*pty*)**

Corrupt a receive or transmit packet. When this function is completed, the protocol type is specified, and if it matches on a transmit or receive packet, the next packet seen is corrupted by changing the last bit in the VMS Communications Request Packet (VCRP) or Complex Chained Buffer (CXB), XORing it with a 1. The protocol type is then cleared so the corruption is not done again. The corruption is only checked when transmit or receive packet data tracing is being done (in the trace routine), so there is no performance hit for checking each packet to verify if it should be corrupted.

The following example shows how to corrupt a PEDRIVER transmit packet:

```
$ MCR LANCPC SET DEVICE/TRACE EIA
$ MCR LANCPC SET DEVICE/DEVICE_SPECIFIC=(FUNCTION="XCOR", VALUE=%x760) EIA
```

DEVICE_SPECIFIC=(FUNCTION="LOOP",VALUE=*loopvalue*)

Specify *loopvalue* to enable (1) or disable (0) loopback mode.

This is useful in test scenarios to run an application in loopback mode, where for standard use the application would not allow selection of loopback mode.

The current setting is displayed in the driver internal counters.

Chapter 6. DEGPA 1Gb Ethernet

The DEGPA Ethernet devices are:

Device	Characteristics
DEGPA-SA	1000BaseSX (multimode fiber)
DEGPA-TA	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)

The driver is SYS\$EW1000A.EXE (Alpha, IA-64).

6.1. DEGPA Driver Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS** displays the entire set of internal counters maintained by the DEGPA driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EWA**.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the DEGPA device.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EWA** command in the quadword driver version field.

Firmware version

Version number in hexadecimal, read from right to left.

Device interrupts

Number of times the driver interrupt service routine was called.

Events completed

Number of events completed from the event ring, such as link state changes, statistics updates, and multicast address changes.

Link transitions

Number of link state up and link state down events.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O requests with error status.

Initialization timeouts

Number of times the driver has timed out an initialization request, reset the device, and completed outstanding I/O requests with error status.

Resets issued

Number of times the driver has reset the device.

Inits initiated

Number of times the driver initiated the device initialization procedure (done at the end of unit initialization and after every device reset and reinitialization).

Inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

User change requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Jumbo transmits issued

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

Transmits issued (using map registers)

Number of transmit requests that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Jumbo receives issued

Number of jumbo receive buffers allocated and assigned to the device.

Receives issued (using map registers)

Number of receive buffers that were described to the device using map registers because part of the request existed in memory outside the DMA window, because the receive buffer crossed a page boundary, or because the receive buffer was a jumbo buffer.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Duplex mode mismatches

Number of times a potential duplex mode mismatch was detected by checking for transmit and receive errors.

Commands outstanding

Number of commands outstanding to the device. Note that this count is adjusted when a command is about to be issued and there are no free command entries (it represents commands that most likely have been completed by the device, but the driver has not freed up the completed entries for reuse yet).

Commands queued

Number of commands issued to the device when the maximum number was outstanding, so the driver had to verify completion of the outstanding commands before queuing or issuing the request.

Command pending mask

Commands that the driver has not yet issued to the device.

Invalid command

Event code of the last invalid or unimplemented command request.

Unexpected event

Event code of the last unrecognized command request.

Rescheduled forks

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

Standard receive buffers

Number of 1518-byte receive buffers owned by the device.

Standard receive buffer deallocations

Number of 1518-byte buffer deallocations done by the driver because the number of outstanding buffers exceeded the maximum number allowed by the driver.

Jumbo receive buffers (current)

Number of large receive buffers owned by the device.

Jumbo receive buffers (minimum)

Minimum number of large receive buffers owned by the device. This is initially set to 1. After the first jumbo receive, the driver sets the minimum to 32.

Jumbo receive buffer allocations

Number of jumbo receive buffer allocations done by the driver.

Jumbo receive buffer deallocations

Number of large buffer deallocations done by the driver because the number of outstanding buffers exceeded the maximum number allowed by the driver.

Standard buffer size (bytes)

Size of the standard receive buffers. The device allows packets up to 1518 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes).

Standard packet size (bytes)

This is the Ethernet packet size (1518 bytes).

Jumbo buffer size (bytes)

Size of the jumbo receive buffers. The device allows packets up to 9018 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes). However, the driver limits the buffer size to the maximum size supported by the pool lookaside lists. The buffer size is 9018 plus 640 bytes of overhead (rounded up to the next 64-byte boundary), which accommodates the full jumbo packet size.

Jumbo packet size (bytes)

This is the jumbo packet size (9018 bytes).

Requested speed

Speed requested by the driver.

Requested link value

Link control bits set by the driver to use during link initialization.

Current link state

Current link state determined by the device.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Jumbo packets

Size of jumbo frames. The LAN_FLAGS system parameter bit 6 or the LANCP command **SET DEVICE / [NO] JUMBO** determines whether the maximum user data size for VMS Communications Interface (VCI) applications is the standard size (1518 bytes, excluding header and CRC) or jumbo size. The default is disabled, LAN_FLAGS bit 6 set to zero.

Link auto negotiation

Determines whether the link state that the driver requests the device to use allows auto-negotiation. The LAN_FLAGS system parameter bit 5 or the LANCP command **SET DEVICE / [NO] AUTO** determines the setting. The default is enabled.

More receive buffers

Set if more receive buffers have been enabled, currently unused.

DMA width (bits)

Device determines whether it is installed in a 32-bit or 64-bit PCI bus and displays the result.

Bus speed (MHz)

Device determines the PCI bus speed and displays the result.

Current PCI state

Current value of the PCI state register which controls the DMA hardware and other PCI characteristics.

Transmit coalesce value**Transmit interrupt delay (µs)**

Transmit interrupts are generated every n "coalesce value" transmit completions, but no later than n "interrupt delay" microseconds after transmit completion.

Receive coalesce value**Receive interrupt delay (µs)**

Receive interrupts are generated every n "coalesce value" receive completions, but no more than n "interrupt delay" microseconds after receipt of a packet.

4GB workaround buffer in use flag

Set if the 4-gigabyte workaround buffer is in use. The DEGPA cannot transmit across a 4-gigabyte boundary in memory. If the driver encounters such a transmit request, it copies the transmit packet to a workaround buffer and transmits the workaround buffer.

4GB workarounds done

Number of 4-gigabyte workarounds done for transmit packets.

4GB workaround buffers orphaned

Number of receive buffers which crossed a 4-gigabyte boundary. The driver orphans these buffers to prevent their use (allocates and never deallocates).

Receive return ring workarounds

On very rare occasions, the DEGPA does not update a receive return ring pointer correctly. The driver detects this condition and resets the device to clear the condition and continue. This is the number of times the workaround was done.

Current time (EXE\$GL_ABSTIM_TICS)

Current time in 10-millisecond ticks of the counters request from LANCP.

Statistics time (EXE\$GL_ABSTIM_TICS)

Time in 10-millisecond ticks of the last statistics update from the device.

Last duplex mode mismatch reported

Time of the last duplex mode mismatch reported.

Previous CRC or late collision error

Time of the last CRC or late collision error recorded.

Transmit time limit

Transmit time limit (in seconds) after which a timeout is declared.

Timer routine interval

Resolution of the transmit timer (the real timeout is limit + interval).

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the DEGPA are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the DEGPA are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and DEGPA, the driver records the time that each transmit buffer is given to the DEGPA. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Minimum buffers requested

Minimum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current minimum limit".

Maximum buffers requested

Maximum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current maximum limit".

Current minimum limit

Minimum number of receive buffers as determined by the driver and by management request, "Minimum buffers requested". The driver will not allow the number of receive buffers in its receive queues to drop below this value. This does not include receive buffers transferred to applications (and not yet returned).

Current maximum limit

Maximum number of receive buffers as determined by the driver and by management request, "Maximum buffers requested". The driver will deallocate receive buffers until the number allocated does not exceed this maximum. When applications return receive buffers, the number may exceed this maximum, at which point they are deallocated. Note that the driver allows this maximum to be exceeded slightly to limit allocation and deallocation activity. See "[Target number of buffers maximum](#)" below.

Current number of buffers

Current number of receive buffers owned by the driver.

Target number of buffers

Desired number of receive buffers. As receive activity does not result in lost packets, this number is gradually decreased toward the "Current minimum limit". If packets are lost, this number is dramatically increased toward the "Current maximum limit". When the driver allocates receive buffers, it allocates them until reaching this target number.

Target number of buffers maximum

Maximum desired number of receive buffers. When an application returns a buffer to the driver, if the current number does not exceed this target maximum, the buffer is kept by the driver. Otherwise, it is deallocated.

MAC Statistics (RFC 1643)

Alignment errors

Number of packets received with CRC errors that are not an integral number of bytes long. These packets are discarded and then counted by the device.

FCS (Frame Check Sequence) errors

Number of packets received with CRC errors that are an integral number of bytes long. These packets are discarded and then counted by the device.

Single collisions

Number of successfully transmitted packets that encountered exactly one collision during transmission (successful after retransmit). These occur in half-duplex mode only.

Multiple collisions

Number of successfully transmitted packets that encountered more than one collision during transmission (successful after multiple retransmits). These occur in half-duplex mode only.

SQE test errors

Number of Signal Quality Error (SQE) test errors generated after successful transmission. These are also called heartbeat errors. Some network hardware does not support this test function, so this error happens on every transmit. These occur in half-duplex mode only.

Deferred transmits

Number of successful transmits that were delayed because the medium was busy. These occur in half-duplex mode only.

Late collisions

Number of times a collision was detected later than 512 bit-times into the transmission of a packet. The transmit fails. These occur in half-duplex mode only.

Excessive collisions

Number of transmits that failed due to excessive collisions. These occur in half-duplex mode only.

Internal MAC transmit errors

Number of transmits that failed because of an internal Media Access Control (MAC) sublayer error other than late collision, excessive collisions, or carrier sense error.

Carrier sense errors

Number of transmits that failed because the carrier was not present during any or all of the transmission attempts.

Frame too long errors

Number of received frames that were longer than the jumbo packet size. These packets are discarded and then counted by the device.

Internal MAC receive errors

Number of receive packets discarded because of an internal MAC sublayer error that is not frame too long, alignment error, or FCS error. These packets are discarded and then counted by the device.

Interface Statistics (RFC 1213)

Index (ifIndex)

A unique value for each interface. Its value ranges between 1 and the value of ifNumber. The value for each interface must remain constant between reinitializations of the entity's network management system.

Type (ifType)

The type of interface, distinguished according to the physical/link protocols immediately "below" the network layer in the protocol stack. The value set by the DEGPA is 6 "ethernet-csmacd".

Mtu (ifMtu)

The size of the largest datagram that can be sent/received on the interface, specified in octets. For interfaces that are used for transmitting network datagrams, this is the size of the largest network datagram that can be sent on the interface.

This field is specified by the driver and is set to the jumbo buffer size previously described.

Speed (ifSpeed)

An estimate of the interface's current bandwidth in bits/second. For interfaces which do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth.

Admin requested status (ifAdminStatus)

The desired state of the interface. The testing state indicates that no operational packets can be passed. The values are: 1 – Up, 2 – Down; 3 – Testing.

This field is always 1 because the driver does not obtain statistical updates from the DEGPA in any other state.

Operational status (ifOperStatus)

The current operational state of the interface. The testing state indicates that no operational packets can be passed. The values are: 1 – Up, 2 – Down; 3 – Testing.

This field is always 1 because the driver does not obtain statistical updates from the DEGPA in any other state.

Last change (ifLastChange)

The value of sysUpTime at the time the interface entered its current operational state. If the current state was entered prior to the last reinitialization of the local network management subsystem, then this object contains a value of 0.

Inbound discards (ifInDiscards)

The number of inbound packets that were chosen to be discarded, even though no errors had been detected to prevent their delivery to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

Inbound errors (ifInErrors)

The number of inbound packets that contained errors preventing them from being deliverable to a higher-level protocol.

Inbound unknown protos (ifInUnknownProtos)

The number of packets received via the interface that were discarded because of an unknown or unsupported protocol.

Outbound discards (ifOutDiscards)

The number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Outbound errors (ifOutErrors)

The number of outbound packets that could not be transmitted because of errors.

Outbound queue length (ifOutQlen)

The length of the output packet queue (in packets).

Physical address (ifPhysAddress)

The interface's address at the protocol layer immediately "below" the network layer in the protocol stack. For interfaces that do not have such an address (for example, a serial line), this object should contain an octet string of zero length.

Description (ifDescr)

A textual string containing information about the interface. This string should include the name of the manufacturer, the product name, and the version of the hardware interface.

The value set by the DEPGA is blank.

Bytes received (ifHCInOctets)

The total number of octets received on the interface, including framing characters. This object is a 64-bit version of ifInOctets.

Unicast packets received (ifHCInUcastPkts)

The number of packets (delivered by this sublayer to a higher sublayer) that were not addressed to a multicast or broadcast address at this sublayer. This object is a 64-bit version of ifInUcastPkts.

Multicast packets received (ifHCInMulticastPkts)

The number of packets (delivered by this sublayer to a higher sublayer) that were addressed to a multicast address at this sublayer. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifInMulticastPkts.

Broadcast packets received (ifHCInBroadcastPkts)

The number of packets (delivered by this sublayer to a higher sublayer) that were addressed to a broadcast address at this sublayer. This object is a 64-bit version of ifInBroadcastPkts.

The DEPGA does not distinguish between broadcast multicast and non-broadcast multicast. As such, this counter is zero.

Bytes sent (ifHCOutOctets)

The total number of octets transmitted out of the interface, including framing characters. This object is a 64-bit version of ifOutOctets.

Unicast packets sent (ifHCOutUcastPkts)

The total number of packets that higher-level protocols requested be transmitted and that were not addressed to a multicast or broadcast address at this sublayer, including those that were discarded or not sent. This object is a 64-bit version of ifOutUcastPkts.

Multicast packets sent (ifHCOutMulticastPkts)

The total number of packets that higher-level protocols requested be transmitted and that were addressed to a multicast address at this sublayer, including those that were discarded or not sent. For a MAC layer protocol, this includes both Group and Functional addresses. This object is a 64-bit version of ifOutMulticastPkts.

Broadcast packets sent (ifHCOutBroadcastPkts)

The total number of packets that high-level protocols requested be transmitted and that were addressed to a broadcast address at this sublayer, including those that were discarded or not sent. This object is a 64-bit version of ifOutBroadcastPkts.

The DEGPA does not distinguish between broadcast multicast and non-broadcast multicast. As such, this counter is zero.

