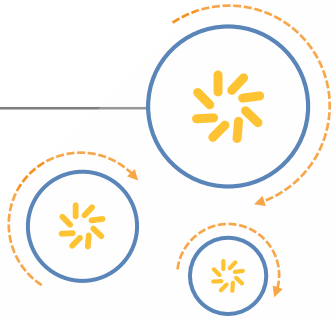
Qualcomm Atheros, Inc.



IPQ4018/IPQ4028/IPQ4019/IPQ4029 Home Switch Software Development Kit

User Guide

80-Y9571-7 Rev. B

November 18, 2015

For additional information or to submit technical questions go to https://createpoint.qti.qualcomm.com/

Qualcomm Atheros, Inc.
1700 Technology Drive
San Jose, CA 95110
U.S.A.

Revision history

Revision	Date	Description
A	June 2015	Initial release
B	November 2015	Update 2.2.1 and 5.2.1.

Contents

1 Overview 6

1.1 Modularity and Hierarchy 6
1.2 Flexibility 6
1.3 Hardware abstraction 6
1.4 Portability 6
1.5 Robustness and Reliability 7

2 Architecture 8

2.1 Basic software architecture 8
2.2 Basic software feature 9
2.2.1 Port Control 9
2.2.2 VLAN 10
2.2.3 Port VLAN 10
2.2.4 FDB 11
2.2.5 ACL 11
2.2.6 QoS 12
2.2.7 IGMP/MLD 12
2.2.8 Leaky 13
2.2.9 Mirror 13
2.2.10 Rate 13
2.2.11 STP 13
2.2.12 MIB 13
2.2.13 LED 13
2.2.14 Misc 14
2.2.15 CosMap 14
2.2.16 IP 15
2.2.17 NAT 15
2.2.18 Trunk 16
2.2.19 Sec 16
2.2.20 Initialization 16

3 Building 17

3.1 Directory structure 17
3.2 Building from the Source 18
3.2.1 Options 18
3.2.2 Build Target 19

4 Porting 21

4.1 Initialization 21
4.2 Register access 21

5 Shell22

5.1 Shell basics 22
5.2 Detailed commands 22
5.2.1 Port control 22
5.2.2 VLAN 24
5.2.3 Port VLAN 25
5.2.4 FDB 26

5.2.5 ACL 27
5.2.6 QoS 27
5.2.7 IGMP/MLD 28
5.2.8 Leaky 30
5.2.9 Mirror 30
5.2.10 Rate 30
5.2.11 STP 31
5.2.12 MIB 31
5.2.13 LED 31
5.2.14 CosMap 31
5.2.15 Misc 32
5.2.16 IP 33
5.2.17 NAT 35
5.2.18 Trunk 36
5.2.19 Security 36
5.2.20 Register Access and Debug 37
5.2.21 Set Device ID 38

1 Overview

The Switch Software Development Kit (SSDK) is a set of drivers to manage Qualcomm® switch. It can be the foundation for customer's application to control the behaviors of the switch or as a reference to build customer's own low-level drivers.

The goals of the design are listed below.

1.1 Modularity and Hierarchy

The SSDK is based on a multi-layer architecture in which every layer has its own targets. This architecture ensures the SSDK can scale from small low-end system to future large multi-CPU and distributed system. Customers can select whether including every layer into their own applications or not by simply changing build options to meet their own requirements.

Meanwhile, the SSDK can be partitioned into separate function modules, thus they can only attach required functions into their systems to archive small-footprint implementations for cost sensitive systems.

1.2 Flexibility

For running on some UNIX-like Operation System in which operation of the CPU is divided into two distinct modes – Kernel mode and User mode, main modules of the SSDK can be executed in the kernel space or the user space by changing the related option on building. In this way, the SSDK can meet different requirements of customers' various systems.

1.3 Hardware abstraction

The SSDK abstracts all hardware details of Qualcomm Atheros switch line by providing consistent APIs to customers, who do not have to understand the implementation and register details and can easily use APIs to build their own applications.

1.4 Portability

The SSDK is built upon a System Abstraction Layer (SAL), which abstracts difference between various OSs and CPU architectures and is easily to be ported to a new OS or CPU system. This can be achieved by adhering OS and Board Support Package (BSP) functions to some elaborated SAL APIs.

1.5 Robustness and Reliability

The SSDK ensures its production-quality from design phase to release phase. In addition, the SSDK handles possible error states and return a well-defined error code to reduce instability.

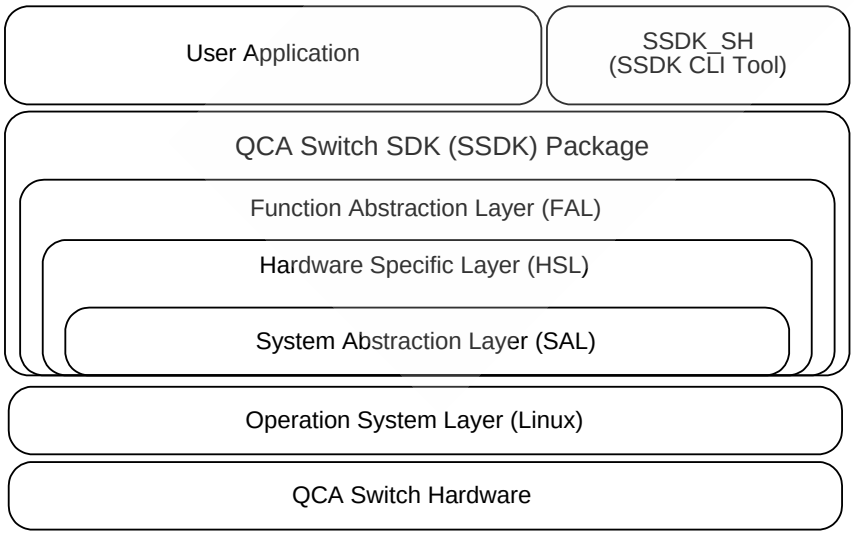
2 Architecture

This chapter gives an overview of the SSDK software architecture and supporting features.

2.1 Basic software architecture

The SSDK can be divided into three layers: FAL, HSL and SAL. Function Abstraction Layer (FAL) abstracts the implement details of Qualcomm Atheros switch chips by providing consistent Function APIs (FAPIs) to customers; Hardware Specific Layer (HSL) provides Hardware APIs (HAPIs) to implement functions of specific chip; System Abstraction Layer (SAL) that consists of OS abstraction APIs (OSAPI) and System Driver APIs (SDAPI) are used to abstract the OS and CPU system. In addition, the SSDK includes a CLI-like simple shell to manage the switch and to provide a reference of the usage of the SSDK to customers.

The basic software architecture can be illustrated in the below figure.



FAL is designed for providing unified APIs for Qualcomm Atheros switch line to user applications such as CLI, web and other protocols, for example, IGMP snooping, DHCP relay etc. Customers can build their applications through the well-defined APIs without the deep understanding of internal implementation and register details. Furthermore, because FAL abstracts the differences among Qualcomm Atheros switch products, when porting a system from one chip to another one, customers can build their applications with only slightly changes.

HSL as what implied by its name, is targeted for providing a set of APIs for specific Qualcomm Atheros switch chip, including for example, Isis the name of the switch core for QCA8337. HSL knows the internal structure of the switch and modifies the related registers directly by invoking SDAPIs in the SAL layer. User applications can invoke APIs provide by HSL directly as well. As the result, the SSDK can serve with impressive small foot-print by simply choosing related build options.

SAL consisting of OS abstraction APIs (OSAPI) and System Driver APIs (SDAPI) exists for portability where OSAPIs provide basic APIs such as memory, timer etc. to abstract differences among common OSs as Linux, xBSD and vxWorks, and SDAPIs abstract the details of specific hardware platform by providing MDIO and future possible PCI functions for register access. By this mean, Efforts to porting to a new OS and hardware platform, which can be extremely relieved, are limited to only this layer, although at this time, the SSDK only provides supports for Linux.

Furthermore, the SSDK includes a simple CLI-like SHELL on Linux, which can be referenced as an example for customers' applications, to expedite experience of Qualcomm Atheros switch.

2.2 Basic software feature

In most cases, APIs of HSL can be a subset of those of FAL, so we only describe features of FAL layer here. Usually APIs in the same feature set are collected to a file with a meaningful suffix as fal_acl, fal_fdb, fal_igmp, fal_leaky, fal_mib, fal_mirror, fal_port_ctrl, fal_portvlan, fal_vlan, fal_qos, fal_rate, fal_stp, fal_misc, fal_cosmap, fal_ip, fal_nat, fal_trunk and fal_sec.

