



User's Guide

qeDiag Diagnostic Tool

45000, 41000, 8500, 8600 Series

83851-546-00 F

November 15, 2019



For more information, visit our website at: <http://www.marvell.com>

Notice

THIS DOCUMENT AND THE INFORMATION FURNISHED IN THIS DOCUMENT ARE PROVIDED "AS IS" WITHOUT ANY WARRANTY. MARVELL EXPRESSLY DISCLAIMS AND MAKES NO WARRANTIES OR GUARANTEES REGARDING THE PRODUCT, WHETHER EXPRESS, ORAL, IMPLIED, STATUTORY, ARISING BY OPERATION OF LAW, OR AS A RESULT OF USAGE OF TRADE, COURSE OF DEALING, OR COURSE OF PERFORMANCE, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE AND NON-INFRINGEMENT.

Information furnished is believed to be accurate and reliable. However, Marvell assumes no responsibility for the consequences of use of such information or for any infringement of patents or other rights of third parties that may result from its use. No license, express or implied, to any Marvell intellectual property rights is granted by this document. Marvell products are not authorized for use as critical components in medical devices, military systems, life or critical support devices, or related systems. Marvell retains the right to make changes to this document at any time, without notice.

Export Control

The user or recipient of this document acknowledges that the information included in this document may be subject to laws including, but not limited to, U.S. export control laws and regulations regarding the export, re-export, transfer, diversion, or release of such information. The user or recipient must comply with all applicable laws and regulations at all times. These laws and regulations include restrictions on prohibited destinations, end users, and end uses.

Patents/Trademarks

Products identified in this document may be covered by one or more Marvell patents and/or patent applications. You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning the Marvell products disclosed herein. Marvell, the Marvell logo, and QLogic registered trademarks of Marvell or its affiliates. Please visit www.marvell.com for a complete list of Marvell trademarks and any guidelines for use of such trademarks. Other names and brands may be claimed as the property of others.

Copyright

Copyright © 2015–2019. Marvell International Ltd. All rights reserved.

Table of Contents

Preface

| | |
|---|-----|
| Intended Audience | ix |
| What Is in This Guide | ix |
| Related Materials | x |
| Documentation Conventions | x |
| Technical Support | xi |
| Downloading Updates and Documentation | xii |
| Knowledgebase | xii |

1 What is qeDiag?

| | |
|------------------------|---|
| Introduction | 1 |
| Functions | 1 |

2 Operating Environments

| | |
|------------------------|---|
| Linux | 3 |
| Windows | 4 |
| UEFI | 4 |
| VMware (ESX) | 5 |

3 Operating Modes

| | |
|---|----|
| Manufacture Mode | 6 |
| Engineering Mode | 7 |
| Driver Debugging Mode (DRV_DBG) | 7 |
| Serial Mode | 7 |
| NO_INIT Mode | 8 |
| Network Mode (DON) | 9 |
| Virtual NVRAM Mode | 12 |
| Per PF Mode | 14 |

4 Command Line Options

5 Engineering Mode Commands

| | |
|--|----|
| add_help | 19 |
| dbgConfig, dbgStart, dbgDump | 19 |
| debug | 20 |

| | |
|-------------------------------|----|
| device | 21 |
| driver | 22 |
| driver load | 22 |
| driver set_loopback | 22 |
| driver link_state | 23 |
| driver set_int_mode | 23 |
| driver unload | 23 |
| driver init | 23 |
| help | 23 |
| extphy | 24 |
| hmem | 24 |
| hmem palloc | 24 |
| hmem paddr | 24 |
| hmem read | 24 |
| hmem write | 25 |
| grcDump | 25 |
| idleChk | 27 |
| l2pkt | 28 |
| l2pkt send | 28 |
| l2pkt receive | 28 |
| led | 29 |
| linkdump | 29 |
| log | 32 |
| log open | 32 |
| log close | 32 |
| mcp | 32 |
| mdump | 34 |
| nictest | 35 |
| nvm | 42 |

| | |
|--|-----|
| nvm cfg | 42 |
| nvm cfg | 42 |
| nvm cfg <option>=<value> | 43 |
| nvm cfg -all <option>=<value> | 43 |
| nvm cfg -all <option>= | 43 |
| nvm cfg -func <func#><op>=<data> | 43 |
| nvm cfg -func <func#><op>= | 43 |
| nvm cfg -find <option board> | 44 |
| nvm cfg <option>- | 44 |
| nvm cfg <board type> | 44 |
| nvm cfg -dump <filename> | 44 |
| nvm chk | 143 |
| nvm crc | 143 |
| nvm dir | 143 |
| nvm dump | 144 |
| nvm dump <filename> | 144 |
| nvm dump -mdump <filename> | 144 |
| nvm prg | 145 |
| nvm read | 145 |
| nvm show | 145 |
| nvm write | 145 |
| nvm fill | 146 |
| nvm upgrade | 146 |
| nvm factory_defaults | 147 |
| nvm update_cfg | 147 |
| nvm trace_modules | 149 |
| pci | 150 |
| pci scan | 150 |
| pci search | 150 |
| pci setdut | 151 |
| pcicfg | 151 |
| pcicfg read | 151 |
| pcicfg show | 151 |
| pcicfg write | 151 |
| Phy | 152 |
| phy raw_read | 152 |
| phy raw_write | 152 |
| phy core_read | 152 |
| phy core_write | 152 |

| | |
|----------------------------|-----|
| phy mac_lb | 152 |
| phy mac_stat | 153 |
| phy info | 153 |
| pretend | 153 |
| reg | 155 |
| reg read | 155 |
| reg show | 155 |
| reg write | 155 |
| serport | 156 |
| serport open | 156 |
| serport redirect | 156 |
| serport close | 156 |
| sfp | 157 |
| sfp read | 157 |
| sfp write | 157 |
| sfp decode | 157 |
| temperature | 158 |
| version | 158 |
| vpd | 158 |
| vpd read | 159 |
| vpd show | 159 |
| vpd write | 159 |
| vpd delete | 160 |

6 Engineering Mode Tcl Utilities

| | |
|------------------------|-----|
| reset_chip | 161 |
| parse_rdf | 161 |
| block_offset | 161 |
| block_regs | 161 |
| reg_read | 162 |
| reg_offset | 162 |
| reg_write | 162 |
| read_regs | 162 |
| efuse_info | 162 |
| profile | 165 |

A Setting NPAR Mode

B macaddr.txt File Format

C Revision History

List of Figures

| Figure | | Page |
|--------|---------------------------|------|
| 3-1 | Operating Modes | 6 |

List of Tables

| Table | | Page |
|-------|--------------------------------------|------|
| 4-1 | Command Line Options | 16 |
| 5-1 | dbgConfig Command Options | 19 |
| 5-2 | grcDump Options Parameters | 25 |
| 5-3 | VPD Tags | 158 |

Preface

Intended Audience

This manual is intended for engineers, testers, technicians, and OEM customers who want to perform manufacturing and engineering level diagnostics on the Marvell® FastLinQ® Controllers.

What Is in This Guide

This guide contains the information you need to use the qeDiag diagnostic tool with the Marvell FastLinQ Controllers.

This preface specifies the intended audience, summarizes the contents of this guide, lists related materials, provides licensing information, describes the typographic conventions used in this guide, and provides technical support and contact information.

The remainder of the user's guide is organized into the following chapters:

- [Chapter 1 What is qeDiag?](#) describes the functions of the qeDiag diagnostic tool.
- [Chapter 2 Operating Environments](#) lists the OSs supported by qeDiag.
- [Chapter 3 Operating Modes](#) describes qeDiag's four operating modes.
- [Chapter 4 Command Line Options](#) defines the qeDiag command line options.
- [Chapter 5 Engineering Mode Commands](#) defines the qeDiag commands.
- [Chapter 6 Engineering Mode Tcl Utilities](#) defines the tool command language (Tcl) utilities.
- [Appendix A Setting NPAR Mode](#) describes how to set NPAR mode.
- [Appendix B macaddr.txt File Format](#) defines the format for the `macaddr.txt` file.
- [Appendix C Revision History](#) contains a list of changes made to this guide since the last revision.

Related Materials

For information about downloading documentation from the Marvell Web site, see [“Downloading Updates and Documentation” on page xii](#).

Documentation Conventions

This guide uses the following documentation conventions:

- **NOTE** provides additional information.
- Text in [blue](#) font indicates a hyperlink (jump) to a figure, table, or section in this guide, and links to Web sites are shown in [underlined blue](#). For example:
 - ❑ [Table 9-2](#) lists problems related to the user interface and remote agent.
 - ❑ See [“Installation Checklist” on page 6](#).
 - ❑ For more information, visit www.marvell.com.
- Text in **bold** font indicates user interface elements such as a menu items, buttons, check boxes, or column headings. For example:
 - ❑ Click the **Start** button, point to **Programs**, point to **Accessories**, and then click **Command Prompt**.
 - ❑ Under **Notification Options**, select the **Warning Alarms** check box.
- Text in `Courier` font indicates a file name, directory path, or command line text. For example:
 - ❑ To return to the root directory from anywhere in the file structure:
Type `cd /root` and press ENTER.
- Key names and key strokes are indicated with UPPERCASE:
 - ❑ Press CTRL+P.
 - ❑ Press the UP ARROW key.
- Text in *italics* indicates terms, emphasis, variables, or document titles. For example:
 - ❑ For a complete listing of license agreements, refer to the *QLogic Software End User License Agreement*.

- ☐ What are *shortcut keys*?
- ☐ To enter the date type *mm/dd/yyyy* (where *mm* is the month, *dd* is the day, and *yyyy* is the year).
- Topic titles between quotation marks identify related topics either within this manual or in the online help, which is also referred to as *the help system* throughout this document.
- Command line syntax conventions include the following:
 - ☐ Plain text indicates items that you must type as shown. For example:
 - `gauccli -pr nic -ei`
 - ☐ `< >` (angle brackets) indicate a variable whose value you must specify. For example:
 - `<serial_number>`
 - ☐ `[]` (square brackets) indicate an optional parameter. For example:
 - `[<file_name>]` means specify a file name, or omit it to select the default file name.
 - ☐ `|` (vertical bar) indicates mutually exclusive options; select one option only. For example:
 - `on|off`
 - `1|2|3|4`

Technical Support

Customers should contact their authorized maintenance provider for technical support of their Marvell FastLinQ products. Technical support for Marvell products under warranty is available with a Marvell support account.

To set up a support account (if needed) and submit a case:

1. Go to www.marvell.com.
2. Click **Support**.
3. Scroll down if needed, and then click **Ethernet Adapters and Controllers**.
4. Under Related Links, click **Customer Support**.
5. Do one of the following:
 - ☐ If you do not have an account, click **Sign Up**.
 - ☐ If you have an account, enter your **User Name** and **Password**, and then click **Log In**.

Downloading Updates and Documentation

The Marvell Web site provides periodic updates to product firmware, software, and documentation.

1. Go to www.marvell.com.
2. Point to **Support**. Under Driver Downloads, click **Marvell QLogic/FastLinQ Drivers**.
3. Click the product type tab: **Adapters**, **Switches**, or **Routers**.
4. Click the corresponding button to search **by Model** or **by Operating System**.
5. To define a search, click an item in each selection column, and then click **Go**.
6. Locate the firmware, software, or document you need, and then click the item's name or icon to download or open the item.

Knowledgebase

The Marvell knowledgebase is an extensive collection of product information that you can search for specific solutions. Marvell is constantly adding to the collection of information in the database to provide answers to your most urgent questions.

To access the knowledgebase:

1. Go to www.marvell.com.
2. Point to **Support**. Under **Support by Product**, click **GET SUPPORT**.
3. On the Support page, select the product line for your adapter:
 - ☐ **QLogic Fibre Channel Adapters and Controllers**
 - ☐ **FastLinQ® | LiquidIO® Ethernet Adapters and Controllers**
4. Under **Related Links**, click **Public Knowledgebase**.

1 What is qeDiag?

This section describes the functions of the qeDiag diagnostic tool.

Introduction

The qeDiag diagnostic tool provides manufacturing and engineering level diagnostics for the following Marvell controllers:

- 45000 Series
- 41000 Series
- 8500 Series
- 8600 Series

These controllers are collectively referred to as the *Marvell FastLinQ Controllers* or simply *controllers* unless otherwise noted.

This guide does the following:

- Describes the qeDiag commands and their usage, as well as all the diagnostic tests supported on the Marvell FastLinQ Controllers.
- Lists the OSs in which qeDiag is supported.
- Describes qeDiag's operation modes.

Functions

The qeDiag tool is used for:

- Diagnostic commands, such as reading registers and NVRAM content
- Configuring the board
- Upgrading images
- Loading the qeDiag driver
- Running board testing
- Running signal integrity (SI) testing
- Running traffic (currently only layer 2 (L2))

The qeDiag tool includes the Tcl shell, which allows you to run Tcl scripts.

NOTE

The qeDiag tool for Marvell FastLinQ Controllers is a different binary from the eDiag tool for the BCM577xx and 8400 Series Controllers. Some of the commands that are available in eDiag have not been implemented in qeDiag, or the commands are used differently. See this guide for more details.

2 Operating Environments

This section lists the OSs supported by qeDiag.

Linux

The supported Linux® 64-bit distributions are:

- Red Hat® 7
- Ubuntu®

Linux 32-bit distributions are supported in version 8.30.1.0 and later.

The supported software is:

- `lediag`
- `slim_driver`

To build and run qeDiag:

1. Open the tar file by issuing the following command:
`tar xzvf qlediag-8.x.x.x.tar.gz`
2. Access the `qlediag` folder by issuing the following command:
`cd qlediag-8.x.x.x`
3. Compile the sources by issuing the following command:
`make`
4. Run the tool by issuing the following command:
`./load.sh`

Linux qeDiag is a user-space application with open source. It uses a `slim_driver` for kernel services. The qeDiag tool communicates with the `slim_driver` using input/output control (IOCTL). The `slim_driver` should stay *slim*, providing only the following kernel services:

- Physical memory allocation
- Kernel memory allocation
- DMA mapping
- Request interrupt request (IRQ)
- Enable IRQ
- Enable MSI-X/MSI

The Linux qeDiag tool supports driver debugging (DRV_DBG) mode. In this mode, the `slim_driver` is not loaded.

Windows

The qeDiag tool is supported in the Windows® driver debugging (DRV_DBG) mode.

The winediag application requires that the Windows driver be installed on the setup. This application communicates with the device through input/output control (IOCTL) requests from the Windows driver.

The supported software is:

- `winediag.exe`

To run qeDiag:

- Run the executable `winediag.exe`.

UEFI

The qeDiag is supported in UEFI for both driver debugging (DRV_DBG) mode and diagnostic engineering (ENG) mode.

The supported software is:

- `qeddiag_x64.efi`

To run qeDiag:

- Run the executable `load.nsh`.

VMware (ESX)

The supported VMware® distributions are:

- ESX® 6.0
- ESX 6.5

To run qeDiag on these distributions, you must have VMware `qedentv` driver version 2.0.6.10 or later.

To run `qeddiag_vmware`:

1. Open the tar file by issuing the following command:

```
tar xzvf qeddiag_vmware_8.x.x.x.tar.gz
```
2. Access the `qeddiag_vmware` folder by issuing the following command:

```
cd qeddiag-vmware_8.x.x.x
```
3. Run the tool by issuing the following command:

```
./load.sh
```

NOTE

`qeddiag_vmware` works only in DRV_DBG mode, over the VMware driver.
`qeddiag_vmware` is already compiled.

3 Operating Modes

This section describes qeDiag's operating modes, which are illustrated in [Figure 3-1](#).

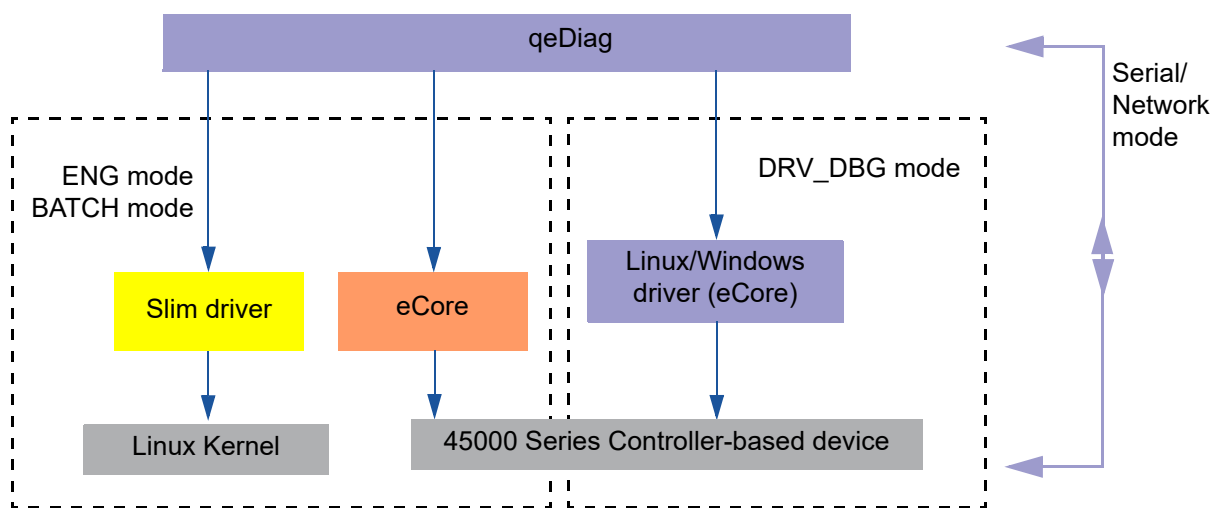


Figure 3-1. Operating Modes

Manufacture Mode

The default mode of operation is the manufacture mode.

The qeDiag tool enters this mode if no command line option is used. That is, running `load.sh` (in Linux), `winediag.exe` (in Windows), or `qeddiag_x64efi` (in UEFI) puts qeDiag in manufacture mode.

In this mode of operation, qeDiag runs the default set of tests (`nictest`) on the board.

Engineering Mode

Engineering mode is for diagnostics, board configuration, and testing.

To enter engineering mode, use the `-eng` command line option. For example:

```
./load.sh -eng (Linux)
qeddiag_x64.efi -eng (UEFI)
```

In Linux, this mode of operation requires `slim_driver` for kernel services.

In Windows, this mode is not supported; only DRV_DBG mode over Windows driver mode is supported.

Driver Debugging Mode (DRV_DBG)

The qeDiag tool automatically enters driver debugging mode (DRV_DBG) in case an operational driver is already running (that is, `qed/qede` in Linux or `evbd` in Windows).

This mode enables diagnostic operations such as register readings, NVRAM readings, and configuration changes.

The following functions are not supported in driver debugging mode:

- Board testing (`nictest`)
- Management control processor (MCP) reset
- Loading the qeDiag internal driver

This mode is similar to the eDebug tool used with the BCM577xx and 8400 Series Controllers.

Serial Mode

Serial mode is useful for debugging in case there is a system halt and debugging is not available through PCI®.

To enter serial mode, use the `-serial` command line option. For example:

```
./load.sh -serial (Linux and ESX)
winediag.exe -serial (Windows)
qeddiag_x64.efi -serial (UEFI)
```

You can use the `-fast_serial` command instead of the `-serial` command for faster serial mode (without the NVRAM readings).

To use serial mode over USB:

- In Linux, use the `-usb` command line option. For example:

```
./load.sh -serial -usb  
./load.sh -usb
```

- In Windows, you can use the relevant COM port.

This mode requires connecting a serial cable between the machine running the qeDiag tool and the universal asynchronous receiver-transmitter (UART) port on the board being tested.

This mode automatically opens a serial connection at a baud rate of 115200.

The default COM port in Linux is `/dev/ttyS0`.

The default COM port in Windows is COM 1.

To change the COM port, issue the following command:

```
-com <comport#>
```

To change the baud rate, issue the `-baud <baudrate>` command. For example:

```
. ./load.sh -serial -com 2 -baud 19200
```

The following commands are supported in serial mode:

- [“device” on page 21](#)
- [“help” on page 23](#)
- [“grcDump” on page 25](#)
- [“idleChk” on page 27](#)
- [“log” on page 32](#)
- [“mcp” on page 32](#)
- [“nvm” on page 42](#)
- [“reg” on page 155](#)
- [“version” on page 158](#)
- [“vpd” on page 158](#)

NO_INIT Mode

NO_INIT mode operation does not perform device initialization. This mode is useful for engineering diagnostics when you do not want to choose or initialize a device.

Network Mode (DON)

Diagnostics over network (DON) mode is useful for debugging if there is a system halt and debugging is not available through PCI.

DON mode provides debug information (for example, `mcp trace`, `grcDump`, `idleChk`) from the Marvell FastLinQ Controllers through the network instead of through a serial cable (the latter of which is very slow).

You can use DON mode when:

- The Marvell FastLinQ Controllers-based target adapter (referred to as *adapter* or *target adapter*) has no serial connection.
- Management firmware (MFW) is running and a link is up.

In DON mode, `qeDiag` can run on the same setup as the target adapter, or on a different setup that is connected to the same network as the target adapter.

Before enabling network mode:

1. Enable DON on the target adapter by issuing the following command:

```
mcp enable_don
```
2. Ensure that the network card is online by issuing the `ifconfig` command, for example:

```
ifconfig <network interface> <IP address>
```

A sample network interface is `p1p1`.

3. Connect a cable from the network card to the target adapter port.
4. Issue the following command:

```
ifconfig -a
```

If this procedure is successful, the command output is:

```
UP BROADCAST RUNNING MULTICAST
```

To enable network mode:

1. Issue the following command:

```
./load.sh -network
```
2. Enter the MAC address of the target adapter.
3. Enter network interface name, for example, `p1p1`.
4. Create a password with 8–12 characters, including at least one digit and one alphabetic character, without repetitions.

The password should be the same DON password programed on the target adapter's NVRAM by the `mcp enable_don` command (see "mcp" on page 32).

To enable network mode in one command line:

```
./load.sh -network -don_mac <target MAC address> -don_if <network  
interface name> -don_pwd <password>
```

The following commands are supported in network mode:

- "device" on page 21
- "grcDump" on page 25
- "idleChk" on page 27
- "mcp" on page 32
- "nvm" on page 42
- "reg" on page 155

Examples

```
./load.sh -network
```

```
*****  
ediag -- Qlogic Everest diagnostic tool, version 8.30.12.0  
*****  
Copyright (c) 2014-2017 QLogic Corporation, all rights reserved.  
  
Please enter target MAC address (XX:XX:XX:XX:XX:XX).  
Please enter new primary MAC address:00:10:18:11:22:33  
User has entered the MAC address 00:10:18:11:22:33  
Please enter source interface name:  
plp1  
Open socket for source interface plp1 and DST MAC address 00:10:18:11:22:33  
Please enter password:  
*****  
Seed: 0x88732d31
```

```
NETWORK:>nvm dir
```

| Image | SRAM Addr | SRAM Run Addr | NVM Offset | Byte Cnt | Version |
|-------|------------|---------------|------------|----------|--------------|
| LIM | 0x08035800 | 0x08035824 | 0x00000040 | 0x00cf8 | LIM 8.18.0.0 |
| DIR1* | 0x00000000 | 0x00000000 | 0x00002000 | 0x001a0 | |

3-Operating Modes

Network Mode (DON)

| | | | | | | |
|------------------------|--------------|------------|------------|------------|---------|---------------|
| | DIR2 | 0x00000000 | 0x00000000 | 0x00003000 | 0x001a0 | |
| 8.30.0.139 | MIM1* | 0x08000F14 | 0x0800FCC0 | 0x00004000 | 0x8fc44 | MIM |
| 8.30.0.137 | MIM2 | 0x08000F14 | 0x0800FC80 | 0x00130000 | 0x8fc44 | MIM |
| | VPD | 0x00000000 | 0x00000000 | 0x0025E000 | 0x00088 | |
| | NVM_CFG1 | 0x00000000 | 0x00000000 | 0x0025F000 | 0x01094 | |
| | DEFAULT_CFG | 0x00000000 | 0x00000000 | 0x00261000 | 0x01094 | |
| | ISCSI_CFG | 0x00000000 | 0x00000000 | 0x00263000 | 0x06008 | ISCSI_CFG 0.7 |
| | FCOE_CFG | 0x00000000 | 0x00000000 | 0x0026A000 | 0x02008 | FCOE_CFG 0.6 |
| 8.30.0.139 | TIM1* | 0x08028000 | 0x080288B4 | 0x0026D000 | 0x03e84 | TIM |
| | ETH_PHY_FW1* | 0x00000000 | 0x00000000 | 0x00271000 | 0x11f0c | 24.0.0.0 |
| | MFW_TRACE1* | 0x00000000 | 0x00000000 | 0x00283000 | 0x0f760 | |
| | CRASH_DUMP | 0x00000000 | 0x00000000 | 0x00293000 | 0x11004 | |
| 8.30.0.137 | TIM2 | 0x08028000 | 0x080288B4 | 0x002A5000 | 0x03e84 | TIM |
| 2.0.12;EFI x64 6.1.1.6 | MBA | 0x00000000 | 0x00000000 | 0x002B6000 | 0x43804 | PCI30 MBA |
| | INIT_HW | 0x00000000 | 0x00000000 | 0x002FA000 | 0x10fa0 | 8.18.2.0 |
| | ETH_PHY_FW2 | 0x00000000 | 0x00000000 | 0x0031E000 | 0x11f0c | 24.0.0.0 |
| | MFW_TRACE2 | 0x00000000 | 0x00000000 | 0x00330000 | 0x0f73c | |
| 8.18.15.0 | RECOVERY | 0x08000880 | 0x08009EA8 | 0x00343000 | 0x12f44 | RCVR |

* Running Image

```

NETWORK:> idleChk
fw-version: 8_20_0_0
fw-image: main
mfw-version: 8_30_0_139
tools-version: 10
chip: ah
platform: asic
pci-func: 0
dump-type: idle-chk

```

FW_IDLE_CHECK:

Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

LSI_IDLE_CHECK:

Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

Idle Check completed successfully (with 1 warnings)

Virtual NVRAM Mode

Virtual NVRAM mode is useful for running qeDiag without the Marvell FastLinQ Controller-based adapter (referred to as *adapter*).

