Software Development Kit
Release Notes
SDK 6.5.3

March, 2016

Broadcom
Network Switching

BROADCOM

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.5.3.

This document provides a general description of the release and its new features. It also describes the chips supported by the release, BCM API additions or changes, resolved issues, and any relevant open issues. The reader should refer to prior release notes for 6.5.x, as only new features or issues are described in this version of the release notes.

Section 2: Product Documentation

The following documents are available through Broadcom's Customer Support Portal, http://support.broadcom.com. They are the primary source of information and should be referenced when using this release:

Table 1: Product Documentation

Document	*Description*
56XX-PG653-R	BCM API Reference Guide. This manual describes the theory of operations of the API and all existing BCM APIs for this release.
56XX-PG707-R	Stacking Software Guide This guide describes how to use the discovery and stacking applications provided in this release.
56XX-PG819-R	Platform Guide This guide describes the SDK source and Makefile structure, abstraction and porting layers, device specific interactions, and the platform/operating system specific features of the SDK. If this is your first time working with the SDK, start with this document.

Section 3: New Devices and Systems

For any given SDK release, support for certain devices may be provided in Preview or Supported status. Devices in Table 2: Supported Switch Devices and Table 4: Supported PHYs have completed the full QA process and are intended for use in production systems. It is expected that customers would integrate the version of the SDK which provides "Supported" status for their use on actual development or production systems.

Devices in Table 3: Preview Switch Devices and Table 5: Preview PHYs are provided to allow early integration of the customer's application with the SDK APIs that support that device. This software has not been fully tested on the physical target device and should not be expected to fully function.

Please reference 56XX-RN649-R, 56XX-RN6410-R, and 56XX-RN6411-R for additional details about supported devices and PHYs in this release.

Table 2: Supported Switch Devices

Family	*Devices*	*Description*
BCM56760	BCM56760 A0	6x40GbE+ 48x10GbE Ethernet Embedded Switch
	BCM56765 A0	6x40GbE+ 48x10GbE Ethernet Embedded Switch
BCM56260	BCM56233 B0	4xGE UNI + 2xGE NNI + 2.5GE to FPGA
BCM56960	BCM56960 B1	32x100 GbE/64x40GbE/128x10 GbE Multilayer Switch
	BCM56930 B1	130-Port RoHS-compliant, high-capacity, 3.2Tbps Switch device with 10/25/40/100G support.
	BCM56961 B1	64x40GbE port configuration
	BCM56962 B1	24x 100GbE/48x 40GbE Multilayer Switch
	BCM56968 B1	32x HG[106]/64x HG[42] Multilayer Switch
BCM53400	BCM53365 A0	16-port 1GbE/2.5GbE/10GbE Multilayer Ethernet Switch
BCM56260	BCM56460 A0/B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks
	BCM56461 A0/B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without iHost support
	BCM56463 A0/B0	34 Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch
	BCM56465 A0/B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without external packet buffer
	BCM56466 A0/B0	24x 1GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 4x 10GbE/HGs[11] Uplinks and Stacking
	BCM56468 A0/B0	34 Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch without external packet buffer
	BCM56260 A0/B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56262 A0/B0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56263 A0/B0	12x 1GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 4x 1GbE Uplinks, Stacking, additional external packet buffer, and Integrated CPU
	BCM56265 A0/B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer
	BCM56267 A0/B0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer
	BCM56268 A0/B0	16 Gbps Integrated Lower Power Carrier Ethernet Access without external packet buffer
BCM56860	BCM56832 A1	27 x 20GbE
	BCM56833 A1	32 x 40GbE/104 x 10GbE
	BCM56836 A1	8 x 100GbE/32 x 10GbE/32 x 40GbE/104 x 10GbE

Table 2: Supported Switch Devices

Family	Devices	Description
	BCM56868 A1	8 x HG127/32 x HG42
BCM53134	BCM53134 B1	4x1GbE (internal PHY), 2 Phyless interface

Table 3: Preview Switch Devices

Family	Devices	Description
BCM56270	BCM56270 A0	4x 2.5GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 8x 10GbE Uplinks, and Stacking
	BCM56271 A0	4x 2.5GbE Port Integrated Low-Power Carrier Ethernet Access Switch 4x 10GbE + 4x 2.5GbE Uplinks, and Stacking
	BCM56272 A0	4x 1GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 2x 10GbE Uplinks, and Stacking.
	BCM53460 A0	4x 2.5GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 8x 10GbE Uplinks, and Stacking.
	BCM53461 A0	4x 1GbE Port Integrated Low-Power Carrier Ethernet Access Switch with 4x 10GbE + 4x 2.5GbE Uplinks, and Stacking.
BCM56160	BCM56160 B0	8x2.5GbE + 16x1GbE + 4x10GbE + 2x20GbE
	BCM56162 B0	24x1GbE + 4x1GbE, 24GbE + 2x1GbE + 2x 13GbE
	BCM56163 B0	8x1GbE + 2x10GbE
	BCM56164 B0	24x1GbE + 4x1GbE
	BCM56166 B0	8x2.5GbE + 16x1GbE + 4x10GbE + 2x20GbE
BCM53440	BCM53440 B0	8x2.5GbE + 16x1GbE + 4x10GbE + 2x20GbE
	BCM53442 B0	24x1GbE + 4x1GbE
	BCM53443 B0	8x1GbE + 2x10GbE
BCM56565	BCM56565 A0	24x10GbE + 24x1GbE+ 4x10GbE+ 2xHG[106] Ethernet Embedded Switch
	BCM56566 A0	12x MGig + 36x 1 / 2.5GE + 8x10G (uplink) + 2xHG[106]
	BCM56567 A0	48x 1 / 2.5 GE + 8x10G (uplink) + 2xHG[106]
BCM56560	BCM56560 A0	2xCAUI + 4x40GbE + 12x10GbE Ethernet Embedded Switch
BCM56760	BCM56762 A0	24x10GE + 6x40GE with support for 25GE

Table 4: Supported PHYs

Device	Driver Family	Description	Switch qualified against
BCM82864	82864	40GbE PMA MUX/DEMUX with 100GbE Bypass	BCM56960
BCM82790 A0	82790	100GbE/OTN CAUI-to-10GbE VSR/CR4/SR4/LR4/KR4 Gearbox PHY	BCM56860
BCM82792 A0	82790	Dual 100G/OTN CAUI-to-100GbE VSR/CR4/SR4/LR4 Gearbox PHY	BCM56860
BCM82796 A0	82790	Dual 100G/Quad 40G VSR/CR4/SR4/LR4 Gearbox PHY	BCM56860
BCM84328	84328	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY	BCM56860

Table 4: Supported PHYs

Device	Driver Family	Description	Switch qualified against
BCM82328	84328	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY	BCM56860
BCM82780F	82780	10G Octal SFI-XFI PHY with 10GbE/40GbE	BCM56860
BCM82752	82752	10G Dual SFI-XFI PHY with 10GbE IEEE 1588 Support	BCM56860
BCM82758	82758	10G Quad SFI-XFI PHY with 10GbE/40GbE and IEEE 1588 Support	BCM56860
BCM54380	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver with MACSEC Half Duplex support	BCM56548,BCM56630

Table 5: Preview PHYs

Device	Driver Family	Description	Switch qualified against
BCM82332	82332	1x 100G/3x 40G/12x 10G Ethernet PHY	BCM56860
BCM82793	82332	100G Gearbox for 100GBASE CR4/SR4/LR4 & SR10	BCM56860

Section 4: New Features per Device

BCM56760/56765 A0 BETA SUPPORT

The BCM56760 family is a class of high performance, nonblocking network switching devices supporting up to 720Gbps to address performance, capacity and service requirements of next generation data center and cloud computing environment. They feature multiple integrated TSC Serdes core with support for 100GE delivering high-bandwidth, glueless network connectivity.

This release supports following port configurations: For 56760 A0

- 48x10GbE + 6x40GbE (ToR configuration)
- 72x10GbE
- 4x50GbE + 48x10GbE + 1x10GbE
- 2x100GbE + 4x40GbE+ 8x25GbE + 12x10GbE + 2x10GbE

For 56765 A0

- 72x10GbE
- 2x100GbE + 48x10GbE
- 16x25GbE + 8x40GbE
- 48x10GbE + 6x40GbE
- 18x40GbE

This release provides Beta support for BCM56760 A0 and BCM56765 A0 devices. BCM56762 A0 is supported as Preview only in this release.