2.2.1 Port Control

Port control provides APIs for relate port control operations. It supports the following functions:

- Set and get duplex/speed mode on one port
- Set and get ability/status of auto negotiation on one port
- Set and get Atheros header status on one port
- Set and get flow control status on one port
- Set and get power saving status on one port
- Set and get hibernate status on one port
- Set and get congestion drop status on one port
- Run cable diagnostic test on one port
- Set and get 802.3az status on one port
- Set and get medium crossover type on one port
- Check current medium crossover status on one port
- Set and get port prefer medium type on one combo port
- Get current active medium type on one combo port
- Set and get fiber mode on one combo port
- Set and get port local loopback status on one port
- Set and get port remote loopback status on one port
- Set reset status on one port
- Set power off on one port
- Set power on on one port
- Set magic frame MAC address on one port

- Get PHY ID on one port
- Set and get port wake on LAN status
- Set and get interface mode on one combo port
- Check current interface mode status on one combo port

2.2.2 VLAN

VLAN provides APIs for operation VLAN entry. It supports the following functions:

- Write VLAN entry in switch chip
- Create delete and update VLAN entry in switch chip
- Next and find VLAN entry in switch chip
- Flush all VLAN entries in switch chip
- Add, deleted and update VLAN port member
- Set and get FID VLAN binding
- Set and get VLAN based source address auto learning status

2.2.3 Port VLAN

Port VLAN provides APIs for port-based VLAN feature and QinQ feature based on VLAN translation entry. It supports the following functions:

- Set and get 802.1q VLAN mode on one port
- Set and get egress VLAN mode on one port
- Add, delete, update and get port VLAN member on one port
- Set and get default VLAN id on one port
- Set and get force port default VLAN id status on one port
- Set and get force port-based VLAN status on one port
- Set and get nest VLAN status on one port
- Set and get TPID for nest VLAN on switch chip
- Set and get ingress VLAN mode on one port
- Set and get TLS status on one port
- Set and get priority propagation status on one port
- Set and get default s-vid on one port
- Set and get default c-vid on one port
- Set and get default VRF lookup ID on one port (available in ESS of IPQ4018/IPQ4019/IPQ4028/IPQ4029 only)
- Set and get VLAN propagation status on one port
- Add, delete and get a VLAN translation entry on one port
- Set and get switch QinQ work mode on switch chip

- Set and get QinQ role on one port
- Set and get dot1q work mode on switch chip
- Iterate all VLAN translation entries on one port

The key points for user application to implement QinQ based on VLAN translation entry feature are below:

- Set switch chip QinQ work mode, s-tag mode or c-tag mode.
- Set switch port QinQ work role, core port or edge port.
- Add one or some VLAN translation entries on one port.

For more details, refer to the QinQ application for the QCA switch family.

2.2.4 FDB

FDB is also called Address resolution lookup (ARL) Table or MAC Table by other vendors, and provides APIs for maintaining forwarding data base in switch chip. It supports the following functions:

- Add and delete FDB entry. Two more parameters are for the load balance feature in FDB entry for IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS.
- First, next and find FDB entry
- Set and get FDB dynamic learning status on one port
- Set and get FDB aging status on switch chip
- Set and get FDB aging timer on switch chip
- Iterate all FDB entries on switch chip
- Next FDB entries in extend mode
- First FDB entries in extend mode
- Set and get source address learning limit on one port
- Set and get forwarding command for learning limit exceed on one port
- Set and get source address learning limit
- Set and get forwarding command for learning limit exceed
- Add, delete, find and iterate reserved FDB entries

2.2.5 ACL

ACL provides APIs for define policy and related actions for specific flow. It supports the following functions:

- Enable and disable ACL engine
- Create and destroy ACL list
- Bind and unbind ACL list on one port
- Add, delete and query ACL rules in a ACL list

The sequence for user applications to implement ACL should as below:

- Create an ACL list
- Add ACL rules to the created ACL list
- Bind the list to a port. The ACL will take effect only when it is bind to port(s)

Since the programming for the ACL engine is complex, contact the local QCA home switch support for further information.

2.2.6 QoS

QoS provides APIs for QoS feature. It supports the following functions:

- Set and get the scheduling mode on switch chip
- Set and get port/queue transmission buffer status/number on one port
- Set and get dot1p tag to queue mapping entry on switch chip
- Set and get DSCP to queue mapping entry on switch chip
- Set and get QoS parameters for precedence of queue mapping modes on one port
- Set and get default dot1p tag on one port
- Set and get port receiving buffer number on one port
- Set and get the scheduling mode on one port
- Set and get port default stag priority
- Set and get port default ctag priority
- Set and get queue based QoS remark

2.2.7 IGMP/MLD

IGMP provides APIs for IGMP/MLD packets identification and hardware join/leave. It supports the following functions:

- Set and get for IGMP/MLD identification status on one port.
- Set and get IGMP/MLD packet forwarding method when enabling IGMP/MLD identification on switch chip.
- Set and get IGMP/MLD hardware join/leave status on one port.
- Set and get router port(s) for IGMP/MLD hardware join/leave on switch chip.
- Set and get the status for creating/deleting a multicast entry in FDB for hardware join/leave on switch chip.
- Set and get the static status of multicast entry which learned by hardware
- Set and get the leaky status of multicast entry which learned by hardware
- Set and get igmpv3/mldv2 packets hardware acknowledgement status on switch chip
- Set and get the queue status of multicast entry which learned by hardware
- Set and get multicast entry learn limit
- Set and get multicast entry learn limit exceed command

2.2.8 Leaky

Leaky provides APIs for leaky function. It supports the following functions:

- Set and get unicast/multicast packets leaky mode (port-based control or FDB-based control)
- Set and get unicast/multicast/ARP packets leaky status on one port

2.2.9 Mirror

Mirror provides APIs for mirror feature. It supports the following functions:

- Set and get analyzer port in switch chip for mirror
- Set and get ingress/egress mirror status for one port

2.2.10 Rate

Rate provides APIs for rate feature. It supports the following functions:

- Set and get port ingress/port egress/queue egress rate limit status on one port
- Set and get packets type of storm control on one port
- Set and get the rate of storm control on one port
- Set and get the rate of storm control on one port
- Set and get port based policer
- Set and get ACL based policer
- Set and get port based shaper
- Set and get queue based shaper

2.2.11 STP

STP provides API for STP feature. It supports the following function:

- Set and get spanning tree state on one port

2.2.12 MIB

MIB provides APIs for getting MIB information from switch chip. It supports the following functions:

- Get MIB information on one port
- Set and get MIB status

2.2.13 LED

On some Qualcomm switch serials, the LED blinking mode can be controlled by setting control pattern. Usually the LEDs are divided into different groups such as WAN groups and LAN groups for each of which has one or several LEDs depending on by switch chips. Different LEDs in different groups may have different control pattern such as always ON, always OFF, always blinking and blinking depending on port speed, port duplex, etc.

The SSDK provides API to control led blinking pattern for each group. It supports the following function:

- Set and get led control pattern on one led group.

2.2.14 Misc

Misc provides APIs for some miscellaneous features. It supports the following functions:

- Set and get ARP status on switch chip
- Set and get forwarding command for unknown source address packets on one port
- Set and get supported max frame size on one chip
- Set and get destination ports for unknown unicast/multicast packets on switch chip
- Set and get CPU port status on switch chip
- Set and get status of PPPoE packets on switch chip
- Set and get status of DHCP packets on one port
- Set and get status of broadcast packets forwarding to CPU on switch chip
- Set and get ARP packets forwarding command on switch chip
- Set and get EAPOL (802.1x) packets forwarding command on switch chip
- Add, delete and get a PPPoE session entry on switch chip
- Set and get EAPOL (802.1x) packets hardware acknowledgement status on one port
- Set and get rip v1 packets hardware acknowledgement status on one port
- Set and get port based ARP request packets hardware acknowledgement status on one port
- Set and get port based ARP ACK packets hardware acknowledgement status on one port
- Add, delete and get PPPoE entries in extend mode

The PPPoE session entry offer capability to forward IP multicast packets encapsulated in PPPoE format to multi switch ports by switch hardware.