This mode can provide NVRAM image changes (for example, firmware upgrade and NVRAM configuration changes) without using a real adapter.

Virtual NVRAM mode requires two inputs:

- An NVRAM binary file (dumped from the adapter or virtual NVRAM mode)
- The controller type, where:
 - AH Marvell FastLinQ 41000 and 8600 Series Controllers
 - BB Marvell FastLinQ 45000 and 8500 Series Controllers

To enable virtual NVRAM mode, issue the following command:

```
./load.sh -vnvm <NVM binary file> -chip_type <AH|BB>
```

Examples

```
./load.sh -vnvm victoria.bin -chip_type AH
*****
ediag -- Qlogic Everest diagnostic tool, version 8.23.0.0
*****
Copyright (c) 2014-2017 QLogic Corporation, all rights reserved.
```

```
1-VIRT_NVM:>nvm upgrade -F -mbi ql_ah_mbi_8.20.09.bin
Upgrading PK image.
Avoiding LIM upgrade
Avoiding RECOVERY upgrade
Upgrading TIM2 image from version TIM 8.20.4.0 to version TIM 8.20.4.0.
Upgrading MIM2 image from version MIM 8.20.4.0 to version MIM 8.20.4.0.
Upgrading ETH_PHY_FW2 image from version 24.0.0.0 to version 24.0.0.0.
Upgrading MFW_TRACE2 image.
Upgrading CRASH_DUMP image.
```


3-Operating Modes

Virtual NVRAM Mode

Upgrading MBA image from version PCI30 MBA 2.0.12;EFI x64 6.1.1.12 to version PCI30 MBA 2.0.12;EFI x64 6.1.1.12.

[set_exp_rom_size:1188] Set expansion ROM size to 0x9

Upgrading INIT_HW image from version 8.18.2.0 to version 8.18.2.0.

[set_mbi_version:749] Updated mbi version and date in NVM_CFG.

[set_mbi_version:778] Updated mbi version and date in DEFAULT_CFG.

Image Loaded Successfully

Note: "mcp reset" or reboot are required for the upgrade to take place.

1-VIRT_NVM:>mcp reset

1-VIRT_NVM:>nvm dir

MBI version: 8.20.9 MBI date: 5/26/2017

| | Image | SRAM Addr | SRAM Run Addr | NVM Offset | Byte Cnt | Version |
|-------------|--------------|------------|---------------|------------|----------|-----------|
| 8.18.0.0 | LIM | 0x08035800 | 0x08035824 | 0x00000040 | 0x00cf8 | LIM |
| | DIR1 | 0x00000000 | 0x00000000 | 0x00002000 | 0x001c8 | |
| | DIR2* | 0x00000000 | 0x00000000 | 0x00003000 | 0x001c8 | |
| 8.20.4.0 | MIM1 | 0x08000EF8 | 0x0800F980 | 0x00004000 | 0x79b98 | MIM |
| 8.20.4.0 | MIM2* | 0x08000EF8 | 0x0800F980 | 0x00130000 | 0x79b98 | MIM |
| | NVM_CFG1 | 0x00000000 | 0x00000000 | 0x0025C000 | 0x01094 | |
| | VPD | 0x00000000 | 0x00000000 | 0x0025E000 | 0x00088 | |
| | PK | 0x00000000 | 0x00000000 | 0x00260000 | 0x000a8 | |
| DEFAULT_CFG | | 0x00000000 | 0x00000000 | 0x00261000 | 0x01094 | |
| 0.7 | ISCSI_CFG | 0x00000000 | 0x00000000 | 0x00263000 | 0x06008 | ISCSI_CFG |
| 0.6 | FCOE_CFG | 0x00000000 | 0x00000000 | 0x0026A000 | 0x02008 | FCOE_CFG |
| 8.18.15.0 | RECOVERY | 0x08000880 | 0x08008E40 | 0x0026D000 | 0x11ddc | RCVR |
| MFW_TRACE1 | | 0x00000000 | 0x00000000 | 0x0027F000 | 0x0f524 | |
| | INIT_HW | 0x00000000 | 0x00000000 | 0x0028F000 | 0x10fa0 | 8.18.2.0 |
| 8.20.4.0 | TIM1 | 0x08028000 | 0x080288B4 | 0x002A4000 | 0x03e74 | TIM |
| | META | 0x00000000 | 0x00000000 | 0x002B9000 | 0x0dd20 | |
| 8.20.4.0 | TIM2* | 0x08028000 | 0x080288B4 | 0x002C7000 | 0x03e74 | TIM |
| | ETH_PHY_FW2* | 0x00000000 | 0x00000000 | 0x002CB000 | 0x11f0c | 24.0.0.0 |

3-Operating Modes

Per PF Mode

```
MFW_TRACE2*    0x00000000    0x00000000    0x002DD000    0x0f524
  CRASH_DUMP    0x00000000    0x00000000    0x002ED000    0x11004
ETH_PHY_FW1    0x00000000    0x00000000    0x00344000    0x11f0c  24.0.0.0
      MBA      0x00000000    0x00000000    0x003A6000    0x4f404  PCI30 MBA
2.0.12;EFI x64 6.1.1.12
-----
* Running Image

1-VIRT_NVM:>nvm cfg 1-

                               Group: board (Group 1)
-----
1: [FUNC] MAC address                               : 00:10:18:11:22:33
```

Per PF Mode

qeDiag version 8.42.6.0 and later supports per physical function (PF) mode. On PFs that are running the operational driver, qeDiag communicates with the PF's driver to get diagnostics information.

On PFs that are not running on the operational driver, qeDiag loads the `slim_driver` and communicates with it to get kernel services. In this case, qeDiag gets diagnostics information from the device through an Ethernet core (eCore).

In following example:

- Device 1 is running with the `qede` driver.
- Device 2 is running without the `qede` driver.

Examples

```
[root@lb-tlvb-miris qlediag_8.42.4.0_mode_per_pf]# ./load.sh -eng
NOTICE: Operational driver is loaded.
*****
ediag -- Cavium Everest diagnostic tool, version 8.42.4.0_mode_per_pf (platform: Linux)
*****
Copyright (c) 2014-2018 Cavium, Inc., all rights reserved.
[os_if_pci_scan:366] Found 21 pci devices
[pci_scan:738] Detected 8 Everest4 devices.
Chip #1:
-----
ASIC:57940 Rev:A2 MFW:8.24.44.0 S.N.:ABC1527A12345
No.ports:2 No.paths:1 No.ports_per_path:2 Is_100G:No
Nvram Size: 16MB
```

3-Operating Modes

Per PF Mode

Board: Dell Duluth (QL41162_Duluth)

| Dev | Path | Port | PF | HwFn | B:D:F | PCI | Spd | MAC | St. | Type | I/F | Drv | DrvVer | Cfg |
|-----|-------------|------|----|------|-------------|--------|-----|-------------------|-----|------|------|------|-----------|-----|
| 1 | 00.00.00.00 | | | | 00:02:00:00 | PCIE-4 | 8.0 | 00:10:18:11:22:33 | D0 | Eth | p4p1 | qede | 8.40.26.0 | |
| 2 | 00.01.01.00 | | | | 00:02:00:01 | PCIE-4 | 8.0 | 00:10:18:11:22:34 | D0 | Eth | | | | |

4 Command Line Options

The available qeDiag command line options are listed in [Table 4-1](#).

Use these options in the command line. For example:

```
qeDiag -log results.log -rc regress.tcl
```

Table 4-1. Command Line Options

| Options | Description |
|---------------------------------|---|
| -arg | Provides arguments to be used by the script |
| -baud | Sets the baud rate for serial mode |
| -board_idx <index> | Selects the requested board index to upgrade |
| -board_type <name> | Selects the requested board type to upgrade |
| -chip_type <AH BB> | Indicates the controller type required for manufacture or virtual NVRAM mode |
| -com | Sets the COM port for serial mode |
| -debug | Adds debug print outs |
| -dev | Selects the device number upon which the tests will be run |
| -don_mac <xx:xx:xx:xx:xx:xx> | Sets MAC address for network mode (DON) |
| -don_if <interface> | Sets interface name for network mode (DON) |
| -don_pwd <password> | Sets password for network mode (DON) |
| -drv_dbg | Changes to DRV_DBG mode (UEFI only) |
| -dup | Enables a duplicate instance of qeDiag in addition to an already running instance of qeDiag |

Table 4-1. Command Line Options (Continued)

| Options | Description |
|-----------------------------------|---|
| -eng | Enters qeDiag engineering mode. A prompt appears. |
| [-F] -fmbi <image> | Upgrades the monolithic binary image (MBI). Also provides an optional force flag ([-F]). |
| fast_serial | Fast serial mode (without NVRAM readings) |
| -fcfg | Source configuration Tcl file |
| -fcfg_dump <file> | Logs the cfg dump into a file |
| -ffcoe_cfg <file> ^a | Programs the FCOE_CFG binary image into the NVRAM |
| -fiscsi_cfg <file> | Programs the ISCSI_CFG binary image into the NVRAM |
| -fmac <macaddr.txt> | Specifies a file that contains the MAC addresses to be programmed. The file <code>macaddr.txt</code> must include the first and last MAC addresses. See “macaddr.txt File Format” on page 169 for more details. |
| -fnvm <image> | Programs a raw image into the NVRAM (preserves the MAC address) |
| -fpga | Provides special handling for field programmable gate array (FPGA) UART. |
| -fvpd | Programs the vital product data (VPD) image into the NVRAM |
| -help | Prints out help and exits |
| -I <iteration#> | Specifies how many iterations of the tests to run |
| -list | Shows the qeDiag version and devices list, and then exits |
| -log <logfile> | Logs the tests' execution results into the specified file |
| -m | Interactively adds the MAC address |
| -nicp_mac | Sets the WWN MAC address option IDs 93 and 94 |
| -no_init | Starts qeDiag without pre-selecting any device |
| -num_mac | Provides the number of MAC addresses for the device |
| -offline ^a | For Windows offline mode usage: <code>-offline <pci_busNo></code> |
| -pmac | Prints the base MAC address into the <code>basemac.bin</code> file, and then exits |

Table 4-1. Command Line Options (Continued)

| Options | Description |
|--|---|
| <code>-rc <script></code> | Specifies a script to source immediately after startup |
| <code>-seq_mac_per_port</code> | Specifies sequential MAC addresses per port: PF0 (port 0) MAC PF1 (port 1) MAC+8 PF2 (port 0) MAC+1 PF3 (port 1) MAC+9, and so on |
| <code>-serial</code> | Enters qeDiag serial mode for debugging |
| <code>-skip_efi_unload</code> | Skips unloading the extensible firmware interface (EFI) driver |
| <code>-skip_id</code> | Skips the 4-process identifier (PID) check in an MBI upgrade |
| <code>-sn_fmt{a n *}</code> | Provides a serial number in the format {n-digit a-char *} |
| <code>-snc</code> | Serial number for specific OEM |
| <code>-snl</code> | Serial number for specific OEM |
| <code>-sno</code> | Serial number for specific OEM |
| <code>-snq <number></code> | Specifies the card's serial number |
| <code>-snt</code> | Serial number for specific OEM |
| <code>-socket <num></code> | Supports remote commands through a socket |
| <code>-stride <number></code> | Checks the primary MAC address alignment |
| <code>-t <grps/tests></code> | Disables certain groups or tests (that is, <code>-t A5</code>) |
| <code>-T <grps/tests></code> | Enables certain tests or groups (that is, <code>-T A5</code>) |
| <code>-usb</code> | Sets serial mode over USB (see “Serial Mode” on page 7) |
| <code>-version</code> | Shows the qeDiag version |
| <code>-vnvm <nvm_image></code> | Enables virtual NVRAM mode (without the adapter) |

^a Marvell FastLinQ 41000 and 45000 Series Controllers

5 Engineering Mode Commands

This section describes the qeDiag commands in alphabetic order.

add_help

Authority `add_help <command name> <description string>`

Syntax This command can be used by Tcl script writers to add help for new commands written in the Tcl script.

Keywords `add_help parse_rdf "Parses the RDF file into global arrays"`

dbgConfig, dbgStart, dbgDump

Authority `dbgConfig <options>`
 `dbgStart`

Run the scenario to be recorded:

`dbgDump <binary dump file>`

The list of <options> for `dbgConfig` can be obtained using “`dbgConfig -help`” and are listed in [Table 5-1](#).

Table 5-1. dbgConfig Command Options

| Options | Description |
|-------------------------------------|---|
| <code>-oneShot</code> | Sets <i>one shot</i> recording mode, where the recording stops when the end of the buffer is reached. If this option is omitted, <i>wrap around</i> mode is used instead. |
| <code>-pci <bufSizeKB></code> | Directs the debug output to a PCI buffer with the specified size. |

Table 5-1. dbgConfig Command Options (Continued)

| Options | Description |
|---|---|
| <code>-rh <stormLetters></code> | Enables the specified STate Optimized RISC Microprocessors (STORMs) for recording handlers <code><stormLetters></code> is a string containing the letters x/y/p/t/m/u (for example, <code>-rh txu</code>) |
| <code>-semFilter <stormLetter></code> <code>eid range <minEid></code> <code><maxEid></code> <code>eid mask <eidVal></code> <code><eidMask></code> <code>cid <cid></code> | Applies the specified STORM processor (SEM) filter. <ul style="list-style-type: none"> ■ Event ID (EID) range filter. Records only data that belongs to handlers with event IDs in the specified range <code><minEid>...<maxEid></code>. ■ EID mask filter. Records only data that belongs to handlers with event IDs that match <code><eidVal></code>. Only bit indexes where <code><eidMask></code> contains ones are compared. ■ Connection ID (CID) filter. Records only data that belongs to handlers with the specified CID <code><cid></code>. |

The recorded binary dump file can be parsed using `parseDebugBusAuto.bat` (in `Z:\\qlogic.org\\il_proj\\`).

debug

Authority `debug set <level> <value>`

Syntax This command changes the debug message level.

The default debug level is NOTICE.

The debug levels (`<level>`) are as follows:

| | |
|---------|---|
| ERR | Only errors are shown. |
| NOTICE | Debug messages with NOTICE level and above are shown. |
| INFO | Debug messages with INFO level and above are shown. |
| VERBOSE | All debug messages are shown. |

device

Authority device [<device number>]

Syntax When used without the <device number> parameter, this command displays a short description of the detected devices of all the Marvell FastLinQ Controllers.

When the <device number> parameter is used, qeDiag selects the device with the given number as the *current* device.

Example

```
1-HwFn0:> device
```

Following is a sample output for Marvell 45000 and 8500 Series Controllers:

```
Chip #1: Board: 57980 Rev: B0 MFW: 8.18.13.0 S.N.: ABC1527A12345 No.ports: 1
-----
Dev Path.Port.PF.HwFn   Bus      PCI   Spd      MAC          Config
-----
  1  00.00.00.00         00:05:00:00 PCIE-8  8.0  00:10:18:00:00:01
  1  00.00.00.01         00:05:00:00 PCIE-8  8.0  00:10:18:00:00:01
```

Following is a sample output for Marvell 41000 and 8600 Series Controllers:

```
Chip #1: Board: 57940 Rev: A2 MFW: 8.18.20.0 S.N.: ABC1527A12345 No.ports: 2
-----
Dev Path.Port.PF.HwFn   Bus      PCI   Spd      MAC          Config
-----
  1  00.00.00.00         00:81:00:00 PCIE-8  8.0  00:10:18:22:33:33
  2  00.01.01.00         00:81:00:01 PCIE-8  8.0  00:10:18:11:11:19
  3  00.00.02.00         00:81:00:02 PCIE-8  8.0  00:10:18:11:11:12
  4  00.01.03.00         00:81:00:03 PCIE-8  8.0  00:10:18:11:11:1A
  5  00.00.04.00         00:81:00:04 PCIE-8  8.0  00:10:18:11:11:13
  6  00.01.05.00         00:81:00:05 PCIE-8  8.0  00:10:18:11:11:1B
```

When the `device` command is issued in `DRV_DBG` mode over the operational driver, it includes additional driver information, as shown in the following example.

```
(1-HwFn0) eth4:> device
```

```
Chip #1: Board: 57940 Rev: A2 MFW: 8.18.20.0 S.N.: ABC1527A12345 No.ports: 2
-----
Dev Path.Port.PF.HwFn   Bus      PCI   Spd      MAC      I/F St. Drv  DrvVer Config
-----
  1  00.00.00.00 00:81:00:00 PCIE-8  8.0  00:10:18:22:33:33 ens4f0 D0 qede 8.18.10.0
  2  00.01.01.00 00:81:00:01 PCIE-8  8.0  00:10:18:11:11:19 ens4f1 D0 qede 8.18.10.0
  3  00.00.02.00 00:81:00:02 PCIE-8  8.0  00:10:18:11:11:12          D0
  4  00.01.03.00 00:81:00:03 PCIE-8  8.0  00:10:18:11:11:1A          D0
  5  00.00.04.00 00:81:00:04 PCIE-8  8.0  00:10:18:11:11:13          D0
  6  00.01.05.00 00:81:00:05 PCIE-8  8.0  00:10:18:11:11:1B          D0
```

driver

The driver commands are described in the following sections.

driver load

Authority `driver load`

Syntax This command sets up the internal qeDiag driver, which is required for initializing the Marvell FastLinQ Controllers and running traffic tests.

The driver loads, initializes the fast path, sets up interrupt requests (IRQs), and uses an Ethernet core (ecore) to initialize the hardware and set up the link. It also starts queues for traffic.

NOTE

The `driver load` command is not available in DRV_DBG mode, where the operational driver is running (`qed/qede` in Linux, `evbd` in Windows).

driver set_loopback

Authority `driver set_loopback <NONE|INT_PHY|EXT_PHY|EXT|MAC>`

Syntax This command sets internal or external loopback.

driver link_state

Authority `driver link_state`

Syntax This command returns the last indicated link state (up or down) and link speed.

driver set_int_mode

Authority `driver set_int_mode <MSIX|MSI|INTA>`

Syntax This command sets the interrupt mode.

NOTE

If the `MSIX` setting fails, the command attempts to set MSI mode. If that fails, the command attempts to set interrupt acknowledge (INTA) mode.

driver unload

Authority `driver unload`

Syntax This command unloads the qeDiag internal driver.

driver init

Authority `driver init`

Syntax This command initializes the device driver, in case it is running in `NO_INIT` mode.

help

Authority `help [<command name>]`

Syntax With no arguments, the `help` command lists all the qeDiag commands in alphabetical order with a short description of each command.

When a `<command name>` parameter is used, the `help` command shows a short description of this command and its usage.

NOTE

The `help` command also shows help on new commands written in Tcl scripts where the script writer used the `add_help` command to add help for the new commands (see [“add_help” on page 19](#)).

extphy

Authority `extphy <fw_upgrade | fw_version | write | read>`
 `extphy fw_upgrade <filename>`
 `extphy fw_version`
 `extphy read <port> <address> <register>`
 `extphy write <port> <address> <register> <value>`

Syntax The types of `extphy` commands are as follows:

| | |
|--------------------------------|---|
| <code>extphy fw_upgrade</code> | Upgrades external PHY firmware with a given file (this upgrade is performed on all relevant ports). |
| <code>extphy fw_version</code> | Gets the current external PHY firmware version |
| <code>extphy read</code> | Reads the register from external PHY (over clause 45) |
| <code>extphy write</code> | Writes the register from external PHY (over clause 45) |

hmem

The `hmem` commands are described in the following sections.

hmem palloc

Authority `hmem palloc <byte count> [<description string>]`

Syntax This command allocates a continuous memory region of `<byte count>` bytes and returns the virtual address of the allocated buffer.

If the `<description string>` parameter is used, the allocated memory is associated with the `<description string>` name.

hmem paddr

Authority `hmem paddr <low32addr> [<high32addr>]`

Syntax This command returns the physical address of a buffer with a virtual address. This command is applicable only when memory is allocated using the `hmem palloc` command (see [“hmem palloc” on page 24](#)).

hmem read

Authority `hmem read <low32addr> [<high32addr>]`

Syntax This command reads a double word (DWORD) at the specified address from host memory.

hmem write

Authority `hmem write <low32addr> [<high32addr>] <value>`

Syntax This command writes a DWORD at the specified address to host memory.

grcDump

Authority `grcDump <dirName> [<options>]`

Syntax This command performs a global register controller (GRC) dump into a binary file.

By default, all memories are included in the dump, except I/O registers (IORs), virtual static classification (VFC), and static debug.

Memories can be included or excluded from the dump, overriding the default.

The supported memories are as follows:

| | |
|------|----------|
| ram | qm |
| pbuf | mcp |
| ior | mac_stat |
| vfc | cfc |
| ctx | igu |
| pxp | brb |
| rss | nig |
| cau | |

Use `regs` as a memory name for a register dump.

Use `static` as a memory name for a static debug dump.

The `<options>` parameters are described in [Table 5-2](#).

Table 5-2. grcDump Options Parameters

| Parameter | Description |
|--|---|
| <code>-exclude <memName1> <memName2>...</code> | Excludes the specified memories from the dump. <memName#> is a memory name (for example, <code>-exclude ctx qm pxp</code>). |
| <code>-include <memName1> <memName2>...</code> | Includes the specified memories in the dump. <memName#> is a memory name (for example, <code>-include ior vfc</code>). |

Table 5-2. grcDump Options Parameters (Continued)

| Parameter | Description |
|---|--|
| <code>-only <memName1> <memName2>...</code> | Includes only the specified memories in the dump. All other memories are excluded. <memName#> is a memory name (for example, <code>-only ctx ram</code>). |
| <code>-info</code> | If specified, only system information is dumped (all memories are excluded). |
| <code>-storms <stormLetters></code> | Includes the specified STORMs in the dump. Other STORMs are excluded. By default, all STORMs are included. <stormLetters> is a string containing x/y/p/t/m/u (for example, <code>-storms txu</code>). |
| <code>-lcids <numLcids></code> | Specifies the number of local connection IDs (LCIDs) to dump. <numLcids> is the number of LCIDs to dump (for example, <code>-lcids 4</code>). |
| <code>-ltids <numLtids></code> | Specifies the number of local task IDs (LTIDs) to dump. <numLtids> is the number of LTIDs to dump (for example, <code>-ltids 4</code>). |
| <code>-unstall</code> | If specified, the STORMs are un-stalled after the dump. This parameter is valid only if IORs or VFC are dumped, in which case the STORMs are stalled automatically before the dump. |

idleChk

Authority idleChk

Syntax This command checks that the Marvell FastLinQ Controllers are in the *idle* state. This command generates an error when it detects register values that are not as expected in the idle state.

NOTE

Perform the `idleChk` command only after the driver is loaded.