Link updown trap enable (ifLinkUpDownTrapEnable)

Indicates whether linkUp/linkDown traps should be generated for this interface. By default, this object should have the value enabled (1) for interfaces which do not operate "on top" of any other interface (as defined in the ifStackTable), and disabled (2) otherwise.

This value is set to 2 by the DEGPA.

High speed status (ifHighSpeed)

An estimate of the interface's current bandwidth in units of 1 000 000 bits/second. If this object reports a value of n , then the speed of the interface is somewhere in the range of $(n - 500\,000)$ to $(n + 499\,999)$. For interfaces that do not vary in bandwidth or for those where no accurate estimation can be made, this object should contain the nominal bandwidth. For a sublayer that has no concept of bandwidth, this object should be 0.

This value is set to 1000 by the DEGPA.

Promiscuous mode status (ifPromiscuousMode)

This object has a value of false (2) if this interface only accepts packets/frames that are addressed to this station. This object has a value of true (1) when the station accepts all packets/frames transmitted on the media. The value 1 is only legal on certain types of media. If legal, setting this object to a value of 1 may require the interface to be reset before becoming effective. The value of ifPromiscuousMode does not affect the reception of broadcast and multicast packets/frames by the interface.

Connector present status (ifConnectorPresent)

This object has the value true (1) if the interface sublayer has a physical connector and the value false (2) otherwise.

Host Commands Statistics

Host state commands

Number of times the driver changed the host state. It can be in one of two states, up or down. The driver issues a host state up command when the first user starts, and a host state down when the last user stops.

FDR (False Discovery Rate) filtering commands

Unused.

Set receive producer index commands

Unused.

Update GENCOMM statistics commands

Unused.

Reset jumbo ring commands

Unused.

Add mcast address commands

Number of times the driver issued a multicast address add command to the device to add a new multicast address to the multicast filtering table. The driver maintains its own multicast address table and updates the device whenever its table changes.

Del mcast address commands

Number of times the driver issued a multicast address delete command to the device to delete a multicast address from the multicast filtering table. The driver maintains its own multicast address table and updates the device whenever its table changes.

Set promiscuous mode commands

Number of times the driver enabled or disabled promiscuous receive mode. The driver issues these commands whenever a user enables or disables promiscuous mode.

Link negotiate commands

Number of times the driver issued a link negotiate command to the device to cause it to set up the link again. The driver issues this command only when a device-specific (debug) command is issued via LANCP to explicitly set a new link value.

Set MAC address commands

Number of times the driver issues a set MAC address command to change the MAC address of the device. Typically, this is issued by the driver once when DECnet starts.

Clear profile commands

Unused.

Set multicast mode commands

Number of times the driver enabled or disabled all multicast receive mode. The driver issues these commands whenever a user enables or disables all multicast mode.

Clear statistics commands

Number of clear statistics commands the driver issued to the device on device startup. Normally, there is one clear statistics command for every reset that is done, plus any device-specific (debug) commands issued via LANCP to explicitly clear the counters.

Set receive jumbo producer index commands

Unused.

Set receive mini producer index commands

Unused.

Refresh statistics commands

Unused.

Unknown commands

Number of commands issued by the driver that were not recognized by the device.

Event Statistics

FW operational events

Number of TG_EVENT_FIRMWARE_OPERATIONAL events generated.

Link state changed events

Number of TG_EVENT_LINK_STATE_CHANGED events generated.

Statistics updated events

Number of TG_EVENT_STATS_UPDATED events generated.

Error events

Number of TG_EVENT_ERROR events generated.

MCast list updated events

Number of TG_EVENT_MULTICAST_LIST_UPDATED events generated.

Reset jumbo ring events

Number of TG_EVENT_RESET_JUMBO_RING events generated.

Set send producer index events

Number of times the DEGPA device has seen updates to the send producer index.

Set send consumer index events

Number of times the send consumer index was updated.

Set receive return producer index events

Number of times the receive return producer index was updated.

Interrupt Statistics

Interrupts generated

Number of interrupts generated by the DEGPA device.

Interrupts avoided

Number of interrupts not generated by the DEPGA device (because of interrupt mitigation).

Event threshold hit

Number of times the event max coalesce buffer descriptors (BD) threshold was reached.

Send BD threshold hit

Number of times the BD_FLAG_COAL_NOW flag was set or that the send max coalesce BD threshold was reached.

Receive BD threshold hit

Number of times the receive max coalesce BD threshold was reached. When this happens, the firmware updates the receive return producer index for the host.

DMA Statistics

DMA read overrun errors

Number of times a DMA read overrun error occurred.

DMA read underrun errors

Number of times a DMA read underrun error occurred.

DMA write overrun errors

Number of times a DMA write overrun error occurred.

DMA write underrun errors

Number of times a DMA write underrun error occurred.

DMA write master abort errors

Number of times a DMA read master abort error occurred.

DMA read master abort errors

Number of times a DMA write master abort error occurred.

Ring Statistics

DMA write ring full

Number of times the DMA write ring was full.

DMA read ring full

Number of times the DMA read ring was full.

Event ring full

Number of times the device event ring was full.

Event producer ring full

Number of times the DMA write ring was full trying to write either the DMA event or event producer to host memory.

MAC transmit descriptor ring full

Number of times the MAC transmit descriptor ring was full.

Transmit buffer space full

Number of times the transmit buffer space was full.

No more DMA write descriptors

Number of times the device ran out of DMA write descriptors.

No more receive BDs

Number of times the device ran out of receive buffer descriptors.

No space in return ring

Number of times the device could not place a buffer descriptor in the return ring because it was full.

Send BDs owned by NIC

Number of send buffer descriptors currently being processed by the device.

Receive BDs owned by NIC

Number of standard (1518 bytes) receive buffer descriptors owned by the device.

Jumbo receive BDs owned by NIC

Number of jumbo receive buffer descriptors owned by the device.

Mini receive BDs owned by NIC

Unused, always 0.

Total receive BDs owned by NIC

Number of receive buffer descriptors owned by the device.

Total send BDs owned by NIC

Number of transmit buffer descriptors owned by the device.

Jumbo frames split into multiple standard BDs

Number of times a jumbo frame was split into multiple standard buffer descriptors.

Bus hang cleared

Number of times an SBus DMA bug was worked around.

Posting an event was delayed

Number of times posting an event was delayed.

Internal MAC Receive Statistics

Packets dropped due to late collisions

Number of receive packets dropped because of late collisions. These occur in half-duplex mode only.

Packets dropped because of loss of link

Number of receive packets dropped because of the loss of a link, such as a cable disconnect, broken cable, NIC hardware failure, or link partner hardware failure.

Packets dropped because of PHY decode errors

Number of receive packets dropped because of Physical Layer (PHY) decode errors, such as a hardware failure that generates so much noise that the PHY cannot recognize the signal, or some other hardware failure.

Packets aborted by MAC because of remote errors

Number of receive packets aborted by MAC because of remote errors, such as receiving a packet smaller than 64 bytes, a PHY decode error, a collision detected during receipt, or an error occurring during gigabit half-duplex frame extension.

Frames dropped due to lack of NIC internal resources

Number of receive packets dropped because of a lack of NIC internal resource, such as memory space or MAC descriptors. This usually occurs because the bus is too slow.

Unicast packets dropped – DA does not match

Number of receive packets dropped because the destination address (DA) in the packet does not match our address. This happens regularly on full-duplex repeaters or during switch flooding.

Multicast packets dropped – DA does not match

Number of receive packets dropped because the destination address in the packet does not match the multicast address list.

Flow control packets received

Number of flow control packets received.

Packets dropped because of lack of space

Number of receive packets dropped because of lack of space. This usually occurs because the bus is too slow.

Packets dropped because of collisions

Number of receive packets dropped because of collisions. This is caused by two nodes sending messages simultaneously. These occur in half-duplex mode only.

MAC receive attentions

Number of MAC receive attentions, including receive descriptor attention, receive buffer attention, flow control XON sent, flow control XOFF sent, and FIFO overrun.

Link state change attentions

Number of link state change attentions, including auto-negotiation changed, link state error, and link ready changed.

Sync lost attentions

Number of sync lost attentions.

Link config attentions

Number of link config attentions, possibly caused by the link partner changing its link configuration. This does not indicate hardware failure unless the actual number of link configuration changes is smaller than this counter.

MAC resets

Number of times the MAC was reset. This is caused by link loss and attempts to reestablish the link.

Receive BD attentions

Unused.

Receive buffer attentions

Number of receive buffer descriptor attentions which is triggered by the number of received buffer descriptors reaching a predefined threshold.

No frame cleanups after receive buffer attention

Number of times the receive buffer got cleaned up when it was full and the frame received count was zero. This indicates that the data in the receive buffer was unreliable; this can happen if the remote mode sent continuous error frames.

One frame cleanups after receive buffer attention

Number of times the receive buffer got cleaned up when it was full and the frame received count was 1. This could happen if the remote node sent continuous error frames.

Multiple frame cleanups after receive buffer attention

Number of times the receive buffer got cleaned up when it was full and the frame received count was greater than 1. This could happen if the remote node sent continuous error frames.

Receive buffer cleanups by time

Number of times the device set up a timer to wait for receive buffer space to be freed up when the receive buffer was full. This can happen if the remote node sent continuous error frames.

DMA cleanups due to DMA attentions

Number of times the DMA buffer got cleaned up when the receive buffer was full. This can happen if the remote node sent continuous error frames.

Internal MAC Transmit Statistics**MAC TRANSMIT 1–15 collisions**

Transmit collision histogram. These are updated in half-duplex mode only.

Transmit attentions

Total number of transmit attentions.

Driver Messages

Console messages issued by the driver.

6.2. DEGPA Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, . . .))** LANCP command provides a mechanism to issue device-specific functions to the DEGPA driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the [VSI OpenVMS System Management Utilities Reference Manual, Volume I: A-L \[https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590\]](https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590) or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the DEGPA using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The DEGPA is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(32bits,16bits))
DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(32bits,16bits))
DEVICE_SPECIFIC=(FUNCTION="FIL3",VALUE=(32bits,16bits))
DEVICE_SPECIFIC=(FUNCTION="FIL4",VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For

example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the DEGPA, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="FXMT",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is limited to the range 2–40. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="LINK",VALUE=*linkvalue*)

Set the requested link settings for the DEGPA, then issue a command to the DEGPA requesting that link negotiation be redone.

DEVICE_SPECIFIC=(FUNCTION="DXMT",VALUE=*delayvalue*)

Change the interrupt mitigation delay on transmit completion value, the "Transmit interrupt delay" value. This value is in microseconds. If out of range, 1–1000, it is set to 1.

DEVICE_SPECIFIC=(FUNCTION="DRCV",VALUE=*delayvalue*)

Change the interrupt mitigation delay on receive completion value, the "Receive interrupt delay" value. This value is in microseconds. If out of range, 1–1000, it is set to 1.

DEVICE_SPECIFIC=(FUNCTION="CXMT",VALUE=*packetvalue*)

Change the interrupt mitigation coalesce on transmit completion value, the "Transmit coalesce value". This value is in number of packets.

DEVICE_SPECIFIC=(FUNCTION="CRCV",VALUE=*packetvalue*)

Change the interrupt mitigation coalesce on receive completion value, the "Receive coalesce value". This value is in number of packets.

Chapter 7. Broadcom 1Gb Ethernet

The Broadcom 5700 Ethernet devices are:

Device	Bus	Characteristics
DEGXA-SA	PCI	1000BaseSX (multimode fiber)
DEGXA-TA	PCI	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
DEGXA-SB	PCI-X	1000BaseSX (multimode fiber)
DEGXA-TB	PCI-X	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
BCM5703 (LOM)	PCI	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
A6847A	PCI	1000BaseSX (multimode fiber)
A6825A	PCI	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
AB352A	PCI-X	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
A9782A	PCI-X	1000BaseSX (multimode fiber)
A9784A	PCI-X	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
AB454A	PCI-X	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)
BCM5701 (LOM)	PCI	10BaseT (UTP), 100BaseTX (UTP), 1000BaseT (UTP)

The driver is SYS\$EW5700.EXE (Alpha, IA-64).

7.1. Broadcom 5700 Driver Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS EWA** displays the entire set of internal counters maintained by the 5700 driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EWA**.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the 5700 device.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EWA** command in the quadword driver version field.

Device revision

Hardware revision level identified by the Broadcom 5700 chip revision. It is displayed as A0, A1, B0, B1, B2, C0, C1, C2. If the version is later than the driver recognizes, it is displayed as >C2, for example. The actual revision is located in bits <31:16> of the Misc Host Control register displayed in the Registers section.

Device PHY ID

ID of the PHY chip found on the device. Bits <31:10> is the Organizationally Unique Identifier (OUI), bits <9:4> is the model number and bits <3:0> is the revision.

Device interrupts

Number of times the driver interrupt service routine was called.

Link transitions

Number of link transitions seen by the driver.

Link transitions avoided

Number of times a second check of the link state avoided a transition from link up to link down.

Status block errors

Number of times the error bit in the status block was set at the same time the fiber link partner was sending configuration data for auto-negotiation.

Status block link state changes

Number of times the link state change bit was set in the status block.

Link checks

Number of times the driver checked the current link status.

Driver auto-negotiation (fiber) sequences done

Number of times the driver executed the auto-negotiation procedure.

Device resets

Number of times the 5700 chip was reset.

Device resets retried (after 1.5-second timeout)

Number of times the 5700 chip reset was retried because the 5700 firmware did not complete initialization the first time.

Device initializations

Number of times the driver initialized the 5700 chip.

Unit inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Unit inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

User start/change/stop requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O with error status.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Jumbo transmits issued

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

Transmits issued (using map registers)

Number of transmit requests that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Jumbo receives issued

Number of jumbo receive buffers allocated and assigned to the device.

Receives issued (using map registers)

Number of receive buffers that were described to the device using map registers because part of the request existed in memory outside the DMA window, because the receive buffer crossed a page boundary, or because the receive buffer was a jumbo buffer.

Receives discarded (bad frame)

Number of receive packets discarded by the driver due to receive error. By default, the 5700 chip does not deliver damaged receive packets to the driver, but can be asked to do so by a device-specific function.

Receive FIFO overflows

Number of times the driver detected a receive FIFO overflow.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Duplex mode mismatches

Number of times a potential duplex mode mismatch was detected by checking for transmit and receive errors.

Rescheduled forks (too long in fork)

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

Standard receive buffers

Number of 1518-byte receive buffers owned by the device.

Standard receive buffer deallocations

Number of 1518-byte buffer deallocations done by the driver because the number of outstanding buffers exceeded the maximum number allowed by the driver.

Jumbo receive buffers (current)

Number of large receive buffers owned by the device.