2.2.15 CosMap

CoSMap provides APIs for cos mapping features. It supports the following functions:

- Set and get DSCP to priority mapping
- Set and get DSCP to drop precedence mapping
- Set and get Dot1p to priority mapping
- Set and get Dot1P to drop precedence mapping (available in IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS only)
- Set and get DSCP to priority mapping for WAN/CPU port (available in IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS only)
- Set and get DSCP to drop precedence mapping for WAN/CPU port (available in IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS only)

- Set and get Dot1p to priority mapping for WAN/CPU port (available in IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS only)
- Set and get Dot1P to drop precedence mapping for WAN/CPU port (available in IPQ4018/IPQ4019/IPQ4028/IPQ4029 ESS only)
- Set and get priority to queue mapping
- Set and get priority to enhanced queue mapping

2.2.16 IP

IP provides APIs for IP features. It supports the following functions:

- Add, delete, get and next host entry
- Bind counter and PPPoE session to host entry
- Set and get port based ARP learn flag
- Set and get ARP learn mode
- Set and get IP/ARP source guard mode
- Set and get routing status
- Add, delete and next interface entry
- Set and get IP/ARP source unknown forwarding command
- Set and get IPv6 base address
- Set and get IPv4 host route
- Set and get IPv6 host route
- Set and get IPv4 default route
- Set and get IPv6 default route
- Set and delete IPv4 load balance
- Set and delete IPv6 load balance
- Set and get default flow command

2.2.17 NAT

NAT provides APIs for NAT/NAPT features. It supports the following functions:

- Add, delete, get and next NAT entry
- Bind counter to NAT entry
- Add, delete, get and next NAPT entry
- Bind counter to NAPT entry
- Set and get NAT status
- Set and get NAPT status
- Set and get NAT hash flag
- Set and get NAPT hash mode

- Set and get NAT private base address
- Set and get NAT private base address mapping mode
- Add, delete and next public address
- Set and get unknown NAT session forwarding command
- Set and delete flow cookie
- Set and delete flow load balance

2.2.18 Trunk

Trunk provides APIs for trunk features. It supports the following functions:

- Set and get trunk group port member information
- Set and get trunk hash mode

2.2.19 Sec

Sec provides APIs for Security features. It supports the following function:

- Set and get security check items information

2.2.20 Initialization

Initialization provides API for user applications to initialize the SSDK.

3 Building

This chapter gives an overview of directory architecture of SSDK and demonstrates how to build it in QSDK.

3.1 Directory structure

[/]	This is the root directory of the SSDK.
[/app]	This directory contains source code files of nathelper.
[/make]	This directory contains target files which define the components of building.
[/src]	All source code files of the SSDK are kept under this directory.
[/src/api]	This directory contains source code files for API access interface.
[/src/fal]	This directory contains source code files for FAL. FAL can provide unified interfaces and wrap the difference among switch chips. Customers can access these interfaces and needn't care internal details.
[/src/fal_uk]	This directory contains an example for user applications that invoke the SSDK built in kernel mode. It can be built with user applications together.
[/src/hsl]	This directory contains source code files for HSL that provides interfaces for a specific chip.
[/src/hsl/athena]	This directory contains source code files for Athena the switch core name for AR8216.
[/src/hsl/garuda]	This directory contains source code files for Garuda the switch core name for AR8316.
[/src/hsl/phy]	This directory contains source code files for PHY driver.
[/src/hsl/shiva]	This directory contains source code files for Shiva the switch core name for AR8227/AR8228/AR8229.
[/src/hsl/horus]	This directory contains source code files for Horus the switch core name for AR8236.
[/src/hsl/isis]	This directory contains source code files for Isis the switch core name for AR8327/AR8328/AR8325.
[/src/hsl/isisc]	This directory contains source code files for ISISC the switch core name for AR8337.
[/src/hsl/dess]	This directory contains source code files for dess the switch core name for ESS module of IPQ4018/IPQ4019/IPQ4028/IPQ4029 platform.
[/src/shell]	This directory contains source code files all source code files for the SHELL.
[/src/init]	This directory contains source code files for the initialization of the SSDK.
[/src/ref]	This directory contains source code files for reference code.
[/src/sal]	This directory contains source code files for SAL.
[/src/sal/os]	This directory contains source code files for OSAPI.
[/src/sal/sd]	This directory contains source code files for SDAPI.
[/src/util]	This directory contains source code files for utility functions.

[/include]	All header files for the SSDK are kept under this directory.
[/include/api]	This directory contains header files for API access interface declarations.
[/include/common]	This directory contains public header files for the SSDK.
[/include/fal]	This directory contains header files of FAL.
[/include/hsl]	This directory contains header files of HSL.
[/include/hsl/athena]	This directory contains header files for Athena the switch core name for AR8216.
[/include/hsl/garuda]	This directory contains header files for Garuda the switch core name for AR8316.
[/include/hsl/shiva]	This directory contains header files for Shiva the switch core name for AR8227/AR8228/AR8229.
[/include/hsl/horus]	This directory contains header files for Horus the switch core name for AR8236.
[/include/hsl/isis]	This directory contains header files for Horus the switch core name for AR8327/AR8328/AR8325.
[/include/hsl/isisc]	This directory contains header files for Horus the switch core name for AR8337.
[/include/hsl/dess]	This directory contains header files for Horus the switch core name for ESS module of IPQ4018/IPQ4019/IPQ4028/IPQ4029 platform.
[/include/init]	This directory contains header files for the initialization of the SSDK.
[/include/ref]	This directory contains header files for reference code declarations.
[/include/sal]	This directory contains header files for SAL.
[/include/sal/os]	This directory contains header files for OSAPI.
[/include/sal/sd]	This directory contains header files for SDAPI.
[/include/shell]	This directory contains header files for SHELL.

3.2 Building from the Source

SSDK can be compiled into two files:

- qca-ssdk.ko: an independent kernel module included all SSDK features.
- ssdk_sh: an user program provided a cli-like simple shell.

3.2.1 Options

Use the following options to satisfy particular requirements and change these options by editing the file config which locates in the root directory:

Switch chip type:

CHIP_TYPE = ALL_CHIP

FAL included in the SSDK:

FAL = TRUE

API_LOCK all APIs locker with locker or not. If you want all APIs with locker, define **API_LOCK=TRUE**. Otherwise, define

API_LOCK=FALSE.

Features included in SSDK:

IN_ACL=TRUE

IN_FDB=TRUE

IN_IGMP=TRUE

IN_LEAKY=TRUE

IN_LED=TRUE

IN_MIB=TRUE

IN_MIRROR=TRUE

IN_MISC=TRUE

IN_PORTCONTROL=TRUE

IN_PORTVLAN=TRUE

IN_QOS=TRUE

IN_RATE=TRUE

IN_STP=TRUE

IN_VLAN=TRUE

IN_REDUCED_ACL=FALSE

IN_COSMAP= FALSE

IN_IP= FALSE

IN_NAT= FALSE

IN_TRUNK= FALSE

IN_SEC= FALSE

IN_NAT_HELPER=FALSE

IN_INTERFACECONTROL=TRUE

IN_MACBLOCK=FALSE

For example, set value of IN_ACL to FALSE if you don't need the ACL feature, then the target of building will exclude the ACL related APIs.

3.2.2 Build Target

- Select SSDK in *make menuconfig*

  make menuconfig

  Use the following path to select qca-ssdk module:

  Kernel modules → Network Devices → Kmod-qca-ssdk

  Then save, exit.
- Building SSDK

make package/qca-ssdk/install V=s

- Clean SSDK

  make package/qca-ssdk/clean

4 Porting

4.1 Initialization

SSDK initialization occurs when qca-ssdk.ko is inserted into kernel, and the initialization process is independent of the other modules.

In QSDK, an auto-load ID is specified to determine the loading time of qca-ssdk.ko.

4.2 Register access

In QSDK, Linux kernel provides the standard MII bus interface. If switch module connects to CPU through MDIO bus, SSDK uses the interface to access switch registers by a specified bus name.

For example:

```
snprintf(busid, MII_BUS_ID_SIZE, "%s.%d", "mdio-gpio", 0);
miidev = bus_find_device_by_name(&platform_bus_type, NULL, busid);
miibus = dev_get_drvdata(miidev);
miibus->write(bus, phy_addr, QCA_MII_MMD_ADDR, addr);
```

For IPQ4018/IPQ4019/IPQ4028/IPQ4029 SOC, switch module is accessed through AHB bus and SSDK maps physical address and use local bus access mode to access switch registers.

```
For example:
ioremap_nocache(ssdk_dt_global.switchreg_base_addr,
                  ssdk_dt_global.switchreg_size);
readl(hw_addr + reg_addr)
writel(reg_val, hw_addr + reg_addr);
```

5 Shell

5.1 Shell basics

The SSDK includes a CLI-like switch shell to configure the switch on Linux. Customer can invoke the shell built per chapter 3 by executing the file ssdk_sh.