Example

```
1-HwFn0:> driver load
```

Following is a sample output for Marvell 45000 and 8500 Series Controllers.

```
1-HwFn0:> idleChk
fw-version: 8_18_20_0
fw-image: main
mfw-version: 8_18_13_0
tools-version: 10
chip: bb
platform: asic
pci-func: 0
dump-type: idle-chk

FW_IDLE_CHECK:
Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at
least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

LSI_IDLE_CHECK:
Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at
least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

Idle Check completed successfully (with 1 warnings)
```

Following is a sample output for Marvell 41000 and 8600 Series Controllers.

```
1-HwFn0:> idleChk
fw-version: 8_18_20_0
fw-image: main
mfw-version: 8_18_20_0
```

```
tools-version: 10
chip: ah
platform: asic
pci-func: 0
dump-type: idle-chk
```

FW_IDLE_CHECK:

Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

LSI_IDLE_CHECK:

Warning: GRC: GRC_REG_TRACE_FIFO_VALID_DATA = 1, the trace FIFO contains at least one valid data.. Registers: GRC_REG_TRACE_FIFO_VALID_DATA=0x1

Idle Check completed successfully (with 1 warnings)

l2pkt

The l2pkt commands are described in the following sections.

l2pkt send

Authority l2pkt send <no. packets> <receive packets 0|1>

Syntax This command sends and receives L2 packets and performs an optional data integrity check.

NOTE

The L2 packet is a fixed length (1,024 bytes) and has fixed MAC addresses.
The data is fixed with the packet index.

l2pkt receive

Authority l2pkt receive <no. packets>

Syntax This command receives L2 packets and performs an optional data integrity check.

led

Authority `led on | off | operate`

Syntax This command turns the device LED on and off. The `led operate` command returns the LED to normal operating mode.

linkdump

Authority `linkdump [port]`

Syntax This command shows link information for the port specified in the `[port]` parameter. If this parameter is not used, this command shows the link information for all ports.

Examples

The following example shows the output for the `linkdump` command when a port is not specified. An explanation of some of the `ITEMS` in this list follows the examples.

```
1-HwFn0:> linkdump
```

```
Chip mode: 2x25
```

```
-----
ITEM                PORT0                PORT1
-----
MFW LINK STATE      UP                    DOWN
  PCS LINK           Yes                    NO
  MAC FAULT          None                   None
  RX SIGNAL DETECT   Yes                    No
LINK SPEED           25G
MODULE               25G Cu Short   Undefined
FLOW CTRL            RX/TX PAUSE
FEC MODE             NONE                   NONE
FEC COR_ERR_CNT      N.A.                   N.A.
FEC UNCOR_ERR_CNT    N.A.                   N.A.
AUTONEG              DISABLED                ENABLED
AN LP ADV SPEEDS     0x00
AN LP ADV FLOW CTRL  No pause adv
```

The following example shows a sample output for the `linkdump` command when port 0 is specified. An explanation of some of the `ITEMS` in this list follows the example.

```
1-HwFn0:> linkdump 0
Chip mode: 2x25
-----
ITEM                                PORT0
-----
MFW LINK STATE                     DOWN
PCS LINK                           NO
MAC FAULT                           None
RX SIGNAL DETECT                    No
LINK SPEED
MODULE                             25G Cu Long
FLOW CTRL
FEC MODE                           NONE
FEC COR_ERR_CNT                     N.A.
FEC UNCOR_ERR_CNT                   N.A.
AUTONEG                             ENABLED
AN LP ADV SPEEDS                    0x00
AN LP ADV FLOW CTRL                 No pause adv
TX PRE FIR                          0
TX MAIN FIR                         0
TX POST FIR                         0
RX DFE                              OFF
LINK CHANGE COUNT                   0
LFA COUNT                           0
LFA STATE                           0
MFW RAW LINK STATUS                 0x00000020
```

Where:

| | |
|------------------|--|
| Chip mode | Speed mode (for example, 2x25, 2x50) |
| MFW LINK STATE | Link state (UP or DOWN) as reported by the management firmware (MFW) |
| PCS LINK | Link state as reported by the physical coding sublayer (PCS) block |
| MAC FAULT | Local or remote MAC fault state |
| RX SIGNAL DETECT | Signal detected by the receiver |

| | |
|----------------------|--|
| LINK SPEED | Link speed. Valid values are: <ul style="list-style-type: none"> ■ 1G ■ 10G ■ 25G ■ 40G (Marvell 45000 Series Controllers) ■ 50G (Marvell 45000 Series Controllers) ■ 100G(Marvell 45000 Series Controllers) |
| MODULE | Type of pluggable transceiver installed |
| FLOW CTRL | Flow control type/setting |
| FEC MODE | Forward error correction (FEC) mode. Valid values are: <ul style="list-style-type: none"> ■ None ■ Firecode ■ Reed Solomon |
| AUTONEG | Auto-negotiation state. Valid values are: <ul style="list-style-type: none"> ■ Complete ■ Enabled ■ Disabled |
| AN LP ADV SPEEDS | Link partner speed advertisements (at the time of publication, this information was not available). |
| AN LP ADV FLOW CTRL | Link partner flow control advertisements (at the time of publication, this information was not available). |
| TX PRE/MAIN/POST FIR | Transmit equalization settings |
| RX DFE <ON OFF> | Part of the receive equalization |
| LINK CHANGE COUNT | Displays the number of times the link status changed from down to up and from up to down. |

log

The log commands save the execution results to a log file. The log commands are described in the following sections.

log open

Authority `log open <logfile>`

Syntax This command opens a log file and starts logging all displayed information to the log file.

log close

Authority `log close`

Syntax This command closes and saves the log file.

mcp

Authority `mcp run`
 `mcp halt`
 `mcp gpr [reg value]`
 `mcp ctrl [reg value]`
 `mcp step [number of steps] [none|both|ctrl|gpr]`
 `mcp trace [-offline | -continuously | -nonstop | -stop] [filename]`
 `mcp access addr [value]`
 `mcp reset`
 `mcp unbreak`
 `mcp parse_trace`
 `mcp enable_don`
 `mcp set_trace_filter`
 `mcp restore_trace_filter`

Syntax This command implements a debugger for the on-chip millions of instructions per second (MIPS) processor.

The `mcp trace` command displays management firmware (MFW) trace messages.

Trace messages are either displayed on the console or saved to text file.

If MFW does not respond, the `mcp trace` data is saved in a binary file named `McpTrace.bin`.

To view the binary trace file on the console, use the command

```
mcp parse_trace <McpTrace.bin> <meta file>
```

Following is an example of the `mcp trace` command output when the MFW does not respond.

```
1-HwFn0:> mcp trace
[ecore_do_mcp_cmd:296(ediag0-hwfn-0)]MFW failed to respond!
[os_if_mcp_nvme_rd_cmd:1706] ecore_mcp_nvme_rd_cmd 196608 failed !
[ecore_do_mcp_cmd:296(ediag0-hwfn-0)]MFW failed to respond!
[os_if_mcp_nvme_rd_cmd:1706] ecore_mcp_nvme_rd_cmd 196608 failed !
dbg_mcp_trace_read_data: reading trace data of size 2076B from GRC
address 0x e20014
[ecore_do_mcp_cmd:296(ediag0-hwfn-0)]MFW failed to respond!
[os_if_mcp_nvme_rd_cmd:1706] ecore_mcp_nvme_rd_cmd 196608 failed !
Reading NVM_TYPE_MFW_TRACE failed. Saved results to
McpTrace8070300.bin
```

```
1-HwFn0:> mcp parse_trace /grc2/McpTrace8070300.bin
/nfs/usr/mcp/579xx/8.7.3.0/b900_trace_meta.img
```

To obtain the meta file of a specific management firmware (MFW) version from the network location, use the command

```
mcp parse_trace <McpTrace.bin> -ah | -bb <MFW version>
```

The `mcp enable_don` command enables the diagnostics over network (DON) mode. A password is required to enable this mode.

The `-nonstop` and `-stop` parameters are available only in DON mode.

The `mcp trace -nonstop` command continuously sends mcp traces into a file with a time stamp.

For example, in Linux:

```
1-HwFn0:> mcp trace -nonstop /tmp/trace.txt
```

For example, in Windows:

```
1-HwFn0:> mcp trace -nonstop "c:\\tmp\\trace.txt"
```

To continuously view the traces, issue the following command on another shell window:

```
tail -F -n+0 /tmp/trace.txt
```

The `mcp set_trace_filter` command changes the level and/or modules that will be tracked by the MFW. This command sends a mailbox to the MFW and does not require an `mcp reset` command for the changes to take effect.

The `mcp restore_trace_filter` command restores the level and modules (according to NVM configuration) that will be tracked by MFW. This command sends a mailbox to the MFW and does not require an `mcp reset` command for the changes to take effect.

Example

Usage: show trace modules according to requested module bits:

1-HwFn0:> mcp_set_trace_filter

`mcp_set_trace_filter <level> [module bit# | all]`

`<level> - 1-DEBUG 2-TRACE 3-ERR`

`<module bit#>:`

| | | | |
|---------------|--------------|---------------|-------------|
| 0-ERROR | 1-DBG | 2-DRV_HSI | 3-INTERRUPT |
| 4-TEMPERATURE | 5-FLR | 6-INIT | 7-NVM |
| 8-PIM | 9-NET | 10-POWER | 11-UTILS |
| 12-RESOURCES | 13-SCHEDULER | 14-PHYMOD | 15-EVENTS |
| 16-PMM | 17-DBG_DRV | 18-ETH | 19-SECURITY |
| 20-PCIE | 21-TRACE | 22-MANAGEMENT | 23-SIM |
| 24-BUF_MGR | | | |

1-HwFn0:> `mcp set_trace_filter 1 0 4 8`

`[mcp_set_trace_filter_cmd:1865] trace level=0x1 modules=0xfffffee`

mdump

Authority `mdump clear_img`

Syntax This command clears the NVRAM mdump to all zeros.

The mdump image is also referred to as `CRASH_DUMP`.

In Linux operating systems, the operational drivers must be unloaded for this command to work.

nictest

Authority This command performs device tests; for example, registers, memory, and traffic tests.

```
nictest [-t <groups>] [-T <groups>] [-I <num>] [-cof] [-debug]
-none] [-mem_fast]
```

Where:

| | |
|-----------|---|
| -t | Excludes specified tests |
| -T | Includes specified tests |
| -I | Specifies how many times the tests are run |
| -cof | Allows testing to continue after a test failure |
| -debug | Runs the test with debug messages |
| -none | Disables all tests |
| -mem_fast | Runs memory tests in fast mode. Only start, middle, and end addresses are tested. |

Syntax This command performs the functional testing of the Marvell FastLinQ Controllers.

| | |
|---------|-------------------|
| Group A | Registers tests |
| Group B | Memory tests |
| Group C | Block tests (TBD) |
| Group D | Traffic tests |

NOTE

To run a specific group of tests, use the `-t` parameter to disable other groups of tests.

At the time of publication, the following new tests have been added to the Marvell QL41000 and 8600 Series Controllers: A3 BMBN, A4 BMB, A9 DBU, A21 MSTAT, A24 NWM, A30 PCIE_EP, A67 WOL, A68 LED, A69 CNIG.

Do not use the `-none -T <group>` parameter, since it enables all tests in the group, including the ones that are known to fail.

Example

Following is an example output for the Marvell QL41000 and 8600 Series Controllers.

```
1-HwFn0:> nictest
```

5-Engineering Mode Commands

nictest

Tests running iteration no. 1 out of 1 iterations on device 1 hwfn 0

Nicetest iteration 1

Group A. Basic Tests (Registers, PCI ect.) on device 1 (hwfn 0)

A01. Block BRB Register Test.....: passed
A02. Block BTB Register Test.....: passed
A03. Block BMBN Register Test.....: passed
A04. Block BMB Register Test.....: passed
A05. Block CAU Register Test.....: passed
A06. Block CCFC Register Test.....: passed
A07. Block CDU Register Test.....: passed
A08. Block DBG Register Test.....: passed
A09. Block DBU Register Test.....: passed
A10. Block DMAE Register Test.....: passed
A11. Block DORQ Register Test.....: passed
A12. Block GRC Register Test.....: passed
A13. Block IGU Register Test.....: passed
A15. Block MCM Register Test.....: passed
A16. Block MCP2 Register Test.....: passed
A17. Block MISC Register Test.....: passed
A18. Block MISCS Register Test.....: passed
A19. Block MSDM Register Test.....: passed
A20. Block MSEM Register Test.....: passed
A21. Block MSTAT Register Test.....: passed
A22. Block NIG Register Test.....: passed
A23. Block NCSI Register Test.....: passed
A24. Block NWM Register Test.....: passed
A25. Block PBF Register Test.....: passed
A26. Block PBF_PB1 Register Test.....: passed
A27. Block PBF_PB2 Register Test.....: passed
A28. Block PTU Register Test.....: passed
A29. Block PCM Register Test.....: passed
A30. Block PCIE_EP Register Test.....: passed
A31. Block PRM Register Test.....: passed
A32. Block PRS Register Test.....: passed
A33. Block PSDM Register Test.....: passed
A34. Block PSEM Register Test.....: passed
A35. Block PSWRQ Register Test.....: passed
A36. Block PSWRQ2 Register Test.....: passed
A37. Block PSWHST2 Register Test.....: passed

A38. Block PSWRD Register Test.....: passed
A39. Block PSWRD2 Register Test.....: passed
A40. Block PSWWR Register Test.....: passed
A41. Block PSWWR2 Register Test.....: passed
A42. Block QM Register Test.....: passed
A43. Block RSS Register Test.....: passed
A44. Block RDIF Register Test.....: passed
A45. Block SRC Register Test.....: passed
A46. Block TCFC Register Test.....: passed
A47. Block TCM Register Test.....: passed
A48. Block TM Register Test.....: passed
A49. Block TSDM Register Test.....: passed
A50. Block TSEM Register Test.....: passed
A51. Block TDIF Register Test.....: passed
A52. Block UCM Register Test.....: passed
A53. Block USDM Register Test.....: passed
A54. Block USEM Register Test.....: passed
A55. Block XCM Register Test.....: passed
A56. Block XPB1 Register Test.....: passed
A57. Block XSDM Register Test.....: passed
A58. Block XSEM Register Test.....: passed
A59. Block YCM Register Test.....: passed
A60. Block YSDM Register Test.....: passed
A61. Block YSEM Register Test.....: passed
A62. Block XYLD Register Test.....: passed
A63. Block TMLD Register Test.....: passed
A64. Block MULD Register Test.....: passed
A65. Block YULD Register Test.....: passed
A67. Block WOL Register Test.....: passed
A68. Block LED Register Test.....: passed
A69. Block CNIG Register Test.....: passed
Group B. Memory Tests on device 1 (hwfn 0)
B01. Block BRB Memory Test.....: passed
B02. Block BTB Memory Test.....: passed
B03. Block CCFC Memory Test.....: passed
B04. Block CDU Memory Test.....: passed
B05. Block DBG Memory Test.....: passed
B06. Block DMAE Memory Test.....: passed
B07. Block DORQ Memory Test.....: passed

```

B08. Block IGU Memory Test.....: passed
B09. Block MCM Memory Test.....: passed
B10. Block MSDM Memory Test.....: passed
B11. Block MSEM Memory Test.....: passed
B12. Block NIG Memory Test.....: passed
B13. Block PBF_PB2 Memory Test.....: passed
B14. Block PCM Memory Test.....: passed
B15. Block PGLUE_B Memory Test.....: passed
B16. Block PRS Memory Test.....: passed
B17. Block PSDM Memory Test.....: passed
B18. Block PSEM Memory Test.....: passed
B19. Block PSWHST Memory Test.....: passed
B20. Block QM Memory Test.....: passed
B21. Block SRC Memory Test.....: passed
B22. Block TCFC Memory Test.....: passed
B23. Block TCM Memory Test.....: passed
B24. Block TM Memory Test.....: passed
B25. Block TSDM Memory Test.....: passed
B26. Block TSEM Memory Test.....: passed
B27. Block UCM Memory Test.....: passed
B28. Block USDM Memory Test.....: passed
B29. Block USEM Memory Test.....: passed
B30. Block XCM Memory Test.....: passed
B31. Block XSDM Memory Test.....: passed
B32. Block XSEM Memory Test.....: passed
B33. Block YCM Memory Test.....: passed
B34. Block YSDM Memory Test.....: passed
B35. Block YSEM Memory Test.....: passed
B36. Block XYLD Memory Test.....: passed
B37. Block TMLD Memory Test.....: passed
B38. Block MULD Memory Test.....: passed
B39. Block YULD Memory Test.....: passed
Group D. Traffic Tests on device 1 (hwfn 0)
D01. L2 Traffic Internal Loopback Test.....: passed
Nictest completed with 0 failures, 107 successes. Total number of tests 107.

```

Following is an example output for the Marvell QL45000 and 8500 Series Controllers.

```

1-HwFn0:> nictest -t
D2

```

Nicetest iteration 1

Group A. Basic Tests (Registers, PCI ect.) on device 1 (hwfn 0)

A01. Block BRB Register Test.....: passed

A02. Block BTB Register Test.....: passed

A05. Block CAU Register Test.....: passed

A06. Block CCFC Register Test.....: passed

A07. Block CDU Register Test.....: passed

A08. Block DBG Register Test.....: passed

A10. Block DMAE Register Test.....: passed

A11. Block DORQ Register Test.....: passed

A12. Block GRC Register Test.....: passed

A13. Block IGU Register Test.....: passed

A15. Block MCM Register Test.....: passed

A16. Block MCP2 Register Test.....: passed

A17. Block MISC Register Test.....: passed

A18. Block MISCS Register Test.....: passed

A19. Block MSDM Register Test.....: passed

A20. Block MSEM Register Test.....: passed

A22. Block NIG Register Test.....: passed

A23. Block NCSI Register Test.....: passed

A25. Block PBF Register Test.....: passed

A26. Block PBF_PB1 Register Test.....: passed

A27. Block PBF_PB2 Register Test.....: passed

A28. Block PTU Register Test.....: passed

A29. Block PCM Register Test.....: passed

A31. Block PRM Register Test.....: passed

A32. Block PRS Register Test.....: passed

A33. Block PSDM Register Test.....: passed

A34. Block PSEM Register Test.....: passed

A35. Block PSWRQ Register Test.....: passed

A36. Block PSWRQ2 Register Test.....: passed

A37. Block PSWHST2 Register Test.....: passed

A38. Block PSWRD Register Test.....: passed

A39. Block PSWRD2 Register Test.....: passed

A40. Block PSWWR Register Test.....: passed

A41. Block PSWWR2 Register Test.....: passed

A42. Block QM Register Test.....: passed
A43. Block RSS Register Test.....: passed
A44. Block RDIF Register Test.....: passed
A45. Block SRC Register Test.....: passed
A46. Block TCFC Register Test.....: passed
A47. Block TCM Register Test.....: passed
A48. Block TM Register Test.....: passed
A49. Block TSDM Register Test.....: passed
A50. Block TSEM Register Test.....: passed
A51. Block TDIF Register Test.....: passed
A52. Block UCM Register Test.....: passed
A53. Block USDM Register Test.....: passed
A54. Block USEM Register Test.....: passed
A55. Block XCM Register Test.....: passed
A56. Block XPB1 Register Test.....: passed
A57. Block XSDM Register Test.....: passed
A58. Block XSEM Register Test.....: passed
A59. Block YCM Register Test.....: passed
A60. Block YSDM Register Test.....: passed
A61. Block YSEM Register Test.....: passed
A62. Block XYLD Register Test.....: passed
A63. Block TMLD Register Test.....: passed
A64. Block MULD Register Test.....: passed
A65. Block YULD Register Test.....: passed

Group B. Memory Tests on device 1 (hwfn 0)

B01. Block BRB Memory Test.....: passed
B02. Block BTB Memory Test.....: passed
B03. Block CCFC Memory Test.....: passed
B04. Block CDU Memory Test.....: passed
B05. Block DBG Memory Test.....: passed
B06. Block DMAE Memory Test.....: passed
B07. Block DORQ Memory Test.....: passed

B08. Block IGU Memory Test.....: passed
B09. Block MCM Memory Test.....: passed
B10. Block MSDM Memory Test.....: passed
B11. Block MSEM Memory Test.....: passed
B12. Block NIG Memory Test.....: passed
B13. Block PBF_PB2 Memory Test.....: passed

```
B14. Block PCM Memory Test.....: passed
B15. Block PGLUE_B Memory Test.....: passed
B16. Block PRS Memory Test.....: passed
B17. Block PSDM Memory Test.....: passed
B18. Block PSEM Memory Test.....: passed
B19. Block PSWHST Memory Test.....: passed
B20. Block QM Memory Test.....: passed
B21. Block SRC Memory Test.....: passed
B22. Block TCFC Memory Test.....: passed
B23. Block TCM Memory Test.....: passed
B24. Block TM Memory Test.....: passed
B25. Block TSDM Memory Test.....: passed
B26. Block TSEM Memory Test.....: passed
B27. Block UCM Memory Test.....: passed
B28. Block USDM Memory Test.....: passed
B29. Block USEM Memory Test.....: passed
B30. Block XCM Memory Test.....: passed
B31. Block XSDM Memory Test.....: passed
B32. Block XSEM Memory Test.....: passed
B33. Block YCM Memory Test.....: passed
B34. Block YSDM Memory Test.....: passed
B35. Block YSEM Memory Test.....: passed
B36. Block XYLD Memory Test.....: passed
B37. Block TMLD Memory Test.....: passed
B38. Block MULD Memory Test.....: passed
B39. Block YULD Memory Test.....: passed
Group C. Block Tests on device 1 (hwfn 0)
  C01. CAM BIST Test.....: passed
Group D. Traffic Tests on device 1 (hwfn 0)
  D01. L2 Traffic Internal Loopback Test.....: passed
Nictest completed with 0 failures, 99 successes. Total number of tests 99.
0
log close
```

nvm

These commands provide access to the Marvell FastLinQ Controllers' NVRAM. The `nvm` commands are described in the following sections.

nvm cfg

Authority

```
nvm cfg
nvm cfg <option=choice>
nvm cfg -all <option=choice>
nvm cfg -all <option=>
nvm cfg -func <func#> <op>=<data>
nvm cfg -func <func#> <op>=
nvm cfg -find <option | board>
nvm cfg <option> -
nvm cfg <board type>
nvm cfg -dump <filename>
```

Syntax This command implements both an interactive and scripted mode of operation that allows you to view and modify the Marvell FastLinQ Controllers' configuration, as described in the following sections.

nvm cfg

Authority `nvm cfg`

Syntax When invoked without any options, this command lists the current NVRAM configuration, and provides an option to change the configuration and save the results.

Each option is described with one of the following:

```
[GLOB]   Global
[PORT]   Per-port option
[FUNC]   Per-function option
```

Example

```
1-HwFn0:> nvm cfg
NVRAM Configuration Groups:
-----
1 .      VF
2 .      board
3 .      board io
4 .      features
```

```

5 .      link settings
6 .      mng_fw
7 .      pcie
8 .      phy
9 .      pre-boot
Select: (q to quit)

```

nvm cfg <option>=<value>

Authority To set a specific configuration option, use the `nvm cfg <option>=<value>` command.

To set an option that includes a string with spaces, use curly braces {}. For example:

```
nvm cfg {option=string}
```

When an option is used that requires a system reboot, the following warning message appears:

```
Please reboot the system for change to take effect
```

Syntax This command configures a specific option in NVRAM.

nvm cfg -all <option>=<value>

Authority `nvm cfg -all <option>=<value>`

Syntax This command sets an NVRAM configuration option with the same value on all devices with a single command.

nvm cfg -all <option>=

Authority `nvm cfg -all <option>=`

Syntax This command retrieves the NVRAM configuration option value from all devices with a single command.

nvm cfg -func <func#><op>=<data>

Authority `nvm cfg -function <func#><op>=<data>`

Syntax This command sets an NVRAM configuration option with a value on a different function than the current function.

nvm cfg -func <func#><op>=

Authority `nvm cfg -function <func#><op>=`

Syntax This command retrieves the NVRAM configuration option value from a different function than the current function.

nvm cfg -find <option | board>

Authority `nvm cfg -find <option | board>`

Syntax This command finds an NVRAM configuration option or a board name using part of the name.

Example To find all NVRAM configuration options that have `wol` in their name, issue the following command:

```
nvm cfg -find wol
```

To find all boards that have `xxx` in their name, issue the following command:

```
nvm cfg -find xxx
```

nvm cfg <option>-

Authority `nvm cfg <option>-`

Syntax This command obtains the configuration value and the description of a specific option.

nvm cfg <board type>

Authority `nvm cfg <board type>`

Syntax This command sets the default choices for the requested board type.

nvm cfg -dump <filename>

Authority `nvm cfg -dump <file name>`

Syntax This command dumps the NVRAM configuration into the file.
The output file is in Tcl format and can be sourced to configure a similar board.

Example Following is the beginning of the output file. In the file, `Eagle` indicates MRS1 Lane and `Falcon` indicates MRS0 Lane.