SDK OVERALL FEATURE STATUS FOR BCM56760

Table 6: BCM56760 A0 Status of SDK features

Features	*Status*
L2	GA
L3	GA
L2 Multicast	GA
IP multicast	GA
DMVOQ	GA
DPVOQ	GA
Legacy Field Processor	GA
L2GRE	GA
Mac-in Mac and SPB	GA
Legacy MPLS	GA
QoS	GA
Mirror	GA
Trunk	GA
Proxy	GA
Higig Proxy	GA
Packet Rx	GA
Packet Tx	GA
Linkscan	GA
Rate	GA
Resilient Hash	GA
RTAG7 flex Hash	GA
Switch Control	GA
STAT legacy & flex	GA

Table 6: BCM56760 A0 Status of SDK features

Features	*Status*
STG	GA
Tunnel	GA
TRILL	GA
Port Extender	GA
VLAN	GA
Legacy protection switching	GA
FCoE	GA
COE subport	GA
VP-VLAN membership	GA
Legacy subport	GA
NIV	GA
UFT	GA
Cosq	GA
System and network level congestion	GA
Base MMU	GA
MMU stats	GA
MMU config tool	GA
MMU Queue structure	GA
BST (buffer statistics)	GA
cut-through	GA
E2ECC	GA
I2C	GA
Port	GA
Flexport	GA
100G Support	GA
VXLAN	GA
MPLS swap to self	GA
RiOT	GA
Multiple split horizon group	GA
MPLS Segment Routing	GA
MPLS entropy & GAL	GA
EP Redirect 2.0	GA
ALPM	Beta*
SER	Beta*
Protection switching enhancement	GA
OAM	GA
Warmboot	GA
IFP & EFP enhancements	GA
Higig-DLB	GA
LED Processor	GA
Warmboot upgrade	N/A - like-to-like only verified

* ALPM is not covered by SER protection at this time. Customers requiring SER protection should use LPM mode.

BCM56565 A0 BETA SUPPORT

BCM56565 family is a purpose built device for access and aggregation deployment in the enterprise network. BCM56565 family offers 48 front panel ports with 24 ports of MGig that offer complete flexibility with support for 1/2.5/5/10G. In addition BCM56565 also offers 200Gbps of stacking bandwidth (HiGiG only) for the new enterprise fixed/stackable market. Config supported in this release:

- 24x10GbE (MGig) + 24x1GbE+ 4x10GbE (Uplink) + 2xHG[106] (stacking)
- 24x10GbE + 24x1/2.5 GbE +1x100GbE/4x25GbE (Uplink) + 4xHG[42]
- 24x10GbE + 24x1GbE + 8x10GbE (Uplink) + 4xHG[42]
- 24x10GbE + 24x1/2.5GbE+8x10GbE (Uplink) + 4xHG[42]
- 24x10GbE + 24x1/2.5GbE +1x100GbE/4x25GbE (Uplink) + 2xHG[106]
- 24x10GbE + 24x1GbE + 8x10GbE (Uplink) + 2xHG[106]
- 24x10GbE + 24x1/2.5GbE + 8x10GbE (Uplink) + 2xHG[106]
- 24x10GbE + 24x1GbE +2x100GbE (Uplink)
- 24x10GbE + 24x1/2.5GbE + 2x100GbE (Uplink)

This release provides only bringup support for BCM56566 A0 and BCM56567 A0 devices. The following config variable needs to be set for these devices: `core_clock_frequency=793.75`

SDK OVERALL FEATURE STATUS FOR BCM56565

The following list of features is supported in this release at GA quality.

Table 7: BCM56565 A0 Status of SDK features

Features	*Status*
L2	GA
L3	GA
L2 Multicast	GA
IP multicast	GA
DMVOQ	GA
DPVOQ	GA
Legacy Field Processor	GA
L2GRE	GA
Mac-in Mac and SPB	GA
MPLS	GA
QoS	GA
Mirror	GA
Trunk	GA
Proxy	GA
Higig Proxy	GA
Packet Rx	GA
Packet Tx	GA
Linkscan	GA
Policer	GA
Rate	GA
Resilient Hash	GA
RTAG7 flex Hash	GA
Switch Control	GA
STAT legacy & flex	GA
STG	GA

Table 7: BCM56565 A0 Status of SDK features

Features	*Status*
Tunnel	GA
Port Extender	GA
VLAN	GA
Legacy protection switching	GA
VP-VLAN membership	GA
Legacy subport	GA
NIV	GA
UFT	GA
Cosq	Beta
System and network level congestion	GA
Base MMU	GA
MMU stats	GA
MMU config tool	GA
MMU Queue structure	GA
BST (buffer statistics)	GA
E2ECC	GA
I2C	GA
Port	Beta
Flexport	Beta
100G Support	GA
VXLAN	GA
RiOT	GA
Service Meter enhancement	Beta
Multiple split horizon group	Beta
EP Redirect 2.0	GA
ALPM	Beta*
SER	Beta*
Hierarchical Protection switching	Beta
OAM	GA
Warmboot	GA
IFP & EFP enhancements	GA
Higig-DLB	GA
LED Processor	GA
Warmboot upgrade	N/A like-to-like only verified

* ALPM is not covered by SER protection at this time. Customers requiring SER protection should use LPM mode.

BCM56560 A0 BETA SUPPORT

The BCM56560 family is a class of high-performance, nonblocking network switching devices supporting up to 700Gbps switch ports bandwidth. Each device family features multiple integrated TSC Serdes Core, each with four integrated 100G SerDes transceivers and associated PCS for native support of SGMII (10/100/1000BT), XFI, XAUI, 10GBASE-KR/CR/SR, 40GBASE-KR4/CR4, XLAUI, CAUI4, and Broadcom's proprietary HiGig2(TM) at different speeds. The BCM56560 delivers high-bandwidth, glueless network connectivity up to 700 Gbps on a single chip. This release supports the following port configurations.

- 2xCAUI + 4x40GbE + 12x10GbE
- 2xCAUI + 48x10bGE + 2x10GbE
- 2xCAUI10 / HG[106] + 16x25GbE
- 6x40GbE + 24x10GbE + 2x10GbE
- 2xCAUI10+ 2xCAUI + 24x10GbE + 2x10GbE

SDK OVERALL FEATURE STATUS FOR BCM56560

The following list of features is supported in this release at GA quality.

Table 8: BCM56560 A0 Status of SDK features

Features	*Status*
L2	GA
L3	GA
L2 Multicast	GA
IP multicast	GA
Legacy Field Processor	GA
Mac-in Mac and SPB	GA
Legacy MPLS	GA
QoS	GA
Mirror	GA
Trunk	GA
Proxy	GA
Higig Proxy	GA
Packet Rx	GA
Packet Tx	GA
Linkscan	GA
Policer	Beta
Rate	GA
Resilient Hash	GA
RTAG7 flex Hash	GA
Switch Control	GA
STAT legacy & flex	GA
STG	GA
Tunnel	GA
Port Extender	GA
VLAN	GA
Legacy protection switching	GA
COE subport	GA
VP-VLAN membership	GA
Legacy subport	GA
NIV	GA

Table 8: BCM56560 A0 Status of SDK features

Features	*Status*
UFT	GA
Cosq	GA
Base MMU	GA
MMU stats	GA
MMU config tool	GA
MMU Queue structure	GA
BST (buffer statistics)	GA
E2ECC	GA
I2C	GA
Port	GA
Flexport	Beta
100G Support	GA
VXLAN	GA
MPLS swap to self	GA
RiOT	GA
SAT Hooks	Beta
Service Meter enhancement	Beta
Multiple split horizon group	GA
MPLS Segment Routing	GA
MPLS entropy & GAL	GA
EP Redirect 2.0	GA
ALPM	Beta*
SER	Beta*
Hierarchical protection switching	GA
OAM (Ethernet and MPLS)	GA
Warmboot	GA
IFP & EFP enhancements	GA
Higig-DLB	GA
LED-Processor	GA
Warmboot upgrade	N/A like-to-like only verified

* ALPM is not covered by SER protection at this time. Customers requiring SER protection should use LPM mode.

BCM56270 BRING-UP SUPPORT

The Broadcom BCM56270 is purpose-built for next-generation Carrier Ethernet Edge and Mobile Backhaul platforms. It combines a feature-rich packet processing engine, integrated hierarchical traffic manager, address management, and a non-blocking switch fabric into a single 28 nm CMOS device. The BCM56270 supports an integrated packet buffer. The BCM56270 is capable of handling L2 switching, L3 routing, metro VPN tunneling, and access control lists (ACLs). The BCM56270 virtualized architecture enables seamless support for VPLS, VPWS, MPLS-TP, PBB, and PBB-TE tunneling. In addition, the BCM56270 supports internal processing of Ethernet OAM, BFD, and BHH to enable a complete host CPU offload for these functions.