The SSDK provides some useful help mechanisms to facilitate the usage of the Shell. To get help specific to a command mode, a command, a keyword, or an argument, use one of following commands:

- Entering a question mark (?) at the Shell prompt allows you to obtain a list of commands available for each command mode.
- Entering "*abbreviated-command*?" the shell shows a list of commands that begin with a particular character string. (No space between command and question mark.)
- Entering "*Command ?*" to get the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)
- Entering "*abbreviated-command<Tab>*", the shell helps you to Completes a partial command name or lists all commands partially matched Help messages of the SSDK use following conventions:
- Required command arguments are inside angle brackets (< >).
- Optional command arguments are in square brackets ([ ]).
- Alternative keywords are separated by vertical bars (|).
- The minimum and the maximum of a value range are separated by horizontal line (-).

To quit the Shell, you can enter "***q***" or "***quit***" at the Shell prompt.

5.2 Detailed commands

5.2.1 Port control

Name	Function	Usage
port duplex get	get duplex mode of a port	***port duplex get <port_id>***
port duplex set	set duplex mode of a port	***port duplex set <port_id> <half | full>***
port speed get	get speed mode of a port	***port speed get <port_id>***
port speed set	set speed mode of a port	***port speed set <port_id> <10 | 100 | 1000>***
port autoAdv get	get auto-negotiation advertisement of a port	***port autoAdv get <port_id>***

Name	Function	Usage
port autoAdv set	set auto-negotiation advertisement of a port	***port autoAdv set <port_id> <cap_bitmap>***
port autoNeg get	get auto-negotiation status of a port	***port autoNeg get <port_id>***
port autoNeg enable	enable auto-negotiation of a port	***port autoNeg enable <port_id>***
port autoNeg restart	restart auto-negotiation process of a port	***port autoNeg restart <port_id>***
port header set	set Atheros header/tag status of a port	***port header set <port_id> <enable | disable>***
port header get	get Atheros header/tag status of a port	***port header get <port_id>***
port txhdr set	set Atheros header/tag status of a port	***port txhdr set <port_id> <noheader|onlymanagement|allframe>***
port txhdr get	get Atheros header/tag status of a port	***port txhdr get <port_id>***
port rxhdr set	set Atheros header/tag status of a port	***port rxhdr set <port_id> <noheader|onlymanagement|allframe>***
port rxhdr get	get Atheros header/tag status of a port	***port rxhdr get <port_id>***
port flowCtrl set	set flow control status of a port	***port flowCtrl set <port_id> <enable | disable>***
port flowCtrl get	get flow control status of a port	***port flowCtrl get <port_id>***
port powersave set	set power saving status of a port	***port powersave set <port_id> <enable | disable>***
port powersave get	get power saving status of a port	***port powersave get <port_id>***
port hibernate set	set hibernate status of a port	***port hibernate set <port_id> <enable | disable>***
port hibernate get	get hibernate status of a port	***port hibernate set <port_id>***
port cdt run	run cable diagnostic test of a port	***port cdt run <port_id> <mdi_pair>***
port congedrop set	set congestion drop status	***port congedrop set <port_id> <enable | disable>***
port congedrop get	get congestion drop status	***port congedrop set <port_id>***
port ieee8023az set	set 802.3 az status of a port	***port ieee8023az set <port_id> <enable|disable>***
port ieee8023az get	get 802.3 az status of a port	***port ieee8023az get <port_id>***
port crossover set	set crossover of a port	***port crossover set <port_id> <auto|mdi|mdix>***
port crossover get	get crossover of a port	***port crossover get <port_id>***
port crossover status check	show current crossover status of a port	***port crossover status <port_id>***
port prefer medium set	set prefer medium of a combo port	***port preferMedium set <port_id> <copper|fiber>***
port prefer medium get	get prefer medium of a combo port	***port preferMedium get <port_id>***
Port current medium get	get port current medium type	***port mediumType get <port_id>***
port fiber mode set	set port fiber mode of a combo port	***port fiberMode set <port_id> <100fx|1000bx>***
port fiber mode get	get port fiber mode of a combo port	***port fiberMode get <port_id>***
port local loopback set	set port local loopback of a port	***port localLoopback set <port_id> <enable|disable>***
port local loopback get	get port local loopback of a port	***port localLoopback get <port_id>***

Name	Function	Usage
port remote loopback set	set port remote loopback of a port	***port remoteLoopback set <port_id> <enable|disable>***
port remote loopback get	get port remote loopback of a port	***port remoteLoopback get***
port reset	set port reset of a port	***port reset set <port_id>***
port power off	set port power off of a port	***port poweroff set <port_id>***
port power on	set port power on of a port	***port poweron set <port_id>***
port magic frame address set	set port magic frame address of a port	***port magicFrameMac set <port_id> <mac_address>***
port magic frame address get	get port magic frame address of a port	***port magicFrameMac get <port_id>***
port phy id get	get port PHY ID of a port	***port phyId get <port_id>***
port wol set	set port WOL status of a port	***port wolstatus set <port_id> <enable|disable>***
port wol get	get port WOL status of a port	***port wolstatus get <port_id>***
port interface mode set	set port interface mode of a combo port	***port interfaceMode set <port_id> <psgmii_baset|psgmii_bx1000| psgmii_fx100|psgmii_amdet|sgmii_baset>***
port interface mode get	get port interface mode of a combo port	***port interfaceMode get <port_id>***
port interface mode status get	get port current interface mode of a combo port	***port interfaceMode status <port_id>***

5.2.2 VLAN

Name	Function	Usage
vlan entry create	create a VLAN entry	***vlan entry create <vlan_id>***
vlan entry del	delete a VLAN entry	***vlan entry del <vlan_id>***
vlan entry update	update port member of a VLAN entry	***vlan entry update <vlan_id> <member_bitmap> <0>***
vlan entry find	find a VLAN entry by VLAN ID	***vlan entry find <vlan_id>***
vlan entry next	find next VLAN entry by VLAN ID	***vlan entry next <vlan_id>***
vlan entry append	append a VLAN entry	***vlan entry append***
vlan entry show	show whole VLAN entries	***vlan entry show***
vlan entry flush	flush all VLAN entries	***vlan entry flush***
vlan fid set	set VLAN entry FID	***vlan fid set <vlan_id> <fid>***
vlan fid get	get VLAN entry FID	***vlan fid get <vlan_id>***
vlan member add	add VLAN entry member	***vlan member add <vlan_id> <port_id> <unmodified|untagged|tagged>***
vlan member del	delete VLAN entry member	***vlan member del <vlan_id> <port_id>***
vlan learnsts set	set VLAN entry learn status	***vlan learnsts set <vlan_id> <enable|disable>***
vlan learnsts get	get VLAN entry learn status	***vlan learnsts get <vlan_id>***