NOTE

This sample output references FCoE and 50G speeds, which are not supported in the Marvell 8500 and 8600 Series Controllers.

The following output file is applicable to the Marvell 41000 and 45000 Series Controllers.

```
#####
# DO NOT EDIT!!! THIS FILE IS AUTOMATICALLY GENERATED.
#
#Copyright(c) 2014-2017 Cavium, Inc., all rights reserved
#Proprietary and Confidential Information.
#
#
#Description: This file contains the NVRAM configuration for <board name>
#             To incorporate these settings into the board's NVRAM, simply
#             source this file in Eddiag as follows
#             source <config_file>
#
#Design Name: <board name>
#Diagnostic Tool Version: eDiag version 8.40.9.0

#Software Release:
#Customer Name:
#Revision History:
#
#Manufacture kit version: 08:03:21:00:09
#Additional Comments:

#####
### Device 1 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                : 00:10:18:11:22:33
# 8: [GLOB] Board Swap
#      {None(0), Path(1), Port(2), Both(3)}      : None
# 9: [GLOB] MF Mode
#      {MF allowed(0), Default(1), SPI04(2),
#      NPAR1.0(3), NPAR1.5(4), NPAR2.0(5), BD(6),
#      UFP(7), DCI_NPAR(8)}                      : Default
# 10: [PORT] Led Mode
#      {Mac1(0), Phy1(1), Phy2(2), Phy3(3), Mac2(4),
#      Phy4(5), Phy5(6), Phy6(7), Mac3(8), Phy7(9),
#      Phy8(10), Phy9(11), Mac4(12), Phy10(13),
#      Phy11(14), Phy12(15), Breakout(16),
```

5-Engineering Mode Commands

nvm

```
#          OCP_3_0(17)}          : Mac1
# 89: [GLOB] Manufacture kit version      : 08:03:21:00:09
# 90: [GLOB] Manufacture timestamp        : 03:06:17
#102: [PORT] LED speed select (S2:S1:S0)
#          {1G(0x1), 10G(0x2), 25G(0x4), 40G(0x8),
#          50G(0x10)}          : 0:1:2
#103: [GLOB] LED port swap (P3:P2:P1:P0) : 1:0:3:2
#105: [GLOB] Override Secure Mode {Disabled(0), Enabled(1)} : Disabled
#125: [GLOB] AUX mode {Default(0), SMBUS only(1)} : Default
#197: [GLOB] NVM_CFG version            : 0
#198: [GLOB] NVM_CFG new option seq      : 21
#199: [GLOB] NVM_CFG removed option seq  : 1
#200: [GLOB] NVM_CFG updated value seq   : 3
#245: [GLOB] Main Clock Frequency
#          {MAIN_CLK_DEFAULT(0), MAIN_CLK_375(1),
#          MAIN_CLK_350(2), MAIN_CLK_325(3),
#          MAIN_CLK_300(4), MAIN_CLK_280(5)} : MAIN_CLK_DEFAULT
#246: [GLOB] Mac Clock Frequency
#          {MAC_CLK_DEFAULT(0), MAC_CLK_782(1),
#          MAC_CLK_516(2)} : MAC_CLK_DEFAULT
#247: [GLOB] Storm Clock Frequency
#          {STORM_CLK_DEFAULT(0), STORM_CLK_1200(1),
#          STORM_CLK_1000(2), STORM_CLK_900(3),
#          STORM_CLK_1100(4)} : STORM_CLK_DEFAULT
#250: [PORT] UID LED Mode Mask
#          {Activity_LED(0x1), Link_LED0(0x2),
#          Link_LED1(0x4), Link_LED2(0x8)} : 0x0
#
#
#          Group: board io (Group 2)
#-----
# 11: [GLOB] Fan Failure Enforcement
#          {Disabled(0), Enabled(1)} : Disabled
#114: [PORT] Transceiver Module Absent
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#126: [PORT] Port Type
```

5-Engineering Mode Commands

nvm

```
#           {Undefined(0), Module(1), Backplane(2),
#           Ext_phy(3), Module_slave(4)}           : Ext_phy
#127: [PORT] Tx Disable
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)}           : NA
#134: [GLOB] Temperature period between checks           : 1
#135: [GLOB] Shutdown threshold temperature           : 110
#136: [GLOB] max count oper threshold           : 5
#137: [GLOB] Dead Temp TH temperature           : 105
#139: [GLOB] Temperature monitoring mode
#           {Disable(0), Internal(1), External(2), Both(3)} : Internal
#144: [GLOB] Fan Failure duration           : 5
#204: [GLOB] Runtime Port Swap GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)}           : NA
#205: [GLOB] Runtime Port Swap Map (P3:P2:P1:P0)           : 0:0:0:0
#206: [GLOB] Thermal Event GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)}           : GPIO18
#207: [GLOB] I2C Interrupt GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
```

5-Engineering Mode Commands

nvm

```
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32) } : NA
#217: [GLOB] ALOM FAN ON AUX GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32) } : NA
#218: [GLOB] ALOM FAN ON AUX value : 95
#219: [GLOB] SLOT ID GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32) } : NA
#220: [GLOB] PMBUS SCL GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32) } : GPIO29
#221: [GLOB] PMBUS SDA GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
```

5-Engineering Mode Commands

nvm

```
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : GPIO31
#272: [PORT] SmartAN FEC Override {Disabled(0), Enabled(1)} : Disabled
#275: [PORT] Transceiver Rate Select 0
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#276: [PORT] Transceiver Rate Select 1
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#277: [PORT] Transceiver Module TX Fault
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#278: [PORT] Transceiver QSFP Module Reset
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
```

5-Engineering Mode Commands

nvm

```
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#279: [PORT] Transceiver QSFP LP Mode
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#280: [PORT] Transceiver Power Enable
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#281: [PORT] LASI Interrupt input
#          {NA(0), LASI0(1), LASI1(2), LASI2(3), LASI3(4)} : NA
#282: [GLOB] EXT_PHY_PGOOD input
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#286: [GLOB] Warning Temperature GPIO
```

5-Engineering Mode Commands

nvm

```
# {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
# GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
# GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
# GPIO12(13), GPIO13(14), GPIO14(15),
# GPIO15(16), GPIO16(17), GPIO17(18),
# GPIO18(19), GPIO19(20), GPIO20(21),
# GPIO21(22), GPIO22(23), GPIO23(24),
# GPIO24(25), GPIO25(26), GPIO26(27),
# GPIO27(28), GPIO28(29), GPIO29(30),
# GPIO30(31), GPIO31(32)} : NA

#288: [GLOB] PERST indication GPIO
# {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
# GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
# GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
# GPIO12(13), GPIO13(14), GPIO14(15),
# GPIO15(16), GPIO16(17), GPIO17(18),
# GPIO18(19), GPIO19(20), GPIO20(21),
# GPIO21(22), GPIO22(23), GPIO23(24),
# GPIO24(25), GPIO25(26), GPIO26(27),
# GPIO27(28), GPIO28(29), GPIO29(30),
# GPIO30(31), GPIO31(32)} : NA
#
#
# Group: pcie (Group 6)
#-----
# 12: [GLOB] Engineering Change :
# 13: [GLOB] Manufacturing ID : 1077
# 14: [GLOB] Serial Number : ABC1527A12345
# 15: [GLOB] PCI Gen : PCI Gen3
# {PCI Gen1(0), PCI Gen2(1), PCI Gen3(2)} : PCI Gen3
# 16: [GLOB] Beacon WOL_Enabled {Disabled(0), Enabled(1)} : Disabled
# 17: [GLOB] ASPM Support
# {L0s L1 enabled(0), L0s disabled(1), L1
# disabled(2), L0s L1 disabled(3)} : L0s disabled
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 76: [GLOB] Vendor ID : 0x1077
# 78: [GLOB] Subsystem Vendor ID : 0x1077
# 79: [GLOB] Subsystem Device ID : 0x0001
# 82: [FUNC] Bar1 size
# {Disabled(0), 64K(1), 128K(2), 256K(3),
# 512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
# 32M(10), 64M(11), 128M(12), 256M(13),
# 512M(14), 1G(15)} : 128K
# 98: [GLOB] PCIE Preemphasis
```

5-Engineering Mode Commands

nvm

```
#           {HW(0), 0dB(1), 3_5dB(2), 6_0dB(3)}           : HW
#128: [GLOB] Max link width
#           {1 lane(1), 2 lanes(2), 4 lanes(3), 8 lanes(4)} : 8 lanes
#130: [GLOB] ASPM L1 mode
#           {Forced(0), Dynamic low latency(1)}           : Forced
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                             : 8M
#201: [GLOB] Extended serial number                       :
#210: [GLOB] OEM1 Number                                  : 123456
#211: [GLOB] OEM2 Number                                  : 123456
#248: [GLOB] PCIE Relaxed Ordering
#           {Disabled(0), Enabled(1)}                     : Disabled
#274: [GLOB] Short PERST Protection
#           {Disabled(0), Enabled(1)}                     : Disabled
#289: [GLOB] PCIe Class Code FCoE                         : 0x0
#290: [GLOB] PCIe Class Code iSCSI                       : 0x0
#
#
#                                     Group: features (Group 3)
#-----
# 20: [PORT] RoCE priority                                 : 0
# 22: [PORT] Enable WoL on ACPI pattern
#           {Disabled(0), Enabled(1)}                     : Disabled
# 23: [PORT] Magic Packet WoL {Disabled(0), Enabled(1)}   : Disabled
# 26: [PORT] DCBX Mode
#           {Disabled(0), IEEE(1), CEE(2), Dynamic(3)}    : Dynamic
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                       : 00:10:18:11:22:33
# 94: [FUNC] FCoe port wwn mac addr                      : 00:10:18:11:22:33
# 95: [FUNC] Bandwidth weight                             : 0
# 96: [FUNC] Max Bandwidth                                : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
# 99: [PORT] LLDP MAC address                             : 00:10:18:11:22:43
#100: [GLOB] FCoE wwn node prefix                         : 20:00
#101: [GLOB] FCoE wwn port prefix                        : 20:01
#107: [GLOB] MBI version                                  : 00:00:00
#108: [GLOB] MBI date                                     : 0x0
#117: [GLOB] Device capabilities
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RoCE(0x8), iWarp(0x10)}                       : 0x1f
#118: [PORT] ETH DID suffix                               : 0x70
#119: [PORT] FCoE DID suffix                             : 0x80
#120: [PORT] iSCSI DID suffix                            : 0x84
```

5-Engineering Mode Commands

nvm

```
#122: [PORT] Default enabled protocols
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4)}           : 0x1
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Disabled
#189: [GLOB] Crash dump trigger enable                       : 0x1
#196: [GLOB] Max MSIX                                       : 0
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}           : RoCE
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                       : 0x1
#222: [GLOB] Reset On Lan {Disabled(0), Enabled(1)}         : Disabled
#223: [GLOB] NCSI Package ID IO {nvram(0), IO_pins(1)}      : nvram
#232: [GLOB] PF Mapping {Continuous(0), Fixed(1)}           : Fixed
#234: [GLOB] Recovery Mode {Disabled(0), Enabled(1)}        : Enabled
#235: [PORT] PHY Module Dead Temp TH                       : 0
#236: [PORT] PHY Module aLOM fan on temp TH                 : 0
#237: [GLOB] Preboot Debug Mode Std                         : 16
#238: [GLOB] Preboot Debug Mode Ext                         : 0
#273: [GLOB] LLDP Disable {Off(0), On(1)}                   : Off
#283: [GLOB] Trace Level
#           {ALL(0), DEBUG(1), TRACE(2), ERROR(3)}          : ALL
#284: [GLOB] Trace Modules
#           {ERROR(0x1), DBG(0x2), DRV_HSI(0x4),
#           INTERRUPT(0x8), VPD(0x10), FLR(0x20),
#           INIT(0x40), NVM(0x80), PIM(0x100), NET(0x200),
#           POWER(0x400), UTILS(0x800), RESOURCES(0x1000),
#           SCHEDULER(0x2000), PHYMOD(0x4000),
#           EVENTS(0x8000), PMM(0x10000),
#           DBG_DRV(0x20000), ETH(0x40000),
#           SECURITY(0x80000), PCIE(0x100000),
#           TRACE(0x200000), PLDM(0x400000),
#           MCTP(0x800000), SIM(0x1000000)}                  : 0x0
#287: [GLOB] Warning Temperature Threshold                  : 0
#291: [GLOB] Number of provisioned MAC                       : 0
#292: [GLOB] Number of provisioned VF MAC                    : 0
#293: [GLOB] Provisioned BMC MAC {Disabled(0), Enabled(1)}  : Disabled
#
#
#           Group: link settings (Group 4)
#-----
# 27: [PORT] Drv Speed capability mask
#           {1G(0x1), 10G(0x2), 20G(0x4), 25G(0x8),
#           40G(0x10), 50G(0x20)}                           : 0x3
# 28: [PORT] MFW Speed capability mask
#           {1G(0x1), 10G(0x2), 20G(0x4), 25G(0x8),
```

5-Engineering Mode Commands

nvm

```
#          40G(0x10), 50G(0x20)}          : 0x3
# 29: [PORT] Drv Link speed
#          {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
#          40G(5), 50G(6)}                : Autoneg
# 30: [PORT] Drv Flow control
#          {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 31: [PORT] MFW Link speed
#          {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
#          40G(5), 50G(6)}                : Autoneg
# 32: [PORT] MFW Flow control
#          {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 33: [PORT] Optic Module Vendor Enforcement
#          {Disabled(0), Enabled(1)}        : Disabled
#140: [PORT] AN 25G/50G OUI {Consortium(0), BAM(1)} : Consortium
#145: [PORT] FEC force mode
#          {None(0), Firecode(1), RS(2), Auto(7)} : None
#146: [GLOB] Multi Network Modes Capability
#          {4x10G(0x1), 1x25G(0x2), 2x25G(0x4),
#          4x25G(0x8), 1x40G(0x10), 2x40G(0x20),
#          2x50G(0x40), 2x10G(0x100)}      : 0x0
#212: [PORT] FEC AN mode
#          {None(0), 10G_Firecode(1), 25G_Firecode(2),
#          10G_and_25G_Firecode(3), 25G_RS(4),
#          25G_Firecode_and_RS(5), All(6)}  : None
#239: [PORT] SmartLinQ Mode {Disabled(0), Enabled(1)} : Disabled
#
#
#
#
#          Group: phy (Group 7)
#-----
# 38: [GLOB] Network Port Mode
#          {2x50G(1), 4x10G_F(3), 1x40G(11), 2x25G(12),
#          1x25G(13), 4x25G(14), 2x10G(15),
#          2x25G_LIO2(16)}                  : 2x10G
# 45: [GLOB] MPS25 RX Lane polarity (L3:L2:L1:L0) : 1:0:0:0
# 46: [GLOB] MPS25 TX Lane polarity (L3:L2:L1:L0) : 1:0:0:0
# 50: [GLOB] MPS25 Preemphasis (L3:L2:L1:L0)      : 0:0:0:0
# 51: [GLOB] MPS25 Driver Current (L3:L2:L1:L0)    : 0:0:0:0
# 52: [GLOB] MPS25 Enforce TX FIR cfg
#          {Disabled(0), Enabled(1)}        : Enabled
# 55: [PORT] External PHY type
#          {None(0), BCM8485x(1), BCM5422x(2), 88X33x0(3)} : BCM8485x
# 56: [PORT] External PHY address                : 0xa
# 87: [GLOB] MPS25 TXFIR Main (L3:L2:L1:L0)       : 8:8:8:8
# 88: [GLOB] MPS25 TXFIR Post (L3:L2:L1:L0)       : 0:0:0:0
```

5-Engineering Mode Commands

nvm

```
#190: [GLOB] MPS25 Lane Swap (L3:L2:L1:L0) : 1:0:3:2
#192: [PORT] Ext PHY reset
# {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
# GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
# GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
# GPIO12(13), GPIO13(14), GPIO14(15),
# GPIO15(16), GPIO16(17), GPIO17(18),
# GPIO18(19), GPIO19(20), GPIO20(21),
# GPIO21(22), GPIO22(23), GPIO23(24),
# GPIO24(25), GPIO25(26), GPIO26(27),
# GPIO27(28), GPIO28(29), GPIO29(30),
# GPIO30(31), GPIO31(32)} : GPIO7
#193: [PORT] EEE power saving mode
# {Disabled(0), Balanced(1), Aggressive(2), Low
# latency(3)} : Balanced
#214: [GLOB] MPS25 Active TXFIR Pre (L3:L2:L1:L0) : 0:0:0:0
#215: [GLOB] MPS25 Active TXFIR Main (L3:L2:L1:L0) : 0:0:0:0
#216: [GLOB] MPS25 Active TXFIR Post (L3:L2:L1:L0) : 0:0:0:0
#224: [GLOB] Tx Rx EQ 25G HLPD (POST:MAIN:PRE1:DFE) : 10:d:3:1
#225: [GLOB] Tx Rx EQ 25G LLPC (POST:MAIN:PRE1:DFE) : a:d:3:1
#226: [GLOB] Tx Rx EQ 25G AC (POST:MAIN:PRE1:DFE) : a:d:3:1
#227: [GLOB] Tx Rx EQ 10G PC (POST:MAIN:PRE1:DFE) : 5:7:0:1
#228: [GLOB] Tx Rx EQ 10G AC (POST:MAIN:PRE1:DFE) : 5:7:0:0
#229: [GLOB] Tx Rx EQ 1G (POST:MAIN:PRE1:DFE) : 0:f:0:0
#230: [GLOB] Tx Rx EQ 25G BT (POST:MAIN:PRE1:DFE) : 0:0:0:0
#231: [GLOB] Tx Rx EQ 10G BT (POST:MAIN:PRE1:DFE) : 0:8:0:0
#249: [PORT] EXT PHY MDI pair swap : 0x0
#
#
# Group: pre-boot (Group 8)
#-----
# 59: [PORT] Preboot OpROM {Disabled(0), Enabled(1)} : Enabled
# 61: [PORT] MBA Delay Time (0-15) : 15
# 62: [PORT] MBA Setup Hot Key {Ctrl S(0), Ctrl B(1)} : Ctrl S
# 63: [PORT] MBA hide setup prompt
# {Disabled(0), Enabled(1)} : Disabled
# 67: [PORT] Preboot Link Speed
# {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
# 40G(5), 50G(6)} : Autoneg
# 69: [FUNC] Preboot Boot Protocol
# {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
# 80: [GLOB] Expansion ROM size
# {Disabled(0), 2K(1), 4K(2), 8K(3), 16K(4),
# 32K(5), 64K(6), 128K(7), 256K(8), 512K(9),
# 1M(10), 2M(11), 4M(12), 8M(13), 16M(14),
```

5-Engineering Mode Commands

nvm

```
#          32M(15)} : 512K
#132: [FUNC] Preboot VLAN value : 1
#133: [FUNC] Preboot VLAN : 0
#242: [PORT] Preboot Link Up Delay : 0
#
#
#          Group: VF (Group 9)
#-----
# 70: [GLOB] Enable SRIOV {Disabled(0), Enabled(1)} : Disabled
# 71: [GLOB] Enable ATC {Disabled(0), Enabled(1)} : Disabled
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#          {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#          64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#          Group: mng_fw (Group 5)
#-----
#109: [GLOB] SMBus Address : 0x32
#110: [GLOB] NCSI Package ID : 0
#111: [GLOB] Sideband Mode {Disabled(0), RMII(1), SGMII(2)} : Disabled
#112: [GLOB] SMBus Mode {Disabled(0), 100Khz(1), 400Khz(2)} : 100Khz
#113: [GLOB] NCSI {Disabled(0), Enabled(1)} : Enabled
#131: [GLOB] On Chip Sensor Mode
#          {Disabled(0), Int_Ext_I2C(1), Int_Only(2),
#          Int_Ext_SMBUS(3)} : Disabled
#141: [GLOB] PLDM sensor mode
#          {internal(0), external(1), both(2)} : internal
#142: [GLOB] External thermal sensor
#          {Disabled(0), Enabled(1)} : Disabled
#143: [GLOB] External thermal sensor address : 0x78
#203: [GLOB] Max cont operating temp : 95
#208: [GLOB] DCI support {Disabled(0), Enabled(1)} : Disabled
#209: [GLOB] PCIE VDM ENABLED {Disabled(0), Enabled(1)} : Disabled
#243: [GLOB] voltage regulator type {Disabled(0), TI(1)} : Disabled
#251: [GLOB] NCSI AUX link
#          {Default(0), 1 port(1), 2 ports(2), 3 ports(3)} : Default
#285: [GLOB] Emulated TMP421 {Disabled(0), Enabled(1)} : Disabled
#
#
#
```

5-Engineering Mode Commands

nvm

```
#####
### Device 2 ###
#####
#
#                                     Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                : 00:10:18:11:22:34
# 10: [PORT] Led Mode
#      {Mac1(0), Phy1(1), Phy2(2), Phy3(3), Mac2(4),
#      Phy4(5), Phy5(6), Phy6(7), Mac3(8), Phy7(9),
#      Phy8(10), Phy9(11), Mac4(12), Phy10(13),
#      Phy11(14), Phy12(15), Breakout(16),
#      OCP_3_0(17)}                    : Mac1
#102: [PORT] LED speed select (S2:S1:S0)
#      {1G(0x1), 10G(0x2), 25G(0x4), 40G(0x8),
#      50G(0x10)}                      : 0:1:2
#250: [PORT] UID LED Mode Mask
#      {Activity_LED(0x1), Link_LED0(0x2),
#      Link_LED1(0x4), Link_LED2(0x8)}  : 0x0
#
#
#                                     Group: features (Group 3)
#-----
# 20: [PORT] RoCE priority              : 0
# 22: [PORT] Enable WoL on ACPI pattern
#      {Disabled(0), Enabled(1)}        : Disabled
# 23: [PORT] Magic Packet WoL {Disabled(0), Enabled(1)} : Disabled
# 26: [PORT] DCBX Mode
#      {Disabled(0), IEEE(1), CEE(2), Dynamic(3)} : Dynamic
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr      : 00:10:18:11:22:34
# 94: [FUNC] FCoE port wwn mac addr      : 00:10:18:11:22:34
# 95: [FUNC] Bandwidth weight            : 0
# 96: [FUNC] Max Bandwidth                : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
# 99: [PORT] LLDP MAC address             : 00:10:18:11:22:44
#118: [PORT] ETH DID suffix               : 0x70
#119: [PORT] FCoE DID suffix              : 0x80
#120: [PORT] iSCSI DID suffix             : 0x84
#122: [PORT] Default enabled protocols
#      {Ethernet(0x1), FCoE(0x2), iSCSI(0x4)} : 0x1
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#      {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
```

5-Engineering Mode Commands

nvm

```
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                     : 0x1
#235: [PORT] PHY Module Dead Temp TH                       : 0
#236: [PORT] PHY Module aLOM fan on temp TH                 : 0
#
#
#                                     Group: link settings (Group 4)
#-----
# 27: [PORT] Drv Speed capability mask
#           {1G(0x1), 10G(0x2), 20G(0x4), 25G(0x8),
#           40G(0x10), 50G(0x20)}                         : 0x3
# 28: [PORT] MFW Speed capability mask
#           {1G(0x1), 10G(0x2), 20G(0x4), 25G(0x8),
#           40G(0x10), 50G(0x20)}                         : 0x3
# 29: [PORT] Drv Link speed
#           {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
#           40G(5), 50G(6)}                               : Autoneg
# 30: [PORT] Drv Flow control
#           {Autoneg(0x1), Rx(0x2), Tx(0x4)}               : 0x7
# 31: [PORT] MFW Link speed
#           {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
#           40G(5), 50G(6)}                               : Autoneg
# 32: [PORT] MFW Flow control
#           {Autoneg(0x1), Rx(0x2), Tx(0x4)}               : 0x7
# 33: [PORT] Optic Module Vendor Enforcement
#           {Disabled(0), Enabled(1)}                     : Disabled
#140: [PORT] AN 25G/50G OUI {Consortium(0), BAM(1)}        : Consortium
#145: [PORT] FEC force mode
#           {None(0), Firecode(1), RS(2), Auto(7)}         : None
#212: [PORT] FEC AN mode
#           {None(0), 10G_Firecode(1), 25G_Firecode(2),
#           10G_and_25G_Firecode(3), 25G_RS(4),
#           25G_Firecode_and_RS(5), All(6)}                : None
#239: [PORT] SmartLinQ Mode {Disabled(0), Enabled(1)}      : Disabled
#
#
#                                     Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                           : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                             : 128K
```