Config supported in this release: 8x1G/2.5G/10G + 4x1G/2.5G

SDK FEATURE STATUS

Table 9: TR Test Status

Description	*Test*	*BCM56270 Status*
Register reset defaults	tr 1	Pass
Register read/write	tr 3	Pass
PCI Compliance	tr 2	Pass
PCI S-Channel	tr 4	Pass
Memory write/read	tr 50	Pass
Memory random Addr/Data	tr 51	Pass
Memory random Addr	tr 52	Pass
Table DMA	tr 71	Pass
Counter width verification	tr 30	Pass
Counter read/write	tr 31	Pass
XGS L2 Delete by Port	tr 35	Pass
XGS L2 Delete by VLAN	tr 36	Pass
MDIO Linkscan	tr 60	Pass
EXT Loopback	tr 20	Pass
CPU Benchmark	tr 21	Pass
CPU S/G, Reload	tr 22	Pass
CPU S/G, Simple	tr 23	Pass
CPU S/G, Random	tr 24	Pass
BCM Packet Send	tr 40	Pass
BCM Packet Receive	tr 41	Pass
CPU DMA loopback	tr 17	Pass
MAC loopback	tr 18 1G, 10G	Pass
PHY loopback	tr 19 1G, 10G	Pass
MAC loopback V2	tr 48 1G, 10G	Pass
PHY loopback V2	tr 49 1G, 10G	Pass
Internal snake MAC loopback	tr 39 1G, 10G	Pass
Internal PHY loopback	tr 39 1G, 10G	Pass
Traffic test MAC loopback	tr 72 1G,10G	Pass
Traffic test PHY loopback	tr 72 1G,10G	Pass

STATUS OF BRINGUP SUPPORT

While the device and legacy feature bring-up are still in progress, the summary of the current base test status as of this release are provided below in Table 13. Where test results are noted as NA, that means that results are not complete

enough to judge overall status. Overall feature completion status is also included. The features below have been unit tested on the chip itself.

Table 10: BCM56270 Status of SDK features

Features	*Status*
L2	Preview
L3	Preview
L2 Multicast	Preview
IP multicast	Preview
DMVOQ	Preview
Legacy Field Processor	Preview
Mac-in Mac and SPB	Preview
Legacy MPLS	Preview
QoS	Preview
Mirror	Preview
Trunk	Preview
Proxy	Preview
Higig Proxy	Preview
Packet Rx	Preview
Packet Tx	Preview
Linkscan	Preview
Policer	Preview
Rate	Preview
Resilient Hash	Preview
RTAG7 flex Hash	Preview
Switch Control	Preview
STAT legacy & flex	Preview
STG	Preview
Tunnel	Preview
Port Extender	Preview
VLAN	Preview
Legacy protection switching	Preview
VP-VLAN membership	Preview
Legacy subport	Preview
NIV	Preview
UFT	Preview
Cosq	Preview
System and network level congestion	Preview
Base MMU	Preview
MMU stats	Preview
MMU Queue structure	Preview
E2ECC	Preview
I2C	Preview
Port	Preview
Flexport	Preview
LED Processor	Preview

BCM88060 A0 BETA SUPPORT

The Broadcom BCM88060 is a second generation Fibre Channel (FC) and Fibre Channel-over-Ethernet (FCoE) mapper. The device enables a flexible chip to be deployed in front of a data center switches to connect to FC endpoints directly. This device provides the flexibility to support FC/FCoE/Ethernet universal port capability as needed. The device incorporates the FC mapping into and out of Ethernet (FCoE) in a physical layer device. As a universal port device, the FCoE capability can optionally connect to a host of switches via a standard Ethernet when the FC universal port is needed. This release supports the following device firmware: BCM8806x version 1.0.3 (rc/firmware/BCM8806x). This firmware binary (BCM8806x) must be present in the SDK boot directory.

STATUS OF SUPPORTED FEATURES

Bring up of the following BCM88060 features have been completed in this SDK release. Some features have reached Beta quality as they have already entered our test regression phase. Table 15 shows the status of features in this release.

Table 11: SDK Feature Status

Feature	*Status*	*Note*
Ethernet Retimer mode - Forced	Beta	Supported speeds: 10G / 20G / 25G / 40G / 50G / 100G
Ethernet Retimer mode - Autoneg	Bringup	Supported speeds: 10G / 20G / 25G / 40G / 50G
Ethernet Gearbox mode - Autoneg/ Forced	Bringup	Supported 40G : System side 2x20G , line side 4x10G
Ethernet EBE (Extended Buffered for Ethernet mode)	Bringup	Supported all speeds: Same speed on system and line side.
Warmboot	Bringup	Tested in Linux environment with Ethernet and FCoE.
Ethernet Flexport	Bringup	Quad, Tri0, Tri1, dual, single supported.
HiGig	Bringup	Supported 42G, 106G
Boot optimization	Bringup	Tested boot and initialization timings with multiple MT2 setup.
Parallel Download with multiple MT2	Bringup	Tested with chained MT2 setup.
FCoE - Fixed mode and AN mode with TTS	Bringup	Supported speeds: 4G / 8G / 16G / 32G. With Fixed Training.
FCoE - Statistics and Events handling	Bringup	Tested inband statistics in FCoE setup.
Diagnostics	Beta	Register read/write, dsc dump, eye scan supported. Phy diag supports multistage loopback.
Traffic tests	Beta	TR19, TR72 supported.

NOTES AND KNOWN LIMITATIONS

- Tested with BCM956960 & BCM56850 SVK on PowerPC with VxWorks and Linux.
- Ethernet Repeater mode is not supported.
- Traffic might terminate on nonflex ports during some flexport configuration.
- Re-configuring FC from 32 to lower speed might require re-initialization.
- FC ports may fail to link up. To recover, power cycle the BCM988060MC board.
- HiGig - tested on BCM956960KM board only.

BCM53134 B1 SUPPORT

The BCM53134 is a highly integrated, cost-effective smart-managed Gigabit switch. The switch design is based on ROBO architecture. The switch offers 4 x 1 GbE and two phyless interface. One of the phyless interface can be configured as IMP port providing connection to the CPU. The other phyless interface can be RGMII or SGMII.

This release supports the following device configuration.

- BCM53134 : 4 x 1GbE ports, + 1 RGMII connected to CPU + 1 RGMII/SGMII

SUPPORTED FEATURES

The following table shows the supported features for this device in this release.

Table 12: BCM53134 B1 Status of SDK features

Feature	*Release with Support*
Legacy	
Cosq	GA
Field Processor	GA
L2	GA
L2 Multicast	GA
Mirror	GA
Packet Rx/TX	GA
Port	GA
Rate	GA
Switch Control	GA
STAT	GA
STG	GA
VLAN	GA

BCM56160 B0 BRINGUP SUPPORT

This release adds the Bringup support for B0 revision of the base BCM56160 and the derivative SKUs.

- B0 SKUs Supported: BCM56160, BCM56162, BCM56163, BCM56164, BCM56166, BCM53440, BCM53442, BCM53443

In addition, this release adds the support of GPHY bypass functionality in B0 feature set.

BCM88670-FAMILY GA RELEASE

This release is for the BCM88370-Family and BCM88670-Family product lines. The 6.5.3 release is the first of 6.5.X train releases where BCM88670-Family and BCM88370-Family product lines are supported. In the continued SDK support, all features introduced in SDK 6.4.11 are also supported in SDK 6.5.3. The subsequent sections describe the increment in available features compared to 6.4.11, major bug-fixes and known issues.

IMPORTANT: SW COMPATIBILITY GUIDELINES 6.4.X TO 6.5.X

- Preserving SW interface of BCM88670 device between SDK versions was a key consideration in 6.5.x device driver design. The SW interface was modified in the following cases: Improved SW design resulting in a more intuitive API, efficiency, significantly modified or extended device functionality.
- To assist the migration process from existing application configuring BCM88670 devices over 6.4.x SDK versions (6.4.10, 6.4.11), to an application configuring BCM88670 devices over 6.5.x SDK versions, a dedicate document was created: "BCM88670 BCM SDK Compatibility Guidelines 6.4.x to 6.5.x". We highly recommend going over the document. In case you don't find the document in docsafe, please approach your AE engineer to get a copy of it.
- Customers that migrate existing applications over 6.4.X releases that are prior to 6.4.10, are required to do two steps: first to go over the release notes of later 6.4.X releases up to 6.4.10 (including) and fit their application configuration accordingly and then start the migration process from 6.4.X to 6.5.X.
- Release is aligned with KBP-SDK 1.4.4 for external TCAM and KAPS.
- Scheduled warmboot is supported.
- ISSU is NOT supported from 6.4.X to 6.5.X

BCM88670-FAMILY B0 REVISION SUPPORT

Following is the list of the expected bug fixes and enhancements that require SDK support and its timeframe.