5.2.3 Port VLAN

Name	Function	Usage
portVlan ingress get	get ingress VLAN mode of a port	***portVlan ingress get <port_id>***
portVlan ingress set	set ingress VLAN mode of a port	***portVlan ingress set <port_id> <disable | secure | check | fallback>***
portVlan egress get	get egress VLAN mode of a port	***portVlan egress get <port_id>***
portVlan egress set	set egress VLAN mode of a port	***portVlan egress set <port_id> <unmodified | untagged | tagged>***
portVlan member add	add a member to the port based VLAN of a port	***portVlan member add <port_id> <memport_id>***
portVlan member del	delete a member from the port based VLAN of a port	***portVlan member del <port_id> <memport_id>***
portVlan member update	update members of the port based VLAN of a port	***portVlan member update <port_id> <port_bitmap>***
portVlan member get	get members of the port based VLAN of a port	***portVlan member get <port_id>***
portVlan defaultVid get	get default VLAN ID of a port	***portVlan defaultVid get <port_id>***
portVlan defaultVid set	set default VLAN ID of a port	***portVlan defaultVid set <port_id> <vid>***
portVlan forceVid set	set VLAN ID enforcement status of a port	***portVlan forceVid set <port_id> <enable | disable>***
portVlan forceVid get	get VLAN ID enforcement status of a port	***portVlan forceVid get <port_id>***
portVlan forceMode set	set port based VLAN enforcement status of a port	***portVlan forceMode set <port_id> <enable | disable>***
portVlan forceMode get	get port based VLAN enforcement status of a port	***portVlan forceMode get <port_id>***
portVlan nestVlan set	set nest VLAN status of a port	***portVlan nestVlan set <port_id> <enable | disable>***
portVlan nestVlan get	get nest VLAN status of a port	***portVlan nestVlan get <port_id>***
portVlan sVlanTPID set	set service VLAN TPID	***portVlan sVlanTPID set <tpid>***
portVlan sVlanTPID get	get service VLAN TPID	***portVlan sVlanTPID get***
portVlan invlan set	set port invlan mode	***portVlan invlan set <port_id> <admit_all|admit_tagged|admit_untagged>***
portVlan invlan get	get port invlan mode	***portVlan invlan get <port_id>***
portVlan tlsMode set	set TLS mode	***portVlan tlsMode set <port_id> <enable|disable>***
portVlan tlsMode get	get TLS mode	***portVlan tlsMode get <port_id>***
portVlan priPropagation set	set priority propagation	***portVlan priPropagation set <port_id> <enable|disable>***
portVlan priPropagation get	get priority propagation	***portVlan priPropagation get <port_id>***
portVlan defaultSVid set	set default SVID	***portVlan defaultSVid set <port_id> <vlan_id>***
portVlan defaultSVid get	get default SVID	***portVlan defaultSVid get <port_id>***

Name	Function	Usage
portVlan defaultCVid set	set default CVID	***portVlan defaultCVid set <port_id> <vlan_id>***
portVlan defaultCVid get	get default CVID	***portVlan defaultCVid get <port_id>***
portVlan vlanPropagation set	set VLAN propagation	***portVlan vlanPropagation set <port_id> <disable|clone|replace>***
portVlan vlanPropagation get	get VLAN propagation	***portVlan vlanPropagation get <port_id>***
portVlan translation add	add VLAN translation	***portVlan translation add <port_id>***
portVlan translation del	del VLAN translation	***portVlan translation del <port_id>***
portVlan translation get	get VLAN translation	***portVlan translation get <port_id>***
portVlan qinqMode set	set qinq mode	***portVlan qinqMode set <ctag|stag>***
portVlan qinqMode get	get qinq mode	***portVlan qinqMode get***
portVlan qinqRole set	set qinq role	***portVlan qinqRole set <port_id> <edge|core>***
portVlan qinqRole get	get qinq role	***portVlan qinqMode get <port_id>***
portVlan translation iterate	iterate VLAN translation tables	***portVlan translation iterate <port_id> <iterator>***
*portvlan vrf_id set	set port VRF ID	***portVlan vrf_id set <port_id> <vrf_id>***
*portvlan vrf_id set	get port VRF ID	***portVlan vrf_id get <port_id>***

1. * Available in ESS of IPQ4018/IPQ4019/IPQ4028/IPQ4029 only.

5.2.4 FDB

Name	Function	Usage
fdb entry add	add a FDB entry	***fdb entry add***
fdb entry flush	flush all FDB entries	***fdb entry flush <0:dynamic only|1:dynamic and static>***
fdb entry show	show whole FDB entries	***fdb entry show***
fdb portEntry flush	flush all FDB entries by a port	***fdb portEntry flush <port_id> <0:dynamic only|1:dynamic and static>***
fdb entry del	delete a FDB entry	***fdb entry del***
fdb firstEntry find	find the first FDB entry	***fdb firstEntry find***
fdb nextEntry find	find next FDB entry	***fdb nextEntry find***
fdb entry find	find a FDB entry	***fdb entry find***
fdb portLearn set	set FDB entry learning status of a port	***fdb portLearn set <port_id> <enable | disable>***
fdb portLearn get	get FDB entry learning status of a port	***fdb portLearn get <port_id>***
fdb ageCtrl set	set FDB entry aging status	***fdb ageCtrl set <enable | disable>***
fdb ageCtrl get	get FDB entry aging status	***fdb ageCtrl get***
fdb ageTime set	set FDB entry aging time	***fdb ageTime set <time:s>***
fdb ageTime get	get FDB entry aging time	***fdb ageTime get***
fdb entry iterate	iterate all FDB entries	***fdb entry iterate <iterator>***
fdb ptlearnlimit set	set port FDB entry learn limit	***fdb ptlearnlimit set <port_id> <enable|disable> <limitcounter>***

Name	Function	Usage
fdb ptlearnlimit get	get port FDB entry learn limit	***fdb ptlearnlimit get <port_id>***
fdb ptlearnexceedcmd set	set port forwarding command when exceed learn limit	***fdb ptlearnexceedcmd set <port_id> <forward|drop|cpycpu|rdtcpu>***
fdb ptlearnexceedcmd get	get port forwarding command when exceed learn limit	***fdb ptlearnexceedcmd get <port_id>***
fdb learnlimit set	set FDB entry learn limit	***fdb learnlimit set <enable|disable> <limitcounter>***
fdb learnlimit get	get FDB entry learn limit	***fdb ptlearnlimit get***
fdb learnexceedcmd set	set forwarding command when exceed learn limit	***fdb ptlearnexceedcmd set <forward|drop|cpycpu|rdtcpu>***
fdb learnexceedcmd get	get FDB entry learn limit	***fdb learnexceedcmd get***
fdb learnexceedcmd get	get FDB entry learn limit	***fdb learnexceedcmd get***
fdb resventry add	add a reserve FDB entry	***fdb resventry add***
fdb resventry del	delete a reserve FDB entry	***fdb resventry del***
fdb resventry find	find a reserve FDB entry	***fdb resventry find***
fdb resventry iterate	iterate all reserve FDB entries	***fdb resventry iterate <iterator>***
fdb resventry show	show whole reserve FDB entries	***fdb resventry show***
fdb ptLearnstatic set	set FDB entry learning static status of a port	***fdb ptLearnstatic set <port_id> <enable|disable>***
fdb ptLearnstatic get	get FDB entry learning static status of a port	***fdb ptLearnstatic get <port_id>***

5.2.5 ACL

Name	Function	Usage
acl list create	create an ACL list	***acl list create <list_id> <priority>***
acl list destroy	destroy an ACL list	***acl list destroy <list_id>***
acl list bind	bind an ACL list to a port	***acl list bind <list_id> <0-0:direction> <0-0:objtype> <objindex>***
acl list unbind	unbind an ACL list from a port	***acl list unbind <list_id> <0-0:direction> <0-0:objtype> <objindex>***
acl rule add	add ACL rules to an ACL list	***acl rule add <list_id> <rule_id> <rule_nr>***
acl rule delete	delete ACL rules from an ACL list	***acl rule delete <list_id> <rule_id> <rule_nr>***
acl rule query	query an ACL rule	***acl rule query <list_id> <rule_id>***
acl status set	set status of ACL engine	***acl status set <enable | disable>***
acl status get	get status of ACL engine	***acl status get***
acl udfprofile set	set port UDF profile	***acl udfprofile set <port_id> <l2/l2snap/l3/l3plus/l4> <offset> <length>***
acl udfprofile get	get port UDF profile	***acl udfprofile get <port_id> <l2/l2snap/l3/l3plus/l4>***