Jumbo receive buffers (minimum)

Minimum number of large receive buffers owned by the device. This is initially set to 32. After the first jumbo receive, the driver sets the minimum to 128.

Jumbo receive buffer allocations

Number of jumbo receive buffer allocations done by the driver.

Jumbo receive buffer deallocations

Number of large buffer deallocations done by the driver because the number of outstanding buffers exceeded the maximum number allowed by the driver.

Standard buffer size (bytes)

Size of the standard receive buffers. The device allows packets up to 1518 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes).

Standard packet size – device standard ring (bytes)

This is the Ethernet packet size (1518 bytes).

Jumbo buffer size (bytes)

Size of the jumbo receive buffers. The device allows packets up to 9018 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes). However, the driver limits the buffer size to the maximum size supported by the pool lookaside lists. The buffer size is 9018 plus 640 bytes of overhead (rounded up to the next 64-byte boundary), which accommodates the full jumbo packet size.

Jumbo packet size – device jumbo ring (bytes)

This is the jumbo packet size (9018 bytes).

Requested link settings

Speed and duplex mode requested by a user.

Current link settings

Current link state.

Remote flow capabilities

Flow control capabilities reported for the link partner.

Current link up timer

Current value of the one-second timer that waits for the link to come back before resetting the device to free pending transmit requests.

Driver flags

Driver flags including the following bits:

- Promiscuous – Promiscuous mode enabled.
- All_Multicast – All multicast mode enabled.
- Jumbo_Frames – Jumbo packets enabled.
- More_Rcv – More receive ring entries enabled.
- Copper – Copper PHY present.
- Fiber – Fiber PHY present.
- 5701 – 5701 chip.
- 5703 – 5703 chip.
- GraniteLOM – Granite 5703 LOM chip.
- Receive_Bad_Frames – Receive bad frames enabled.
- Bugcheck_XTMO – Bugcheck on transmit timeout.

- Setup_Link_In_Progress – Currently setting up the link.
- All_Messages_Enabled – Display all messages.
- Device_Link_Handling – Device does link handling.
- Driver_Link_Handling – Driver does link handling (fiber only).

Driver state

Current driver state, as follows:

- 0 – Driver state is undefined.
- 1 – Driver state is resetting.
- 2 – Driver is running, but the link is down. Completing transmits with error status.
- 4 – Driver is running, and the link is up. Processing transmits and receives.
- 8 – Device is not usable.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

DMA width (bits)

Device determines whether it is installed in a 32-bit or 64-bit PCI bus and displays the result.

Bus speed (MHz)

Device determines whether it is installed in a low-speed (33-MHz PCI or 66-MHz PCI-X) bus or high-speed (66-MHz PCI or 133-MHz PCI-X) bus and displays the result.

Bus type

Device determines whether it is installed in a PCI or PCI-X bus and displays the result.

MSI control (Alloc<6:4>, Req<3:1>, Enable<0>)

Contents of the MSI_CONTROL register that stores the Message Signaled Interrupts (MSI) status. If MSI is enabled, bit <0> is set. Bits <3:1> are the number of requested messages (always set to 3 by the device). Bits <6:4> are the number of messages allocated to the device by the system. If the /DEBUG qualifier is specified, more MSI context is displayed, such as MSI Address and Data registers.

DMA control value

Contents of the DMA control register which regulates read and write DMA operations.

Transmit coalesce value**Transmit interrupt delay (µs)**

Transmit interrupts are generated every *n* "coalesce value" transmit completions, but no later than *n* "interrupt delay" microseconds after transmit completion.

Receive coalesce value**Receive interrupt delay (µs)**

Receive interrupts are generated every *n* "coalesce value" receive completions, but no more than *n* "interrupt delay" microseconds after receipt of a packet.

Initialization delay (µs)

Total time (in microseconds) waiting for the device during initialization including shutdown.

Link setup delay (µs)

Total time (in microseconds) waiting for the device during setup of the link.

Transmit time limit

Transmit time limit (in seconds) after which a timeout is declared.

Timer routine interval

Resolution of the transmit timer (the real timeout is limit + interval).

Registers (wrote/read)

Misc Host Control

Last value written to and read from the Misc Host Control register.

MAC Mode

Last value written to and read from the MAC Mode register.

MAC Status

Last value written to and read from the MAC Status register.

MI Status

Last value written to and read from the MI Status register.

RX Mode

Last value written to and read from the RX Mode register.

TX Status

Last value written to and read from the TX Status register.

Time Stamps

Current uptime

Current system uptime.

Last reset

Last time the device was reset.

Last link up

Last time a link up transition occurred.

Last link down

Last time a link down transition occurred.

Total link uptime

Total time the link has been up.

Total link downtime

Total time the link has been down.

Last driver auto-negotiation (fiber) initiated

Last time the driver started auto-negotiation on the fiber NIC.

Last driver auto-negotiation (fiber) completed

Last time the driver completed auto-negotiation on the fiber NIC.

Time spent doing driver auto-negotiation (fiber)

Total time spent doing auto-negotiation on the fiber NIC.

Last transmit timeout

Time of the last transmit timeout.

Last soft error

Time of the last soft error.

Last duplex mode mismatch reported

Time of the last duplex mode mismatch reported.

Previous CRC or late collision error

Time of the last CRC or late collision error recorded.

Driver Auto-Negotiation Context (fiber)

Auto-negotiation context for driver auto-negotiation on the fiber NIC.

Minimum buffers requested

Minimum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current minimum limit".

Maximum buffers requested

Maximum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current maximum limit".

Current minimum limit

Minimum number of receive buffers as determined by the driver and by management request, "Minimum buffers requested". The driver will not allow the number of receive buffers in its receive queues to drop below this value. This does not include receive buffers transferred to applications (and not yet returned).

Current maximum limit

Maximum number of receive buffers as determined by the driver and by management request, "Maximum buffers requested". The driver will deallocate receive buffers until the number allocated does not exceed this maximum. When applications return receive buffers, the number may exceed this maximum, at which point they are deallocated. Note that the driver allows this maximum to be exceeded slightly to limit allocation and deallocation activity. See "[Target number of buffers maximum](#)" below.

Current number of buffers

Current number of receive buffers owned by the driver.

Target number of buffers

Desired number of receive buffers. As receive activity does not result in lost packets, this number is gradually decreased toward the "Current minimum limit". If packets are lost, this number is dramatically increased toward the "Current maximum limit". When the 5700 driver allocates receive buffers, it allocates them until reaching this target number.

Target number of buffers maximum

Maximum desired number of receive buffers. When an application returns a buffer to the driver, if the current number does not exceed this target maximum, the buffer is kept by the driver. Otherwise, it is deallocated.

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the 5700 chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the 5700 chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and 5700 chip, the driver records the time that each transmit buffer is given to the 5700 chip. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Status Block

Status

After an interrupt, this longword is nonzero. The driver fork process clears the status longword so the driver can recognize whether there is pending work to be done (such as checking the ring indexes for completions or checking the link status). The bits in the longword are defined as follows:

- Bit 1 – Link changed status.
- Bit 2 – Status block updated flag.
- Bit 4 – Status block error.

Receive mini consumer index

Unused.

Receive jumbo consumer index

Current position of the 5700 driver in the jumbo receive buffer ring.

Receive standard consumer index

Current position of the 5700 driver in the standard receive buffer ring.

Receive ring 0 producer index

Current position of the 5700 driver in the receive return ring.

Send ring 0 consumer index

Current position of the 5700 driver in the send ring.

Statistics Block

Statistics – Receive MAC

Bytes received

Number of bytes successfully received.

Runt packets received (bad CRC)

Number of runt packets less than 64 bytes long received and discarded.

Unicast packets received

Number of packets successfully received that were not addressed to a multicast or broadcast address.

Multicast packets received

Number of packets successfully received that were addressed to a multicast address, but not to the broadcast address.

Broadcast packets received

Number of packets successfully received that were addressed to the broadcast address.

FCS errors

Number of packets received with CRC errors that are an integral number of bytes long. These packets are discarded and are counted by the device.

Alignment errors

Number of packets received with CRC errors that are not an integral number of bytes long. These packets are discarded and are counted by the device.

MAC Control XON frames received

Number of XON frames received. An XON frame is a MAC control frame with a pause command and a length equal to zero.

MAC Control XOFF frames received

Number of XOFF frames received. An XOFF frame is a MAC control frame with a pause command and a length greater than zero.

MAC Control other frames received

Number of other MAC control frames received. These are MAC control frames with no pause command.

Transmitting disabled (XOFF)

Number of times the transmitter was disabled because of a receive of an XOFF frame.

Frame too long errors

Number of received frames that were longer than the jumbo packet size. These packets are discarded and are counted by the device.

Frame exceed jabber time errors

Number of frames received that are longer than the specified maximum frame time (for example, a jabber frame).

Runt packets received (good CRC)

Number of frames received with a size less than 64 bytes (including a correct CRC field).

Length errors (length field not equal actual)

Number of frames received where the 802.3 length field does not match the total frame size.

Length errors (type field 1523–1535)

Number of frames received where the field is in the range of 1523 to 1535 (for example, an invalid type field and a length error).

Packets (64 bytes)

Packets (65–127 bytes)

Packets (128–255 bytes)

Packets (256–511 bytes)

Packets (512–1023 bytes)

Packets (1024–1522 bytes)

Packets (1523–2047 bytes)

Packets (2048–4095 bytes)

Packets (4096–8191 bytes)

Packets (8192–9018 bytes)

Number of packets of this size received.

Statistics – Transmit MAC

Bytes sent

Number of bytes successfully transmitted.

Collisions experienced

Number of collisions experienced.

XON packets sent

Number of XON packets sent in support of flow control.

XOFF packets sent

Number of XOFF packets sent in support of flow control.

Flow control sequences done

Number of flow control sequences done.

Internal MAC transmit errors

Number of transmits that failed because of an internal MAC sublayer error other than late collision, excessive collisions, or carrier sense error.

Single collisions

Number of successfully transmitted packets that encountered exactly one collision during transmission (successful after retransmit). These occur in half-duplex mode only.

Multiple collisions

Number of successfully transmitted packets that encountered more than one collision during transmission (successful after multiple retransmits). These occur in half-duplex mode only.

Deferred transmits

Number of successful transmits that were delayed because the medium was busy. These occur in half-duplex mode only.

Excessive collisions

Number of transmits that failed because of excessive collisions. These occur in half-duplex mode only.

Late collisions

Number of times a collision was detected later than 512 bit-times into the transmission of a packet. The transmit fails. These occur in half-duplex mode only.

MAC transmit 2–15 collisions

Number of successfully transmitted packets that encountered n collisions during transmission (successful after n retransmits). These occur in half-duplex mode only.

Unicast packets sent

Number of successfully transmitted packets that were not addressed to a multicast or broadcast address.

Multicast packets sent

Number of successfully transmitted packets that were addressed to a multicast address but not to the broadcast address.

Broadcast packets sent

Number of successfully transmitted packets that were addressed to a broadcast address.

Carrier sense errors

Number of transmits that failed because a carrier was not present during any or all of the transmission attempts.

Outbound discards

Number of outbound packets that were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

Outbound errors

Number of outbound packets that could not be transmitted because of errors.

Statistics – Receive List Placement State Machine**Frames received onto return ring 1–16**

Number of frames received onto each of the return rings.

Frames discarded using filters

Number of frames received but discarded after validation by the receive filters.

DMA write queue full

Number of times the DMA write queue was full.

DMA high priority write queue full

Number of times the DMA write high-priority queue was full.

No more receive BDs

Number of times the NIC ran out of receive buffer descriptors.

Inbound discards

Number of inbound packets that were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

Inbound errors

Number of inbound packets that contained errors that prevented them from being deliverable to a higher-layer protocol.

Receive threshold hit

Number of times the receive max coalesce frames threshold was reached, resulting in a status block update and interrupt.

Statistics – Send Data Initiator State Machine

Frames sent from send ring 1–16

Number of frames sent from each of the send rings.

DMA read queue full

Number of times the DMA read queue was full.

DMA high priority read queue full

Number of times the DMA read high-priority queue was full.

Send data completion queue full

Number of times the send data completion flow-through-queue (FTQ) was full.

Statistics – Host Coalescing State Machine

Send producer index updates

Number of times the NIC has seen updates to any send producer ring index.

Ring status updates

Number of times the status block was updated (written to host memory). If the driver is not currently in its interrupt service routine, an interrupt is generated after the update.

Interrupts generated

Number of interrupts generated by the NIC.

Interrupts avoided

Number of interrupts avoided by the NIC (because of interrupt mitigation).

Send threshold hit

Number of times the send max coalesce frames threshold was reached, resulting in a status block update and interrupt.

Debug Data

Consists of a list of the contents-significant 5700 chip registers, collected when internal driver counters were requested.

EEPROM Contents

Consists of the Electrically Erasable Programmable Read-Only Memory (EEPROM) contents of the non-volatile memory on the 5700 NIC, which contains manufacturing information.

Driver Messages

Console messages issued by the driver.

7.2. Broadcom 5700 Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, . . .))** LANCP command provides a mechanism to issue device-specific functions to the 5700 driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the [VSI OpenVMS System Management Utilities Reference Manual, Volume I: A-L \[https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590\]](https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590) or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the 5700 chip using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The 5700 chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FILA", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILB", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILC", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILD", VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL1", VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL2", VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the 5700 chip, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="FXMT",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is limited to the range 2–40. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=FUNCTION="FCRC"

Send the next packet without a valid CRC.

This is used to generate CRC errors for test purposes.

DEVICE_SPECIFIC=(FUNCTION="RBD",VALUE=*value*)

If *value* is zero, the chip will discard bad received packets. If *value* is nonzero, the bad packets will be received and discarded by the driver. This allows the packets to be traced and inspected.

DEVICE_SPECIFIC=(FUNCTION="DXMT",VALUE=*delayvalue*)

Change the interrupt mitigation delay on transmit completion value, the "Transmit interrupt delay" value. This value is in microseconds. If less than 1, it is set to 1. If greater than 1000, it is set to 1000.

DEVICE_SPECIFIC=(FUNCTION="DRCV",VALUE=*delayvalue*)

Change the interrupt mitigation delay on receive completion value, the "Receive interrupt delay" value. This value is in microseconds. If less than 1, it is set to 1. If greater than 1000, it is set to 1000.

DEVICE_SPECIFIC=(FUNCTION="CXMT",VALUE=*packetvalue*)

Change the interrupt mitigation coalesce on transmit completion value, the "Transmit coalesce value". This value is in number of packets. If less than 1, it is set to 1. If greater than 64, it is set to 64.