Table 13: SDK support timeline for BCM88670

Feature/Bug fix	*SDK JIRA*	*Documentation*	*Timeframe*
OAM UP-MEP LIF-profile	SDK-87294	Errata#1	6.5.4
ETPP Counter Pointer by OutLIF	SDK-76171	Errata#3	Done
Hierarchical FEC Does Not Work When the 1st FEC Hierarchy is Unprotected	SDK-75505	Errata#17	Done
OAM Classifier Can Not Count Per Priority	SDK-84890	Errata#18	Done
ROO - remarking of Native IP Header	SDK-76172	Errata#19	Done
Routing out of Mac-in-MAC Tunnel (part of ROO SPB)	SDK-76178	Errata#20	Done
IP-MC Fallback to Bridge - PCP/DEI	SDK-76174	Errata#21	Done
MPLS LSR - EXP remark	SDK-76175	Errata#22	Done
GLEM Trap is Not Reliable	SDK-75507	Errata#23	Done
Mapping TC-DP to EXP/ TOS at the ETPP is using wrong profile	SDK-88249	Errata#33	Done
Egress VLAN Editing (EVE) Used to Remove VLAN Tags Causes Outbound Mirroring to Work Improperly	SDK-76177	Errata#34	Done

Table 13: SDK support timeline for BCM88670

Feature/Bug fix	*SDK JIRA*	*Documentation*	*Timeframe*
MPLS EEDB Utilization Improvement, MPLS PUSH2SWAP	SDK-81282	Section in 88670-AG304-R document	Done
Forwarding and Link-Layer MTU per Out-LIF	SDK-76184	Section in 88670-AG304-R document	TBD
Egress VLAN Membership Check	SDK-76177	Section in 88670-AG304-R document	Done
14 MPLS Push Commands	SDK-79923	Section in next 88670-AG304-R document (not published yet)	Done
LB Key Enhancements	SDK-82302	Section in next 88670-AG304-R document (not published yet)	Done

VALIDATED FEATURES

Basic data path, connectivity and Traffic Management features:

- None

Packet Processing:

- None

NEW FEATURES

Basic data path, connectivity and Traffic Management features:

- None

Packet Processing:

- KBP - support up to 6 lookups (added 2 more ACL lookups)
- KBP - LEM scale, support both lookup LEM and KBP for IPv4 UC packets.
- Added new FEC diagnostics (diag pp fec) to print fec table and last FEC hit
- Improvaments for diagnostics "diag dbal"
- MPLS - MPLS Additional Label feature added, which enables more than 4 labels egress encapsulation
- VPWS tagged mode is supported (from PWE to AC, from PWE to PWE) through new flexible cross connect api
- Extender E-Channel Registration and Routing over Extender (for UC cases) were added
- Improved performance for IPMC/L3 traverse functions
- Utilization of descriptor DMA mechanism to enhance performance for FEC DB (in Jericho and above) and KAPS (in Q-AX) entries insertion
- Added GLEM database to shadowing
- Added support for OAMP CCM/BFD per endpoint statistics. Note: UDH with statistics is not functional.
- OAMP transmission when UDH is supported for most types of packets. For list of dispensations see Errata section.

Availability:

- ISSU - Added the Ability to apply service packs in an In-Srevice-Upgrade manner. Service packs Won't be ISSUable by default, if there is a need to apply a service pack in an ISSU fashion, it need to be communicated to Broadcom in Advance.

MAJOR BUGFIXES

The list below refers to major bugfixes, and does not provide a comprehensive coverage of various bugfixes on all levels.

Basic data path, connectivity and Traffic Management features:

- None

Packet Processing:

- KBP - IPv4/6 MC Fix issues related to the entry mask.
- KBP - DIP SIP sharing, Fix issue related to ACL group with only SIP/DIP in the group.
- In MPLS ingress protection, the filtering for a protected backup outer label wasn't taken if the inner label was unprotected (protection pointer was accidentally taken from inner label).
- When performing CPU based MACT learning, entries with payload FEC + out-LIF were not parsed correctly, resulting in a false invalid out-LIF indication of the learned entry. The MACT event parsing is now correct as it reflects the real value of the out-LIF valid indication.

ERRATA

The list below relates to major open bugs that are not resolved: Basic data path, connectivity and Traffic Management features:

- WB isn't supported.

Packet Processing:

- A call to `bcm_l3_egress_create()` with `failover_id` out of legal range causes the SDK to crash.
- Diag pp eve doesn't display correct Out-LIF information when the per device default Out-LIF is used.
- Tunnel: When `bcm_tunnel_initiator_create()` is called with type bcmTunnelTypeIpAnyIn4=3, the retrieve API `bcm_tunnel_initiator_get()` returns type bcmTunnelTypeIp4In4=1.
- OAMP transmission is not functional for the following types of packets: BFD echo, BFD PWE GAL, Micro-BFD.
- OAM/BFD statistics when using UDH.
- BFD: Remote Multiplier detection is not functional.

BCM88470-FAMILY BETA RELEASE

This release is a beta version for the BCM88470-Family product line, following previously released bring-up EA versions. This release is meant for the silicone bring-up, and is not meant as a stable or feature reach release. The subsequent sections describe the features validated for this release, known issues and bring-up guidelines.

VALIDATED FEATURES

Unless stated otherwise, features validation for this release covered legacy (BCM88660-compatible) modes, and not the new modes added for BCM88370-Family and BCM88670-Family devices.

Basic data path, connectivity and Traffic Management features:

- Register & Memory access including DMA
- Supported SKUs: Qumran-AX: 88470, 88470P, 88471,88473 Kalia: and 88476, 88476P
- Device core mode: dual-core symmetrical
- Interfaces:
- SRAM, DDR4@1.6GHz
- CPU RX/TX (packet DMA)
- NIF 156.25MHz and 125MHz
- NIF:
  - 10GE: KR/XFI, RXAUI, XAUI
  - 40GE: XLGE
  - 100GE
  - ILKN
  - Only 12.5G
  - The following restriction are imposed on sharing ILKN PMs.
  - PM4-6 can’t share the same ILKN with PM8-10.
  - PM7 can’t share the same ILKN with any other PM.
  - PM11 can’t share the same ILKN with any other PM.
  - QSGMII
- Forwarding: Unicast Multicast mirroring
- CosQ: End-to-end scheduling Ingress and Egress compensation VOQ ingress queue creation. Mapping by-destination and by-flow-id Tail Drop Credit watch dog Egress queuing: number of priorities per port: 1/2/8 Undersubscribed CoE (mapping multiple ports, no Flow Control)
- Counters: SNMP counters Internal (diagnostics) counters
- Packet forwarding - ITMH, Force-FWD
- Fabric connectivity with FE device (Kalia only)
- Metering/Policer
- Link Bonding (Data path only)

Packet Processing:

- Diagnostics
- Port
- STG
- L2-Forwarding
- L2-Learning
- VLAN-Translation
- VLAN

- VSWITCH
- L3-IPv4-UC
- L3-IPv6-UC
- L3-Interface-RIF
- L3-Egress-ARP
- L3-Egress-FEC
- L3-ECMP
- L2VPN
- L3VPN
- MPLS
- VPLS
- VPWS
- Tunnel-IPv4
- Protection (Ingress, FEC, Egress, 1:1, 1+1)
- VXLAN

KNOWN ISSUES

Network Interface:

- Only the interfaces indicated above are validated. If any interface that is currently not supported is blocking initial bring-up, please consult with Broadcom AE.
- Auto-training and Auto-negotiation protocols are not validated at this stage
- SAT is not functional
- Control message of Link Bonding is not supported.

Packet Processing:

- ROO/RIOT is not functional
- OAM is not functional. Only basic scenarios work like ETH-OAM.
- BFD is not functional. Only basic scenarios work like BFDv4 multi-hop.
- KBP is not functional
- FCOE is not functional
- MIM is not functional
- ERSPAN is not functional
- Hashing (Configured,SLB) is not functional
- DCB (Extender, EVB) is not functional
- EVPN is not functional
- L2GRE is not functional
- TRILL is not functional
- QOS is not tested yet
- Reflector is not functional
- ECN-PP is not functional

BRING-UP GUIDELINES

REFERENCE DOCUMENTATION

Multiple documents describing relevant HW and SW aspects are available, including:

SW DOCUMENTATION

- 88X7X-PG2xx: Traffic Manager Theory of Operation This document describes theory of operation and provides driver reference for the BCM88670 device series.
- 88X7X-PG1xx: BCM-API Packet Processing: Theory of Operation This document describes BCM-API for the BCM88670 device packet processing capabilities, and how to configure it for networking applications.