5.2.6 QoS

Name	Function	Usage
qos qTxBufSts set	set queue Tx buffer counting status of a port	***qos qTxBufSts set <port_id> <enable | disable>***
qos qTxBufSts get	get QoS queue Tx buffer counting status of a port	***qos qTxBufSts get <port_id>***
qos qTxBufNr set	set queue Tx buffer number	***qos qTxBuf set <port_id> <queueid:0-3> <number>***
qos qTxBufNr get	get queue Tx buffer number	***qos qTxBuf get <port_id> <queueid:0-3>***
qos ptTxBufSts set	set port Tx buffer counting status of a port	***qos ptTxBufSts set <port_id> <enable | disable>***
qos ptTxBufSts get	set port Tx buffer counting status of a port	***qos ptTxBufSts get <port_id>***
qos ptTxBufNr set	set port Tx buffer number	***qos ptTxBufNr set <port_id> <number>***
qos ptTxBufNr get	get port Tx buffer number	***qos ptTxBufNr get <port_id>***
qos ptMode set	set QoS mode of a port	***qos ptMode set <port_id> <da | up | dscp | flow> <enable | disable>***
qos ptMode get	get QoS mode of a port	***qos ptMode get <port_id> <da | up | dscp | flow>***
qos ptModePri set	set the priority of QoS modes of a port	***qos ptModePri set <port_id> <da | up | dscp | flow> <priority:0-3>***
qos ptModePri get	get the priority of QoS modes of a port	***qos ptModePri get <port_id> <da | up | dscp | flow>***
qos ptschMode set	set port traffic scheduling mode	***qos ptschMode set <port_id> <sp|wrr|mix|mixplus> <q0,q1,q2,q3>***
qos ptschMode get	get port traffic scheduling mode	***qos ptschMode get <port_id> <sp|wrr|mix|mixplus> <q0,q1,q2,q3>***
qos ptRxBufNr set	set port Rx buffer number	***qos ptRxBufNr set <port_id> <number:0-60>***
qos ptRxBufNr get	get port Rx buffer number	***qos ptRxBufNr get <port_id>***
qos ptDefaultSpri set	set default stag priority for received frames of a port	***qos ptDefaultSpri set <port_id> <spri:0-7>***
qos ptDefaultSpri get	get default stag priority for received frames of a port	***qos ptDefaultSpri get <port_id>***
qos ptDefaultCpri set	set default ctag priority for received frames of a port	***qos ptDefaultCpri set <port_id> <spri:0-7>***
qos ptDefaultCpri get	get default ctag priority for received frames of a port	***qos ptDefaultCpri get <port_id>***
qos ptQuRemark set	set egress queue based remark	***qos ptQuRemark set <port_id> <queue_id> <table_id> <enable|disable>***
qos ptQuRemark get	get egress queue based remark	***qos ptQuRemark get <port_id> <queue_id>***

5.2.7 IGMP/MLD

Name	Function	Usage
igmp mode set	set IGMP/MLD snooping status of a port	***igmp mode set <port_id> <enable | disable>***
igmp mode get	get port IGMP/MLD snooping status	***igmp mode get <port_id>***
igmp cmd set	set IGMP/MLD frames forwarding command	***igmp cmd set <forward | drop | cpycpu | rdtcpu>***

Name	Function	Usage
igmp cmd get	get IGMP/MLD frames forwarding command	***igmp cmd get***
igmp portJoin set	set IGMP/MLD hardware joining status	***igmp portJoin set <port_id> <enable | disable>***
igmp portJoin get	get IGMP/MLD hardware joining status	***igmp portJoin get <port_id>***
igmp portLeave set	set IGMP/MLD hardware leaving status	***igmp portLeave set <port_id> <enable | disable>***
igmp portLeave get	get IGMP/MLD hardware leaving status	***igmp portLeave get <port_id>***
igmp rp set	set IGMP/MLD router ports	***igmp rp set port_bit_map***
igmp rp get	get IGMP/MLD router ports	***igmp rp get***
igmp createStatus set	set IGMP/MLD ability for creating entry	***igmp createStatus set <enable | disable>***
igmp createStatus get	get IGMP/MLD ability for creating entry	***igmp createStatus get***
igmp static set	set IGMP/MLD static status for creating entry	***igmp static set <enable|disable>***
igmp static get	get IGMP/MLD static status for creating entry	***igmp static get***
igmp leaky set	set IGMP/MLD leaky status for creating entry	***igmp leaky set <enable|disable>***
igmp leaky get	get IGMP/MLD leaky status for creating entry	***igmp leaky get***
igmp version3 set	set IGMP v3/MLD v2 status for creating entry	***igmp version3 set <enable|disable>***
igmp version3 get	get IGMP v3/MLD v2 status for creating entry	***igmp version3 get***
igmp queue set	set IGMP/MLD queue status for creating entry	***igmp queue set <enable|disable>***
igmp queue get	get IGMP/MLD queue status for creating entry	***igmp queue get***
igmp ptlearnlimit set	set port Multicast entry learn limit	***igmp ptlearnlimit set <port_id> <enable|disable> <limitcounter>***
igmp ptlearnlimit get	get port Multicast entry learn limit	***igmp ptlearnlimit get <port_id>***
igmp ptlearnexceedcmd set	set port forwarding command when exceed multicast learn limit	***igmp ptlearnexceedcmd set <port_id> <forward|drop|cpycpu|rdtcpu>***
igmp ptlearnexceedcmd get	get port forwarding command when exceed multicast learn limit	***igmp ptlearnexceedcmd get <port_id>***

5.2.8 Leaky

Name	Function	Usage
leaky ucMode set	set unicast packets leaky mode	***leaky ucMode set <port | fdb>***
leaky ucMode get	get unicast packets leaky mode	***leaky ucMode get***
leaky mcMode set	set multicast packets leaky mode	***leaky mcMode set <port | fdb>***
leaky mcMode get	get multicast packets leaky mode	***leaky mcMode get***
leaky arp set	set ARP packets leaky mode	***leaky arp set <port_id> < enable | disable>***
leaky arp get	get ARP packets leaky mode	***leaky arp get <port_id>***
leaky ptUcMode set	set unicast packets leaky status of a port	***leaky ptUcMode set <port_id> <enable | disable>***
leaky ptUcMode get	get unicast packets leaky status of a port	***leaky ptUcMode get <port_id>***
leaky ptMcMode set	set multicast packets leaky status of a port	***leaky ptMcMode set <port_id> <enable | disable>***
leaky ptMcMode get	get multicast packets leaky status of a port	***leaky ptMcMode get <port_id>***

5.2.9 Mirror

Name	Function	Usage
mirror analyPt set	set mirror analysis port	***mirror analyPt set <port_id>***
mirror analyPt get	get mirror analysis port	***mirror analyPt get***
mirror ptIngress set	set ingress mirror status of a port	***mirror ptIngress set <port_id> <enable | disable>***
mirror ptIngress get	get ingress mirror status of a port	***mirror ptIngress get <port_id>***
mirror ptEgress set	set egress mirror status of a port	***mirror ptEgress set <port_id> <enable | disable>***
mirror ptEgress get	get egress mirror status of a port	***mirror ptEgress get <port_id>***

5.2.10 Rate

Name	Function	Usage
rate portpolicer set	set port policer	***rate portpolicer set <port_id>***
rate portpolicer get	get storm control rate	***rate portpolicer get <port_id>***
rate portshaper set	set port egress shaper	***rate portshaper set <port_id> <enable|disable>***
rate portshaper get	get port egress shaper	***rate portshaper get <port_id>***
rate queueshaper set	set queue egress shaper	***rate queueshaper set <port_id> <queue_id> <enable|disable>***
rate queueshaper get	get queue egress shaper	***rate queueshaper get <port_id> <queue_id>***
rate aclpolicer set	set ACL policer	***rate aclpolicer set <policer_id>***

Name	Function	Usage
rate aclpolicer get	get ACL policer	***rate aclpolicer get <policer_id>***

5.2.11 STP

Name	Function	Usage
stp ptState set	set STP state of a port	***stp ptState set st_id <port_id> <disable | block | listen | learn | forward>***
stp portState get	get STP state of a port	***stp ptState get st_id <port_id>***

5.2.12 MIB

Name	Function	Usage
mib statistics get	get statistics information of a port	***mib statistics get <port_id>***
mib status set	get MIB status	***mib status set <enable | disable>***
mib status get	get MIB status	***mib status set***

5.2.13 LED

Name	Function	Usage
led ctrlpattern set	set LED control pattern	***led ctrlpattern set <group_id> <led_id>***
led ctrlpattern get	get LED control pattern	***led ctrlpattern get <group_id> <led_id>***

5.2.14 CosMap

Name	Function	Usage
cosmap dscp2pri set	set DSCP to priority map table	***cosmap dscp2pri set <dscp> <priority>***
cosmap dscp2pri get	get DSCP to priority map table	***cosmap dscp2pri get <dscp>***
cosmap dscp2dp set	set DSCP to DP map table	***cosmap dscp2dp set <dscp> <dp>***
cosmap dscp2dp get	get DSCP to DP map table	***cosmap dscp2dp get <dscp>***
cosmap up2pri set	set dot1p to priority map table	***cosmap up2pri set <up> <priority>***
cosmap up2pri get	set dot1p to priority map table	***cosmap up2pri get <up>***
cosmap up2dp set	set dot1p to DP map table	***cosmap up2dp set <up> <dp>***
cosmap up2dp get	set dot1p to DP map table	***cosmap up2dp get <up>***
*cosmap dscp2ehpri set	set DSCP to priority map table for 0, 5, 6 ports	***cosmap dscp2ehpri set <dscp> <priority>***
*cosmap dscp2ehpri get	get DSCP to priority map table for 0, 5, 6 ports	***cosmap dscp2ehpri get <dscp>***
*cosmap dscp2ehdp set	set DSCP to DP map table for 0, 5, 6 ports	***cosmap dscp2ehdp set <dscp> <dp>***
*cosmap dscp2ehdp get	get DSCP to DP map table for 0, 5, 6 ports	***cosmap dscp2ehdp get <dscp>***