5-Engineering Mode Commands

nvm

```
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)} : 8M
#
#
#
#           Group: phy (Group 7)
#-----
# 55: [PORT] External PHY type
#           {None(0), BCM8485x(1), BCM5422x(2), 88X33x0(3)} : BCM8485x
# 56: [PORT] External PHY address : 0xb
#192: [PORT] Ext PHY reset
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : GPIO7
#193: [PORT] EEE power saving mode
#           {Disabled(0), Balanced(1), Aggressive(2), Low
#           latency(3)} : Balanced
#249: [PORT] EXT PHY MDI pair swap : 0x0
#
#
#
#           Group: pre-boot (Group 8)
#-----
# 59: [PORT] Preboot OpROM {Disabled(0), Enabled(1)} : Enabled
# 61: [PORT] MBA Delay Time (0-15) : 15
# 62: [PORT] MBA Setup Hot Key {Ctrl S(0), Ctrl B(1)} : Ctrl S
# 63: [PORT] MBA hide setup prompt
#           {Disabled(0), Enabled(1)} : Disabled
# 67: [PORT] Preboot Link Speed
#           {Autoneg(0), 1G(1), 10G(2), 20G(3), 25G(4),
#           40G(5), 50G(6)} : Autoneg
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value : 1
#133: [FUNC] Preboot VLAN : 0
#242: [PORT] Preboot Link Up Delay : 0
#
#
```

5-Engineering Mode Commands

nvm

```
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                      : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)}                                     : Disabled
# 84: [FUNC] VF PCI Device ID                          : 0x8090
#
#
#                                     Group: board io (Group 2)
#-----
#114: [PORT] Transceiver Module Absent
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)}                     : NA
#126: [PORT] Port Type
#           {Undefined(0), Module(1), Backplane(2),
#           Ext_phy(3), Module_slave(4)}                 : Ext_phy
#127: [PORT] Tx Disable
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)}                     : NA
#272: [PORT] SmartAN FEC Override {Disabled(0), Enabled(1)} : Disabled
#275: [PORT] Transceiver Rate Select 0
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
```


5-Engineering Mode Commands

nvm

```
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}                                : NA
#276: [PORT] Transceiver Rate Select 1
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}                                : NA
#277: [PORT] Transceiver Module TX Fault
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}                                : NA
#278: [PORT] Transceiver QSFP Module Reset
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}                                : NA
#279: [PORT] Transceiver QSFP LP Mode
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
```

5-Engineering Mode Commands

nvm

```
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#280: [PORT] Transceiver Power Enable
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#281: [PORT] LASI Interrupt input
#          {NA(0), LASI0(1), LASI1(2), LASI2(3), LASI3(4)} : NA
#
#
#
#

#####
### Device 3 ###
#####

#          Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:22:35
#
#
#          Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#          {Disabled(0), 64K(1), 128K(2), 256K(3),
#          512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#          32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
#          {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#          16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)} : 1M
#
```

5-Engineering Mode Commands

nvm

```
#
#                                     Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value                          : 1
#133: [FUNC] Preboot VLAN                                : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                          : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID                             : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)}      : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                          : 00:10:18:11:22:35
# 94: [FUNC] FCoE port wwn mac addr                          : 00:10:18:11:22:35
# 95: [FUNC] Bandwidth weight                                : 0
# 96: [FUNC] Max Bandwidth                                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}    : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Disabled
#202: [FUNC] RDMA enablement
#                   {None(0), RoCE(1), iWARP(2), Both(3)}     : RoCE
#213: [FUNC] NPAR enabled protocol
#                   {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#                   RDMA(0x8)}                                : 0x3
#
#
#

#####
### Device 4 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                      : 00:10:18:11:22:36
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                            : 0x0000
# 82: [FUNC] Bar1 size
#                   {Disabled(0), 64K(1), 128K(2), 256K(3),
#                   512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#                   32M(10), 64M(11), 128M(12), 256M(13),
#                   512M(14), 1G(15)}                        : 128K
#191: [FUNC] Bar2 size
#                   {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#                   16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#                   512M(14), 1G(15)}                        : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#                   {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
```

5-Engineering Mode Commands

nvm

```
#132: [FUNC] Preboot VLAN value : 1
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:10:18:11:22:36
# 94: [FUNC] FCoE port wwn mac addr : 00:10:18:11:22:36
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x3
#
#
#####
### Device 5 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:22:37
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID           : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}                  : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}                  : 1M
#
#
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value             : 1
#133: [FUNC] Preboot VLAN                   : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF           : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                           : Disabled
# 84: [FUNC] VF PCI Device ID               : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr         : 00:10:18:11:22:37
# 94: [FUNC] FCoE port wwn mac addr         : 00:10:18:11:22:37
# 95: [FUNC] Bandwidth weight               : 0
# 96: [FUNC] Max Bandwidth                  : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}      : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)}            : RoCE
#213: [FUNC] NPAR enabled protocol
```

5-Engineering Mode Commands

nvm

```
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                               : 0x5
#
#
#

#####
### Device 6 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                               : 00:10:18:11:22:38
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                       : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                           : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                           : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value                       : 1
#133: [FUNC] Preboot VLAN                             : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                       : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
```

5-Engineering Mode Commands

nvm

```
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#          Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:10:18:11:22:38
# 94: [FUNC] FCoE port wwn mac addr : 00:10:18:11:22:38
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)} : 0x5
#
#
#
```

```
#####
### Device 7 ###
#####
```

```
#          Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:22:39
#
#
#          Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#          {Disabled(0), 64K(1), 128K(2), 256K(3),
#          512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#          32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
#          {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#          16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
```


5-Engineering Mode Commands

nvm

```
#          512M(14), 1G(15)}          : 1M
#
#
#          Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#          {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value          : 1
#133: [FUNC] Preboot VLAN                : 0
#
#
#          Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF          : 16
# 75: [FUNC] VF PCI BAR2 size
#          {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#          64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID              : 0x8090
#
#
#          Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr          : 00:10:18:11:22:39
# 94: [FUNC] FCoE port wwn mac addr          : 00:10:18:11:22:39
# 95: [FUNC] Bandwidth weight                : 0
# 96: [FUNC] Max Bandwidth                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)} : 0x1
#
#
#
```

5-Engineering Mode Commands

nvm

```
#####
### Device 8 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address          : 00:10:18:11:22:3a
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value   : 1
#133: [FUNC] Preboot VLAN        : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                  : Disabled
# 84: [FUNC] VF PCI Device ID     : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
```

5-Engineering Mode Commands

nvm

```
# 93: [FUNC] FCoE node wwn mac addr : 00:10:18:11:22:3a
# 94: [FUNC] FCoe port wwn mac addr : 00:10:18:11:22:3a
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
# {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
#213: [FUNC] NPAR enabled protocol
# {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
# RDMA(0x8)} : 0x1
#
#
#
```

```
#####
### Device 9 ###
#####
```

```
# Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:22:3b
#
#
# Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
# {Disabled(0), 64K(1), 128K(2), 256K(3),
# 512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
# 32M(10), 64M(11), 128M(12), 256M(13),
# 512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
# {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
# 16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
# 512M(14), 1G(15)} : 1M
#
#
# Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
# {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value : 1
```

5-Engineering Mode Commands

nvm

```
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:10:18:11:22:3b
# 94: [FUNC] FCoE port wwn mac addr : 00:10:18:11:22:3b
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Enabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x1
#
#
#
#####
### Device 10 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:22:3c
#
#
```

5-Engineering Mode Commands

nvm

```
#                                     Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID           : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}                  : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}                  : 1M
#
#
#                                     Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value             : 1
#133: [FUNC] Preboot VLAN                   : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF           : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                            : Disabled
# 84: [FUNC] VF PCI Device ID               : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr         : 00:10:18:11:22:3c
# 94: [FUNC] FCoE port wwn mac addr        : 00:10:18:11:22:3c
# 95: [FUNC] Bandwidth weight              : 0
# 96: [FUNC] Max Bandwidth                  : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}      : Enabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)}            : None
#213: [FUNC] NPAR enabled protocol
```

5-Engineering Mode Commands

nvm

```
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                               : 0x1
#
#
#

#####
### Device 11 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                               : 00:10:18:11:22:3d
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                       : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                           : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                           : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value                       : 1
#133: [FUNC] Preboot VLAN                             : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                       : 8
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
```

5-Engineering Mode Commands

nvm

```
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)}                                : Disabled
# 84: [FUNC] VF PCI Device ID                       : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                   : 00:10:18:11:22:3d
# 94: [FUNC] FCoE port wwn mac addr                   : 00:10:18:11:22:3d
# 95: [FUNC] Bandwidth weight                         : 0
# 96: [FUNC] Max Bandwidth                           : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}   : Enabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)}      : None
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)}                                : 0x1
#
#
#
```

```
#####
### Device 12 ###
#####
```

```
#                                     Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                             : 00:10:18:11:22:3e
#
#
#                                     Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                     : 0x0000
# 82: [FUNC] Bar1 size
#          {Disabled(0), 64K(1), 128K(2), 256K(3),
#          512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#          32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)}                         : 128K
#191: [FUNC] Bar2 size
#          {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#          16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
```

5-Engineering Mode Commands

nvm

```
#          512M(14), 1G(15)}          : 1M
#
#
#          Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#          {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value          : 1
#133: [FUNC] Preboot VLAN                : 0
#
#
#          Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF          : 8
# 75: [FUNC] VF PCI BAR2 size
#          {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#          64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID              : 0x8090
#
#
#          Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr          : 00:10:18:11:22:3e
# 94: [FUNC] FCoE port wwn mac addr          : 00:10:18:11:22:3e
# 95: [FUNC] Bandwidth weight                : 0
# 96: [FUNC] Max Bandwidth                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Enabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)} : 0x1
#
#
#
```


5-Engineering Mode Commands

nvm

```
#####
### Device 13 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address          : 00:10:18:11:22:3f
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value   : 1
#133: [FUNC] Preboot VLAN        : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                  : Disabled
# 84: [FUNC] VF PCI Device ID     : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)}      : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                          : 00:10:18:11:22:3f
# 94: [FUNC] FCoE port wwn mac addr                          : 00:10:18:11:22:3f
# 95: [FUNC] Bandwidth weight                                : 0
# 96: [FUNC] Max Bandwidth                                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}    : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Enabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}           : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                       : 0x1
#
#
#

#####
### Device 14 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                      : 00:10:18:11:22:40
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                            : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
```

5-Engineering Mode Commands

nvm

```
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value : 1
#133: [FUNC] Preboot VLAN : 0
#
#
#           Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#           Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:10:18:11:22:40
# 94: [FUNC] FCoE port wwn mac addr : 00:10:18:11:22:40
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Enabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)} : 0x1
#
#
#
```

5-Engineering Mode Commands

nvm

```
#####
### Device 15 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address          : 00:10:18:11:22:41
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value   : 1
#133: [FUNC] Preboot VLAN        : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                  : Disabled
# 84: [FUNC] VF PCI Device ID      : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
```

5-Engineering Mode Commands

nvm

```
# 93: [FUNC] FCoE node wwn mac addr           : 00:10:18:11:22:41
# 94: [FUNC] FCoe port wwn mac addr           : 00:10:18:11:22:41
# 95: [FUNC] Bandwidth weight                  : 0
# 96: [FUNC] Max Bandwidth                     : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Enabled
#202: [FUNC] RDMA enablement
#         {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#         {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#         RDMA(0x8)} : 0x1
#
#
#
```

```
#####
### Device 16 ###
#####
```

```
#                                     Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                 : 00:10:18:11:22:42
#
#
#                                     Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID         : 0x0000
# 82: [FUNC] Bar1 size
#         {Disabled(0), 64K(1), 128K(2), 256K(3),
#         512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#         32M(10), 64M(11), 128M(12), 256M(13),
#         512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
#         {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#         16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#         512M(14), 1G(15)} : 1M
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value                               : 1
#133: [FUNC] Preboot VLAN                                   : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                               : 8
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID                               : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality {Ethernet(0), iSCSI(1), FCoE(2)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                       : 00:10:18:11:22:42
# 94: [FUNC] FCoE port wwn mac addr                       : 00:10:18:11:22:42
# 95: [FUNC] Bandwidth weight                             : 0
# 96: [FUNC] Max Bandwidth                               : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}      : Enabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}         : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)} : 0x1
#
#
#
```

5-Engineering Mode Commands

nvm

```
#####  
### Shared ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(IS_SPECIAL_NVM_MODE) 1}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 1} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 0}  
nvm cfg \  
{8=0}\  
{9=1}\  
{11=0}\  
{13=1077}\  
{15=2}\  
{16=0}\  
{17=1}\  
{38=15}\  
{45=1:0:0:0}\  
{46=1:0:0:0}\  
{50=0:0:0:0}\  
{51=0:0:0:0}\  
{52=1}\  
{70=0}\  
{71=0}\  
{76=0x1077}\  
{78=0x1077}\  
{79=0x0001}\  
{80=9}\  
{87=8:8:8:8}\  
{88=0:0:0:0}\  
{89=08:03:21:00:09}\  
{90=03:06:17}\  
{98=0}\  
{100=20:00}\  
{101=20:01}\  
{103=1:0:3:2}\  
{105=0}\  
{107=00:00:00}\  
{108=0x0}\  
{109=0x32}\  
{110=0}\  
{111=0}\  
{112=1}\  
{113=1}\
```

5-Engineering Mode Commands

nvm

```
{117=0x1f}\  
{125=0}\  
{128=4}\  
{130=0}\  
{131=0}\  
{134=1}\  
{135=110}\  
{136=5}\  
{137=105}\  
{139=1}\  
{141=0}\  
{142=0}\  
{143=0x78}\  
{144=5}\  
{146=0x0}\  
{189=0x1}\  
{190=1:0:3:2}\  
{196=0}\  
{197=0}\  
{198=21}\  
{199=1}\  
{200=3}\  
{203=95}\  
{204=0}\  
{205=0:0:0:0}\  
{206=19}\  
{207=0}\  
{208=0}\  
{209=0}\  
{210=123456}\  
{211=123456}\  
{214=0:0:0:0}\  
{215=0:0:0:0}\  
{216=0:0:0:0}\  
{217=0}\  
{218=95}\  
{219=0}\  
{220=30}\  
{221=32}\  
{222=0}\  
{223=0}\  
{224=10:d:3:1}\  
{225=a:d:3:1}\  
{226=a:d:3:1}\  
{227=5:7:0:1}
```


5-Engineering Mode Commands

nvm

```
{228=5:7:0:0}\
{229=0:f:0:0}\
{230=0:0:0:0}\
{231=0:8:0:0}\
{232=1}\
{234=1}\
{237=16}\
{238=0}\
{243=0}\
{245=0}\
{246=0}\
{247=0}\
{248=0}\
{251=0}\
{273=0}\
{274=0}\
{282=0}\
{283=0}\
{284=0x0}\
{285=0}\
{286=0}\
{287=0}\
{288=0}\
{289=0x0}\
{290=0x0}\
{291=0}\
{292=0}\
{293=0}\
```

```
#####
### Device 1 ###
#####
```

```
nvm cfg \
{10=0}\
{20=0}\
{22=0}\
{23=0}\
{26=3}\
{27=0x3}\
{28=0x3}\
{29=0}\
```

5-Engineering Mode Commands

nvm

```
{ 30=0x7 }\n{ 31=0 }\n{ 32=0x7 }\n{ 33=0 }\n{ 55=1 }\n{ 56=0xa }\n{ 59=1 }\n{ 61=15 }\n{ 62=0 }\n{ 63=0 }\n{ 67=0 }\n{ 102=0:1:2 }\n{ 114=0 }\n{ 118=0x70 }\n{ 119=0x80 }\n{ 120=0x84 }\n{ 122=0x1 }\n{ 126=3 }\n{ 127=0 }\n{ 140=0 }\n{ 145=0 }\n{ 192=8 }\n{ 193=1 }\n{ 212=0 }\n{ 235=0 }\n{ 236=0 }\n{ 239=0 }\n{ 242=0 }\n{ 249=0x0 }\n{ 250=0x0 }\n{ 272=0 }\n{ 275=0 }\n{ 276=0 }\n{ 277=0 }\n{ 278=0 }\n{ 279=0 }\n{ 280=0 }\n{ 281=0 }\n{ 37=0x0000 }\n{ 69=7 }\n{ 74=16 }\n{ 75=0 }\n{ 82=2 }\n{ 84=0x8090 }\n{ 92=0 }
```

5-Engineering Mode Commands

nvm

```
{95=0}\
{96=100}\
{97=0}\
{132=1}\
{133=0}\
{187=0}\
{191=8}\
{202=1}\
{213=0x1}\
```

```
#####
### Device 2 ###
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 2} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 1}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{10=0}\
{20=0}\
{22=0}\
{23=0}\
{26=3}\
{27=0x3}\
{28=0x3}\
{29=0}\
{30=0x7}\
{31=0}\
{32=0x7}\
{33=0}\
{55=1}\
{56=0xb}\
{59=1}\
{61=15}\
{62=0}\
{63=0}\
{67=0}\
{102=0:1:2}\
{114=0}\
{118=0x70}\
{119=0x80}\
{120=0x84}\
```

5-Engineering Mode Commands

nvm

```
{122=0x1}\  
{126=3}\  
{127=0}\  
{140=0}\  
{145=0}\  
{192=8}\  
{193=1}\  
{212=0}\  
{235=0}\  
{236=0}\  
{239=0}\  
{242=0}\  
{249=0x0}\  
{250=0x0}\  
{272=0}\  
{275=0}\  
{276=0}\  
{277=0}\  
{278=0}\  
{279=0}\  
{280=0}\  
{281=0}\  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=8}\  
{202=1}\  
{213=0x1}\
```

5-Engineering Mode Commands

nvm

```
#####  
### Device 3 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 3} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 2}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x3}
```

```
#####  
### Device 4 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 4} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 3}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}
```

5-Engineering Mode Commands

nvm

```
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x3}\
```

```
#####  
### Device 5 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 5} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 4}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x5}\
```

5-Engineering Mode Commands

nvm

```
#####  
### Device 6 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 6} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 5}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x5}\  

```

```
#####  
### Device 7 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 7} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 6}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  

```

5-Engineering Mode Commands

nvm

```
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x1}\
```

```
#####  
### Device 8 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 8} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 7}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=16}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=1}\  
{213=0x1}\
```


5-Engineering Mode Commands

nvm

```
#####  
### Device 9 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 9} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 8}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\  

```

```
#####  
### Device 10 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 10} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 9}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  

```

5-Engineering Mode Commands

nvm

```
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\
```

```
#####  
### Device 11 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 11} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 10}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\
```

5-Engineering Mode Commands

nvm

```
#####  
### Device 12 ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 12} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 11}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\  
  
  
#####  
### Device 13 ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 13} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 12}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}
```

5-Engineering Mode Commands

nvm

```
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\
```

```
#####  
### Device 14 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 14} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 13}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\
```

5-Engineering Mode Commands

nvm

```
#####  
### Device 15 ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 15} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 14}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\  
  
  
#####  
### Device 16 ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 16} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 15}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}
```

```
{96=100}\  
{97=0}\  
{132=1}\  
{133=0}\  
{187=1}\  
{191=5}\  
{202=0}\  
{213=0x1}\
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}} {} else {set  
::current(IS_SPECIAL_NVM_MODE) 0}  
device 1
```

The following file applies to the Marvell 8500 and 8600 Series Controllers.

```
#####  
# DO NOT EDIT!!! THIS FILE IS AUTOMATICALLY GENERATED.  
  
#  
#Copyright(c) 2014-2016 QLogic Corporation, all rights reserved  
  
#Proprietary and Confidential Information.  
#  
#  
#Description: This file contains the NVRAM configuration for <board name>  
#             To incorporate these settings into the board's NVRAM, simply  
#             source this file in Eddiag as follows  
#             source <config_file>  
#  
#Design Name: <board name>  
#Diagnostic Tool Version:  
#Software Release:  
#Customer Name:  
#Revision History:  
#  
#Manufacture kit version: 08:01:22:02:01  
#Additional Comments:
```

5-Engineering Mode Commands

nvm

```
#####
### Device 1 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address           : 00:10:18:22:33:33
# 8: [GLOB] Board Swap
#       {None(0), Path(1), Port(2), Both(3)} : None
# 9: [GLOB] MF Mode
#       {MF allowed(0), Default(1), SPI04(2),
#       NPAR1.0(3), NPAR1.5(4), NPAR2.0(5), BD(6),
#       UFP(7)} : Default
# 10: [PORT] Led Mode
#       {Mac1(0), Phy1(1), Phy2(2), Phy3(3), Mac2(4),
#       Phy4(5), Phy5(6), Phy6(7), Mac3(8), Phy7(9),
#       Phy8(10), Phy9(11), Mac4(12), Phy10(13),
#       Phy11(14), Phy12(15), Breakout(16)} : Mac1
# 89: [GLOB] Manufacture kit version : 08:01:22:02:01
# 90: [GLOB] Manufacture timestamp  : 03:09:17
#102: [PORT] LED speed select (S2:S1:S0)
#       {1G(0x1), 10G(0x2), 25G(0x4), 40G(0x8),
#       50G(0x10)} : 0:4:3
#103: [GLOB] LED port swap (P3:P2:P1:P0) : 0:2:1:3
#105: [GLOB] Override Secure Mode {Disabled(0), Enabled(1)} : Enabled
#125: [GLOB] AUX mode {Default(0), SMBUS only(1)} : Default
#197: [GLOB] NVM_CFG version         : 0
#198: [GLOB] NVM_CFG new option seq  : 21
#199: [GLOB] NVM_CFG removed option seq : 1
#200: [GLOB] NVM_CFG updated value seq : 3
#
#
#                               Group: board io (Group 2)
#-----
# 11: [GLOB] Fan Failure Enforcement
#       {Disabled(0), Enabled(1)} : Disabled
#114: [PORT] Transceiver Module Absent
#       {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#       GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#       GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#       GPIO12(13), GPIO13(14), GPIO14(15),
#       GPIO15(16), GPIO16(17), GPIO17(18),
#       GPIO18(19), GPIO19(20), GPIO20(21),
#       GPIO21(22), GPIO22(23), GPIO23(24),
#       GPIO24(25), GPIO25(26), GPIO26(27),
#       GPIO27(28), GPIO28(29), GPIO29(30),
```

5-Engineering Mode Commands

nvm

```
#          GPIO30(31), GPIO31(32)}          : GPIO15
#126: [PORT] Port Type
#          {Undefined(0), Module(1), Backplane(2),
#          Ext_phy(3), Module_slave(4)}          : Module
#127: [PORT] Tx Disable
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}          : GPIO16
#134: [GLOB] Temperature period between checks          : 1
#135: [GLOB] Shutdown threshold temperature          : 110
#136: [GLOB] max count oper threshold          : 5
#137: [GLOB] Caution threshold temperature          : 105
#139: [GLOB] Temperature monitoring mode
#          {Disable(0), Internal(1), External(2), Both(3)} : Internal
#144: [GLOB] Fan Failure duration          : 5
#204: [GLOB] Runtime Port Swap GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}          : NA
#205: [GLOB] Runtime Port Swap Map (P3:P2:P1:P0)          : 0:0:0:0
#206: [GLOB] Thermal Event GPIO
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}          : GPIO18
```

5-Engineering Mode Commands

nvm

```
#207: [GLOB] I2C Interrupt GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : NA
#217: [GLOB] ALOM FAN ON AUX GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : NA
#218: [GLOB] ALOM FAN ON AUX value : 95
#219: [GLOB] SLOT ID GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : NA
#220: [GLOB] PMBUS SCL GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : GPIO29
```

5-Engineering Mode Commands

nvm

```
#221: [GLOB] PMBUS SDA GPIO
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : GPIO31
#
#
#
#           Group: pcie (Group 6)
#-----
# 12: [GLOB] Engineering Change :
# 13: [GLOB] Manufacturing ID : 1077
# 14: [GLOB] Serial Number :
# 15: [GLOB] PCI Gen :
#           {PCI Gen1(0), PCI Gen2(1), PCI Gen3(2)} : PCI Gen3
# 16: [GLOB] Beacon WOL_Enabled {Disabled(0), Enabled(1)} : Disabled
# 17: [GLOB] ASPM Support :
#           {L0s L1 enabled(0), L0s disabled(1), L1
#           disabled(2), L0s L1 disabled(3)} : L0s disabled
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 76: [GLOB] Vendor ID : 0x1077
# 78: [GLOB] Subsystem Vendor ID : 0x1590
# 79: [GLOB] Subsystem Device ID : 0x021d
# 82: [FUNC] Bar1 size :
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)} : 128K
# 98: [GLOB] PCIE Preemphasis :
#           {HW(0), 0dB(1), 3_5dB(2), 6_0dB(3)} : HW
#128: [GLOB] Max link width :
#           {1 lane(1), 2 lanes(2), 4 lanes(3), 8 lanes(4)} : 8 lanes
#130: [GLOB] ASPM L1 mode :
#           {Forced(0), Dynamic low latency(1)} : Forced
#191: [FUNC] Bar2 size :
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)} : 8M
#201: [GLOB] Extended serial number :
```