HW ARCHITECTURE SPECS:

- `88470_88476-AG2xx:` Traffic Management Architecture This document describes the BCM88470/6 traffic management architecture and fabric adapter. It is intended for system architects and anyone else seeking an understanding of the features and capabilities that the BCM8847x traffic management architecture provides.
- 8847X-AG3xx: BCM8847X Packet Processor Enhancements This document is intended for system architects and anyone else seeking an understanding of the features and capabilities that the BCM8847x Packet processing architecture provides.
- 88470-AG1xx: OAM enhancements This document provides an overview of the OAM functions supported in BCM8847x and explains how each OAM function is supported. It is intended for use by system architects and anyone else seeking an understanding of the OAM features in the BCM8847x switching device.

DIAGNOSTICS SHELL

The diagnostics commands below are useful for debugging. All the diagnostics are called from the BCM shell. If you type the diagnostic incorrectly, usage will be printed. The following diag shell commands are validated:

- General diag count g - display a graphical representation of the device counters show counters: MIB counters clear counters
- NIF/Ports: diag nif - display link status information. diag `port_db` (diag port) - display ports information, mapping to PMs, channels etc.
- Packet DMA: "tx" shell command
- PP: diag pp kbp `kaps_show` diag dbal

PORT MAPPING

Core association must be defined upon initialization, using the new format of BCM8867x:

- `ucode_port.port<logical-port-id>.<unit>` = `<Interface_name>.<channel_num>:core_<core-id>.<tm-` port-id>

DRAM INTERFACE

It is recommended to start the bring-up in "SRAM-only" mode. Before trying to use DRAM. SOC properties: `ext_ram_present=0` to work in SRAM-only mode.

- When working with DRAM, change `ext_ram_present=<0|1|21|22|3>` according to the number of DRAM interfaces - seed details in the UM (88x7x-PG2xx).
- When working with DRAM, please note that for each device, for the first time you need to run DRAM PHY calibration. Once calibration parameters are determined, they can be saved and restored upon later initializations to significantly reduce initialization time. See Driver Reference- DRAM PHY Tuning section in the UM for details.

NETWORK LANE SWAP AND POLARITY CONFIGURATION

Network Interface

- Lane swap and polarity - the direction of configuration for both RX and TX is from front panel to device. For example, for the swap configuration below the correct swap-mapping is 0x0321. Device lane0 -> front panel lane3 Device lane1 -> front panel lane0 Device lane2 -> front panel lane1 Device lane3 -> front panel lane2

PROGRAMMABLE ITMH

ITMH (Ingress Traffic Management Header) in 8867x format works.

CINT EXAMPLES MATURITY LEVEL

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/arad_snake_internal_lpbk.c	none
src/examples/dpp/cint_ac_1to1_coupled_protection.c	high
src/examples/dpp/cint_additional_tpids.c	high
src/examples/dpp/cint_advanced_vlan_translation_mode.c	medium
src/examples/dpp/cint_arad_pfc_reception_mapping.c	none
src/examples/dpp/cint_benchmarking_methodology.c	high
src/examples/dpp/cint_bfd.c	medium
src/examples/dpp/cint_bfd_ipv6.c	None
src/examples/dpp/cint_cmic_rcpu.c	none
src/examples/dpp/cint_compensation_setting.c	low
src/examples/dpp/cint_counter_processor.c	none
src/examples/dpp/cint_dynamic_port_add_remove.c	medium
src/examples/dpp/cint_e2e_composite_connectors.c	none
src/examples/dpp/cint_e2e_composite_se.c	none
src/examples/dpp/cint_e2e_dual_shaper.c	high
src/examples/dpp/cint_e2e_symmetric_connection.c	none
src/examples/dpp/cint_e2e_wfq.c	none
src/examples/dpp/cint_ecn_example.c	none
src/examples/dpp/cint_egress_acceptable_frame_type.c	high
src/examples/dpp/cint_egress_transmit_scheduler.c	high
src/examples/dpp/cint_egr_mtu.c	high
src/examples/dpp/cint_evb_example.c	high
src/examples/dpp/cint_fap_vlan_translation.c	high
src/examples/dpp/cint_fcoe_route.c	high
src/examples/dpp/cint_field_bfd_echo.c	None
src/examples/dpp/cint_field_bfd_ipv4_single_hop.c	none

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/cint_field_bfd_ipv6_single_hop.c	none
src/examples/dpp/cint_field_change_upper_layer_protocol_for_pop_mpls.c	high
src/examples/dpp/cint_field_copy_pcp_to_dscp_for_ip_packets.c	high
src/examples/dpp/cint_field_dir_ext_compare_result.c	none
src/examples/dpp/cint_field_dir_ext_counter_inlif.c	low
src/examples/dpp/cint_field_ecn_cni_extract.c	none
src/examples/dpp/cint_field_ecn_dm.c	none
src/examples/dpp/cint_field_egress_modify_tc_per_port.c	none
src/examples/dpp/cint_field_fiber_channel.c	high
src/examples/dpp/cint_field_flexible_qinq_example.c	high
src/examples/dpp/cint_field_gre_learn_data_rebuild.c	medium
src/examples/dpp/cint_field_ipmc_bidir.c	high
src/examples/dpp/cint_field_ipv6_extension_header.c	high
src/examples/dpp/cint_field_learn_data_rebuild.c	high
src/examples/dpp/cint_field_learn_extension.c	none
src/examples/dpp/cint_field_mim_learn_info_set.c	medium
src/examples/dpp/cint_field_mpls_speculative_parse.c	high
src/examples/dpp/cint_field_mtu_filtering.c	high
src/examples/dpp/cint_field_oam_bfd_advanced.c	none
src/examples/dpp/cint_field_pwe_gal_lif_update.c	high
src/examples/dpp/cint_field_reflector_1pass.c	high
src/examples/dpp/cint_field_vpls_gre_wa.c	high
src/examples/dpp/cint_flexible_hashing.c	none
src/examples/dpp/cint_gal_o_pwe_o_mpls.c	high
src/examples/dpp/cint_igmp_example.c	high
src/examples/dpp/cint_ipmc_bidir.c	high

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/cint_ipmc.c	high
src/examples/dpp/cint_ipmc_example.c	high
src/examples/dpp/cint_ipmc_example_rpf.c	high
src/examples/dpp/cint_ipmc_flows.c	medium
src/examples/dpp/cint_ip_route.c	medium
src/examples/dpp/cint_ip_route_explicit_rif.c	high
src/examples/dpp/cint_ip_route_rpf.c	high
src/examples/dpp/cint_ip_route_tunnel.c	medium
src/examples/dpp/cint_ip_route_tunnel_mpls_additional_labels.c	none
src/examples/dpp/cint_ip_tunnel.c	medium
src/examples/dpp/cint_ip_tunnel_term.c	medium
src/examples/dpp/cint_ipv4_fap.c	high
src/examples/dpp/cint_ipv6_fap.c	high
src/examples/dpp/cint_ipv6_tunnel.c	high
src/examples/dpp/cint_ipv6_tunnel_term.c	high
src/examples/dpp/cint_jer_fc_coe_config_example.c	none
src/examples/dpp/cint_jer_fc_ilkn_inb_config_example.c	none
src/examples/dpp/cint_jer_fc_oob_config_example.c	none
src/examples/dpp/cint_jer_fc_trigger_config_example.c	none
src/examples/dpp/cint_l2_auth.c	high
src/examples/dpp/cint_l2_cache.c	high
src/examples/dpp/cint_l2_cpu_learning.c	high
src/examples/dpp/cint_l2_fast_flush.c	high
src/examples/dpp/cint_l2gre.c	medium
src/examples/dpp/cint_l2_learn_event_parsing.c	none
src/examples/dpp/cint_l2_learning.c	high
src/examples/dpp/cint_l2_learning_multi_device.c	high
src/examples/dpp/cint_l2_mact.c	High
src/examples/dpp/cint_l2_mact_sa_discard.c	High
src/examples/dpp/cint_l2_traverse.c	High
src/examples/dpp/cint_l2_vlan_learning.c	High
src/examples/dpp/cint_l3_multipath.c	medium
src/examples/dpp/cint_l3vpn.c	High
src/examples/dpp/cint_l3_vrrp.c	high