Name	Function	Usage
*cosmap up2ehpri set	set dot1p to priority map table for 0, 5, 6 ports	***cosmap up2ehpri set <up> <priority>***
*cosmap up2ehpri get	set dot1p to priority map table for 0, 5, 6 ports	***cosmap up2ehpri get <up>***
*cosmap up2ehdp set	set dot1p to DP map table for 0, 5, 6 ports	***cosmap up2ehdp set <up> <dp>***
*cosmap up2ehdp get	set dot1p to DP map table for 0, 5, 6 ports	***cosmap up2ehdp get <up>***
cosmap pri2q set	set priority to queue mapping	***cosmap pri2q set <priority> <queueid>***
cosmap pri2q get	get priority to queue mapping	***cosmap pri2q get <priority>***
cosmap pri2ehq set	set priority to enhanced queue mapping	***cosmap pri2ehq set <priority> <queueid>***
cosmap pri2ehq get	get priority to enhanced queue mapping	***cosmap pri2ehq get <priority>***
cosmap egRemark set	set egress remark table	***cosmap egRemark set <tableid>***
cosmap egRemark get	get egress remark table	***cosmap egRemark get <tableid>***

1. * Available in ESS of IPQ4018/IPQ4019/IPQ4028/IPQ4029 only.

5.2.15 Misc

Name	Function	Usage
misc frameMaxSize set	set the maximal received frame size of the device	***misc frameMaxSize set <size:byte>***
misc frameMaxSize get	get the maximal received frame size of the device	***misc frameMaxSize get***
misc ptUnkUcFilter set	set flooding status of unknown unicast frames	***misc ptUnkUcFilter set <port_id> <enable | disable>***
misc ptUnkUcFilter get	get flooding status of unknown unicast frames	***misc ptUnkUcFilter get <port_id>***
misc ptUnkMcFilter set	set flooding status of unknown multicast frames	***misc ptUnkMcFilter set <port_id> <enable | disable>***
misc ptUnkMcFilter get	get flooding status of unknown multicast frames	***misc ptUnkMcFilter get <port_id>***
misc cpuPort set	set CPU port status	***misc cpuPort set <enable | disable>***
misc cpuPort get	get CPU port status	***misc cpuPort get***
misc PppoeCmd set	set PPPoE frames forwarding command	***misc PppoeCmd set <forward | rdtcpu>***
misc PppoeCmd get	get PPPoE frames forwarding command	***misc PppoeCmd get***
misc Pppoe set	set PPPoE frames hardware identification status	***misc Pppoe set <enable | disable>***
misc Pppoe get	get PPPoE frames hardware identification status	***misc Pppoe get***
misc ptDhcp set	set DHCP frames hardware identification status	***misc ptDhcp set <port_id> <enable | disable>***
misc ptDhcp get	get DHCP frames hardware identification status	***misc ptDhcp get <port_id>***

Name	Function	Usage
misc arpcmd set	set ARP packets forwarding command	***misc arpcmd set <forward| cpycpu|rdtcpu>***
misc arpcmd get	get ARP packets forwarding command	***misc arpcmd get***
misc eapolcmd set	set EAPOL packets forwarding command	***misc eapolcmd set < cpycpu|rdtcpu>***
misc eapolcmd get	set EAPOL packets forwarding command	***misc eapolcmd get***
misc eapolstatus set	set EAPOL frames hardware identification status	***misc eapolstatus set <port_id> <enable|disable>***
misc eapolstatus get	get EAPOL frames hardware identification status	***misc eapolstatus get <port_id>***
misc rip set	set RIP packets hardware identification status	***misc rip set <enable|disable>***
misc rip get	get RIP packets hardware identification status	***misc rip get***
misc ptarpreq set	set ARP request packets hardware identification status	***misc ptarpreq set <port_id> <enable|disable>***
misc ptarpreq get	get ARP request packets hardware identification status	***misc ptarpreq get <port_id>***
misc ptarpack set	set ARP ACK packets hardware identification status	***misc ptarpack set <port_id> <enable|disable>***
misc ptarpack get	get ARP ACK packets hardware identification status	***misc ptarpack get <port_id>***
misc extendpppoe add	add a PPPoE session entry	***misc extendpppoe add***
misc extendpppoe del	delete a PPPoE session entry	***misc extendpppoe del***
misc extendpppoe get	get a PPPoE session entry	***misc extendpppoe get***
misc glomacaddr set	set the global MAC address	***misc glomacaddr set***
misc glomacaddr get	get the global MAC address	***misc glomacaddr get***
misc lldp set	set LLDP frames hardware identification status	***misc lldp set***
misc lldp get	get LLDP frames hardware identification status	***misc lldp get***
misc framecrc set	set frame CRC reserve status	***misc framecrc set***
misc framecrc get	get frame CRC reserve status	***misc framecrc get***

5.2.16 IP

Name	Function	Usage
ip hostentry add	add host entry	***ip hostentry add***
ip hostentry del	delete host entry	***ip hostentry del <del_mode>***
ip hostentry get	get host entry	***ip hostentry get <get_mode>***
ip hostentry next	next host entry	***ip hostentry next <next_mode>***
ip hostentry show	show whole host entries	***ip hostentry show***

Name	Function	Usage
ip hostentry bindcnt	bind counter to host entry	***ip hostentry bindcnt <host entry id> <cnt id> <enable|disable>***
ip hostentry bindpppoe	bind PPPoE to host entry	***ip hostentry bindpppoe <host entry id> <pppoe id> <enable|disable>***
ip ptarplearn set	set port ARP learn flag, bit[0] requests bit[1] acknowledgement	***ip ptarplearn set <port_id> <flag>***
ip ptarplearn get	get port ARP learn flag, bit[0] requests bit[1] acknowledgement	***ip ptarplearn get <port_id>***
ip arplearn set	set ARP learn mode	***ip arplearn set <learnlocal|learnall>***
ip arplearn get	get ARP learn mode	***ip arplearn get <port_id>***
ip ptipsrcguard set	set IP source guard mode	***ip ptipsrcguard set <port_id> <mac_ip|mac_ip_port|mac_ip_vlan|mac_ip_port_vlan|no_guard>***
ip ptipsrcguard get	get IP source guard mode	***ip ptipsrcguard get <port_id>***
ip ptarpsrcguard set	set ARP source guard mode	***ip ptarpsrcguard set <port_id> <mac_ip|mac_ip_port|mac_ip_vlan|mac_ip_port_vlan|no_guard>***
ip ptarpsrcguard get	get ARP source guard mode	***ip ptarpsrcguard get <port_id>***
ip routestatus set	set IP route status	***ip routestatus set <enable|disable>***
ip routestatus get	get IP route status	***ip routestatus get***
ip intfentry add	add interface MAC address	***ip intfentry add <enable|disable>***
ip intfentry del	delete interface MAC address	***ip intfentry del***
ip intfentry add	add interface MAC address	***ip intfentry add <enable|disable>***
ip intfentry show	IP INTF entry show	***ip intfentry show***
ip ipunksrc set	set IP unknown source command	***ip ipunksrc set <forward|drop|cpycpu|rdtcpu>***
ip ipunksrc get	get IP unknown source command	***ip ipunksrc get***
ip arpunksrc set	set ARP unknown source command	***ip arpunksrc set <forward|drop|cpycpu|rdtcpu>***
ip arpunksrc get	get ARP unknown source command	***ip arpunksrc get***
ip ip6baseaddr set	set IP6 base address	***ip ip6baseaddr set <forward|drop|cpycpu|rdtcpu>***
ip ip6baseaddr get	get IP6 base address	***ip ip6baseaddr get***
ip vrfbaseaddr set	Set IP4 base address	***ip vrfbaseaddr set <ip_addr>***
ip vrfbaseaddr get	Get IP4 base address	***ip vrfbaseaddr get***
ip defaultroute set	Set default route	***ip defaultroute set***
ip defaultroute get	Get default route	***ip defaultroute get***
ip hosttroute set	Set host route	***ip hostroute set***