5-Engineering Mode Commands

nvm

```
#210: [GLOB] OEM1 Number           : 867328-B21
#211: [GLOB] OEM2 Number           : 869570-001
#
#
#                               Group: features (Group 3)
#-----
# 20: [PORT] RoCE priority           : 0
# 22: [PORT] Enable WoL on ACPI pattern
#           {Disabled(0), Enabled(1)} : Disabled
# 23: [PORT] Magic Packet WoL {Disabled(0), Enabled(1)} : Disabled
# 26: [PORT] DCBX Mode
#           {Disabled(0), IEEE(1), CEE(2), Dynamic(3)} : Dynamic
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr  : 11:11:11:11:11:11
# 94: [FUNC] FCoE port wwn mac addr  : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight        : 0
# 96: [FUNC] Max Bandwidth           : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
# 99: [PORT] LLDP MAC address        : 00:10:18:11:11:21
#100: [GLOB] FCoE wwn node prefix    : 10:00
#101: [GLOB] FCoE wwn port prefix    : 20:00
#107: [GLOB] MBI version             : 00:00:00
#108: [GLOB] MBI date                : 0x0
#117: [GLOB] Device capabilities
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RoCE(0x8), iWarp(0x10)}   : 0x19
#118: [PORT] ETH DID suffix          : 0x70
#119: [PORT] FCoE DID suffix         : 0x80
#120: [PORT] iSCSI DID suffix        : 0x84
#122: [PORT] Default enabled protocols
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4)}   : 0x1
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#189: [GLOB] Crash dump trigger enable : 0x1
#196: [GLOB] Max MSIX                : 0
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}     : RoCE
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                       : 0x9
#222: [GLOB] Reset On Lan {Disabled(0), Enabled(1)} : Disabled
#223: [GLOB] NCSI Package ID IO {nvram(0), IO_pins(1)} : nvram
#232: [GLOB] PF Mapping {Continuous(0), Fixed(1)} : Fixed
#233: [GLOB] Thermal Alarm gpio
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
```

5-Engineering Mode Commands

nvm

```
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)} : NA
#234: [GLOB] Recovery Mode {Disabled(0), Enabled(1)} : Enabled
#235: [PORT] PHY Module Dead Temp TH : 0
#236: [PORT] PHY Module aLOM fan on temp TH : 0
#237: [GLOB] Preboot Debug Mode Std : 16
#238: [GLOB] Preboot Debug Mode Ext : 0
#
#
#          Group: link settings (Group 4)
#-----
# 27: [PORT] Drv Speed capability mask
#          {1G(0x1), 10G(0x2), 25G(0x8), 40G(0x10),
#          50G(0x20)} : 0xa
# 28: [PORT] MFW Speed capability mask
#          {1G(0x1), 10G(0x2), 25G(0x8), 40G(0x10),
#          50G(0x20)} : 0xa
# 29: [PORT] Drv Link speed
#          {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
#          50G(6)} : Autoneg
# 30: [PORT] Drv Flow control
#          {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 31: [PORT] MFW Link speed
#          {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
#          50G(6)} : Autoneg
# 32: [PORT] MFW Flow control
#          {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 33: [PORT] Optic Module Vendor Enforcement
#          {Disabled(0), Enabled(1)} : Disabled
#140: [PORT] AN 25G/50G OUI {Consortium(0), BAM(1)} : Consortium
#145: [PORT] FEC force mode
#          {None(0), Firecode(1), RS(2), Auto(7)} : None
#146: [GLOB] Multi Network Modes Capability
#          {4x10G(0x1), 1x25G(0x2), 2x25G(0x4),
#          4x25G(0x8), 1x40G(0x10), 2x40G(0x20),
#          2x50G(0x40), 2x10G(0x100)} : 0x0
#212: [PORT] FEC AN mode
#          {None(0), 10G_Firecode(1), 25G_Firecode(2),
```

5-Engineering Mode Commands

nvm

```
#          10G_and_25G_Firecode(3), 25G_RS(4),
#          25G_Firecode_and_RS(5), All(6)}                : All
#239: [PORT] SmartLinQ Mode {Disabled(0), Enabled(1)}      : Disabled
#
#
#                                     Group: phy (Group 7)
#-----
# 38: [GLOB] Network Port Mode
#          {2x50G(1), 4x10G_F(3), 1x40G(11), 2x25G(12),
#          1x25G(13), 4x25G(14), 2x10G(15)}                : 2x25G
# 45: [GLOB] MPS25 RX Lane polarity (L3:L2:L1:L0)          : 0:0:0:0
# 46: [GLOB] MPS25 TX Lane polarity (L3:L2:L1:L0)          : 0:0:0:0
# 50: [GLOB] MPS25 Preemphasis (L3:L2:L1:L0)               : 3:3:3:3
# 51: [GLOB] MPS25 Driver Current (L3:L2:L1:L0)            : 0:0:0:0
# 52: [GLOB] MPS25 Enforce TX FIR cfg
#          {Disabled(0), Enabled(1)}                      : Enabled
# 55: [PORT] External PHY type {None(0), BCM8485x(1)}      : None
# 56: [PORT] External PHY address                          : 0x0
# 87: [GLOB] MPS25 TXFIR Main (L3:L2:L1:L0)                : d:d:d:d
# 88: [GLOB] MPS25 TXFIR Post (L3:L2:L1:L0)                : a:a:a:a
#190: [GLOB] MPS25 Lane Swap (L3:L2:L1:L0)                 : 0:2:1:3
#192: [PORT] Ext PHY reset
#          {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#          GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#          GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#          GPIO12(13), GPIO13(14), GPIO14(15),
#          GPIO15(16), GPIO16(17), GPIO17(18),
#          GPIO18(19), GPIO19(20), GPIO20(21),
#          GPIO21(22), GPIO22(23), GPIO23(24),
#          GPIO24(25), GPIO25(26), GPIO26(27),
#          GPIO27(28), GPIO28(29), GPIO29(30),
#          GPIO30(31), GPIO31(32)}                        : NA
#193: [PORT] EEE power saving mode
#          {Disabled(0), Balanced(1), Aggressive(2), Low
#          latency(3)}                                     : Disabled
#214: [GLOB] MPS25 Active TXFIR Pre (L3:L2:L1:L0)          : 3:3:3:3
#215: [GLOB] MPS25 Active TXFIR Main (L3:L2:L1:L0)         : d:d:d:d
#216: [GLOB] MPS25 Active TXFIR Post (L3:L2:L1:L0)         : a:a:a:a
#224: [GLOB] Tx Rx EQ 25G HLPC (POST:MAIN:PRE:DFE)        : 10:d:3:1
#225: [GLOB] Tx Rx EQ 25G LLPC (POST:MAIN:PRE:DFE)        : a:d:3:1
#226: [GLOB] Tx Rx EQ 25G AC (POST:MAIN:PRE:DFE)          : a:d:3:1
#227: [GLOB] Tx Rx EQ 10G PC (POST:MAIN:PRE:DFE)          : 5:7:0:1
#228: [GLOB] Tx Rx EQ 10G AC (POST:MAIN:PRE:DFE)          : 5:7:0:0
#229: [GLOB] Tx Rx EQ 1G (POST:MAIN:PRE:DFE)              : 0:f:0:0
#230: [GLOB] Tx Rx EQ 25G BT (POST:MAIN:PRE:DFE)          : 0:0:0:0
```

5-Engineering Mode Commands

nvm

```
#231: [GLOB] Tx Rx EQ 10G BT (POST:MAIN:PRE:DFE) : 0:8:0:0
#
#
#                               Group: pre-boot (Group 8)
#-----
# 59: [PORT] Preboot OpROM {Disabled(0), Enabled(1)} : Enabled
# 61: [PORT] MBA Delay Time (0-15) : 15
# 62: [PORT] MBA Setup Hot Key {Ctrl S(0), Ctrl B(1)} : Ctrl S
# 63: [PORT] MBA hide setup prompt
#           {Disabled(0), Enabled(1)} : Disabled
# 67: [PORT] Preboot Link Speed
#           {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
#           50G(6)} : Autoneg
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : PXE
# 80: [GLOB] Expansion ROM size
#           {Disabled(0), 2K(1), 4K(2), 8K(3), 16K(4),
#           32K(5), 64K(6), 128K(7), 256K(8), 512K(9),
#           1M(10), 2M(11), 4M(12), 8M(13), 16M(14),
#           32M(15)} : 256K
#132: [FUNC] Preboot VLAN value : 0
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 70: [GLOB] Enable SRIOV {Disabled(0), Enabled(1)} : Enabled
# 71: [GLOB] Enable ATC {Disabled(0), Enabled(1)} : Disabled
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: mng_fw (Group 5)
#-----
#109: [GLOB] SMBus Address : 0x49
#110: [GLOB] NCSI Package ID : 0
#111: [GLOB] Sideband Mode {Disabled(0), RMII(1), SGMII(2)} : Disabled
#112: [GLOB] SMBus Mode {Disabled(0), 100Khz(1), 400Khz(2)} : 100Khz
#113: [GLOB] NCSI {Disabled(0), Enabled(1)} : Enabled
#131: [GLOB] On Chip Sensor Mode
```

5-Engineering Mode Commands

nvm

```
#           {Disabled(0), Int_Ext_I2C(1), Int_Only(2),
#           Int_Ext_SMBUS(3)}                               : Disabled
#141: [GLOB] PLDM sensor mode
#           {internal(0), external(1), both(2)}             : internal
#142: [GLOB] External thermal sensor
#           {Disabled(0), Enabled(1)}                       : Disabled
#143: [GLOB] External thermal sensor address                 : 0x78
#203: [GLOB] Max cont operating temp                         : 95
#208: [GLOB] DCI support {Disabled(0), Enabled(1)}          : Enabled
#209: [GLOB] PCIE VDM ENABLED {Disabled(0), Enabled(1)}     : Disabled
#243: [GLOB] voltage regulator type {Disabled(0), TI(1)}    : Disabled
#
#
#
#####
### Device 2 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                     : 00:10:18:11:11:19
# 10: [PORT] Led Mode
#           {Mac1(0), Phy1(1), Phy2(2), Phy3(3), Mac2(4),
#           Phy4(5), Phy5(6), Phy6(7), Mac3(8), Phy7(9),
#           Phy8(10), Phy9(11), Mac4(12), Phy10(13),
#           Phy11(14), Phy12(15), Breakout(16)}             : Mac1
#102: [PORT] LED speed select (S2:S1:S0)
#           {1G(0x1), 10G(0x2), 25G(0x4), 40G(0x8),
#           50G(0x10)}                                       : 0:4:3
#
#
#
#                               Group: features (Group 3)
#-----
# 20: [PORT] RoCE priority                                   : 0
# 22: [PORT] Enable WoL on ACPI pattern
#           {Disabled(0), Enabled(1)}                       : Disabled
# 23: [PORT] Magic Packet WoL {Disabled(0), Enabled(1)}     : Disabled
# 26: [PORT] DCBX Mode
#           {Disabled(0), IEEE(1), CEE(2), Dynamic(3)}      : Dynamic
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}       : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                         : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                         : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                               : 0
# 96: [FUNC] Max Bandwidth                                  : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}   : Disabled
```

5-Engineering Mode Commands

nvm

```
# 99: [PORT] LLDP MAC address : 00:10:18:11:11:22
#118: [PORT] ETH DID suffix : 0x70
#119: [PORT] FCoE DID suffix : 0x80
#120: [PORT] iSCSI DID suffix : 0x84
#122: [PORT] Default enabled protocols
# {Ethernet(0x1), FCoE(0x2), iSCSI(0x4)} : 0x1
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
# {None(0), RoCE(1), iWARP(2), Both(3)} : RoCE
#213: [FUNC] NPAR enabled protocol
# {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
# RDMA(0x8)} : 0x9
#235: [PORT] PHY Module Dead Temp TH : 0
#236: [PORT] PHY Module aLOM fan on temp TH : 0
#
#
# Group: link settings (Group 4)
#-----
# 27: [PORT] Drv Speed capability mask
# {1G(0x1), 10G(0x2), 25G(0x8), 40G(0x10),
# 50G(0x20)} : 0xa
# 28: [PORT] MFW Speed capability mask
# {1G(0x1), 10G(0x2), 25G(0x8), 40G(0x10),
# 50G(0x20)} : 0xa
# 29: [PORT] Drv Link speed
# {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
# 50G(6)} : Autoneg
# 30: [PORT] Drv Flow control
# {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 31: [PORT] MFW Link speed
# {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
# 50G(6)} : Autoneg
# 32: [PORT] MFW Flow control
# {Autoneg(0x1), Rx(0x2), Tx(0x4)} : 0x7
# 33: [PORT] Optic Module Vendor Enforcement
# {Disabled(0), Enabled(1)} : Disabled
#140: [PORT] AN 25G/50G OUI {Consortium(0), BAM(1)} : Consortium
#145: [PORT] FEC force mode
# {None(0), Firecode(1), RS(2), Auto(7)} : None
#212: [PORT] FEC AN mode
# {None(0), 10G_Firecode(1), 25G_Firecode(2),
# 10G_and_25G_Firecode(3), 25G_RS(4),
# 25G_Firecode_and_RS(5), All(6)} : All
#239: [PORT] SmartLinQ Mode {Disabled(0), Enabled(1)} : Disabled
#
```

5-Engineering Mode Commands

nvm

```
#
#
# Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
# {Disabled(0), 64K(1), 128K(2), 256K(3),
# 512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
# 32M(10), 64M(11), 128M(12), 256M(13),
# 512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
# {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
# 16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
# 512M(14), 1G(15)} : 8M
#
#
# Group: phy (Group 7)
#-----
# 55: [PORT] External PHY type {None(0), BCM8485x(1)} : None
# 56: [PORT] External PHY address : 0x0
#192: [PORT] Ext PHY reset
# {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
# GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
# GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
# GPIO12(13), GPIO13(14), GPIO14(15),
# GPIO15(16), GPIO16(17), GPIO17(18),
# GPIO18(19), GPIO19(20), GPIO20(21),
# GPIO21(22), GPIO22(23), GPIO23(24),
# GPIO24(25), GPIO25(26), GPIO26(27),
# GPIO27(28), GPIO28(29), GPIO29(30),
# GPIO30(31), GPIO31(32)} : NA
#193: [PORT] EEE power saving mode
# {Disabled(0), Balanced(1), Aggressive(2), Low
# latency(3)} : Disabled
#
#
# Group: pre-boot (Group 8)
#-----
# 59: [PORT] Preboot OpROM {Disabled(0), Enabled(1)} : Enabled
# 61: [PORT] MBA Delay Time (0-15) : 15
# 62: [PORT] MBA Setup Hot Key {Ctrl S(0), Ctrl B(1)} : Ctrl S
# 63: [PORT] MBA hide setup prompt
# {Disabled(0), Enabled(1)} : Disabled
# 67: [PORT] Preboot Link Speed
# {Autoneg(0), 1G(1), 10G(2), 25G(4), 40G(5),
# 50G(6)} : Autoneg
```

5-Engineering Mode Commands

nvm

```
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : PXE
#132: [FUNC] Preboot VLAN value : 0
#133: [FUNC] Preboot VLAN : 0
#
#
#           Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#           Group: board io (Group 2)
#-----
#114: [PORT] Transceiver Module Absent
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : GPIO11
#126: [PORT] Port Type
#           {Undefined(0), Module(1), Backplane(2),
#           Ext_phy(3), Module_slave(4)} : Module
#127: [PORT] Tx Disable
#           {NA(0), GPIO0(1), GPIO1(2), GPIO2(3), GPIO3(4),
#           GPIO4(5), GPIO5(6), GPIO6(7), GPIO7(8),
#           GPIO8(9), GPIO9(10), GPIO10(11), GPIO11(12),
#           GPIO12(13), GPIO13(14), GPIO14(15),
#           GPIO15(16), GPIO16(17), GPIO17(18),
#           GPIO18(19), GPIO19(20), GPIO20(21),
#           GPIO21(22), GPIO22(23), GPIO23(24),
#           GPIO24(25), GPIO25(26), GPIO26(27),
#           GPIO27(28), GPIO28(29), GPIO29(30),
#           GPIO30(31), GPIO31(32)} : GPIO8
#
```

5-Engineering Mode Commands

nvm

```
#
#
#####
### Device 3 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address           : 00:10:18:11:11:12
#
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID   : 0x0000
# 82: [FUNC] Bar1 size
#      {Disabled(0), 64K(1), 128K(2), 256K(3),
#      512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#      32M(10), 64M(11), 128M(12), 256M(13),
#      512M(14), 1G(15)}           : 128K
#191: [FUNC] Bar2 size
#      {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#      16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#      512M(14), 1G(15)}           : 1M
#
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#      {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value    : 0
#133: [FUNC] Preboot VLAN          : 0
#
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF   : 16
# 75: [FUNC] VF PCI BAR2 size
#      {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#      64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#      2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#      64M(15)}                     : Disabled
# 84: [FUNC] VF PCI Device ID       : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}           : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                             : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                             : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                   : 0
# 96: [FUNC] Max Bandwidth                                       : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}      : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}           : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}              : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                           : 0x9
#
#
#
#####
### Device 4 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                         : 00:10:18:11:11:1a
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                               : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}     : NONE
#132: [FUNC] Preboot VLAN value                                 : 0
```

5-Engineering Mode Commands

nvm

```
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#       {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x9
#
#
#####
### Device 5 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:11:13
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
```

5-Engineering Mode Commands

nvm

```
#          512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#          32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)}                                : 128K
#191: [FUNC] Bar2 size
#          {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#          16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)}                                : 1M
#
#
#                                     Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#          {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}    : NONE
#132: [FUNC] Preboot VLAN value                               : 0
#133: [FUNC] Preboot VLAN                                    : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                             : 16
# 75: [FUNC] VF PCI BAR2 size
#          {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#          64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)}                                           : Disabled
# 84: [FUNC] VF PCI Device ID                                 : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#          {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}         : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                           : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                           : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                 : 0
# 96: [FUNC] Max Bandwidth                                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}    : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Disabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)}            : None
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)}                                           : 0x9
#
#
```

5-Engineering Mode Commands

nvm

```
#
#####
### Device 6 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address           : 00:10:18:11:11:1b
#
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID   : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 1M
#
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value    : 0
#133: [FUNC] Preboot VLAN          : 0
#
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF   : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                     : Disabled
# 84: [FUNC] VF PCI Device ID       : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}           : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                             : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                             : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                    : 0
# 96: [FUNC] Max Bandwidth                                       : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}      : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}           : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}               : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                           : 0x9
#
#
#
#####
### Device 7 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                         : 00:10:18:11:11:14
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                               : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                     : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                     : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}      : NONE
```


5-Engineering Mode Commands

nvm

```
#132: [FUNC] Preboot VLAN value           : 0
#133: [FUNC] Preboot VLAN                 : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF           : 16
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                           : Disabled
# 84: [FUNC] VF PCI Device ID               : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#       {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr         : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr        : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight               : 0
# 96: [FUNC] Max Bandwidth                  : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)}                           : 0x9
#
#
#
#####
### Device 8 ###
#####
#                                     Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                     : 00:10:18:11:11:1c
#
#
#                                     Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID           : 0x0000
# 82: [FUNC] Bar1 size
```

5-Engineering Mode Commands

nvm

```
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                               : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                               : 1M
#
#
#                                     Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}   : NONE
#132: [FUNC] Preboot VLAN value                               : 0
#133: [FUNC] Preboot VLAN                                     : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                             : 16
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)}                                         : Disabled
# 84: [FUNC] VF PCI Device ID                                 : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}        : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                          : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                          : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                : 0
# 96: [FUNC] Max Bandwidth                                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}    : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}           : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                         : 0x9
#
```

5-Engineering Mode Commands

nvm

```
#
#
#####
### Device 9 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address           : 00:10:18:11:11:15
#
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID   : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 1M
#
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value    : 0
#133: [FUNC] Preboot VLAN          : 0
#
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF   : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                     : Disabled
# 84: [FUNC] VF PCI Device ID       : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}           : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                             : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                             : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                   : 0
# 96: [FUNC] Max Bandwidth                                       : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}      : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}           : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}              : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                           : 0x9
#
#
#
#####
### Device 10 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                         : 00:10:18:11:11:1d
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                               : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                     : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                     : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}      : NONE
```

5-Engineering Mode Commands

nvm

```
#132: [FUNC] Preboot VLAN value : 0
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#       {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x9
#
#
#
#####
### Device 11 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:11:16
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
```

5-Engineering Mode Commands

nvm

```
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                               : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                               : 1M
#
#
#                                     Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}   : NONE
#132: [FUNC] Preboot VLAN value                               : 0
#133: [FUNC] Preboot VLAN                                     : 0
#
#
#                                     Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF                             : 8
# 75: [FUNC] VF PCI BAR2 size
#           {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#           64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#           2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#           64M(15)}                                         : Disabled
# 84: [FUNC] VF PCI Device ID                                 : 0x8090
#
#
#                                     Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}        : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                           : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                           : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                 : 0
# 96: [FUNC] Max Bandwidth                                    : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}    : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}         : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}           : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                       : 0x9
#
```

5-Engineering Mode Commands

nvm

```
#
#
#####
### Device 12 ###
#####
#
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address           : 00:10:18:11:11:1e
#
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID   : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}           : 1M
#
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value    : 0
#133: [FUNC] Preboot VLAN          : 0
#
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF   : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                     : Disabled
# 84: [FUNC] VF PCI Device ID       : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}           : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                             : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                             : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                   : 0
# 96: [FUNC] Max Bandwidth                                       : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}      : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}           : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}              : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                           : 0x9
#
#
#
#####
### Device 13 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                         : 00:10:18:11:11:17
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                               : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}     : NONE
#132: [FUNC] Preboot VLAN value                                : 0
```


5-Engineering Mode Commands

nvm

```
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#       {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x9
#
#
#####
### Device 14 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address : 00:10:18:11:11:1f
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
```

5-Engineering Mode Commands

nvm

```
#          512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#          32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)} : 128K
#191: [FUNC] Bar2 size
#          {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#          16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#          512M(14), 1G(15)} : 1M
#
#
#          Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#          {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value : 0
#133: [FUNC] Preboot VLAN : 0
#
#
#          Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#          {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#          64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#          2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#          64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#          Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#          {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#          {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#          {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#          RDMA(0x8)} : 0x9
#
#
```

5-Engineering Mode Commands

nvm

```
#
#####
### Device 15 ###
#####

#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address          : 00:10:18:11:11:18
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID : 0x0000
# 82: [FUNC] Bar1 size
#       {Disabled(0), 64K(1), 128K(2), 256K(3),
#       512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#       32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 128K
#191: [FUNC] Bar2 size
#       {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#       16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#       512M(14), 1G(15)}          : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#       {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)} : NONE
#132: [FUNC] Preboot VLAN value   : 0
#133: [FUNC] Preboot VLAN        : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)}                  : Disabled
# 84: [FUNC] VF PCI Device ID     : 0x8090
#
#
```

5-Engineering Mode Commands

nvm

```
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#           {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)}           : Ethernet
# 93: [FUNC] FCoE node wwn mac addr                             : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr                             : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight                                   : 0
# 96: [FUNC] Max Bandwidth                                     : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)}      : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)}          : Disabled
#202: [FUNC] RDMA enablement
#           {None(0), RoCE(1), iWARP(2), Both(3)}              : None
#213: [FUNC] NPAR enabled protocol
#           {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#           RDMA(0x8)}                                          : 0x9
#
#
#
#####
### Device 16 ###
#####
#                               Group: board (Group 1)
#-----
# 1: [FUNC] MAC address                                         : 00:10:18:11:11:20
#
#
#                               Group: pcie (Group 6)
#-----
# 37: [FUNC] MF Vendor Device ID                               : 0x0000
# 82: [FUNC] Bar1 size
#           {Disabled(0), 64K(1), 128K(2), 256K(3),
#           512K(4), 1M(5), 2M(6), 4M(7), 8M(8), 16M(9),
#           32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 128K
#191: [FUNC] Bar2 size
#           {Disabled(0), 1M(5), 2M(6), 4M(7), 8M(8),
#           16M(9), 32M(10), 64M(11), 128M(12), 256M(13),
#           512M(14), 1G(15)}                                   : 1M
#
#
#                               Group: pre-boot (Group 8)
#-----
# 69: [FUNC] Preboot Boot Protocol
#           {PXE(0), iSCSI Boot(3), FCoE Boot(4), NONE(7)}     : NONE
```