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/cint_linkscan.c	none
src/examples/dpp/cint_mact_bulk_operations.c	high
src/examples/dpp/cint_mact.c	medium
src/examples/dpp/cint_mim_mp.c	Medium
src/examples/dpp/cint_mim_p2p.c	High
src/examples/dpp/cint_mirror_erspan.c	medium
src/examples/dpp/cint_mirror_tests.c	high
src/examples/dpp/cint_mldp.c	high
src/examples/dpp/cint_mpls_ecmp.c	medium
src/examples/dpp/cint_mpls_lsr.c	medium
src/examples/dpp/cint_mpls_mc.c	high
src/examples/dpp/cint_mpls_nop_action_encapsulation.c	high
src/examples/dpp/cint_mpls_protection.c	high
src/examples/dpp/cint_mpls_tunnel_initiator.c	high
src/examples/dpp/cint_mpls_various_scenarios.c	medium
src/examples/dpp/cint_multicast_application_port_lvl_example.c	none
src/examples/dpp/cint_multicast_applications.c	none
src/examples/dpp/cint_multi_device_utils.c	medium
src/examples/dpp/cint_multiple_mymac_term.c	high
src/examples/dpp/cint_mvpn.c	high
src/examples/dpp/cint_oam_acceleration_advanced_features.c	none
src/examples/dpp/cint_oam.c	medium
src/examples/dpp/cint_oam_cfm_o_eth_o_pwe_o_eth.c	None
src/examples/dpp/cint_oam_endpoint_over_lag.c	None
src/examples/dpp/cint_oam_up_mep_over_vpls.c	None
src/examples/dpp/cint_oam_y1731_over_tunnel.c	none
src/examples/dpp/cint_pmf_2pass_snoop.c	none
src/examples/dpp/cint_pmf_trill_2pass_snoop.c	none
src/examples/dpp/cint_policer_metering_example.c	medium
src/examples/dpp/cint_port_eg_filter_example.c	high

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/cint_port_extender_cb_uc.c	high
src/examples/dpp/cint_port_extender_dynamic_switching.c	none
src/examples/dpp/cint_port_extender_mapping.c	none
src/examples/dpp/cint_port_match.c	high
src/examples/dpp/cint_port_tpid.c	low
src/examples/dpp/cint_port_vlan_range.c	high
src/examples/dpp/cint_pp_config2.c	high
src/examples/dpp/cint_push_mpls_gre.c	high
src/examples/dpp/cint_qos.c	medium
src/examples/dpp/cint_qos_initial.c	high
src/examples/dpp/cint_qos_l2.c	high
src/examples/dpp/cint_qos_l3.c	high
src/examples/dpp/cint_qos_l3_rif_cos.c	medium
src/examples/dpp/cint_qos_marking.c	high
src/examples/dpp/cint_qos_mspw.c	high
src/examples/dpp/cint_qos_vpls.c	high
src/examples/dpp/cint_queue_tests.c	low
src/examples/dpp/cint_rate_policer_example.c	high
src/examples/dpp/cint_route_over_ac.c	high
src/examples/dpp/cint_snake.c	high
src/examples/dpp/cint_stateful_load_balancing.c	none
src/examples/dpp/cint_stg.c	high
src/examples/dpp/cint_system_red.c	none
src/examples/dpp/cint_tdm_example.c	none
src/examples/dpp/cint_tm_fap_config2.c	low
src/examples/dpp/cint_tm_fap_with_id.c	none
src/examples/dpp/cint_trap_mgmt.c	none
src/examples/dpp/cint_trill_advanced_vlan_translation_mode.c	none
src/examples/dpp/cint_trill.c	none
src/examples/dpp/cint_trill_roo.c	none
src/examples/dpp/cint_trill_tts.c	none
src/examples/dpp/cint_trunk.c	high
src/examples/dpp/cint_twamp.c	high
src/examples/dpp/cint_vlan_basic_bridge.c	high
src/examples/dpp/cint_vlan_control_config.c	high
src/examples/dpp/cint_vlan_edit_port_default_lif.c	high
src/examples/dpp/cint_vlan_egress_membership.c	high

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/cint_vlan_egress_tag_acceptance.c	high
src/examples/dpp/cint_vlan_gport_add.c	high
src/examples/dpp/cint_vlan_port_initial_vid.c	high
src/examples/dpp/cint_vsq_gl_example.c	high
src/examples/dpp/cint_vsq_tc_examples.c	low
src/examples/dpp/cint_vswitch_cross_connect_p2p.c	high
src/examples/dpp/cint_vswitch_cross_connect_p2p_directional.c	high
src/examples/dpp/cint_vswitch_cross_connect_p2p_multi_device.c	high
src/examples/dpp/cint_vswitch_metro_mp.c	medium
src/examples/dpp/cint_vswitch_metro_p2p.c	high
src/examples/dpp/cint_vswitch_metro_simple.c	high
src/examples/dpp/cint_vswitch_vpls.c	medium
src/examples/dpp/cint_vswitch_vpls_ecmp.c	high
src/examples/dpp/cint_vswitch_vpls_gre.c	high
src/examples/dpp/cint_vswitch_vpls_hvpls.c	high
src/examples/dpp/cint_vswitch_vpls_roo.c	none
src/examples/dpp/cint_vswitch_vpws_tagged.c	high
src/examples/dpp/cint_vxlan.c	medium
src/examples/dpp/cint_vxlan_roo.c	None
src/examples/dpp/cint_vxlan_roo_mc.c	None
src/examples/dpp/cint_xgs_mac_extender_mappings.c	high
src/examples/dpp/pon/cint_pon_anti_spoofing.c	none
src/examples/dpp/pon/cint_pon_application.c	high
src/examples/dpp/pon/cint_pon_cos_remark.c	none
src/examples/dpp/pon/cint_pon_cpu_rx_tx.c	high
src/examples/dpp/pon/cint_pon_general_anti_spoofing.c	none
src/examples/dpp/pon/cint_pon_ipv4_acl.c	none

Table 14: Cint examples

cint	*maturity level*
src/examples/dpp/pon/cint_pon_ipv6_acl.c	none
src/examples/dpp/pon/cint_pon_local_route.c	high
src/examples/dpp/pon/cint_pon_mac_management.c	high
src/examples/dpp/pon/cint_pon_multicast.c	none
src/examples/dpp/pon/cint_pon_tunnel_mapping.c	high
src/examples/dpp/pon/cint_pon_tunnel_profile.c	high
src/examples/dpp/pon/cint_pon_up_ratelimit.c	high
src/examples/dpp/pon/cint_pon_up_scheduler.c	none
src/examples/dpp/pon/cint_pon_utils.c	medium
src/examples/dpp/pon/cint_pon_vlan_range.c	high
src/examples/dpp/utility/cint_utils_extender.c	high
src/examples/dpp/utility/cint_utils_field.c	medium
src/examples/dpp/utility/cint_utils_global.c	low
src/examples/dpp/utility/cint_utils_l2.c	medium
src/examples/dpp/utility/cint_utils_l3.c	medium
src/examples/dpp/utility/cint_utils_mpls.c	medium
src/examples/dpp/utility/cint_utils_mpls_port.c	medium
src/examples/dpp/utility/cint_utils_multicast.c	medium
src/examples/dpp/utility/cint_utils_oam.c	medium
src/examples/dpp/utility/cint_utils_port.c	medium
src/examples/dpp/utility/cint_utils_roo.c	none
src/examples/dpp/utility/cint_utils_vlan.c	medium

BCM88770 (FE3600) RELEASE

The Broadcom BCM88770 (formerly named BCM88950) is the fourth generation in the Dune product line of Fabric Element (FE) devices. This is a sustain release of BCM88950 driver, with all major features supported.

MAJOR BUGFIXES

- None

BCM88660 (ARAD+), BCM88650 (ARAD) RELEASE

This is a sustain release of BCM88660, BCM88650 driver, with all major features supported.

IMPORTANT NOTES

See BCM88670-Family GA release section, important: SW compatibility guidelines 6.4.X to 6.5.X.

MAJOR BUGFIXES

- None

ERRATA

The list below relates to major open bugs that are not resolved: Basic data path, connectivity and Traffic Management features:

- None

Packet Processing:

- `bcm_port_match_delete()` fails for matches that are added to a LIF that was created with no matches using `BCM_VLAN_PORT_MATCH_NONE`
- A call to `bcm_l3_egress_create()` with `failover_id` out of legal range causes the SDK to crash.
- Diag pp eve doesn't display correct Out-LIF information when the per device default Out-LIF is used.
- Tunnel: When `bcm_tunnel_initiator_create()` is called with type bcmTunnelTypeIpAnyIn4=3, the retrieve API `bcm_tunnel_initiator_get()` returns type bcmTunnelTypeIp4In4=1.

NEW EXTERNAL PHY SUPPORT

BCM8279X ADDITIONAL SUPPORT

The BCM82790/2/6 is a dual 100Gbps Gearbox PHY that multiplexes and demultiplexes eight 25Gbps channels to/from twenty 10Gbps channels supporting Ethernet and Optical Transport Networking(OTN). The BCM82790/2/6 is compliant with the IEEE 802.3ba standard for 100GbE and ITU OTL4.4 signaling, as well as 40GbE, OTL 3.4, and OTL 4.10. The Lane interface supports four bidirectional lanes at 25.78125/27.952493 Gbps for 100GbE/OTL4.4 applications. This release is considered GA for BCM8279x. Additional functionality supported in this release:

- HIGIG2

All flexport modes including these added in the release:

- 100G GB to 40G Mux
- 100G GB to 10G PT Enzo mode
- 40G Mux to 100G
- 10G PT Enzo mode to 100G GB
- 40G PT Enzo mode to 40G Mux

Limitations:

- Chip reset will reset all the ports that share the same PHY-ID (MDIO ADDR).