DEVICE_SPECIFIC=(FUNCTION="CRCV",VALUE=*packetvalue*)

Change the interrupt mitigation coalesce on receive completion value, the "Receive coalesce value". This value is in number of packets. If less than 1, it is set to 1. If greater than 128, it is set to 128.

DEVICE_SPECIFIC=(FUNCTION="DYNI",VALUE=*value*)

If *value* is clear, disable dynamic interrupt mitigation (DIM). If set, enable DIM. The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="AUTO"

Restart auto-negotiation.

DEVICE_SPECIFIC=FUNCTION="INTR"

Generate an interrupt.

DEVICE_SPECIFIC=(FUNCTION="XDEL",VALUE=*timevalue*)

Delay issuing the next transmit request for the time specified in 10-millisecond units. A maximum of 2000 (20 seconds) is allowed.

This function is useful for investigating application behavior when a transmit request is delayed.

DEVICE_SPECIFIC=(FUNCTION="XTOB",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a transmit timeout occurs. If 0 (disable), transmit timeout results in a device reset.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="XLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long transmit occurs. If 0 (disable), a long transmit completion is counted.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="FLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long fork occurs. If 0 (disable), a long time to run a fork process is recorded in the counters.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="SLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long one second occurs. If 0 (disable), a long time before the one-second timer fork process is run is recorded in the counters.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="ALLM",VALUE=*setting*)

Change the setting of the driver flag "all messages". If the "all messages" flag is set, the driver displays all console messages. The default setting is bit 31 of the LAN_FLAGS system parameter. This function allows the "all messages" setting to be set for an individual device rather than all LAN devices on the system.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="LOOP",VALUE=*loopvalue*)

Specify *loopvalue* to enable (1) or disable (0) loopback mode.

This is useful in test scenarios to run an application in loopback mode, where for standard use the application would not allow selection of loopback mode.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="OMSG"

Console messages are displayed by the driver during initialization, auto-negotiation, and link handling. However, the driver limits the total number of messages displayed. This function resets the count, allowing additional messages to be displayed.

DEVICE_SPECIFIC=(FUNCTION="RCOR",VALUE=*pty*)

DEVICE_SPECIFIC=(FUNCTION="XCOR",VALUE=*pty*)

Corrupt a receive or transmit packet. When this function is completed, the protocol type is specified, and if it matches on a transmit or receive packet, the next packet seen is corrupted by changing the last bit in the VCRP or CXB, XORing it with a 1. The protocol type is then cleared so the corruption is not done again. The corruption is only checked when transmit or receive packet data tracing is being done (in the trace routine), so there is no performance hit for checking each packet to verify if it should be corrupted.

The following example shows how to corrupt a PEDRIVER transmit packet:

```
$ MCR LANCNP SET DEVICE/TRACE EWA
$ MCR LANCNP SET DEVICE/DEVICE_SPECIFIC=(FUNCTION="XCOR", VALUE=%x760) EWA
```

DEVICE_SPECIFIC=(FUNCTION="FLOW",VALUE=*value*)

Disable flow control (if *value* is zero) or enable flow control (if *value* is nonzero).

DEVICE_SPECIFIC=(FUNCTION="SCOU",VALUE=*offset*)

Simulate a counter value by writing to device memory where the counters are located and incrementing a value.

DEVICE_SPECIFIC=FUNCTION="HBUG"

Force a hardware bug.

DEVICE_SPECIFIC=(FUNCTION="DMAC",VALUE=*value*)

Write the DMA Control Register with the *value* given.

DEVICE_SPECIFIC=(FUNCTION="WCSR",VALUE=(*registeraddress,value*))

Write the device register specified with the *value* given.

DEVICE_SPECIFIC=FUNCTION="FLSC"

Force a link state change interrupt.

(SYS\$EW5700_MON.EXE driver only).

DEVICE_SPECIFIC=(FUNCTION="LSCI",VALUE=*value*)

If *value* is zero, disable link state change interrupts. If *value* is nonzero, enable link state change interrupts.

DEVICE_SPECIFIC=(FUNCTION="FLIN",VALUE=*setting*)

Force link indication.

- If *value* is 0 (disabled), do not force the link state.
- If *value* is 1, force the link to be up.
- If *value* is 2, force the link to be down.

DEVICE_SPECIFIC=(FUNCTION="BOUN",VALUE=(*shorttime,longtime,blocktime*))

Change the link bounce parameters. The values are defined as follows:

- *shorttime* – Link up time to consider link bounce too short (in ABSTIM_TICS).

Minimum/Default = 200 / Maximum = 1000

- *longtime* – Link up time to disable link bounce handling (in ABSTIM_TICS).

Minimum = *shorttime* / Default = 1000 / Maximum = 6000

- *blocktime* – Maximum link check block time (in ABSTIM_TICS).

Minimum/Default = 6000 / Maximum = 360 000

If any of the times are not supplied, all of the values are set to the defaults.

Chapter 8. Intel 1Gb Ethernet

The Intel 1Gb devices are:

Table 8.1. Intel 1Gb Ethernet Chip Designations

Chip Name	Chip Number
Kenai32	82540EM
Kenai32M	82540EP
Cordova	82544EI
Cordova-R	82544GC
Kenai64M	82545EM
Attla	82545GM
Anvik	82546EB
Anvik II	82546GB
Ophir	82571EB
Zoar	82575EB

The A7011s and ABxxxx are PCI-X devices. The ADxxxx are PCI-e devices. The LOMs may be either depending on the system.

Table 8.2. Intel 1Gb Ethernet Device Type Translation

Vendor ID	Subsystem/ Subvendor ID	Device Name	Chip
107A8086	12A8103C	A7011A	82546GB (Fiber 2-port)
10798086	12A6103C	A7012A	82546GB (Copper 2-port)
10798086	12CF103C	AB352A	82546GB (Copper 2-port)
10798086	12CD103C	82546LOM	82546GB (Copper embedded on rx2620)
10798086	12C6103C	82546LOM	82546GB (Copper embedded on rx1620)
100E8086	1274103C	82540LOM	82540EM (Copper embedded on rx2600)
10798086	12D1103C	AB290A	82546GB (Castor Copper 2-port + SCSI U320 combo)
10798086	12D3103C	AB545A	82546GB (Cerberus Copper 4-port)
10268086	12DE103C	AD193A	82546GB (Vega Copper 1-port + 1-port 4G QLogic FC)
10798086	12DF103C	AD194A	82546GB (Vega Copper 2-port + 2-port 4G QLogic FC)
10268086	1333103C	AD331A	82545GM (Cyclops Copper 1-port)
10278086	1334103C	AD332A	82545GM (Cyclops Fiber 1-port)

Vendor ID	Subsystem/ Subvendor ID	Device Name	Chip
105E8086	704E103C	AD337A	82571EB (Ashta Copper 2-port) (Enchanted Rock)
105F8086	704F103C	AD338A	82571EB (Ashta Fiber 2-port)
10BC8086	704B103C	AD339A	82571EB (Ashta Copper 4-port) (Lost Maples)
10A78086	171A103C	82575LOM	82575EB (Kauai LOM)
10A98086	171A103C	82575LOM	82575EB (Buccaneer LOM)
10A78086	7042103C	AD221A	82575EB (Deneb Copper 1-port + 1-port 4G Emulex FC)
10A78086	7043103C	AD222A	82575EB (Deneb Copper 2-port + 2-port 4G Emulex FC)
10A98086	7054103C	AD393A	82575EB (Deneb Fiber 2-port + 2-port 4G Emulex FC)
10D98086	1716103C	NC360M	82571EB (Blade Fiber mezzanine 2-port) (Cedar Hill)
10DA8086	1717103C	NC364M	82571EB (Blade Fiber mezzanine 4-port) (Mother Neff)
100E8086	001E103C	HPVM VIO	82546GB (HPVM emulated VIO device)

The driver is SYS\$EI1000.EXE (IA-64).

8.1. Intel 1Gb Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS EWA** displays the entire set of internal counters maintained by the Intel 1Gb driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EIA**.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the Intel 1Gb device.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EIA** command in the quadword driver version field.

Device revision

Hardware revision level of the chip.

Device PHY ID

ID of the PHY chip found on the device. Bits <31:10> is the OUI, bits <9:4> is the model number and bits <3:0> is the revision.

Device interrupts

Number of times the driver interrupt service routine was called.

Link transitions

Number of link transitions seen by the driver.

Cabling errors

Number of times a cable fault was detected for the fiber NIC.

Interrupt link state changes

Number of link state changes reported in the interrupt control register.

Link checks

Number of times the driver checked the current link status.

Device resets

Number of times the device was reset.

Unit inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Unit inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

User start/change/stop requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O with error status.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Jumbo transmits issued

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

Transmits issued (using map registers)

Number of transmit requests that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Jumbo receives issued

Number of jumbo receive buffers allocated and assigned to the device.

Receives issued (using map registers)

Number of receive buffers that were described to the device using map registers because part of the request existed in memory outside the DMA window, because the receive buffer crossed a page boundary, or because the receive buffer was a jumbo buffer.

Receives discarded (bad frame)

Number of receive packets discarded by the driver due to receive error. By default, the chip does not deliver damaged receive packets to the driver, but can be asked to do so by a device-specific function.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Duplex mode mismatches

Number of times a potential duplex mode mismatch was detected by checking for transmit and receive errors.

Rescheduled forks (too long in fork)

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

PHY register read/write errors

Number of errors reading or writing a PHY register.

Standard buffer size (bytes)

Size of the standard receive buffers. The device allows packets up to 1518 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes).

Standard packet size (bytes)

This is the Ethernet packet size (1518 bytes).

Jumbo buffer size (bytes)

Size of the jumbo receive buffers. The device allows packets up to 16 384 bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes). However, the driver limits the packet size to 8192 bytes, so the buffer size is 8192 plus 640 bytes of overhead (rounded up to the next 64-byte boundary).

Jumbo packet size (bytes)

This is the jumbo packet size (8192 bytes).

Requested link settings

Speed and duplex mode requested by a user.

Current link settings

Current link state.

Current link up timer

Current value of the one-second timer that waits for the link to come back before resetting the device to free pending transmit requests.

Driver flags

Driver flags.

Driver state

Current driver state, as follows:

- 0 – Driver state is undefined.
- 1 – Driver state is resetting.
- 2 – Driver is running, but the link is down. Completing transmits with error status.
- 4 – Driver is running, and the link is up. Processing transmits and receives.
- 8 – Device is not usable.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

DMA width (bits)

Device determines whether it is installed in a 32-bit or 64-bit PCI bus and displays the result.

Bus speed (MHz)

Device determines whether it is installed in a low-speed (33-MHz PCI or 66-MHz PCI-X) bus or high-speed (66-MHz PCI or 133-MHz PCI-X) bus and displays the result.

Bus type

Device determines whether it is installed in a PCI or PCI-X bus and displays the result.

Interrupt delay value

Minimum delay between interrupts, in units of 256 nanoseconds.

Transmit descriptor threshold

Number of descriptors that the chip will accumulate before writing them back.

Transmit interrupt delay (µs)

Amount of delay after transmit completion that an interrupt is generated.

Receive interrupt delay (µs)

Amount of delay after receive completion that an interrupt is generated.

Transmit time limit

Transmit time limit (in seconds) after which a timeout is declared.

Timer routine interval

Resolution of the transmit timer (the real timeout is limit + interval).

Time Stamps

Current uptime

Current system uptime.

Last reset

Last time the device was reset.

Last link up

Last time a link up transition occurred.

Last link down

Last time a link down transition occurred.

Total link uptime

Total time the link has been up.

Total link downtime

Total time the link has been down.

Last transmit timeout

Time of the last transmit timeout.

Last soft error

Time of the last soft error.

Last duplex mode mismatch reported

Time of the last duplex mode mismatch reported.

Previous CRC or late collision error

Time of the last CRC or late collision error recorded.

Receive Buffers

Minimum buffers requested

Minimum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current minimum limit".

Maximum buffers requested

Maximum number of receive buffers as set by default or by management request (SETMODE or LANCP command). This is used to determine the "Current maximum limit".

Current minimum limit

Minimum number of receive buffers as determined by the driver and by management request, "Minimum buffers requested". The driver will not allow the number of receive buffers in its receive queues to drop below this value. This does not include receive buffers transferred to applications (and not yet returned).

Current maximum limit

Maximum number of receive buffers as determined by the driver and by management request, "Maximum buffers requested". The driver will deallocate receive buffers until the number allocated does not exceed this maximum. When applications return receive buffers, the number may exceed this maximum, at which point they are deallocated. Note that the driver allows this maximum to be exceeded slightly to limit allocation and deallocation activity. See ["Target number of buffers maximum"](#) below.

Current number of buffers

Current number of receive buffers owned by the driver.

Target number of buffers

Desired number of receive buffers. As receive activity does not result in lost packets, this number is gradually decreased toward the "Current minimum limit". If packets are lost, this number is dramatically increased toward the "Current maximum limit". When the driver allocates receive buffers, it allocates them until reaching this target number.

Target number of buffers maximum

Maximum desired number of receive buffers. When an application returns a buffer to the driver, if the current number does not exceed this target maximum, the buffer is kept by the driver. Otherwise, it is deallocated.

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and chip, the driver records the time that each transmit buffer is given to the chip. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Statistics Block

CRC errors

Number of packets received with CRC errors that are an integral number of bytes long. These packets are discarded and are counted by the device.

Alignment errors

Number of packets received with CRC errors that are not an integral number of bytes long. These packets are discarded and are counted by the device.

Symbol errors

Number of symbol errors detected. These occur on fiber NICs only.

Receive errors

Number of packets received with error status.

Missed packets count

Number of packets missed because the receive FIFO had insufficient space to store incoming packets, either because too few buffers were provided by the driver or because there was insufficient bandwidth on the PCI bus.

Single collision count

Number of successfully transmitted packets that encountered exactly one collision during transmission (successful after retransmit). These occur in half-duplex mode only.

Excessive collision count

Number of transmits that failed because of excessive collisions. These occur in half-duplex mode only.

Multiple collision count

Number of successfully transmitted packets that encountered more than one collision during transmission (successful after multiple retransmits). These occur in half-duplex mode only.

Late collision count

Number of times a collision was detected later than 512 bit-times into the transmission of a packet. The transmit fails. These occur in half-duplex mode only.

Collision count

Total number of collisions that are not late collisions seen by the transmitter.

Defer count

Number of successful transmits that were delayed because the medium was busy. These occur in half-duplex mode only.

XMT – No CRS

Number of packet transmissions in which the carrier signal was lost shortly after the start of transmission. These occur in UTP, full-duplex mode only.

Sequence errors

Number of sequence errors detected where the frame delimiter symbols were seen in an illegal sequence. These occur on fiber NICs only.