5-Engineering Mode Commands

nvm

```
#132: [FUNC] Preboot VLAN value : 0
#133: [FUNC] Preboot VLAN : 0
#
#
#                               Group: VF (Group 9)
#-----
# 74: [FUNC] Number of VFs per PF : 8
# 75: [FUNC] VF PCI BAR2 size
#       {Disabled(0), 4K(1), 8K(2), 16K(3), 32K(4),
#       64K(5), 128K(6), 256K(7), 512K(8), 1M(9),
#       2M(10), 4M(11), 8M(12), 16M(13), 32M(14),
#       64M(15)} : Disabled
# 84: [FUNC] VF PCI Device ID : 0x8090
#
#
#                               Group: features (Group 3)
#-----
# 92: [FUNC] Personality
#       {Ethernet(0), iSCSI(1), FCoE(2), RoCE(3)} : Ethernet
# 93: [FUNC] FCoE node wwn mac addr : 00:00:00:00:00:00
# 94: [FUNC] FCoE port wwn mac addr : 00:00:00:00:00:00
# 95: [FUNC] Bandwidth weight : 0
# 96: [FUNC] Max Bandwidth : 100
# 97: [FUNC] Pause on host ring {Disabled(0), Enabled(1)} : Disabled
#187: [FUNC] Function hide {Disabled(0), Enabled(1)} : Disabled
#202: [FUNC] RDMA enablement
#       {None(0), RoCE(1), iWARP(2), Both(3)} : None
#213: [FUNC] NPAR enabled protocol
#       {Ethernet(0x1), FCoE(0x2), iSCSI(0x4),
#       RDMA(0x8)} : 0x9
#
#
#
#####
### Shared ###
#####
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(IS_SPECIAL_NVM_MODE) 1}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 1} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 0}
nvm cfg \
{8=0}\
{9=1}\
{11=0}\
```

5-Engineering Mode Commands

nvm

```
{13=1077}\  
{15=2}\  
{16=0}\  
{17=1}\  
{38=12}\  
{45=0:0:0:0}\  
{46=0:0:0:0}\  
{50=3:3:3:3}\  
{51=0:0:0:0}\  
{52=1}\  
{70=1}\  
{71=0}\  
{76=0x1077}\  
{78=0x1590}\  
{79=0x021d}\  
{80=8}\  
{87=d:d:d:d}\  
{88=a:a:a:a}\  
{89=08:01:22:02:01}\  
{90=03:09:17}\  
{98=0}\  
{100=10:00}\  
{101=20:00}\  
{103=0:2:1:3}\  
{105=1}\  
{107=00:00:00}\  
{108=0x0}\  
{109=0x49}\  
{110=0}\  
{111=0}\  
{112=1}\  
{113=1}\  
{117=0x19}\  
{125=0}\  
{128=4}\  
{130=0}\  
{131=0}\  
{134=1}\  
{135=110}\  
{136=5}\  
{137=105}\  
{139=1}\  
{141=0}\  
{142=0}\  
{143=0x78}\
```

5-Engineering Mode Commands

nvm

```
{144=5}\
{146=0x0}\
{189=0x1}\
{190=0:2:1:3}\
{196=0}\
{197=0}\
{198=21}\
{199=1}\
{200=3}\
{203=95}\
{204=0}\
{205=0:0:0:0}\
{206=19}\
{207=0}\
{208=1}\
{209=0}\
{210=867328-B21}\
{211=869570-001}\
{214=3:3:3:3}\
{215=d:d:d:d}\
{216=a:a:a:a}\
{217=0}\
{218=95}\
{219=0}\
{220=30}\
{221=32}\
{222=0}\
{223=0}\
{224=10:d:3:1}\
{225=a:d:3:1}\
{226=a:d:3:1}\
{227=5:7:0:1}\
{228=5:7:0:0}\
{229=0:f:0:0}\
{230=0:0:0:0}\
{231=0:8:0:0}\
{232=1}\
{233=0}\
{234=1}\
{237=16}\
{238=0}\
{243=0}\
```

5-Engineering Mode Commands

nvm

```
#####  
### Device 1 ###  
#####
```

```
nvm cfg \  
{10=0}\  
{20=0}\  
{22=0}\  
{23=0}\  
{26=3}\  
{27=0xa}\  
{28=0xa}\  
{29=0}\  
{30=0x7}\  
{31=0}\  
{32=0x7}\  
{33=0}\  
{55=0}\  
{56=0x0}\  
{59=1}\  
{61=15}\  
{62=0}\  
{63=0}\  
{67=0}\  
{102=0:4:3}\  
{114=16}\  
{118=0x70}\  
{119=0x80}\  
{120=0x84}\  
{122=0x1}\  
{126=1}\  
{127=17}\  
{140=0}\  
{145=0}\  
{192=0}\  
{193=0}\  
{212=6}\  
{235=0}\  
{236=0}\  
{239=0}\  
{37=0x0000}\  
{69=0}\  
{74=16}\  
{75=0}\  
{82=2}
```


5-Engineering Mode Commands

nvm

```
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=8}\
{202=1}\
{213=0x9}\

#####
### Device 2 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 2} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 1}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{10=0}\
{20=0}\
{22=0}\
{23=0}\
{26=3}\
{27=0xa}\
{28=0xa}\
{29=0}\
{30=0x7}\
{31=0}\
{32=0x7}\
{33=0}\
{55=0}\
{56=0x0}\
{59=1}\
{61=15}\
{62=0}\
{63=0}\
{67=0}\
{102=0:4:3}\
{114=12}\
{118=0x70}\
{119=0x80}\
{120=0x84}\
{122=0x1}\
```

5-Engineering Mode Commands

nvm

```
{126=1}\
{127=9}\
{140=0}\
{145=0}\
{192=0}\
{193=0}\
{212=6}\
{235=0}\
{236=0}\
{239=0}\
{37=0x0000}\
{69=0}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=8}\
{202=1}\
{213=0x9}\

#####
### Device 3 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 3} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 2}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
```

5-Engineering Mode Commands

nvm

```
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 4 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 4} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 3}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 5 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 5} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 4}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
```

5-Engineering Mode Commands

nvm

```
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 6 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 6} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 5}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 7 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 7} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 6}
```

5-Engineering Mode Commands

nvm

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 8 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 8} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 7}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=16}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\
```

5-Engineering Mode Commands

nvm

```
#####
### Device 9 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 9} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 8}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=8}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 10 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 10} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 9}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=8}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
```

5-Engineering Mode Commands

nvm

```
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 11 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 11} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 10}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=8}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 12 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 12} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 11}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=8}\
{75=0}\
{82=2}\
```

5-Engineering Mode Commands

nvm

```
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 13 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 13} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 12}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
nvm cfg \
{37=0x0000}\
{69=7}\
{74=8}\
{75=0}\
{82=2}\
{84=0x8090}\
{92=0}\
{95=0}\
{96=100}\
{97=0}\
{132=0}\
{133=0}\
{187=0}\
{191=5}\
{202=0}\
{213=0x9}\

#####
### Device 14 ###
#####

if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 14} else {set
::current(SPECIAL_NVM_MODE_FUNC_NUM) 13}
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}
```


5-Engineering Mode Commands

nvm

```
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=0}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=0}\  
{213=0x9}\  

```

```
#####  
### Device 15 ###  
#####
```

```
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 15} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 14}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=0}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=0}\  
{213=0x9}\  

```

5-Engineering Mode Commands

nvm

```
#####  
### Device 16 ###  
#####  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {device 16} else {set  
::current(SPECIAL_NVM_MODE_FUNC_NUM) 15}  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(SPECIAL_NVM_MODE_NUM_PORTS) 2}  
nvm cfg \  
{37=0x0000}\  
{69=7}\  
{74=8}\  
{75=0}\  
{82=2}\  
{84=0x8090}\  
{92=0}\  
{95=0}\  
{96=100}\  
{97=0}\  
{132=0}\  
{133=0}\  
{187=0}\  
{191=5}\  
{202=0}\  
{213=0x9}\  
  
if {[catch {set x $::current(IS_SPECIAL_NVM_MODE)}}] {} else {set  
::current(IS_SPECIAL_NVM_MODE) 0}  
device 1
```

nvm chk

Authority nvm chk <board type> [-fix]

Syntax This command compares the existing NVRAM configuration to the default configuration setting expected for this board type. The `-fix` option corrects mismatching configuration on the board.

Example 1-HwFn0:> **nvm chk T8004G_10GE**
 The following errors were found in the configuration:
 ID 9: "MF Mode" mismatch:
 Expected 1
 Existing 0

nvm crc

Authority nvm crc [-fix <entry>]

Syntax This command computes the image CRC and compares it to the stored CRC in the NVRAM end-of-image. The `-fix` option corrects mismatching configuration on the board.

nvm dir

Authority nvm dir [-delete <entry>]

Syntax This command does the following:

- Displays a listing of the firmware programmed in NVRAM
- Shows the image offset in NVRAM, size (in byte count), and version
- Deletes a particular image (optional)

Example

```
1-HwFn0:> nvm dir
```

| Image | SRAM Addr | SRAM Run Addr | NVM Offset | Byte Cnt | Version |
|-------------|------------|---------------|------------|----------|-------------|
| LIM | 0x08035800 | 0x08035824 | 0x00000040 | 0x00aec | LIM 8.3.1.0 |
| DIR1* | 0x00000000 | 0x00000000 | 0x00002000 | 0x0018c | |
| DIR2 | 0x00000000 | 0x00000000 | 0x00003000 | 0x0018c | |
| MIM1 | 0x08000A40 | 0x08011994 | 0x00004000 | 0x6cd4c | MIM 8.5.3.0 |
| MIM2* | 0x08000A40 | 0x08011994 | 0x00130000 | 0x6d128 | MIM 8.5.4.0 |
| NVM_CFG1 | 0x00000000 | 0x00000000 | 0x0025C000 | 0x01094 | |
| TIM2* | 0x08028000 | 0x080284A0 | 0x0025E000 | 0x025d0 | TIM 8.5.4.0 |
| EAGLE_FW2* | 0x00000000 | 0x00000000 | 0x00261000 | 0x07404 | |
| FALCON_FW2* | 0x00000000 | 0x00000000 | 0x00269000 | 0x07c14 | |

| | | | | | |
|-------------|------------|------------|------------|-----------------|-------------|
| PCIE_FW2* | 0x00000000 | 0x00000000 | 0x00271000 | 0x01bb8 | |
| AVS_FW2* | 0x00000000 | 0x00000000 | 0x00273000 | 0x03c04 | |
| TIM1 | 0x08028000 | 0x080284A0 | 0x00288000 | 0x02564 | TIM 8.5.3.0 |
| EAGLE_FW1 | 0x00000000 | 0x00000000 | 0x0028B000 | 0x07404 | |
| FALCON_FW1 | 0x00000000 | 0x00000000 | 0x00293000 | 0x07c14 | |
| PCIE_FW1 | 0x00000000 | 0x00000000 | 0x0029B000 | 0x01bb8 | |
| AVS_FW1 | 0x00000000 | 0x00000000 | 0x0029D000 | 0x03c04 | |
| MFW_TRACE2* | 0x00000000 | 0x00000000 | 0x002A1000 | 0x115d4 | |
| MBA | 0x00000000 | 0x00000000 | 0x002B3000 | 0x7d804 | |
| | | | | EFI x64 1.0.0.2 | |
| MFW_TRACE1 | 0x00000000 | 0x00000000 | 0x00331000 | 0x115d4 | |

 * Running Image

The following example deletes an image from the NVRAM.

```
1:> nvm dir -delete MBA
```

nvm dump

The nvm dump commands read all or part of the NVRAM into a file or a specific image type into a file, as described in the following sections.

nvm dump <filename>

Authority nvm dump [<offset> <length>] <filename>

Syntax This command reads from the NVRAM and writes the contents to the file <filename>.

If the <offset> and <length> parameters are not provided, this command reads the NVRAM starting at offset zero.

If the <offset> and <length> parameters are provided, this command reads the NVRAM, starting at <offset> and ending at <offset+length>, into a file.

nvm dump -mdump <filename>

Authority nvm dump -mdump <filename>

Syntax This command extracts the current mdump (CRASH_DUMP) binary image to a file.

nvm prg

Authority `nvm prg [options] <offset> <filename>`

The `[options]` parameter value is:

`-raw` Programs the complete raw image, including the MAC address and serial number. If this option is not used, the MAC address and serial number are preserved.

Syntax This command programs the NVRAM starting at the `<offset>` parameter with the data read from the file specified by the `<filename>` parameter.

nvm read

Authority `nvm read <addr>`

Syntax This command reads a DWORD from NVRAM at the offset `<addr>` parameter and returns the value read.

Example `1:> nvm read 0`
 `0x669955aa`

nvm show

Authority `nvm show <begin_addr> [nbytes]`

Syntax This command performs the same operation as the `nvm read` command, except that it formats the output in a more user-friendly format. The optional argument `[nbytes]` (which must be a multiple of four) specifies the number of bytes to display.

Example

```
1:> nvm show 0 0x40
00000000:669955aa 08006c00 0000030c 00000ec8 3a1090f5 08000000 07001f98 00002000
0000020:00000000 11010840 0000c400 00000000 00000000 00000000 00000000 00000000
```

nvm write

Authority `nvm write <addr> <value1> [value2...]>`

Syntax This command writes the NVRAM, beginning at the offset `<addr>` parameter, with the 32-bit value `value1` [`value 2...`].

nvm fill

Authority `nvm fill <addr> <bytecount> <value32 | inc | addr>`

Syntax This command fills the NVRAM, starting at the offset `<addr>` parameter, for a length of `<bytecount>` bytes.

If the argument `value32` is specified, this command fills each NVRAM DWORD with that 32-bit value.

If the argument `inc` is specified, this command fills each NVRAM DWORD with an incrementing value starting from zero.

If the argument `addr` is specified, this command fills each NVRAM DWORD with its offset value.

Example

```
1:> nvm fill 0 0x100 0x55aa55aa
Programming from: 0x00000000 to 0x00000100

Done
```

nvm upgrade

Authority `nvm upgrade [-F] <option> <filename>`

Syntax This command upgrades the firmware for the Marvell FastLinQ Controllers.

The `<filename>` parameter specifies the name of the file that contains the appropriate image.

If the upgrade version is the same or older than the version in NVRAM, the upgrade is aborted.

The `-F` option forces the upgrade without checking the version.

The `<option>` parameter is one of the following:

| Option | Firmware Name |
|----------------------|--|
| <code>-mfw</code> | management firmware |
| <code>-l2b</code> | L2 firmware |
| <code>-mba</code> | multiple boot agent (MBA) (pre-execution (PXE)) code |
| <code>-ccm</code> | comprehensive configuration management (CCM) image |
| <code>-optm</code> | qualified optic modules information |
| <code>-vpd</code> | virtual product data |
| <code>-hw_set</code> | hardware set commands |
| <code>-pk</code> | public key image |

| Option | Firmware Name |
|-----------------------|---|
| <code>-mdump</code> | mdump image (also referred to as <code>CRASH_DUMP</code>) in NVRAM. |
| <code>-skip_id</code> | MBI image (upgraded without performing the four-part ID check) |
| <code>-mbi</code> | monolithic binary image. When this upgrade is successful, a new MBI version and date are set. When this upgrade is not successful, the MBI version and date are set to zero. A single-image upgrade also sets the MBI version and date to zero. |
| <code>-init_hw</code> | initialize binary image values |

nvm factory_defaults

Authority `nvm factory_defaults`

Syntax This command restores the NVRAM configurations that were set during the board's manufacturing process.

This command uses a new `DEFAULT_CFG` image (programmed during the manufacturing process) to restore the default board configuration.

nvm update_cfg

Authority `nvm update_cfg <NVM_CFG1 binary file> <nvm_meta text file>`

Syntax This command upgrades the NVRAM configuration (on an already configured board) to a new configuration, retaining option changes made by the user.

This command upgrades both the NVRAM configuration image `NVM_CFG1` (per the new incoming `NVM_CFG1` binary file) and the new incoming `META` text file, taking into account the onboard `NVM_CFG1`, `DEFAULT_CFG`, and `META` images. In addition, this command also upgrades the `DEFAULT_CFG` and `META` images, as described in the following paragraphs.

The `update_cfg` command reads the incoming `<nvm_meta text file>`, which includes NVRAM configuration options and their descriptions; for example, the offset in NVRAM, size, entity type (glob/path/port/function), configuration type (hardware/user), and so on.

For each NVRAM configuration option in the incoming `<nvm_meta text file>`, the command checks if this option already exists in the onboard `META` image. If option already exists and it is a user-configuration type, the command checks if the default value has been changed by the user.

For the user-configuration type, if the option value differs from the default, the command updates the NVM_CFG1 image option with the onboard value. Otherwise, the command updates the NVM_CFG1 image option with the new value from the NVM_CFG1 image.

Example

1. Prepare a board using a kit with an older NVRAM configuration (for example, 45212_25v21.01.03, which includes qeDiag 8.8.4.0).
 - a. Erase the board.
 - b. Issue the following command to create a DEFUAL_CFG image:
`run ./update.sh`
 - c. Reboot
 - d. Change a few options (per port) from their default values. For example:

```
nvm cfg 10=1, (glob) nvm cfg 24-40
```

2. Install qeDiag version 8.10.12.0 or later.
3. Install the kit *old* META image on board using qeDiag version 8.10.12.0 by issuing the following command:

```
nvm upgrade -meta  
45212_25v21.01.03/qllediag_8.8.4.0/nvm_meta.txt
```

4. Issue the following command to compare the NVRAM configuration to the expected configuration before running the `nvm update_cfg` command:

```
nvm chk <board type>
```

The results should show multiple differences.

5. Create an NVM_CFG1 binary file <vba_board_type.bin> and META text file <nvm_meta.txt> from the NVM configuration excel file <E4_nvm_cfg.xlsm>.

6. Issue the `nvm update_cfg` command:

```
nvm update_cfg <vba_qboard_type.bin> <nvm_meta.txt>
```

7. Compare the NVRAM configuration.

The results of the `nvm chk <board type>` command should show only differences in the options that were changed from their default value in [Step 1](#) (for example, 10 and 24).

nvm trace_modules

Authority `nvm trace_modules`
 `nvm trace_modules show [module bit# | all]`
 `nvm trace_modules hide [module bit# | all]`

Syntax The `nvm trace_modules` command retrieves the current trace module status (which module traces are shown and which module traces are hidden).

The `nvm trace_modules show [module bit# | all]` command shows traces of either the requested modules (according to module bits) or all modules. Issue the `mcp reset` command for these changes to take effect.

The `nvm trace_modules hide [module bit# | all]` command hides traces of the requested modules (according to module bits) or all modules. Issue the `mcp reset` command for these changes to take effect.

Example

```
1-HwFn0:> nvm trace_modules
Current trace modules status: 0xffffffff
( 0) ERROR      =Show ( 1) DBG          =Hide ( 2) DRV_HSI    =Hide ( 3) INTERRUPT =Hide
( 4) TEMPERATURE=Show ( 5) FLR          =Hide ( 6) INIT      =Hide ( 7) NVM        =Hide
( 8) PIM         =Show ( 9) NET          =Hide (10) POWER      =Hide (11) UTILS     =Hide
(12) RESOURCES  =Hide (13) SCHEDULER    =Hide (14) PHYMOD     =Hide (15) EVENTS    =Hide
(16) PMM         =Hide (17) DBG_DRV     =Hide (18) ETH         =Hide (19) SECURITY  =Hide
(20) PCIE        =Hide (21) TRACE       =Hide (22) MANAGEMENT=Hide (23) SIM        =Hide
(24) BUF_MGR     =Hide
```

```
1-HwFn0:> nvm trace_modules show 9 18
[nvm_trace_modules_cmd:3715] Set trace modules 0xffffbfcee
```

```
1-HwFn0:> nvm trace_modules
Current trace modules status: 0xffffbfcee
( 0) ERROR      =Show ( 1) DBG          =Hide ( 2) DRV_HSI    =Hide ( 3) INTERRUPT =Hide
( 4) TEMPERATURE=Show ( 5) FLR          =Hide ( 6) INIT      =Hide ( 7) NVM        =Hide
( 8) PIM         =Show ( 9) NET          =Show (10) POWER      =Hide (11) UTILS     =Hide
(12) RESOURCES  =Hide (13) SCHEDULER    =Hide (14) PHYMOD     =Hide (15) EVENTS    =Hide
(16) PMM         =Hide (17) DBG_DRV     =Hide (18) ETH         =Show (19) SECURITY  =Hide
(20) PCIE        =Hide (21) TRACE       =Hide (22) MANAGEMENT=Hide (23) SIM        =Hide
(24) BUF_MGR     =Hide
```

pci

These commands search for PCI devices and displays their information. The `pci` commands are described in the following sections.

pci scan

Authority `pci scan`

Syntax This command scans all PCI devices in the system.

Example

```
1-HwFn0:> pci scan
```

| Domain | Bus | Dev | Func | Vendor ID | Device ID | Base/IO Address | IRQ |
|--------|-----|-----|-------|-----------|-----------|-------------------|-----|
| ===== | === | === | ===== | ===== | ===== | ===== | === |
| 0 | 0 | 0 | 0 | 8086 | 2774 | 00000000:00000000 | 0 |
| 0 | 0 | 1 | 0 | 8086 | 2775 | 00000000:00000000 | 0 |
| 0 | 1 | 0 | 0 | 10DE | 0165 | 00000000:00000000 | 0 |
| 0 | 2 | 0 | 0 | 1077 | 1635 | F0780000:00000010 | 0 |
| 0 | 2 | 0 | 1 | 1077 | 1635 | F0680000:00000011 | 0 |
| 0 | 2 | 0 | 2 | 1077 | 1635 | F0580000:00000010 | 0 |
| 0 | 2 | 0 | 3 | 1077 | 1635 | F0480000:00000011 | 0 |
| 0 | 2 | 0 | 4 | 1077 | 1635 | F0380000:00000010 | 0 |
| 0 | 2 | 0 | 5 | 1077 | 1635 | F0280000:00000011 | 0 |
| 0 | 2 | 0 | 6 | 1077 | 1635 | F0180000:00000010 | 0 |
| 0 | 2 | 0 | 7 | 1077 | 1635 | F0080000:00000011 | 0 |
| 0 | 4 | 0 | 0 | 14E4 | 1677 | 00000000:00000000 | 0 |

pci search

Authority `pci search [-did <deviceID>] [-vid <vendorID>] [-class <class>]`

Syntax This command searches the entire PCI or PCIe® bus for all devices with either a device ID, a vendor ID, or a class code of `<class>`. If the command finds any device, the corresponding bus number, device number, and function number are displayed.

Example

```
1-HwFn0:> pci search -vid 0x1077
```

```
{0 2 0 0} {0 2 0 1} {0 2 0 2} {0 2 0 3} {0 2 0 4} {0 2 0 5} {0 2 0 6} {0 2 0 7}
```

pci setdut

Authority `pci setdut <domain> <bus> <device> <function>`

Syntax This command sets the address of the device to be used with the `pcicfg` command. After the command, the current device will be unset (no device will be set as *current*).

pcicfg

The `pcicfg` commands provide a mechanism to read, modify, display, and write the PCI configuration space registers. These commands are described in the following sections.

pcicfg read

Authority `pcicfg read <addr>`

Syntax This command reads data from PCI configuration space using the PCI configuration cycle.

Example `1-HwFn0:> pcicfg read 0`
`0x16351077`

pcicfg show

Authority `pcicfg show <begin address> <number of bytes>`

Syntax This command performs a `pcicfg read` command with the same argument and formats user-friendly output. Adding the additional parameter `<number of bytes>` (must be a multiple of four) indicates how much data to display.

Example

```
1-HwFn0:> pcicfg show 0 0x40
0000000:16351077 00100406 02000000 00800010 f008000c 00000000 f000000c 00000000
0000020:f080000c 00000000 00000000 e4f21077 00000000 00000048 00000000 0000020a
```

pcicfg write

Authority `pcicfg write <address> <value>`

Syntax This command treats `<value>` as a 32-bit value and writes it into the configuration register at the `<address>` parameter.

Phy

The `phy` commands described in the following sections perform PHY and MAC operations.

phy raw_read

Authority `phy raw_read <port> <lane> <reg>`

Syntax This command reads the PHY registers.

phy raw_write

Authority `phy raw_write <port> <lane> <reg> <data low>`

Syntax This command reads the PHY registers.

phy core_read

Authority `phy core_read <port> <reg>`

Syntax This command reads MAC and port registers, for example:

| | |
|----------------|--------------------------|
| Port registers | 200h–209h (port type) |
| Port registers | 20Ah–22Bh (general type) |
| MAC registers | 600h–62Dh |

phy core_write

Authority `phy core_write <port> <reg> <data low> <data high>`

Syntax This command writes to the MAC and port registers.

phy mac_lb

Authority `phy mac_lb <port> <en|dis>`

Syntax This command enables MAC on this port with loopback enabled or disabled.

phy mac_stat

Authority `phy mac_stat <port> <reg>`

Syntax This command displays MAC statistics for a port (registers 00h–70h).

phy info

Authority `phy info`

Syntax This command displays the core information for all ports. For example:

| | |
|---------------------------|--|
| <code><port></code> | Global port ID, which identifies the core |
| <code><lane></code> | 0 single lane (logical lane 0) |
| | 1 single lane (logical lane 1) |
| | 2 single lane (logical lane 2) |
| | 3 single lane (logical lane 3) |
| | 4 dual lane (logical lanes 0 and 1) |
| | 5 dual lane (logical lanes 2 and 3) |
| | 6 quad lane (logical lanes 0, 1, 2, and 3) |

pretend

Authority `pretend pf/vf <pf_id/vf_id>`

Syntax The `pretend vf <vf_id>` command enables a *pretend* virtual function (VF) through its VF ID. This command is used in Linux DRV_DBG mode, over the qed driver.

The `pretend pf <pf_id>` command enables a *pretend* to a different present function and reads registers from a different physical function (PF).

Example To set four VFs, issue the following command:

```
echo 4 > /sys/bus/pci/devices/0000\:05\:00\.o/sriov_numvfs
```

Issue the following command to start qeDiag and use `pretend`:

```
[root@lb-tlvb-pcie44 qllediag_8.10.2.0]#./load.sh -eng
```

Following is a sample output.