BCM82864 ADDITIONAL SUPPORT

The BCM82864 is a 40Gbps PMA Gearbox phy that demultiplexes eight 20Gbps channels to sixteen 10Gbps channels supporting Ethernet and Optical Transport Networking(OTN). Additionally the device will operate in 100GbE mode where the BCM82864 supports two full-duplex 100Gbps ports. Additional features supported in this release:

- LR2/SR2/CR2/KR2 are supported in this release.
- CAUI4, CAUI-C2C, CAUI-C2M interfaces are supported in this release.
- Retimer is supported in this release.

Driver Limitations:

- Chip reset will reset all the ports that share the same PHY-ID (MDIO ADDR).
- Flexport and Higig are supported at Beta level in this release.

BCM82328/BCM84328 SUPPORT

BCM82328 is a low power, low latency PHY integrating retime and equalizer functions supporting 40-GbR and 10GbE applications. In 40G mode it supports two full-duplex 40GbE ports (XLPPI-to-XLAUI/KR4) for SR4, LR4, and copper CR4 QSFP+ line-card applications. In 10G mode, it supports eight full-duplex 10GbE ports (SFI-to-XFI/KR) for SR, LR and copper CR SFP+ line-card applications. Operates with with the following reference clocks: Single 156.25 MHz differential clock for 1.25 Gbps, 10.3125 Gpbs and 11.5 Gbps Functionality Supported:

- Link status
- Speed and Interface configuration
- port enable/disable
- Duplex set/get
- Polarity inversion
- Power set/get
- PRBS

- Loopback set/get
- Diagnostics
- Tx set/Get
- Rx set/Get
- Auto negotiation
- Reg read/write
- PHY Reset
- I2C module controller
- Auto module detect
- Probe

Driver Limitations:

- Repeater mode only, no retimer operation
- Clause 73 only for starting Clause 72 and requesting FEC. No speed resolution performed. No Clause 73 in 11.5 Gbps line rate, only Clause 72 supported. Clause 72 may be enabled standalone for close systems. Clause 37 is not supported. No on-chip FEC encoding/decoding, but shall pass-through FEC-encoded data.

BCM84328 is a low power, low latency PHY integrating retime and equalizer functions supporting 40-GbR and 10GbE applications. In 40G mode it supports two full-duplex 40GbE ports (XLPPI-to-XLAUI/KR4) for SR4, LR4, and copper CR4 QSFP+ line-card applications. In 10G mode, it supports eight full-duplex 10GbE ports (SFI-to-XFI/KR) for SR, LR and copper CR SFP+ line-card applications. Operates with the following reference clocks: Single 156.25 MHz differential clock for 1.25 Gbps, 10.3125 Gpbs and 11.5 Gbps. Functionality Supported:

- Link status
- Speed and Interface configuration
- port enable/disable
- mdix set/get
- Polarity inversion
- Power set/get
- PRBS
- Loopback set/get
- Diagnostics
- Tx set/Get
- Rx set/Get
- Auto negotiation
- Reg read/write
- PHY Reset
- Probe
- I2C module controller
- Auto module detect

Driver Limitations:

- Repeater mode only, no retimer operation.
- Clause 73 only for starting Clause 72 and requesting FEC. No speed resolution performed. No Clause 73 in 11.5 Gbps line rate, only Clause 72 supported. Clause 72 may be enabled standalone for close systems. Clause 37 is not supported. No on-chip FEC encoding/decoding, but shall pass-through FEC-encoded data

BCM54380 ADDITIONAL SUPPORT

GA level support has been added to make switches, which do not have half-duplex capability, work with half-duplex link partners. Interoperability between PHY and switch testing occurred on the following switches: BCM56548, BCM56630.

Section 5: Things to note

This section lists items that require special attention.

DEVICE DEPRECATION IN SDK 6.5.X

Support for 65nm and older switch devices has been removed from SDK 6.5.x and subsequent releases in this branch with the following exceptions:

Table 15: Device list

Helix+	***56312-56319***
Scorpion	5682X
HumV	56700, 56701
Humv+/conqueror	56720,56721, 56725
Firebolt	56500-503,56505-509,56322,324,
Triumph/Triumph2/Apollo/Firebolt3	5662X,5663X, 56534,56535, 5652X
Firebolt2	56510-56519
Raven	56220, 56024, 56025, 56225,226,227,229,56624, 56020
Enduro	5633X
Starfighter/Blackbird2	53128

Section 11 of this document has been updated to remove the devices no longer supported.

CPU DEPRECATION IN SDK 6.5.X

Support for the following CPUs are being deprecated in SDK 6.5.x.

Table 16: CPU list

BCM98245	***CPCI 32-bit PPC with Motorola 8245 Processor***
BCM953003C	XMC 32-bit MIPS74Kc with BCM53003 Processor
BCM5300X	32-bit MIPS74Kc with BCM5300X Processor
BCM5301X	Integrated ARM Cortex-A9 CPU on BCM5301X Switch Devices
BCM5302X	Integrated ARM Cortex-A9 CPU on BCM5302X Switch Devices
BCM5621X	Integrated MIPS CPU on BCM5621X Switch Devices
BCM5622X	Integrated MIPS CPU on BCM5622X Switch Devices

SBX DEVICE DEPRECATION IN SDK 6.5.X

In future SDK releases in the 6.5.x stream, all SBX code is no longer being built.

API DEPRECATION IN SDK 6.5.X

Support for the Authorization module and related APIs are deprecated in this release. This affects all APIs that begin `bcm_auth.` The BCM API Reference Guide (56XX-PG650-R) has been updated to reflect this decision.

Support for the CES Module and related APIs are deprecated in this release. This affects all APIs that begin with `bcm_ces.` The BCM API Reference Guide (56XX-PG650-R) has been updated to reflect this decision.

ROUTE INSTALLATION CONSIDERATIONS ON BCM56960 IN ALPM MODE

In BCM56960, due to the relatively small size of UFT and the design of sharing one UFT bucket between multiple pivots, the bcmSwitchObjectL3RouteV4RoutesMax, bcmSwitchObjectL3RouteV6Routes64bMax and bcmSwitchObjectL3RouteV6Routes128bMax in ALPM mode return the number of maximum possible routes that could be installed respectively. The actual number of routes that could be installed may be smaller than those objects.

BCM56860 FLEXPORT MAXIMUM NUMBER LOGICAL PORTS WHEN SUPPORTING PORT SPEEDS OF >= 100GE

BCM56860 supports only up to 44 logical ports per pipeline (i.e. 88 for entire device) when system is configured for port speeds >= 100GE.

SLIGHT DEVIATION OF TRAFFIC DISTRIBUTION ACCURACY ON BCM56860

BCM56860 customers may see some slight deviations with the accuracy of traffic distribution in MMU. The amount of deviation is less than 1%, and depends on the packet size and traffic conditions. The issue will remain intermittent until Broadcom completes internal verification and tuning of MMU parameters as a part of BCM56860 PRA preparation. Once optimized MMU settings become available, they will be integrated into a future SDK release

MINIMUM VXLAN VPN ID FOR BCM56860 AND BCM56960

VPN ID 0x7000 is allowed when calling `bcm_vxlan_vpn_create` with `BCM_VXLAN_PORT_WITH_ID` on BCM56850. It is not allowed for BCM56860 and BCM56960 due to VPN ID 0x7000 is reserved on those devices.

BCM56960 - MIX OF AUTONEG AND FORCED SPEED PORTS WITHIN A SINGLE FALCON CORE IS NOT SUPPORTED

Due to a single PLL constraint inside the Falcon, a mix of both autoneg and forced speed on a single core is currently not supported. All the ports on a single core should be all in autoneg or forced speed mode. The first port of a core doing autoneg will disable the rest of the forced speed ports on the same core. Once the first port on a core completes autoneg and the VCO is locked, the rest of ports on the same core can only do autoneg with the same VCO.

CHANGES IN TRUNK FIELD ACTION IN BCM56960

BCM56960 supports only RoundRobin and Random hashing unlike older devices. Hence, Field actions - bcmFieldActionDynamicHgTrunkCancel, bcmFieldActionTrunkLoadBalanceCancel, bcmFieldActionEcmpLoadBalanceCancel, which were used to disable dynamic hashing are not supported on BCM56960. These are updated to new field actions bcmFieldActionHgTrunkRandomRoundRobinHashCancel, bcmFieldActionTrunkRandomRoundRobinHashCancel, bcmFieldActionEcmpRandomRoundRobinHashCancel which can be used to disable HG Trunk load balancing, Trunk load balancing and ECMP load balancing respectively

SPN_PHY_PORT_PRIMARY_AND_OFFSET

Setting of the config property `spn_PHY_PORT_PRIMARY_AND_OFFSET` is absolutely required for the following PHYs.