Carrier extension errors

Number of packets received in which a carrier extension error was detected.

Receive length errors

Number of packets shorter than 64 bytes or longer than 16 384 bytes.

XON receive count

Number of valid XON packets received.

XON transmit count

Number of XON packets transmitted.

XOFF receive count

Number of valid XOFF packets received.

XOFF transmit count

Number of XOFF packets transmitted.

FC Receive Unsupported count

Number of unsupported flow control frames received.

Packets received (64 bytes)

Packets received (65–127 bytes)

Packets received (128–255 bytes)

Packets received (256–511 bytes)

Packets received (512–1023 bytes)

Packets received (1024... bytes)

Number of packets of this size received.

Good packets received

Number of good packets received of any legal length.

Broadcast packets received

Number of good broadcast packets received.

Multicast packets received

Number of good multicast packets received.

Good packets transmitted

Number of good packets transmitted.

Good octets received

Number of good bytes received.

Good octets transmitted

Number of good bytes transmitted.

No receive buffer count

Number of packets received where there were no available buffers in host memory. The packets are still received as long as there is sufficient space in the receive FIFO and the driver eventually provides more buffers.

Receive undersize count

Number of good packets received that were less than 64 bytes in length.

Receive fragment count

Number of packets received with a CRC error that were less than 64 bytes in length.

Receive oversize count

Number of good packets received that were larger than the maximum packet size.

Receive jabber count

Number of packets received with a CRC error that were larger than the maximum packet size.

Management packets received

Number of management packets received.

Management packets dropped

Number of management packets received but lost because the management receive FIFO was full or the packet was longer than 200 bytes.

Management packets transmitted

Number of management packets sent.

Total octets received

Total number of bytes received, regardless of errors or type of packet.

Total octets transmitted

Total number of bytes transmitted, regardless of errors or type of packet.

Total packets received

Total number of packets received, regardless of errors or type of packet.

Total packets transmitted

Total number of packets transmitted, regardless of errors or type of packet.

Packets transmitted (64 bytes)

Packets transmitted (65–127 bytes)

Packets transmitted (128–255 bytes)

Packets transmitted (256–511 bytes)

Packets transmitted (512–1023 bytes)

Packets transmitted (1024... bytes)

Number of packets of this size transmitted.

Multicast packets transmitted

Number of multicast packets transmitted.

Broadcast packets transmitted

Number of broadcast packets transmitted.

TCP segmentation xmt count

Number of TCP segmentation offload transmissions.

TCP segmentation xmt fail

Number of TCP segmentation offload transmissions that failed to transmit all of the data.

Driver Messages

Console messages issued by the driver.

8.2. Intel 1Gb Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, . . .))** LANCP command provides a mechanism to issue device-specific functions to the Intel 1Gb driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the [VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L \[https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590\]](https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590) or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the chip using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FILA", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILB", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILC", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILD", VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EIA/DEVICE_SPECIFIC=(FUNCTION="FIL1", VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EIA/DEVICE_SPECIFIC=(FUNCTION="FIL2", VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the chip, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="FXMT",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is limited to the range 2–40. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="DXMT",VALUE=*delayvalue*)

Change the interrupt mitigation delay on transmit completion value, the "Transmit interrupt delay (μ s)" value. This value is in microseconds, limited to 1000. If zero, the delay is disabled.

DEVICE_SPECIFIC=(FUNCTION="DRCV",VALUE=*delayvalue*)

Change the interrupt mitigation delay on receive completion value, the "Receive interrupt delay (μ s)" value. This value is in microseconds, limited to 1000. If zero, the delay is disabled.

DEVICE_SPECIFIC=(FUNCTION="DINT",VALUE=*delayvalue*)

Change the interrupt delay value. The value is in microseconds, limited to 1000. If zero, the delay is disabled.

**DEVICE_SPECIFIC=FUNCTION="DMIG"
(8254x and 82571 Devices Only)**

Disable interrupt mitigation. Note that the Transmit Interrupt Delay Value (TIDV) register is set to the minimum value of 1 (specifications state that it cannot be set to 0).

This is used for performance testing when it is desirable to get an interrupt immediately upon transmit and receive completion.

DEVICE_SPECIFIC=(FUNCTION="FLOW",VALUE=*value*)

Disable flow control (if *value* is zero) or enable flow control (if *value* is nonzero).

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=FUNCTION="FCRC"

Send the next packet without a valid CRC.

This is used to generate CRC errors for test purposes.

DEVICE_SPECIFIC=(FUNCTION="RBAD",VALUE=*value*)

If *value* is zero, the chip will discard bad received packets. If *value* is nonzero, the bad packets will be received and discarded by the driver. This allows the packets to be traced and inspected.

DEVICE_SPECIFIC=FUNCTION="CVME"**DEVICE_SPECIFIC=FUNCTION="SVME"**

IEEE VLAN Mode Enable (VME) bit set/clear. If clear, the VLAN tag will not be stripped from the packet so it can be seen in the receive buffer.

DEVICE_SPECIFIC=(FUNCTION="RCRC",VALUE=(*low32bits,high32bits*))

Set receive validation CRC check counter. It can be set from 0 to 0xFFFFFFFFFFFFFFFE. 0xFFFFFFFFFFFFFFF is used to indicate that the next CRC validation should fail, so we can verify what happens after a failure.

Some devices had a manufacturing issue that the driver detects by enabling CRC validation of the first 50 000 received packets. If a receive packet is declared good by the chip, but the driver CRC validation fails, the device is marked offline and unusable.

This allows the number of CRC validations to be changed.

DEVICE_SPECIFIC=FUNCTION="ARST"

Restart auto-negotiation.

This is used for link handling testing.

DEVICE_SPECIFIC=FUNCTION="FLSC"

Force a link state change interrupt.

This is used for link handling testing.

DEVICE_SPECIFIC=(FUNCTION="LSCI",VALUE=*value*)

If zero, disables link state change interrupts.

If nonzero, enables link state change interrupts.

This is used for link handling testing. Note that whether or not this function is disabled, the driver still does a periodic link state check. As such, it would detect a state change, albeit not immediately.

Chapter 9. Neterion XFRAME 10Gb Ethernet

The Neterion XFRAME devices are:

HP Part #	Media	Vendor ID	Device ID	SubSystem Vendor ID	SubSystem ID
AB287A	850nm	17D5	5831	12D5	103C
AD385A	850nm	17D5	5832	1337	103C
Other	850nm	17D5	5832	6020	Other

The driver is SYS\$EWXFRAME.EXE (IA-64).

9.1. XFRAME 10Gb Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS EWA** displays the entire set of internal counters maintained by the XFRAME 10Gb driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EIA**.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the XFRAME device.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EWA** command in the quadword driver version field.

Device interrupts

Number of times the driver interrupt service routine was called.

Link transitions

Number of link transitions seen by the driver.

Device resets

Number of times the device was reset.

First user initializations

Number of times a user started with no other users initialized (initialization requires more work than if the user is not the first user).

Unit inits (not using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were not used to map the device data structures.

Unit inits (using map registers)

Number of unit initializations executed; since unit initialization is only executed once, this counter will be 1 if map registers were used to map the device data structures.

User start/change/stop requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O with error status.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Transmit copy failures

Number of transmit failures needing a transmit copy, but memory could not be allocated for the transmit buffer.

Jumbo transmits issued

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

Transmits issued (using map registers)

Number of transmit requests that were described to the device using map registers because part of the request existed in memory outside the DMA window.

Jumbo receives issued

Number of jumbo receive buffers allocated and assigned to the device.

Receives issued (using map registers)

Number of receive buffers that were described to the device using map registers because part of the request existed in memory outside the DMA window, because the receive buffer crossed a page boundary, or because the receive buffer was a jumbo buffer.

Receives discarded (bad frame)

Number of receive packets discarded by the driver due to receive error. By default, the chip does not deliver damaged receive packets to the driver, but can be asked to do so by a device-specific function.

Rescheduled forks (too long in fork)

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Buffer size (bytes)

Size of the receive buffers. The device allows packets up to 1518 (or jumbo size) bytes (including header and CRC), plus the VMS receive buffer structure overhead (640 bytes).

Packet size (bytes)

This is the Ethernet packet size (1518 or jumbo frame size bytes).

Receive mapping boundary

Mapping value for receive frames, the maximum value of BOFF (Buffer Offset) that can result in the need for only two map registers. If BOFF is any larger, the buffer spills into a third page (displayed with the `/DEBUG` qualifier).

Current link state

Current link state.

Driver flags

Driver flags including the following bits:

- Promiscuous – Promiscuous mode enabled.
- All_Multicast – All multicast mode enabled.
- Jumbo_Frames – Jumbo packets enabled.
- Fiber – Fiber PHY found.
- 850nm – Multimode Fiber (MMF).
- 1300nm – Single-mode Fiber (SMF).
- PCI-X 266MHz capable – Device capable of operating at 266 MHz.
- Bugcheck_XTMO – Bugcheck on transmit timeout.
- Bugcheck_LongXmt – Bugcheck on a long transmit.
- Bugcheck_LongFork – Bugcheck on a long fork.
- All_Messages_Enabled – Display all messages.
- Receive_Bad_Frames – Receive bad frames enabled.
- AdapterEnableDone – Adapter initialization complete.
- LinkChangeIntEna – Link state change interrupts enabled.
- ForcePromiscuous – Promiscuous mode enabled even if not requested by a user.
- Disable_Interrupt_Mitigation – Interrupt mitigation disabled.
- PCI-X 266MHz actual – Device operating at 266 MHz.
- PCI 33MHz – Device operating at 33 MHz.
- PeriodicReset – Periodically reset the device.

Driver state

Current driver state, as follows:

- 0 – Driver state is undefined.
- 1 – Driver state is resetting.
- 2 – Driver is running, but the link is down. Completing transmits with error status.
- 4 – Driver is running, and the link is up. Processing transmits and receives.
- 8 – Device is not usable.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Interrupt Mitigation data

- Transmit traffic, % utilization / frame count, range 0 *percent, count*
- Transmit traffic, % utilization / frame count, range 1 *percent, count*

- Transmit traffic, % utilization / frame count, range 2 *percent, count*
- Transmit traffic, % utilization / frame count, range 3 *percent, count*
- Receive traffic, % utilization / frame count, range 0 *percent, count*
- Receive traffic, % utilization / frame count, range 1 *percent, count*
- Receive traffic, % utilization / frame count, range 2 *percent, count*
- Receive traffic, % utilization / frame count, range 3 *percent, count*

Miscellaneous driver data

Miscellaneous fields (displayed with the **/DEBUG** qualifier; intended for use by driver developers):

- Driver delay (μ s)
- Device quiesce time (μ s)
- Quiesce operations
- Performance monitor running
- Performance buffer S2 VA

LSB size

Size of the driver structure LAN Station Block (LSB) in bytes (displayed with the **/DEBUG** qualifier).

Transmit Rings, Receive Rings, Statistics addresses (debug)

Addresses of transmit, receive, and mailbox ring structures for the following (displayed with the **/DEBUG** qualifier):

- Map registers allocated
- Transmit time limit
- Timer routine interval
- Statistics virtual address (VA)
- Receive rings VA
- Transmit rings VA
- EEPROM data VA

Time Stamps

These fields provide timer information, the transmit timeout value, the periodic timer frequency, and time stamps for link transitions and errors.

- Current uptime
- Last reset

- Last link up
- Last link down
- Total link uptime
- Total link downtime
- Last transmit timeout
- Last soft error

PCI Config Registers (debug)

PCI config and Base Address Register (BAR) registers (displayed with the **/DEBUG** qualifier):

- PCI_ID_REG
- PCI_CMD_STATUS
- PCI_REVISION_ID
- PCI_CACHE_LINE_SIZE
- PCI_BAR0
- PCI_BAR1
- SUBSYSTEM_ID
- EXPANSION_ROM
- PCI_INT_LINE
- PCIX_COMMAND

DMA Window (debug)

DMA window address information (displayed with the **/DEBUG** qualifier):

- DMA window base address (PCI bus space)
- DMA window base address (memory space)
- DMA window base offset (PCI bus space)
- DMA window size
- DMA window usable size
- DMA window maximum physical address
- DMA window base PFN
- DMA window maximum PFN
- DMA monster window offset

Receive Buffers

Receive buffer information (displayed with the `/DEBUG` qualifier):

- Minimum buffers requested
- Maximum buffers requested
- Current minimum limit
- Current maximum limit
- Current buffers
- Target buffers
- Target buffers maximum

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the device are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the device are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and XFRAME, the driver records the time that each transmit buffer is given to the device. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Receive completion time

To help understand the operation of the system, driver, and XFRAME, the driver records the time that each receive buffer is transferred to an application. When the application returns, the driver calculates the elapsed time, creating a histogram of receive completion times from 10 to 150 milliseconds.