```

NOTICE: Operational driver is loaded.
*****
      eddiag -- Qlogic Everest diagnostic tool, version 8.10.2.0
*****
      Copyright (c) 2014-2016 Qlogic Corporation, all rights reserved.

[pci_scan:626] Detected 6 devices.
Chip #1: Board: 57980   Rev: B0           MFW: 8.7.7.0   S.N.: ABC1527A12345
-----
Dev Chip.PF.HwFn   Bus      PCI      Spd      MAC      I/F  St. Drv  DrvVer Config
-----
  1    1.0.0      00:05:00:00 PCIE-16  8.0  00:10:18:11:22:33  eth19 D0 qede 8.7.9.0
  2    1.1.0      00:05:00:01 PCIE-16  8.0  00:10:18:11:22:34  eth20 D0 qede 8.7.9.0
  3    1.0.0      00:05:02:00 PCIE-0   (unknown) 00:00:00:00:00:00  eth1  D0 qede 8.7.9.0
  4    1.1.0      00:05:02:01 PCIE-0   (unknown) 00:00:00:00:00:00  eth2  D0 qede 8.7.9.0
  5    1.2.0      00:05:02:02 PCIE-0   (unknown) 00:00:00:00:00:00  eth3  D0 qede 8.7.9.0
  6    1.3.0      00:05:02:03 PCIE-0   (unknown) 00:00:00:00:00:00  eth4  D0 qede 8.7.9.0

(1-HwFn0) eth19:> pretend 1
Pretend to VF 1
(1-HwFn0).VF1 eth19:> reg_write -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT -v 1
(1-HwFn0).VF1 eth19:> pretend 2
Pretend to VF 2
(1-HwFn0).VF2 eth19:> reg_write -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT -v 2
(1-HwFn0).VF2 eth19:> pretend 3
Pretend to VF 3
(1-HwFn0).VF3 eth19:> reg_write -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT -v 3
(1-HwFn0).VF3 eth19:> pretend 4
Pretend to VF 4
(1-HwFn0).VF4 eth19:> reg_write -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT -v 4
(1-HwFn0).VF4 eth19:> pretend 1
Pretend to VF 1
(1-HwFn0).VF1 eth19:> reg_read -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT
0x1
(1-HwFn0).VF1 eth19:> pretend 2
Pretend to VF 2
(1-HwFn0).VF2 eth19:> reg_read -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT
0x2
(1-HwFn0).VF2 eth19:> pretend 3
Pretend to VF 3
(1-HwFn0).VF3 eth19:> reg_read -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT
0x3
(1-HwFn0).VF3 eth19:> pretend 4
Pretend to VF 4
(1-HwFn0).VF4 eth19:> reg_read -n IGU_REG_STATISTIC_NUM_VF_MSG_SENT
0x4

```

reg

The `reg` commands enable access to registers for read and write operations, as described in the following sections. Number 0 of the PF translation table (PTT) is reserved for the qeDiag tool to access/read the registers.

Direct read and indirect read operations (using `PCICFG_REG_WINDOW_ADDRESS` and `PCICFG_REG_WINDOW_DATA`) will be added in a future release.

reg read

Authority `reg read <address>`

Syntax This command reads the register's data.

Example `1-HwFn0:> reg read 0x2000`
`0x16351077`

reg show

Authority `reg show <begin address> <number of bytes>`

Syntax This command performs a `reg read` command from the `<begin address>` parameter.

Adding the `<number of bytes>` parameter (must be a multiple of four) indicates how much data to display.

Example

```
1-HwFn0:> reg show 0x2000 0x40
0002000:16351077 00100406 02000000 00800010 f078000c 00000000 f070000c 00000000
0002020:f087000c 00000000 00000000 e4f21077 00000000 00000048 00000000 0000010b
```

reg write

Authority `reg write <address> <value>`

Syntax This command treats the `<value>` parameter as a 32-bit value and writes it into the register at the `<address>` parameter.

There are Tcl utilities that allow reading and writing to the register by its name (and not by its address). See [Chapter 6](#) for more details.

- `reg_read -n <register name>`
- `reg_write -n <register name> -v <value>`
- `reg_offset -n <register name>`

serport

The `serport` commands allow qeDiag to control the system's serial port and redirect console input and output through the serial port. These commands are described in the following sections.

serport open

Authority `serialport open <comport> <baudrate> <settings>`

Syntax This command attempts to open a communication port for I/O to be redirected to a dumb terminal.

The baud rate can be optionally specified as one of the following:

- 1,200
- 4,800
- 9,600
- 19,200
- 38,400
- 57,600
- 115,200 (default)

The setting can also be optionally specified with either 7 or 8 data bits; any parity setting (odd, even, or none); and 1 or 2 stop bits. The default setting is 8N1.

Example `1:> serialport open 2`
opened port 2 settings: 115200, 8N1

serport redirect

Authority `serialport redirect`

Syntax This command moves the console input and output to the serial port that is open by the `serport open` command. As soon as the command is executed, all input and output are redirected to the port selected in the `serport open` command.

serport close

Authority `serialport close <comport>`

Syntax This command closes the communication port specified by the `<comport>` parameter and restores all console input and output to the standard display.

sfp

The `sfp` commands read, write, and decode SFP and quad small form pluggable (QSFP) electrically erasable programmable read-only memory (EEPROM).

The valid I²C address is A0h or A2h.

sfp read

Authority `sfp read <port> <I2C address> <offset> <size>`

Syntax This command reads and displays bytes from the EEPROM.

Example

```
1-HwFn0:> sfp read 0 0xA0 148 13
00000090: .. .. .. 46 49 4e 49 53 41 52 20 43 4f 52 50
FINISAR CORP
000000a0: 20
```

sfp write

Authority `sfp write <port> <I2C address> <offset> <size> <value>`

Syntax This command writes a byte, word, or double word (DWORD) to the EEPROM.

sfp decode

Authority `sfp decode <port>`

Syntax This command displays SFP and QSFP information.

Example

```
1-HwFn0:> sfp decode 0
QSFP+ inserted.
Transceiver compliance code 0x4
40G SR-4 detected.
Extended rate select bytes 0x0
Length (SMF) 0x0
Length (OM3 50 um) 0x32
Length (OM2 50 um) 0x0
Length (OM1 62.5 um) 0x0
Length (Passive or active) 0x0
Device technology byte 0x0
Vendor name: FINISAR CORP
Vendor OUI: 009065
Vendor PN: FTL410QE2C
Vendor rev: A
```

temperature

Authority temperature

Syntax This command displays the Marvell FastLinQ Controllers' temperature.

Keywords 1-HwFn0:> **temperature**
[tcl_al_read_temperature_cmd:135] Number of sensors 1
[tcl_al_read_temperature_cmd:143] [sensor 0] sensor_location 1,
threshold_high 105, critical 110, current_temp 87

version

Authority version

Syntax This command displays the version string for qeDiag.

vpd

These commands read, modify, display, and write a vital product data (VPD) image in the NVRAM.

The `vpd` commands are described in the following sections. Some of the commands have a `<tag name>` parameter; valid values are listed [Table 5-3](#).

Table 5-3. VPD Tags

| Tag Name | Tag Name |
|----------|--------------|
| PN | EC |
| FG | LC |
| MN | PG |
| SN | CP |
| RV | YA |
| RW | FN |
| CC | FC |
| YB | Vx |
| Yx | PRODUCT_NAME |

vpd read

Authority vpd read <tag name>

Syntax This command reads and displays a VPD tag (if it exists) from a VPD image in the NVRAM.

Valid values for the <tag name> parameter are listed in [Table 5-3](#).

Example 1-HwFn0:> vpd read SN
ABC1527A12345

vpd show

Authority vpd show

Syntax This command displays all VPD tags.

Example

```
1-HwFn0:> vpd show
vpd show
VPD information:
PRODUCT_NAME = QLogic QL45212-DE 25 Gigabit Ethernet Converged Adapter
FN = 88Y5922
EC = N32372
CC = 2CC5
PN = 88Y5921
FC = 1763
V0 = FFV0.0.0
SN = ABC1527A12345
MN = 000102030405
YB = 4f 46 4d 45 4e 41 02 08 00 00 00 00 00 00 07 00 0f 00 01 00 01 00 07 00
0f 00 02 00 02 00 06 00 0f 00 01 00 03 00 06 00 0f 00 02 00 04 00 06 00 09 00
01 00 05 00 06 00 09 00 02 00 06 00 06 00 09 00 01 00 07 00 06 00 09 00 02 00
```

vpd write

Authority vpd write <tag name> <value>

Syntax This command changes the VPD parameter <tag name> value to the <value> parameter.

Valid values for the <tag name> parameter are listed in [Table 5-3](#).

A new VPD image will be created in the NVRAM if one does not exist.

vpd delete

Authority vpd delete <tag name>

Syntax This command deletes a VPD tag (if it exists) in the VPD image in the NVRAM.
Valid values for the <tag name> parameter are listed in [Table 5-3](#).
The VPD product name tag cannot be deleted.

6 Engineering Mode Tcl Utilities

This section defines the Tcl utilities.

Engineering mode Tcl utilities can be executed from the engineering mode command prompt.

reset_chip

Authority `reset_chip`

Syntax This utility resets the Marvell FastLinQ Controllers and puts them in a stable state.

parse_rdf

Authority `parse_rdf`

Syntax This utility parses the register's register definition file (RDF) into global arrays for future use.

block_offset

Authority `block_offset -b <block name>`

Syntax This utility returns the block's offset from the base address register (BAR) by its name.

block_regs

Authority `block_regs -b <block name>`

Syntax This utility returns a list of registers of this block.

reg_read

Authority `reg_read -n <reg name>`

Syntax This utility reads a register's value by its name.

reg_offset

Authority `reg_offset -n <reg name>`

Syntax This utility retrieves the register's offset address from the BAR by its name.

reg_write

Authority `reg_write -n <reg name> -v <value>`

Syntax This utility writes a value to a register by its name.

read_regs

Authority `read_regs`

Valid arguments are:

- `-b <blockName>` Read only this block of registers.
- `-silent` Do not display the output on the screen.
- `-out <filename>` Output to the file in the `<filename>` variable.

Syntax This utility reads all the registers' values and prints them to a file.

efuse_info

Authority `efuse_info`

Syntax This utility reads the efuse information.

Example

The following example applies to the Marvell FastLinQ 41000 and 8600 Series Controllers.

```
line 0: 0xff000000
line 1: 0x0
line 2: 0x0
```

```
line 3: 0x0
line 4: 0x0
line 5: 0x0
line 6: 0x0
line 7: 0x0
line 8: 0x80f0
line 9: 0x0
line 10: 0x0
line 11: 0x0
line 12: 0x0
line 13: 0x43375438
line 14: 0x3132
line 15: 0x0
line 16: 0xff000000
line 17: 0x0
line 18: 0x0
line 19: 0x0
line 20: 0x0
line 21: 0x0
line 22: 0x0
line 23: 0x0
line 24: 0x80f0
line 25: 0x0
line 26: 0x0
line 27: 0x0
line 28: 0x0
line 29: 0x43375438
line 30: 0x3132
line 31: 0x0
```

The following example applies to the Marvell 45000 and 8500 Series Controllers.

```
Row 0 (bits 0-32): 0x0
Row 1 (bits 32-64): 0x0
Row 2 (bits 64-96): 0x0
Row 3 (bits 96-128): 0x1e87
Row 4 (bits 128-160): 0x40000000
Row 5 (bits 160-192): 0x0
Row 6 (bits 192-224): 0x0
Row 7 (bits 224-256): 0x0
Row 8 (bits 256-288): 0x4
```

```
Row 9 (bits 288-320): 0x0
Row 10 (bits 320-352): 0x0
Row 11 (bits 352-384): 0x0
Row 12 (bits 384-416): 0x0
Row 13 (bits 416-448): 0x0
Row 14 (bits 448-480): 0x210a
Row 15 (bits 480-512): 0x0
Row 16 (bits 512-544): 0x0
Row 17 (bits 544-576): 0x0
Row 18 (bits 576-608): 0x0
Row 19 (bits 608-640): 0x0
Row 20 (bits 640-672): 0x0
Row 21 (bits 672-704): 0x0
Row 22 (bits 704-736): 0x0
Row 23 (bits 736-768): 0x0
Row 24 (bits 768-800): 0x0
Row 25 (bits 800-832): 0x0
Row 26 (bits 832-864): 0x0
Row 27 (bits 864-896): 0x0
Row 28 (bits 896-928): 0x0
Row 29 (bits 928-960): 0x0
Row 30 (bits 960-992): 0x0
Row 31 (bits 992-1024): 0x0
Row 32 (bits 1024-1056): 0x0
Row 33 (bits 1056-1088): 0x0
Row 34 (bits 1088-1120): 0x0
Row 35 (bits 1120-1152): 0x0
Row 36 (bits 1152-1184): 0x0
Row 37 (bits 1184-1216): 0x0
Row 38 (bits 1216-1248): 0x0
Row 39 (bits 1248-1280): 0x0
Row 40 (bits 1280-1312): 0x0
Row 41 (bits 1312-1344): 0x0
Row 42 (bits 1344-1376): 0x0
Row 43 (bits 1376-1408): 0x0
Row 44 (bits 1408-1440): 0x0
Row 45 (bits 1440-1472): 0x0
Row 46 (bits 1472-1504): 0x0
Row 47 (bits 1504-1536): 0x0
```



```

Row 48 (bits 1536-1568): 0x0
Row 49 (bits 1568-1600): 0x0
Row 50 (bits 1600-1632): 0x0
Row 51 (bits 1632-1664): 0x0
Row 52 (bits 1664-1696): 0x0
Row 53 (bits 1696-1728): 0x0
Row 54 (bits 1728-1760): 0x0
Row 55 (bits 1760-1792): 0x0
Row 56 (bits 1792-1824): 0x0
Row 57 (bits 1824-1856): 0x0
Row 58 (bits 1856-1888): 0x0
Row 59 (bits 1888-1920): 0x0
Row 60 (bits 1920-1952): 0x0
Row 61 (bits 1952-1984): 0x0
Row 62 (bits 1984-2016): 0x0
Row 63 (bits 2016-2048): 0x0

```

```
exit
```

profile

Authority profile

Valid arguments are:

| | |
|----------------------|---|
| -forever | Runs in a loop without stopping (will never exit to user prompt) |
| -freq <frequency> | Updates actual Marvell FastLinQ controller's frequency for script calculations |
| -f <fileName> | Writes results to a file (mode 1 only) |
| -h <fileName> | Creates histograms and writes them into a file |
| -s <samples> | Specifies number of samples per iteration (default is 1000 samples) |
| -d <dev0> <dev1> | Specifies the device for engine 0 and engine 1 for joint profiling |
| -i <iterations> | Specifies the number of iterations to display profiling data (default is 10 iterations) |
| -u | When the Marvell FastLinQ controller is in stall count mode, displays real time STORM CPU utilization |

| | |
|----------|---|
| -k | Keeps results from all iterations in the terminal (without this flag, each iteration overrides the last one in the terminal) |
| -ah | Specifies the Marvell FastLinQ controller as 4100 Series |
| -bb | Specifies the Marvell FastLinQ controller as 45000 Series |
| -e5 | Specifies the Marvell FastLinQ controller as 55000 Series |
| -dec | Shows histogram values in decimal instead of hexadecimal |
| -sbucket | Uses simple bucket modes in histogram presentation |
| -ext | Includes additional registers from an external file to show in histogram mode |
| -r | Emulates a frequency ratio modifier; that is, the fastest clock speed divided by the actual speed (emulates the GUI clocks tab) |
| -storms | Chooses specific STORMs and displays their order to analyze utilization/stall count |
| -tx | Chooses transmit STORMs and displays their order to analyze utilization/stall count. |
| -rx | Chooses receive STORMs and displays their order to analyze utilization/stall count |
| -nl | Prints STORM utilization in new lines instead of a running line |
| -vert | Prints STORM utilization as a table instead of a line |
| -nc | Strips all colors |
| -help | Shows this text |

Syntax This utility analyzes the Marvell FastLinQ Controllers' performance.

The `profile` utility has the following modes:

| Mode | Description |
|------|-------------|
|------|-------------|

| | |
|---|--|
| 1 | Displays real time STORM CPU utilization |
|---|--|

- 2 Samples registers and saves a histogram to a file
- 3 Samples registers and saves both a histogram and a full CSV dump to files
- 4 Displays the real time STORM stall count
- 5 Displays PCIe core statistics (Marvell FastLinQ QL45000 Series Controllers)

Mode 1

```
profile [-forever | -i <iterations>] [-f <fileName>] [-freq
<frequency>] [-d <dev0> <dev1>] [-ah | -bb] [-storms <storm
letters> | -tx | -rx]
```

Mode 2

```
profile -h <filename> [-forever | -i <iterations>] [-s <samples>]
[-d <dev0> <dev1>] [-ah | -bb | -e5] [-ext <external filename>]
[-dec] [-sbucket]
```

Mode 3

```
profile -v <filename> [-forever | -i <iterations>] [-s <samples>]
[-d <dev0> <dev1>] [-ah | -bb | -e5] [-ext <external filename>]
[-dec] [-sbucket]
```

Mode 4

```
profile -sc [-u] [-forever | -i <iterations>] [-f <fileName>]
[-freq <frequency>] [-k] [-d <dev0> <dev1>] [-ah | -bb] [-storms
<storm letters> | -tx | -rx]
```

Mode 5

```
profile -ps [-forever | -i <iterations>] [-f <fileName>] [-freq
<frequency>] [-k] [-d <dev0> <dev1>]
```

Example `profile -h histOut.txt -i 5`

A Setting NPAR Mode

This appendix describes how to set NPAR mode.

To set NPAR mode:

1. Configure MF (NPAR) MAC addresses by issuing the following command:

```
use ./load.sh -m -nicp_mac
```

This command configures the MAC address and WWN MAC addresses of all 16 functions. Following is a sample output:

```
1: [FUNC] MAC address           : 00:10:18:22:22:00
93: [FUNC] FCoE node wwn mac addr : 00:10:18:22:22:00
94: [FUNC] FCoE port wwn mac addr : 00:10:18:22:22:00
```

2. Issue the `nvm cfg <board type>` command to configure the default values of the following options:

```
95: [FUNC] Bandwidth weight      : 3
96: [FUNC] Max Bandwidth         : 100
100: [GLOB] FCoE wwn node prefix : 10:00
101: [GLOB] FCoE wwn port prefix : 20:00
```

3. Set the function personality for each function (`nvm cfg 92`), and then set the vendor device ID to the value for the corresponding personality (`nvm cfg 77`):

```
#define DEVICE_ID_57980          0x1634
#define DEVICE_ID_57980_FCOE     0x1638
#define DEVICE_ID_57980_ISCSI    0x1639
```

4. Change MF mode (`nvm cfg option 9`) to 3 for NPAR1.0.
5. Reboot the system.

After the reboot, you should see 16 devices.

B macaddr.txt File Format

The `macaddr.txt` file programs the MAC addresses of all Marvell FastLinQ Controllers during OEM manufacturing. This file contains a range of MAC addresses that are automatically updated after the MAC address is programmed and is invoked from the `qeDiag` command line as follows:

```
./load.sh -fmac macaddr.txt
```

After the MAC addresses have been programmed, the `macaddr.txt` file will be updated such that the `mac_addr_start` value is incremented by the number of allocated MAC addresses, which is 16 (for 16 PFs), plus the number of LLDP addresses (as the number of ports), plus the VFs addresses, plus the BMC addresses (if enabled).

The following parameters comprise the `macaddr.txt` file:

| <u>Parameter</u> | <u>Description</u> |
|-----------------------------|---|
| <code>mac_addr_pref</code> | Specifies the first 3 bytes of the MAC address. Values are in hexadecimal without the <code>0x</code> prefix. |
| <code>mac_addr_start</code> | Specifies the start of the address range. The first value forms the last 3 bytes of the primary MAC address. Values are in hexadecimal without the <code>0x</code> prefix. |
| <code>mac_addr_end</code> | Specifies the end of the address range. Values are in hexadecimal without the <code>0x</code> prefix. |

Following is a sample `macaddr.txt` file:

```
mac_addr_pref = 001018
mac_addr_start = 042240
mac_addr_end = 0422FF
```

C Revision History

| Document Revision History | |
|---|---|
| Rev 01, March 20, 2015 | |
| Rev 02, October 9, 2015 | |
| Rev A, November 21, 2016 | |
| Rev B, October 27, 2017 | |
| Rev C, October 5, 2018 | |
| Rev D, October 12, 2018 | |
| Rev E, November 20, 2018 | |
| Rev F, November 15, 2019 | |
| Changes | Sections Affected |
| Updated the instructions for getting technical support for Marvell products. | “Technical Support” on page xi |
| Updated the instructions for downloading updates and documentation. | “Downloading Updates and Documentation” on page xii |
| Updated the instructions for accessing the knowledgebase. | “Knowledgebase” on page xii |
| Removed Red Hat 6 from the supported OS list. | “Linux” on page 3 |
| Corrected Step 1 to enable DON on the target adapter. | “Network Mode (DON)” on page 9 |
| In the command to enable virtual NVRAM mode and the corresponding example, changed chip parameter to <code>-chip_type</code> . | “Virtual NVRAM Mode” on page 12 |
| Added support for per PF mode. | “Per PF Mode” on page 14 |
| In the Examples section, <code>LINK SPEED</code> parameter, changed the 40G/50G/100G values to apply only to 45000 Series Adapters. | “linkdump” on page 29 |

| | |
|---|---|
| <p>Added <code>mcp set_trace_filter</code> command.</p> <p>Added <code>mcp restore_trace_filter</code> command.</p> <p>In the Syntax section:</p> <p>In the second paragraph, removed the sentence “The <code>-continuously</code>, <code>-nonstop</code>, and <code>-stop</code> parameters are available only in network (DON) mode.”</p> <p>After the paragraph about the <code>mcp enable_don</code> command, added that “The <code>-nonstop</code> and <code>-stop</code> parameters are available only in DON mode.:</p> <p>Added information about the <code>mcp trace -nonstop</code>, <code>mcp set_trace_filter</code> and <code>mcp restore_trace_filter</code> commands.</p> <p>Added an Example section.</p> <p>Added <code>nvm cfg -all <option>=></code> command.</p> <p>Added <code>nvm cfg -func <func#> <op>=<data></code> command.</p> <p>Added <code>nvm cfg -func <func#> <op>=</code> command.</p> <p>Added <code>nvm trace_modules</code> commands.</p> | <p>“mcp” on page 32</p> <p>“nvm cfg” on page 42, “nvm cfg -all <option>=>” on page 43</p> <p>“nvm cfg” on page 42, “nvm cfg -func <func#><op>=<data>” on page 43</p> <p>“nvm cfg” on page 42, “nvm cfg -func <func#><op>=” on page 43</p> <p>“nvm trace_modules” on page 149</p> |
|---|---|



Marvell Technology Group
<http://www.marvell.com>

Marvell. Moving Forward Faster