Name	Function	Usage
ip hosttroute get	Get host route	***ip hostroute get***
ip rfsip4 set	Set IP4 load balance	***ip rfsip4 set***
ip rfsip4 get	Get IP4 load balance	***ip rfsip4 get***
ip rfsip6 set	Set IP6 load balance	***ip rfsip6 set***
ip rfsip6 get	Get IP6 load balance	***ip rfsip6 get***
ip defaultflowcmd set	Set default flow command	***ip defaultflowcmd set <vrf id> <lan2lan|wan2lan|lan2wan|wan2wan> <forward|drop|rdtcpu|admit_all>***
ip defaultflowcmd get	Get default flow command	***ip defaultflowcmd set <vrf id> <lan2lan|wan2lan|lan2wan|wan2wan>***
ip defaultrtflowcmd set	Set default flow command	***ip defaultrtflowcmd set <vrf id> <lan2lan|wan2lan|lan2wan|wan2wan> <forward|drop|rdtcpu|admit_all>***
ip defaultrtflowcmd get	Get default flow command	***ip defaultrtflowcmd get <vrf id> <lan2lan|wan2lan|lan2wan|wan2wan>***

5.2.17 NAT

Name	Function	Usage
nat natentry add	add NAT entry	***nat natentry add***
nat natentry del	delete NAT entry	***nat natentry del <del_mode>***
nat natentry get	get NAT entry	***nat natentry get <get_mode>***
nat natentry show	show whole NAT entries	***nat natentry show***
nat natentry bindcnt	bind counter to NAT entry	***nat natentry bindcnt <nat entry id> <cnt id> <enable|disable>***
nat naptentry add	add NAPT entry	***nat naptentry add***
nat naptentry del	delete NAPT entry	***nat naptentry del <del_mode>***
nat naptentry get	get NAPT entry	***nat naptentry get <get_mode>***
nat naptentry show	show whole NAPT entries	***nat naptentry show***
nat naptentry bindcnt	bind counter to NAPT entry	***nat naptentry bindcnt <napt entry id> <cnt id> <enable|disable>***
nat natstatus set	set NAT status	***nat natstatus set <enable|disable>***
nat natstatus get	get NAT status	***nat natstatus get***
nat naptstatus set	set NAPT status	***nat naptstatus set <enable|disable>***
nat naptstatus get	get NAPT status	***nat naptstatus get***
nat nathash set	set NAT hash mode	***nat nathash set <flag>***
nat nathash get	get NAT hash mode	***nat nathash get***
nat naptmode set	set NAPT mode	***nat naptmode set <fullcone|strictcone|portstrict|synmatric>***
nat naptmode get	get NAPT mode	***nat naptmode get***
nat prvbaseaddr set	set NAT PRV base address	***nat prvbaseaddr set <ip4 addr>***
nat prvbaseaddr get	get NAT PRV base address	***nat prvbaseaddr get***

Name	Function	Usage
nat prvaddrmode set	set NAT PRV address map mode	***nat prvaddrmode set <enable|disable>***
nat prvaddrmode get	get NAT PRV address map mode	***nat prvaddrmode get***
nat pubaddr add	add PUB address	***nat pubaddr add <enable|disable>***
nat pubaddr del	delete PUB address	***nat pubaddr del <del_mode>***
nat pubaddr show	show whole PUB address entries	***nat pubaddr show***
nat natunksess set	set NAT unknown session command	***nat natunksess set <forward|drop|cpycpu|rdtcpu>***
nat natunksess get	get NAT unknown session command	***nat natunksess get***
nat flowentry add	add flow entry	***nat flowentry add***
nat flowentry del	delete flow entry	***nat flowentry del <del_mode>***
nat flowentry get	get flow entry	***nat flowentry get <get_mode>***
nat flowentry show	show whole flow entries	***nat flowentry show***
nat flowentry bindcnt	bind counter to flow entry	***nat flowentry bindcnt <napt entry id> <cnt id> <enable|disable>***

5.2.18 Trunk

Name	Function	Usage
trunk group set	set trunk group member information	***trunk group set <trunk_id> <disable|enable> <port_bitmap>***
trunk group get	get trunk group member information	***trunk group get <trunk_id>***
trunk hashmode set	set trunk hash mode	***trunk hashmode set <hash_mode>***
trunk hashmode get	get trunk hash mode	***trunk hashmode get <trunk_id>***

5.2.19 Security

Name	Function	Usage
sec mac set	set MAC part security filter	***sec mac set <resv_vid/invalid_src_addr> <value>***
sec mac get	get MAC part security filter	***sec mac get <resv_vid/invalid_src_addr>***
sec ip set	set IP related security filter	***sec ip set <invalid_ver/same_addr/ttl_change_status/ttl_val> <value>***
sec ip get	get IP related security filter	***sec ip get <invalid_ver/same_addr/ttl_change_status/ttl_val>***
sec ip4 set	set IPv4 related security filter	***sec ip4 set <invalid_hl/hdr_opts/invalid_df/frag_offset_min_len/frag_offset_min_size/frag_offset_max_len/invalid_frag_offset/invalid_sip/invalid_dip/invalid_chksum/invalid_pl/df_clear_status/ipid_random_status> <value>***

Name	Function	Usage
sec ip4 get	get IPv4 related security filter	***sec ip4 get <invalid_hl/hdr_opts/invalid_df/frag_offset_min_len/frag_offset_min_size/frag_offset_max_len/invalid_frag_offset/invalid_sip/invalid_dip/invalid_chksum/invalid_pl/df_clear_status/ipid_random_status>***
sec ip6 set	set IPv6 related security filter	***sec ip6 set <invalid_dip/invalid_sip/invalid_pl> <value>***
sec ip6 get	get IPv6 related security filter	***sec ip6 get <invalid_dip/invalid_sip/invalid_pl>***
sec tcp set	set TCP related security filter	***sec tcp set <blat/invalid_hl/min_hdr_size/invalid_syn/su_block/sp_block/sap_block/xmas_scan/null_scan/sr_block/sf_block/sar_block/rst_scan/rst_with_data/fa_block/pa_block/ua_block/invalid_chksum/invalid_urgptr/invalid_opts> <value>***
sec tcp get	get TCP related security filter	***sec tcp get <blat/invalid_hl/min_hdr_size/invalid_syn/su_block/sp_block/sap_block/xmas_scan/null_scan/sr_block/sf_block/sar_block/rst_scan/rst_with_data/fa_block/pa_block/ua_block/invalid_chksum/invalid_urgptr/invalid_opts>***
sec udp set	set UDP related security filter	***sec udp set <blat/invalid_len/invalid_chksum> <value>***
sec udp get	get UDP related security filter	***sec udp get <blat/invalid_len/invalid_chksum>***
sec icmp4 set	set ICMP4 related security filter	***sec icmp4 set <ping_pl_exceed/ping_frag/ping_max_pl> <value>***
sec icmp4 get	get ICMP4 related security filter	***sec icmp4 get <ping_pl_exceed/ping_frag/ping_max_pl>***
sec icmp6 set	set ICMP6 related security filter	***sec icmp6 set <ping_pl_exceed/ping_frag/ping_max_pl> <value>***
sec icmp6 get	get ICMP6 related security filter	***sec icmp6 get <ping_pl_exceed/ping_frag/ping_max_pl>***

5.2.20 Register Access and Debug

Name	Function	Usage
debug phy get	read PHY register	***debug phy get <ph_id> <reg_addr>***
debug phy set	write PHY register	***debug phy set <ph_id> <reg_addr> <value>***
debug reg get	read switch register	***debug reg get <reg_addr>***
debug reg set	write switch register	***debug reg set <reg_addr> <value>***
debug entry get	read switch register entry	***debug entry get <entry_name>***
debug entry set	write switch register entry	***debug entry set <entry_name>***
debug field get	read switch register field	***debug field get <field_name>***
debug field set	write switch register field	***debug field set <field_name>***
debug aclList dump	dump all ACL lists	***debug aclList dump***
debug aclRule dump	dump all ACL rules	***debug aclRule dump***

5.2.21 Set Device ID

Name	Function	Usage
device id set	set device ID	***device id set <dev_id>***