Statistics Block

Statistics – MAC Transmit

- 802.3: FramesTransmittedOk
- 802.3: OctetsTransmittedOk
- 802.3: FramesLostDueToIncMACXmitError
- 802.3: MulticastFramesXmittedOk
- 802.3: BroadcastFramesXmittedOk
- 802.3: PauseMacCtrlFramesTransmitted

- MIB-II: ifOutOctets
- MIB-II: ifOutUcastPkts
- MIB-II: ifOutNUcastPkts
- MIB-II: ifOutErrors
- RMON-2: etherStatsOctets
- XFRAMEII: Frames transmitted with VLAN tag

Statistics – MAC Receive

- 802.3: FramesReceivedOk
- 802.3: OctetsReceivedOk
- 802.3: FrameCheckSequenceErrors
- 802.3: FramesLostDueToIntMACRcvError
- 802.3: MulticastFramesReceivedOk
- 802.3: BroadcastFramesReceivedOk
- 802.3: InRangeLengthErrors
- 802.3: OutOfRangeLengthField
- 802.3: FrameTooLongErrors
- 802.3: PauseMACCtrlFramesReceived
- 802.3: UnsupportedOpCodesReceived, MACControlFramesReceived
- MIB-II: ifInOctets
- MIB-II: ifInUcastPkts
- MIB-II: ifInNUcastPkts
- MIB-II: ifInErrors
- RMON-2: etherStatsDropEvents
- RMON-2: etherStatsOctets
- RMON-2: etherStatsPkts
- RMON-2: etherStatsUndersizePkts
- RMON-2: etherStatsOversizePkts
- RMON-2: etherStatsFragments
- RMON-2: etherStatsJabbers

- RMON-2: etherStatsPkts64Octets
- RMON-2: etherStatsPkts65to127Octets
- RMON-2: etherStatsPkts128to255Octets
- RMON-2: etherStatsPkts256to511Octets
- RMON-2: etherStatsPkts512to1023Octets
- RMON-2: etherStatsPkts1024to1518Octets
- XFRAMEII: Received MAC frames with length 1519–4095
- XFRAMEII: Received MAC frames with length 4096–8191
- XFRAMEII: Received MAC frames with length 8191...
- XFRAMEII: Received MAC frames with length (including FCS)
- XFRAMEII: Received jumbo frames
- XFRAMEII: Received jumbo frames with bad FCS, i.e. jabbers
- XFRAMEII: Received jumbo frames longer than maximum
- XFRAMEII: Frames received with VLAN tag
- XFRAMEII: Received frames discarded because of invalid frame length
- XFRAMEII: Received frames discarded because of bad FCS
- XFRAMEII: Received pause frames discarded
- XFRAMEII: Received frames discarded because of invalid DA
- XFRAMEII: Received frames discarded because of WOL (Wake-on-LAN)
- XFRAMEII: Received frames discarded because target queue full
- XFRAMEII: Received frames discarded because of ingress mem full
- XFRAMEII: Received frames discarded because of RED (Receive Error Detected)
- XFRAMEII: Link faults
- Reserved
- 802.3: SymbolErrorDuringCarrier
- Frames that passed through queue 0 of receive buffer
- Times receive buffer queue 0 has filled up
- Pause quanta that the MAC has been in paused state
- Unexpected control characters during normal data transmission
- Unexpected or misplaced control characters outside of normal data transmission

Statistics – PCI/PCI-X Read Transaction

- Total read requests made by the device
- Requests made for a new read sequence
- Retry responses receive on the start of the new read sequences
- Retry responses received for read requests
- Read requests accepted by target after a write terminated with retry

Statistics – PCI/PCI-X Write Transaction

- Total write requests made by the device
- Requests made for a new write sequence
- Retry responses receive on the start of the new write sequences
- Retry responses received for write requests
- Target initiated disconnects on write transactions
- Write requests accepted by target after a read terminated with retry

Statistics – DMA Transaction

- Host write transactions to Tx Pointer FIFOs
- Transmit Descriptor (TxD) read requests
- Transmit Descriptor (TxD) write requests
- Receive Descriptor (RxD) read requests
- Receive Descriptor (RxD) write requests
- Transmit frame read requests
- Receive frame read requests

Debug Data (device register contents)

Device register contents, either 32 or 64-bits (displayed with the **/DEBUG** qualifier):

- 0004 PCI_CMD_STATUS
- 0062 PCI-X_COMMAND
- 0064 PCI-X_STATUS
- GENERAL_INT_STATUS
- GENERAL_INT_MASK
- ADAPTER_STATUS
- ADAPTER_CONTROL

- SERR_SOURCE
- PIC_INT_STATUS
- TXPIC_INT_REG
- RXPIC_INT_REG
- FLSH_INT_REG
- MDIO_INT_REG
- IIC_INT_REG
- GPIO_INT_REG
- PIC_CONTROL
- SWAPPER_CTRL
- TXDMA_INT_STATUS
- PFC_ERR_REG
- TDA_ERR_REG
- PCC_ERR_REG
- TTI_ERR_REG
- LSO_ERR_REG
- TPA_ERR_REG
- SM_ERR_REG
- RXDMA_INT_STATUS
- RDA_ERR_REG
- RC_ERR_REG
- PRC_PCIX_ERR_REG
- RPA_ERR_REG
- RTI_ERR_REG
- MAC_INT_STATUS
- MAC_TMAC_ERR_REG
- MAC_RMAC_ERR_REG
- MAC_CFG
- RMAC_ERR_CFG
- MC_INT_STATUS

- MC_ERR_REG
- XGXS_INT_STATUS
- XGXS_TXGXS_ERR_REG
- XGXS_RXGXS_ERR_REG
- XGXS_CFG
- XGXS_STATUS
- PCI_MODE
- RCSR (Receive Control and Status Register) contents
- ISR (Interrupt Status Register) undefined state

Driver Messages

Console messages issued by the driver.

9.2. XFRAME 10Gb Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC",VALUE=(V1,V2,...))** LANCP command provides a mechanism to issue device-specific functions to the XFRAME 10Gb driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A-L* [<https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590>] or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="MAPR"

Force all transmit and receive buffers to be described to the chip using map registers.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="RCOR",VALUE=*pty*)

DEVICE_SPECIFIC=(FUNCTION="XCOR",VALUE=*pty*)

Corrupt a receive or transmit packet. When this function is completed, the protocol type is specified, and if it matches on a transmit or receive packet, the next packet seen is corrupted by changing the last bit in the VCRP or CXB, XORing it with a 1. The protocol type is then cleared so the corruption is not done again. The corruption is only checked when transmit or receive packet data tracing is being done (in the trace routine), so there is no performance hit for checking each packet to verify if it should be corrupted.

The following example shows how to corrupt a PEDRIVER transmit packet:

```
$ MCR LANCP SET DEVICE/TRACE EWA
$ MCR LANCP SET DEVICE/DEVICE_SPECIFIC=(FUNCTION="XCOR", VALUE=%x760) EWA
```

DEVICE_SPECIFIC=(FUNCTION="RBD",VALUE=*value*)

If *value* is zero, the chip will discard bad received packets. If *value* is nonzero, the bad packets will be received and discarded by the driver. This allows the packets to be traced and inspected.

DEVICE_SPECIFIC=(FUNCTION="FLOW",VALUE=*value*)

Disable flow control (if *value* is zero) or enable flow control (if *value* is nonzero).

DEVICE_SPECIFIC=(FUNCTION="XTMO",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is not range checked. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=(FUNCTION="FILA",VALUE=(*32bits,16bits*))

DEVICE_SPECIFIC=(FUNCTION="FILB",VALUE=(*32bits,16bits*))

DEVICE_SPECIFIC=(FUNCTION="FILC",VALUE=(*32bits,16bits*))

DEVICE_SPECIFIC=(FUNCTION="FILD",VALUE=(*32bits,16bits*))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL1", VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EWA/DEVICE_SPECIFIC=(FUNCTION="FIL2", VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=(FUNCTION="ALLM",VALUE=*setting*)

Change the setting of the driver flag "all messages". If the "all messages" flag is set the driver displays all console messages. The default setting is bit 31 of the LAN_FLAGS system parameter. This function allows the "all messages" setting to be set for an individual device rather than all LAN devices on the system.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=FUNCTION="STOP"

Stop the chip, thereby inducing a transmit timeout on the next transmit request.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="XDEL",VALUE=*timevalue*)

Delay issuing the next transmit request for the time specified in 10-millisecond units. A maximum of 2000 (20 seconds) is allowed.

This function is useful for investigating application behavior when a transmit request is delayed.

DEVICE_SPECIFIC=(FUNCTION="XTOB",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a transmit timeout occurs. If 0 (disable), transmit timeout results in a device reset.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="XLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long transmit occurs. If 0 (disable), a long transmit completion is counted.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="FLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long fork occurs. If 0 (disable), a long time to run a fork process is recorded in the counters.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="LOOP",VALUE=*loopvalue*)

Specify *loopvalue* to enable (1) or disable (0) loopback mode.

This is useful in test scenarios to run an application in loopback mode, where for standard use the application would not allow selection of loopback mode.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=FUNCTION="OMSG"

Console messages are displayed by the driver during initialization, auto-negotiation, and link handling. However, the driver limits the total number of messages displayed. This function resets the count, allowing additional messages to be displayed.

DEVICE_SPECIFIC=(FUNCTION="WCSR",VALUE=(*registeraddress,value*))

Write the device register specified with the *value* given.

DEVICE_SPECIFIC=(FUNCTION="RCSR",VALUE=(*registeraddress*))

Read the device register specified. The value is returned in the internal counters.

DEVICE_SPECIFIC=(FUNCTION="TTIN",VALUE=(*range,utilization,fcvalue*))

DEVICE_SPECIFIC=(FUNCTION="RTIN",VALUE=(*range,utilization,fcvalue*))

Sets the transmit and receive traffic interrupt values, which are in three ranges (0, 1, 2) where the line rate utilization and the frame count values specify when transmit interrupts should be generated.

DEVICE_SPECIFIC=(FUNCTION="RVLA",VALUE=*setting*)

If clear (0), do not strip the VLAN tag from the packet, so it can be seen in the receive buffer. If set (1), strip the tag from the packet. The default is 1.

Chapter 10. Broadcom BCM57711 10Gb Ethernet

The Broadcom BCM57711 10Gb devices are:

Name	Part	Phy	Interface
Krypton	NC532m	BCM8073	10GBE-KR
Kenny	AM225A	BCM8726	10GBE-SR
Kenny	AM232A	BCM8726	10GBE-LR
Kenny	AM233A	BCM8726	10GBE-CU
Kauai LOM	LOM	BCM8073	10GBE-KR
Buccaneer LOM	LOM	BCM8073	10GBE-KR

The driver is `SYS$EW57711.EXE`, `SYS$ER57711.EXE` (IA-64).

10.1. Broadcom BCM57711 10Gb Internal Counters

The LANCP command `SHOW DEVICE/INTERNAL_COUNTERS EWA` displays the entire set of internal counters maintained by the BCM57711 driver. Some counters are special debug counters. These are not displayed unless the additional qualifier `/DEBUG` is specified. Counters that are zero are not displayed unless the additional qualifier `/ZERO` is specified.

The LAN SDA extension also displays the complete set of internal counters with the command `LAN INTERNAL/DEVICE=EWA`.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the Broadcom BCM57711 10Gb device.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the `ANALYZE/IMAGE` command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA `LAN/DEVICE=EWA` command in the quadword driver version field.

Device interrupts

Number of times the driver interrupt service routine was called.

Spurious interrupts

Number of times the driver interrupt service routine was called when the driver was not running (for example, in the "undefined" state).

Invalid interrupts

Number of times the driver interrupt service routine was called but the interrupt was not valid.

Link transitions

Number of link transitions seen by the driver.

Device resets

Number of times the device was reset.

Driver initializations

Number of times the driver went through its initialization procedure.

User start/change/stop requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O with error status.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Transmit copy failures

Number of transmit failures needing a transmit copy, but memory could not be allocated for the transmit buffer.

Jumbo transmits issued

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

Jumbo receives issued

Number of jumbo receive buffers allocated and assigned to the device.

Receives discarded (bad frame)

Number of receive packets discarded by the driver due to receive error. By default, the chip does not deliver damaged receive packets to the driver, but can be asked to do so by a device-specific function.

Rescheduled forks (too long in fork)

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Current Link state

Current link state.

Requested Link state

Current link state requested.

Driver flags

Driver flags including the following bits:

- Promiscuous – Promiscuous mode enabled.
- All_Multicast – All multicast mode enabled.
- Jumbo_Frames – Jumbo packets enabled.
- RCV_Fork – A fork to process receive completion is requested.
- XMT_Fork – A fork to process transmit completion is requested.
- Bugcheck_XTMO – Bugcheck on transmit timeout.
- Bugcheck_LongXmt – Bugcheck on a long transmit.
- Bugcheck_LongFork – Bugcheck on a long fork.
- PeriodicReset – Periodically reset the device.

Driver state

Current driver state, as follows:

- 0 – Driver state is undefined.
- 1 – Driver state is resetting.
- 2 – Driver is running, but the link is down. Completing transmits with error status.
- 4 – Driver is running, and the link is up. Processing transmits and receives.
- 8 – Device is not usable.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Miscellaneous fields

Miscellaneous fields (many of which are displayed with the **/DEBUG** qualifier):

- Interrupt Status
- Port Number
- Function Number
- XRING VA
- XRING_EXT VA
- RRING VA
- RRING_EXT VA
- RRING_COMPL VA
- Slow Path Ctx VA
- Slow Path Ring VA
- Status Block VA
- Default Status Block VA
- Doorbell VA
- Xmt count
- Xmt packet index
- XMT compl. sw index
- XMT compl. hw index
- RXD (Receive Descriptor) hw put index
- RCQE (Receive Completion Queue Entry) hw put index
- RCQE hw get index

- RCQE sw get index
- CSTORM index
- USTORM index
- Client id

Time Stamps

These fields provide timer information, the transmit timeout value, the periodic timer frequency, and time stamps for link transitions and errors.

- Transmit time limit
- Timer routine interval
- Last link up
- Last link down
- Total link uptime
- Total link downtime
- Last transmit timeout
- Last soft error

PCI Config Registers (debug)

PCI config and BAR registers (displayed with the **/DEBUG** qualifier):

- PCI_ID_REG
- PCI_CMD_STATUS
- PCI_REVISION_ID
- PCI_LATENCY_TIMER
- PCI_BAR0
- PCI_BAR2
- BAR0 Size
- BAR2 Size
- SUBSYSTEM_ID
- PCI_CACHE_LINE_SIZE

DMA Window (debug)

DMA window address information (displayed with the **/DEBUG** qualifier):

- DMA window base address (PCI bus space)

- DMA window base address (memory space)
- DMA window base offset (PCI bus space)
- DMA window size
- DMA window usable size
- DMA window maximum physical address
- DMA window base PFN
- DMA window maximum PFN
- DMA monster window offset

Receive Buffers

Receive buffer information:

- Minimum buffers requested
- Maximum buffers requested
- Current minimum limit
- Current maximum limit
- Current buffers
- Target buffers
- Target buffers maximum

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the BCM57711 chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the BCM57711 chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and BCM57711 chip, the driver records the time that each transmit buffer is given to the device. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Receive completion time

To help understand the operation of the system, driver, and BCM57711 chip, the driver records the time that each receive buffer is transferred to an application. When the application returns, the

driver calculates the elapsed time, creating a histogram of receive completion times from 10 to 150 milliseconds.

LSB extension fields

- Maximum packet size
- Port Number
- Function Number
- RCV buffer size
- Chip revision
- Link state
- IfAdmin state
- IfOper state
- Link up wait ctr
- Validity map
- ISCP/LSB VA (VA of device context)
- Slowpath VA
- Slowpath PA
- Slowpath interrupts
- Spurious Slowpath interrupts
- Link interrupts
- Link bounces
- Shmem base address
- Shmem2 base address
- Firmware sequence number
- Shared configuration data
- External link up
- Link status register
- Hardware config.
- Hardware config2.
- Bootcode revision
- Board type

- Lane config.
- External PHY config.
- External PHY rev.
- External ROM2 ver.
- 10gig mode
- Firmware return code
- Firmware state
- PHY flags
- PHY address
- SER lane
- Link config.
- Serdes config.
- Speed cap mask
- Default SBLK VA
- Default SBLK PA
- RX lane swap
- TX lane swap
- Autoneg
- Requested Autoneg
- Medium type
- SW Config.
- Speed
- Requested speed mask
- Speed mask
- Duplex
- Requested Duplex
- Advertising
- Invalid speed
- MIN_BW 0

- MIN_BW 1
- MIN_BW 2
- MIN_BW 3
- MAX_BW 0
- MAX_BW 1
- MAX_BW 2
- MAX_BW 3
- AEU mask
- SPQ max
- SPQ put idx
- SPQ get idx
- SPQ PA
- SPQ VA
- SPQ avail
- ATTN idx
- CST idx
- UST idx
- XST idx
- TST idx
- ATTN state
- Supported Loopback mode
- Requested flow control
- Flow control
- Outer VLAN
- Multi-function mode
- Multi-function config.
- Primary function flag
- IEEE fc
- Function link status

- DCC (Dynamic Control Channel) driver status
- DCC ifadmin state
- Last DCC event
- DCC events
- DCC enable events
- DCC disable event
- DCC adjust bw events
- DCC events failures
- Gunzip PA
- Gunzip VA
- DMAE (Direct Memory Access Engine) ready flag
- Port stx
- Function stx
- FW pulse write seq
- Stats state
- Stats update
- Executer idx
- Stats pending
- Stats counter
- Multimode
- WSUM
- RX mode
- Cap msix offset
- Vectbl offset
- MSI-X count
- GRC (Global Register Controller) dump buffer VA
- GRC dump buffer size
- Fatal init
- MII (Media Independent Interface) control
- Link chk wrkaround

- Link reset wrkaround
- Anal_trigger
- VN_WSUM

Driver Messages

Console messages issued by the driver.

10.2. Broadcom BCM57711 10Gb Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, ...))** LANCP command provides a mechanism to issue device-specific functions to the BCM57711 10Gb driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the *VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L* [<https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590>] or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FILA", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILB", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILC", VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILD", VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EBA/DEVICE_SPECIFIC=(FUNCTION="FIL1", VALUE=(%x000400aa,%x4fef))
```

```
$ MCR LANCX SET DEVICE EBA/DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="XTOB",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a transmit timeout occurs. If 0 (disable), transmit timeout results in a device reset.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="XLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long transmit occurs. If 0 (disable), a long transmit completion is counted.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="FLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long fork occurs. If 0 (disable), a long time to run a fork process is recorded in the counters.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=FUNCTION="OMSG"

Console messages are displayed by the driver during initialization, auto-negotiation, and link handling. However, the driver limits the total number of messages displayed. This function resets the count, allowing additional messages to be displayed.

DEVICE_SPECIFIC=(FUNCTION="XTMO",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is not range checked. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="XDEL",VALUE=*timevalue*)

Delay issuing the next transmit request for the time specified in 10-millisecond units. A maximum of 2000 (20 seconds) is allowed.

This function is useful for investigating application behavior when a transmit request is delayed.

Chapter 11. Emulex BE3 10Gb Ethernet

The Emulex BE 10Gb devices are:

Table 11.1. Emulex BE3 10Gb Ethernet Chip Designations

Device	Form Factor	Chip	Function	PCI IDs	Media
NC550	Standup	BE2	NIC	19A2 0700 103C 174B	SR LR LRM DAC
NC550SFP	Standup	BE2	NIC	19A2 0700 103C 1747	SR LR LRM DAC
NC550m	Mezzanine	BE2	NIC	19A2 0700 103C 1746	KR
NC551m	Mezzanine	BE2	NIC	19A2 0700 103C 174A	KR
NC551i	LOM	BE2	NIC	19A2 0702 103C 3314	KR
NC552	Standup	BE3	NIC	19A2 0710 103C 3344	SR LR LRM DAC
AT111A	Standup	BE3	NIC	19A2 0710 103C 17A3	SR LR LRM DAC
NC552SFP	Standup	BE3	NIC	19A2 0710 103C 3340	SR LR LRM DAC
AT118A	Standup	BE3	NIC	19A2 0710 103C 17A6	SR LR LRM DAC
NC552m	Mezzanine	BE3	NIC	19A2 0710 103C 3341	KR
NC553m	Mezzanine	BE3	NIC	19A2 0710 103C 3345	KR
AT093A	Standup	BE3	NIC	19A2 0710 103C 3391	SR LR LRM DAC
AT094A	Standup	BE3	NIC	19A2 0710 103C 3392	SR LR LRM DAC
NC553i	LOM	BE3	NIC	19A2 0710 103C 177B	KR

Table 11.2. Emulex BE3 10Gb Ethernet PCI Device IDs

Device	NIC	FC	Marketing Description
NC550	Y	Y	HP StorageWorks NC550 Dual Port Converged Network Adapter
NC550SFP	Y	N	HP NC550SFP Dual Port 10GbE Server Adapter
NC550m	Y	N	HP NC550m Dual Port Flex-10 10GbE BL-c Adapter
NC551m	Y	Y	HP NC551m Dual Port FlexFabric 10Gb Converged Network Adapter
NC551i	Y	Y	HP NC551i Dual Port FlexFabric 10Gb Converged Network Adapter
NC552	Y	N	HP NC552 2p 10GbE PCIe Adapter
AT111A	Y	Y	HP Integrity CN1100E Dual Port Converged Network Adapter
NC552SFP	Y	N	HP NC552SFP 10Gb 2-port Flex-10 Server Adapter
AT118A	Y	N	HP Integrity NC552SFP 2P 10GbE Adapter
NC552m	Y	N	HP NC552m 10Gb 2-port Flex-10 Server Adapter

Device	NIC	FC	Marketing Description
NC553m	Y	Y	HP NC553m 10Gb 2-P Flex-10 Server Adapter
AT093A	Y	Y	HP AT093A PCIe 1P 8Gb FC and 1P 1/10GbE Adapter
AT094A	Y	Y	HP AT094A PCIe 2P 8Gb FC and 2P 1/10GbE Adapter
NC553i	Y	Y	HP Dual Port FlexFabric 10Gb BL8X0c i4 Embedded CNIC

The driver is SYS\$EWBE3.EXE (IA-64).

11.1. Emulex BE3 10Gb Internal Counters

The LANCP command **SHOW DEVICE/INTERNAL_COUNTERS EBA** displays the entire set of internal counters maintained by the Emulex BE3 driver. Some counters are special debug counters. These are not displayed unless the additional qualifier **/DEBUG** is specified. Counters that are zero are not displayed unless the additional qualifier **/ZERO** is specified.

The LAN SDA extension also displays the complete set of internal counters with the command **LAN INTERNAL/DEVICE=EBA**.

Note

The definitions of the following counters may change between versions of the driver.

Counters

Device name

Name of the Emulex 10Gb device.

Description

Full name of the device returned by the device firmware.

Driver timestamp

Compilation timestamp of the driver.

Driver version

Driver version numbered 1–*n* that is usually identical to the *x-n* ID displayed in the **ANALYZE/IMAGE** command output for the driver image. It includes variant information, if applicable. The full driver version includes the target OpenVMS release and is displayed by the SDA **LAN/DEVICE=EBA** command in the quadword driver version field.

Device interrupts

Number of times the driver interrupt service routine was called.

Spurious interrupts

Number of times the driver interrupt service routine was called when the driver was not running (for example, in the "undefined" state).

Link transitions

Number of link transitions seen by the driver.

Device resets

Number of times the device was reset.

PTE to PFN translations

Number of times a global page was encountered during a chained transmit request, causing the driver to convert an invalid PTE to a valid PFN.

Transmits queued

Number of transmit requests queued because the link was not available or because too many transmit requests were already outstanding.

Transmit timeouts

Number of times the driver has timed out a transmit, reset the device, and completed outstanding I/O with error status.

Transmit errors (too few segments)

Number of transmit requests completed with error status (SS\$_INCSEGTRA) because the application did not specify the transmit buffer completely.

Transmit copies (too many segments)

Number of transmit requests that exceeded the maximum number of chain segments that the driver can handle; the driver then copied some of them to a temporary buffer so that it could transmit the packet.

Transmit copy failures

Number of transmit failures needing a transmit copy, but memory could not be allocated for the transmit buffer.

Jumbo transmits

Number of transmit requests with a packet length exceeding 1514 bytes (excluding CRC).

CKO TCP transmits

Number of TCP transmit requests specifying checksum offload (CKO).

CKO UDP transmits

Number of UDP transmit requests specifying CKO.

CKO IP transmits

Number of IP transmit requests specifying CKO.

TSO transmits

Number of TCP transmit requests specifying TCP segmentation offload (TSO).

Jumbo receives

Number of jumbo receive buffers allocated and assigned to the device.

Receives discarded (bad frame)

Number of receive packets discarded by the driver due to receive error. By default, the chip does not deliver damaged receive packets to the driver, but can be asked to do so by a device-specific function.

CKO receives

Number of receive packets processed using hardware validated checksums.

LRO receives

Number of receive packets processed using driver large receive offload (LRO) functionality, transferred to a user concatenated to reduce protocol overhead.

Rescheduled forks (too long in fork)

Number of times that a rescheduled fork was done. In transmit and receive processing, the driver limits the amount of time spent in the fork process before rescheduling.

User start/change/stop requests

Number of user startup and shutdown requests processed by the driver; this is typically one or two requests when a user starts up and one when a user stops.

Soft errors

Number of times errors were recovered in the driver by resetting and reinitializing the device without notifying user applications.

Transmit coalesce value (μ s)

Transmit interrupts are generated every n "coalesce value" transmit completions, but no later than n "interrupt delay" microseconds after transmit completion.

Receive coalesce value (μ s)

Receive interrupts are generated every n "coalesce value" receive completions, but no more than n "interrupt delay" microseconds after receipt of a packet.

Link state

Current link state.

Requested Link state

Current link state requested.

Driver flags

Driver flags including the following bits:

- Promiscuous – Promiscuous mode enabled.

- All_Multicast – All multicast mode enabled.
- Jumbo_Frames – Jumbo packets enabled.
- Bugcheck_XTMO – Bugcheck on transmit timeout.
- Bugcheck_LongXmt – Bugcheck on a long transmit.
- Bugcheck_LongFork – Bugcheck on a long fork.
- PeriodicReset – Periodically reset the device.
- DebugQnQ – Force QnQ¹ driver workaround on every transmit.

Driver state

Current driver state, as follows:

- 0 – Driver state is undefined.
- 1 – Driver state is resetting.
- 2 – Driver is running, but the link is down. Completing transmits with error status.
- 4 – Driver is running, and the link is up. Processing transmits and receives.
- 8 – Device is not usable.

LAN_FLAGS system parameter

Current value of the LAN_FLAGS system parameter.

Port Number

Port number for this device; can be 0 or 1.

Function Number

Function number for this device; can be 0, 1, 2, or 3 depending on device configuration.

LSB size

Size of the driver structure LSB in bytes.

Time Stamps

These fields provide timer information, the transmit timeout value, the periodic timer frequency, and time stamps for link transitions and errors.

- Transmit time limit
- Timer routine interval
- Last link up
- Last link down
- Total link uptime

¹"QnQ" in this document refers to what Emulex calls the *QnQ workaround*. There is a bug in the BE3 chip that the driver works around, and the driver uses the Emulex name QnQ when describing the workaround. This is distinct from "QinQ", which refers to VLAN stacking and applies to Blade environments where there is an inner and outer VLAN tag (double or stacked tags).

- Total link downtime
- Last transmit timeout
- Last soft error

TX, RX, MQ, queues, rings, BMBX (debug)

Addresses of transmit, receive, and mailbox ring structures for the following (displayed with the **/DEBUG** qualifier):

- XRING
- XRING_EXT
- RRING_EXT
- TXQ
- TXCQ
- TXEQ
- TXQ_PA
- TXCQ_PA
- TXEQ_PA
- XRING_EXT_PA
- RRING
- RXQ
- RXCQ
- RXEQ
- RXQ_PA
- RXCQ_PA
- RXEQ_PA
- MRING
- MBQ
- MBQ_PA
- MBCQ_PA
- BMBOX
- BMBOX_PA
- MBCQ

PCI Config Registers (debug)

PCI config and BAR registers (displayed with the **/DEBUG** qualifier):

- PCI_CFID
- PCI_CFCR
- PCI_CFRV
- PCI_CFLT
- PCI_BAR0
- PCI_BAR2
- PCI_BAR4
- BAR0 size
- BAR2 size
- BAR4 size
- SLI_INTF
- SUBSYSTEM_ID
- PCI_CACHE_LINE_SIZE

DMA Window (debug)

DMA window address information (displayed with the **/DEBUG** qualifier):

- DMA window base address (PCI bus space)
- DMA window base address (memory space)
- DMA window base offset (PCI bus space)
- DMA window size
- DMA window usable size
- DMA window maximum physical address
- DMA window base PFN
- DMA window maximum PFN
- DMA monster window offset

Receive Buffers

Receive buffer information:

- Minimum buffers requested

- Maximum buffers requested
- Current minimum limit
- Current maximum limit
- Current buffers
- Target buffers
- Target buffers maximum

Fork Delay (after scheduled)

To help determine whether the buffering requirements of the driver and the BE3 chip are sufficient for the system configuration, the driver records the amount of time from fork scheduled to the time the fork is actually run. The data is recorded in 10-millisecond increments from 10 to 310 milliseconds.

This data can be used in conjunction with the number of packets discarded due to insufficient buffers to determine whether the buffering settings of the driver (minimum and maximum receive buffers) and the amount of buffering on the BE3 chip are sufficient for normal operation. If packets are being discarded, the buffering should be increased until the number of packets lost is minimal.

Transmit Time

To help understand the operation of the system, driver, and BE3 chip, the driver records the time that each transmit buffer is given to the device. When the transmit completes, the driver calculates the elapsed time, creating a histogram of transmit times from 10 to 310 milliseconds.

Receive completion time

To help understand the operation of the system, driver, and BE3 chip, the driver records the time that each receive buffer is transferred to an application. When the application returns, the driver calculates the elapsed time, creating a histogram of receive completion times from 10 to 150 milliseconds.

Firmware information

Device firmware is initiated at power up and reset, getting to a firmware state where it is ready to accept commands. Commands are written to a device mailbox called the BMBX (boot mailbox) and other structures such as the CQ (Command Queue), CQE (Command Queue Entry), MQ (Mailbox Queue), and MQE (Mailbox Queue Entry).

These counters record the firmware version and timing of firmware ready and BMBX usage information. Some of these counters are only displayed if the **/DEBUG** qualifier is included.

- Firmware version
- Time of last wait for fw ready
- Elapsed time for fw ready wait (last)
- Elapsed time for fw ready wait (max)
- Start time of fw ready wait (max)

- func_wait_cnt (Monitors initialization time)
- BMBX_OWNER
- BMBX_LAST_OWNER
- CMD_UNSUP_STATE_CNT
- Last BMBX command opcode
- Last BMBX command time
- Last BMBX command status
- BMBX command timeouts, cleared on reset
- Max BMBX command timeouts
- BMBX commands issued
- BMBX not initially ready
- Maximum time to allow for BMBX commands
- BMBX command timeouts, not cleared on reset
- CQE_INVALID
- CMD_BAD_STATUS
- CQE_BAD_STATUS
- CQE_UNSUPPORTED_ATTRIB
- MQE_UNAVAILABLE
- MQ_REQUESTS
- MQ_NOT_READY
- MQ_BLOCKED
- Last time MQE not available when needed
- Last time MQ blocked
- Last time MQ GET_STATS command issued
- Last time MQ GET_LINK_STATUS command issued

Link information

These fields give more link information (some fields displayed with the **/DEBUG** qualifier):

- Link state
- PHY link state

- Last link check
- Last link info value
- Last QoS info value
- DCC bandwidth events
- Link transitions
- PHY link transitions
- Last bandwidth event

Card PORT0 Statistics

Card PORT1 Statistics

Counters for port 0, 1 of this device.

- rx bytes
- rx_ipv4 bytes
- rx_ipv6 bytes
- tx bytes
- rx_total packets
- rx_unicast packets
- rx_multicast packets
- rx_broadcast packets
- rx_crc_errors
- rx_alignment_symbol_errors
- rx_pause packets
- rx_priority_pause packets
- rx_control packets
- rx_in_range_errors
- rx_out_range_errors
- rx_frame_too_long
- rx_address_match_errors
- rx_dropped_too_small
- rx_dropped_too_short
- rx_dropped_header_too_small

- rx_dropped_tcp_length
- rx_dropped_runts
- rx_64 byte packets
- rx_65_127 byte packets
- rx_128_256 byte packets
- rx_256_511 byte packets
- rx_512_1023 byte packets
- rx_1024_1518 byte packets
- rx_1519_2047 byte packets
- rx_2048_4095 byte packets
- rx_4096_8191 byte packets
- rx_8192_9216 byte packets
- rx_ip_checksum_errs
- rx_tcp_checksum_errs
- rx_udp_checksum_errs
- rx_non_rss packets
- rx_ipv4 packets
- rx_ipv6 packets
- rx_chute1 packets
- rx_chute2 packets
- rx_chute3 packets
- rx_mgmt packets
- rx_switched_unicast packets
- rx_switched_multicast packets
- rx_switched_broadcast packets
- tx_unicast packets
- tx_multicast packets
- tx_broadcast packets
- tx_pause packets

- tx_priority_pause packets
- tx_control packets
- tx_64 byte packets
- tx_65_127 byte packets
- tx_128_256 byte packets
- tx_256_511 byte packets
- tx_512_1023 byte packets
- tx_1024_1518 byte packets
- tx_1519_2047 byte packets
- tx_2048_4095 byte packets
- tx_4096_8191 byte packets
- tx_8192_9216 byte packets
- rx_fifo_overflow
- rx_input_fifo_overflow
- pmem_fifo_overflow
- jabber_event
- rsvd1
- rsvd2
- rsvd3

Card Error Statistics

A number of statistics include errors detected by the device.

- mgmt_rx_port bytes
- mgmt_tx_port bytes
- rx_drops_no_pbuf
- rx_drops_no_txpb
- rx_drops_no_ern_desc
- rx_drops_no_tpre_desc
- mgmt_rx_port packets
- mgmt_rx_port_pause packets

- mgmt_rx_port_errors
- mgmt_tx_port packets
- mgmt_tx_port_pause packets
- mgmt_rx_port_pmem_fifo_oflo
- rx_drops_too_many_frags
- rx_drops_invalid_ring
- forwarded packets
- rx_drops_mtu
- mgmt_rx_port_rxpp_fifo_oflo
- mgmt_rxpp_drops_too_big

LSB Extension Miscellaneous

- Port number (attrib addl command)
- ASIC (Application-Specific Integrated Circuit) revision
- RSS (Receive Side Scaling) capable
- Function mode
- Function capabilities
- HW interface ID
- Unicast MAC ID
- Serial number
- Requested offload setting
- Currently rx queue
- On die temperature
- Last temperature time
- UEFI (Unified Extensible Firmware Interface) NIC version
- Ext PHY version
- PHY type
- PHY interface type
- PHY misc params
- PHY ext PHY details

- PHY auto speeds
- PHY fixed speeds
- SFP (Small Form-factor Pluggable) vendor name
- SFP vendor PN (part number)
- SFP vendor revision

Flow Control settings

- Requested tx flow control setting
- Requested rx flow control setting
- Actual tx flow control setting
- Actual rx flow control setting

QnQ workaround information

The QnQ issue is similar to UERR, but it is a hardware condition rather than a firmware-detected fault. The effect is the same. When device firmware supports a driver workaround, the driver can avoid the failure condition by detecting the condition and specifying the transmit request in such a way that the firmware recognizes the condition and can work around it.

- QnQ tag enabled in HW
- QnQ firmware present
- QnQ VLAN tag
- QnQ VLAN tag and QnQ firmware present
- QnQ VLAN tag valid time
- Dummy WQEs (Work Queue Entries) used
- QnQ workaround transmits
- QnQ workarounds done (IPv4)
- QnQ workarounds done (IPv6)
- Last QnQ workaround (IPv4)
- Last QnQ workaround (IPv6)

Miscellaneous counters and timestamps

- MQ interrupts
- MQ interrupts (sync)
- MQ interrupts (async)

- MQ interrupts (unrecognized)
- MQ failed commands
- Last MQ failed command
- Link state change MQ async events
- Last Link state change MQ async event
- DCC BW MQ async events
- Last DCC BW MQ async event
- QnQ VLAN tag MQ async events
- Last QnQ VLAN tag MQ async event
- Other MQ async events
- Last Other MQ async event
- Last reset start time
- Last reset end time
- Last reset elapsed time
- Max reset elapsed time
- Last max reset elapsed time
- Previous reset start time
- Previous reset end time
- Last cmd timeout
- Last CQE invalid
- Last cmd bad status
- Last CQE bad status
- Last CQE unsupported

UERR information

Unrecoverable error (UERR) is a fatal firmware condition, where the firmware encounters an error that it cannot proceed from. When a UERR occurs, the firmware stops and cannot be restarted. The UERR status registers (lo and hi) provide more detail. The UERR mask register indicates which UERR status bits are valid.

- Last UERR status (lo)
- Last UERR status (hi)
- Last UERR mask (lo)

- Last UERR mask (hi)

BMBX Statistics (debug)

Boot Mailbox (BMBX) commands are issued in three steps; pre-command (Part 1), write high address (Part 2), and write low address (Part 3). This statistics track the timing for each part.

- BMBX max time
- BMBX max when
- BMBX last time
- BMBX last when
- BMBX accum time

Cmd Statistics (debug)

This table lists each command that can be issued to the device, the number of commands issued, and timing information for each (displayed with the **/DEBUG** qualifier):

- MAC_QUERY
- MAC_SET
- MULTICAST_SET
- VLAN_CONFIG
- LINK_QUERY
- READ_FLASH
- WRITE_FLASH
- CQ_CREATE
- EQ_CREATE
- MCC_CREATE
- READ_EEPROM
- GET_CNTL_ATTRIB
- RX_FILTER
- GET_FW_VERSION
- SET_FLOW_CONTROL
- GET_FLOW_CONTROL
- SET_FRAME_SIZE
- MODIFY_EQ_DELAY

- FW_CONFIG
- INTERFACE_CREATE
- INTERFACE_DESTROY
- MCC_DESTROY
- CQ_DESTROY
- EQ_DESTROY
- SET_LINK_SPEED
- QUERY_FW_CONFIG
- PMAC_ADD
- PMAC_DEL
- FUNC_RESET
- READ_TRANS_DATA
- CREATE_MQ_EXT
- GET_PHY_CONFIG
- GET_ADDL_ATTRIB
- CONFIG_RSS
- CONFIG_PROMISC
- GET_STATS
- WQ_CREATE
- RQ_CREATE
- WQ_DELETE
- RQ_DELETE

Driver Messages

Console messages issued by the driver.

11.2. Emulex BE3 10Gb Device-Specific Functions

The **SET DEVICE /DEVICE_SPECIFIC=(FUNCTION="FUNC", VALUE=(V1, V2, ...))** LANCP command provides a mechanism to issue device-specific functions to the Emulex BE3 driver. While some functions are common to many LAN drivers, most are primarily useful in a diagnostic context and are not typically included in LANCP command documentation.

For further information on the LANCP utility and LANCP commands, refer to the relevant section in the [VSI OpenVMS System Management Utilities Reference Manual, Volume I: A–L \[https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590\]](https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-l/#d0e28590) or the LANCP online Help.

Notes

The function name is case sensitive and is always four characters in length.

The command requires the SYSPRV privilege.

The definitions of the following functions may change between versions of the driver.

Device-Specific Functions

DEVICE_SPECIFIC=FUNCTION="CCOU"

Clear the driver counters.

This is useful in debug or test scenarios.

DEVICE_SPECIFIC=FUNCTION="SOFT"

Force a non-fatal (soft) error. The chip is stopped and restarted.

DEVICE_SPECIFIC=(FUNCTION="FILA",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILB",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILC",VALUE=(32bits,16bits))

DEVICE_SPECIFIC=(FUNCTION="FILD",VALUE=(32bits,16bits))

Specify a receive or transmit packet trace address for use in driver tracing. If the source or destination address does not match one of the four filter addresses, the packet data is not traced. For example, to filter on addresses AA-00-04-00-EF-4F and AB-00-04-01-D2-01, issue the following LANCP commands:

```
$ MCR LANCP SET DEVICE EBA/DEVICE_SPECIFIC=(FUNCTION="FIL1",VALUE=(%x000400aa,%x4fef))
$ MCR LANCP SET DEVICE EBA/DEVICE_SPECIFIC=(FUNCTION="FIL2",VALUE=(%x010400ab,%x01d2))
```

DEVICE_SPECIFIC=FUNCTION="SYSI"

Send the periodic System ID messages immediately instead of waiting until expiry of the 8–12-minute System ID timer.

This is a convenient way to get the driver to send two packets.

DEVICE_SPECIFIC=FUNCTION="SYSZ"

Disable the sending of periodic System ID messages.

This is used to disable transmits that may obscure other traffic of interest.

DEVICE_SPECIFIC=(FUNCTION="WAIT",VALUE=*timevalue*)

Wait the specified time (in nanoseconds) at IPL 8 with the driver port lock held.

This is a convenient way to introduce a delay at IPL 8.

DEVICE_SPECIFIC=(FUNCTION="XTOB",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a transmit timeout occurs. If 0 (disable), transmit timeout results in a device reset.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="XLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long transmit occurs. If 0 (disable), a long transmit completion is counted.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="FLON",VALUE=*setting*)

If set to 1 (enable), the driver crashes the system with a NETDLLERR Bugcheck if a long fork occurs. If 0 (disable), a long time to run a fork process is recorded in the counters.

If enabled, this setting is displayed in the driver flags in the driver internal counters.

DEVICE_SPECIFIC=(FUNCTION="PRES",VALUE=*periodicmask*)

Schedule periodic random resets. The value supplied is AND'ed with (RSCC × 69 069) and, if the result is 3, a reset is done. For example, if the value supplied is 7, a reset is done, on average, every 8 seconds.

A value of 0 disables this function, and a value of *n* enables it.

This is used to test error handling.

DEVICE_SPECIFIC=FUNCTION="OMSG"

Console messages are displayed by the driver during initialization, auto-negotiation, and link handling. However, the driver limits the total number of messages displayed. This function resets the count, allowing additional messages to be displayed.

DEVICE_SPECIFIC=(FUNCTION="FLOW",VALUE=*value*)

Sets flow control according to the *value*:

- 0 – Disable flow control.
- 1 – Enable flow control.
- 2 – Force receive flow control only.
- 3 – Force transmit flow control only.

DEVICE_SPECIFIC=(FUNCTION="XTMO",VALUE=*ticks*)

Override the default transmit timer. The *ticks* value is in timer ticks and is not range checked. If the fast transmit timer is enabled, each tick is 0.25 seconds; otherwise, it is 1 second.

The current setting is displayed in the driver internal counters.

**DEVICE_SPECIFIC=FUNCTION="FXTO"
(SYS\$EWBE3_MON.EXE Driver Only)**

Force a transmit timeout on the next transmit.

DEVICE_SPECIFIC=(FUNCTION="XDEL",VALUE=*timevalue*)

Delay issuing the next transmit request for the time specified in 10-millisecond units. A maximum of 2000 (20 seconds) is allowed.

This function is useful for investigating application behavior when a transmit request is delayed.

DEVICE_SPECIFIC=(FUNCTION="CXMT",VALUE=*delayvalue*)

Change the interrupt mitigation delay on transmit completion, the "Transmit coalesce value (μ s)" value. This allowed range is 0 to 1000000 microseconds. If zero, the delay is set to the default of 20 microseconds.

DEVICE_SPECIFIC=(FUNCTION="CRCV",VALUE=*delayvalue*)

Change the interrupt mitigation delay on receive completion, the "Receive coalesce value (μ s)" value. This allowed range is 0 to 1000000 microseconds. If zero, the delay is set to the default of 32 microseconds.

**DEVICE_SPECIFIC=(FUNCTION="CORR",VALUE=*bytetocorrupt*)
(SYS\$EWBE3_MON.EXE Driver Only)**

Corrupt a byte in a transmit packet (increment it by 1).

This can be used to lose packets by selecting a byte in the destination address (0–5) to validate handling of lost packets for a protocol.

This can be used to validate IP checksums by corrupting the packet data protected by the checksum after the network stack has calculated the checksum.

DEVICE_SPECIFIC=(FUNCTION="QDEB",VALUE=*setting*)

If set to 1 (enable), the driver issues every transmit with the QnQ driver workaround, where the driver creates an additional transmit segment for the VLAN tag with special transmit descriptor bits set.

If set to 0 (disable), the driver only does the QnQ workaround if the transmit request requires it.

This is used to validate the QnQ workaround driver support.

DEVICE_SPECIFIC=(FUNCTION="QOFF",VALUE=*setting*)

If set to 1 (enable), enables the QnQ workaround in the driver.

If set to 0 (disable), disables the QnQ workaround.

By disabling the workaround and issuing a particular transmit request, the BE3 chip locks up. This helps validate the QnQ workaround support, by verifying the problem packet really does cause a lockup, and then verifying the QnQ workaround actually does prevent the lockup.