- BCM54880E
- BCM54680E
- BCM54682E
- BCM54685E
- BCM54640E
- BCM542XX

NOTES AND CONSIDERATIONS OF WARMBOOT FOR SPECIFIC DEVICES AND FEATURES

WARMBOOT: VALIDATED WARMBOOT UPGRADES.

Following warmboot upgrades have been validated in this release.

Table 17: Validated Warmboot upgrades

Software upgrade	*Supported*
6.5.2 to 6.5.3	Yes
6.4.11 to 6.5.3	Yes

Warmboot testing and issue resolution has focused on the following family of devices:

- BCM53400/BCM56060
- BCM56150
- BCM56220
- BCM56340
- BCM56440
- BCM56450
- BCM56640
- BCM56840
- BCM56850
- BCM56860
- BCM56960
- BCM56160
- BCM56760 - like-to-like tested only

WARMBOOT SUPPORT FOR EMBEDDED APPLICATIONS

This section describes warmboot support and procedures for associated firmware supported in SDK 6.5.x. Below are the various embedded applications (FW) available and supported with SDK 6.5.x:

- BFD
- BHH LM/DM
- MPLS LM/DM
- ETH LM/DM
- Broadsync
- 1588/PTP

All applications support warmboot with the exception of 1588/PTP. Warmboot for the above apps follows the same process like any other module on SDK side. Eapps warmboot support is through Level-2 warmboot i.e. requires information to be stored in sCache. SDK warmboot doesn't affect execution of embedded applications (except 1588/PTP firmware) on internal ARM cores and they keep running as is and maintain their state.

WARMBOOT PROCEDURES

- Users are expected to invoke "sc ControlSync=1" command in the BCM diag-shell to store the information in

sCache

- The corresponding API is `bcm_switch_control_set(int` unit, `bcm_switch_control_t` type, `int` arg)
- Note : Typically, storing of information in sCache is required when there is a module or session or context creation on SDK side. Calling of the `bcm_switch_control_set( )` API is left to the discretion of user/customer depending on their system requirements for eg. after each session creation or module init or before the warmboot procedure initiation
- Free up the resources attached to the unit by calling below API equivalent of "exit clean" BCM shell command. i.e, "`soc_shutdown` (int unit)" Note : Broadsync warmboot procedure mandates usage of the above API. As part of API, SDK informs uC1 about the warmboot process initiation so that Broadsync FW waits for the SDK warmboot procedure to complete for further communication

STEPS TO FOLLOW DURING THE SDK WARMBOOT

Below is an example sequence of diag-shell commands executed (through reload.soc) as part of the SDK warmboot sequence for initializing SDK HOST and firmware communication

- MCSMsg INIT;
- MCSMsg 0;
- MCSMsg 1;
- "MCSMsg INIT" and "MCSMsg x" commands invokes the below APIs respectively:
- `soc_cmic_uc_msg_start(int` unit) /* Allocates Messaging resources/Mutexes */
- `soc_cmic_uc_msg_uc_start(int` unit, `int` uc) /* Starts Messaging with specific core.*/

Customers have to include the above calls in their SW.

WARMBOOT UPGRADE OF EAPPS

In SDK WB upgrades, the HW/switching pipe-line remains functional and there is no interruption during the WB upgrades. In case of FW apps, there is no possibility to upgrade the FW without disruption to the service (sessions established by FW apps). This is because, when the FW WB upgrade happens, FW is reloaded by the customer application. This causes all the sessions with peer entities to go down. To get back to the same state as before the FW upgrade (i.e. FW reload), the customer application will have to re-establish all the sessions with peer entities after the FW reload. As the FW upgrade is fully controlled by the customer application, it is left to customer applications to gracefully tear-down the session before FW upgrade (reload) and then re-establish the sessions after the upgrade as they require.

WARMBOOT - SCACHE MEMORY REQUIREMENTS FOR BCM56560, BCM56565 AND BCM56760

To avoid warmboot failure due to out of memory, BCM56560, BCM56565 and BCM56760 devices require 4MB size of scache memory.

WARMBOOT COSQ CONSIDERATION FOR BCM56340

In BCM56340 device family, CosQNew implementation was not incorrect in previous version. This programming error has been corrected using SDK-82083/SDK-83152 in 6.5.2 and 6.4.8. Due to this while upgrading from older version, CosQNew action will be recovered as UcastCosQNew.

WARMBOOT COSQ CONSIDERATION FOR BCM56260/BCM56450 FAMILY

In BCM56260 and BCM56450 device family, when upgrading from 6.5.2 to 6.5.3, incorrect values of multicast subscriber gport are stored in scache update. This programming error has been corrected using SDK-86593 in 6.5.2, and would need to be patched on 6.5.3 to avoid warmboot upgrade issues to 6.5.3.

SCACHE SIZE REQUIREMENTS FOR BCM56960

In BCM56960 the field related software data structures is saved into the s-cache and software recovers and link them back after warm boot. The s-cache size needs to be increased to save all the data structures available in field module. Software depends only on the saved data for the recovery of IFP configurations after warm boot in BCM56960.

SCACHE SIZE REQUIREMENTS FOR DEVICES IN LEGACY IFP SUPPORT

In warmboot design, the software data structures are saved into the sCache during warmboot sync. When the SDK boots in warmboot mode, it recovers the data structures from sCache and initializes the structures. The sCache size is calculated based on total size of various data structures, considering the max value of every field in the data structure. In BCM56260, the total scache required for all modules together is 573408 bytes. In BCM53400, the total scache required for all modules together is 380188 bytes.

10G LOOPBACK FUNCTION ONLY WORKS WHEN LINKS ARE UP ON THE PORT ON WHICH LOOPBACK IS DESIRED

For PHY 84848 a recent change in the software driver impacts how the speed value of a port is returned. The port speed is only returned correctly when the port has links up and PHY loopback functions use this returned value of speed. When loopback is set for at 10G speeds without link up on the port, loopback does not get set correctly. As as result any packets that are sent on the loopback path may not be received back properly. To workaround this issue, connect a cable on the port that you need a loopback on so that the link is up, and then run loopback. Broadcom is working on a fix in a future release for this problem

BCM82780 INTERFACE CONSIDERATIONS

If using BCM82780 40G mode with interface type CR4/LR4/SR4/KR4, it is required to set the system interface config side to XLAUI/KR4. For example:

```
    phy_sys_interface_xe0=15   /* to set it to XLAUI */
```

If using BCM82780 10G mode, it is required to set the external PHY primary core number config variable in the config file. If the MDIO address of the first port for a given core is 0x20.., the config variable should be set as follows:

```
        Xphy_primary_core_num_32=0x20
        Xphy_primary_core_num_33=0x20
        Xphy_primary_core_num_34=0x20
        Xphy_primary_core_num_35=0x20
    The syntax is : Xphy_primary_core_num_<mdio address in decimal>=<primary
port mdio address>
   In BCM82780 if the port is in KR4 mode, should the user intend to change the
interface the command needs to be executed twice:
        >port xe0,xe1 if=xlaui sp=40000 an=0
       >port xe0,xe1 if=xlaui sp=40000 an=0 /* to change interface from KR4 to
XLAUI */
```

RECENT PHY INTERFACE LIBRARY CHANGES

The 100G interface for BCM56860, BCM86375, and BCM88560 devices required extensive changes to the interface library called "Portmod". Between SDK 6.5.2 and 6.5.3, BCM5686x operations for "Portmod" with BCM8279x family was verified and BCM8279x has achieved GA quality. For legacy 10G/40G devices, the "Portmod" changes required the shim layer to be created to enable the drivers to connect to the Falcon and Eagle cores. This work was completed in between SDK 6.5.2 and 6.5.3, thus made these legacy PHYs operational with the Portmod capable switches noted above.

ENGINEERING SUPPORT STATEMENT FOR OLDER XGS SDK RELEASES

The following releases are now out of XGS engineering support since they are older than 12 months since release:

SDK 6.4.x releases: 6.4.4, 6.4.3, 6.4.2, 6.4.1, 6.4.0 SDK 6.3.x releases: 6.3.11 and older All 6.2.x and older releases. Service impacting defects will be reviewed and potentially addressed on these older releases. Support for non-service impacting defects will only be provided to customers running a SDK release that is less than 12 months old. The following table Table 22: Resolved issues and improvements per older XGS device family shows the number of issues and improvements that have been added to our supported SDK releases by device over the past 12 months. While the table shows individual devices, many issues and improvements will apply to multiple products, e.g. BCM56850 and BCM56860, or all XGS products.

Table 18: Resolved issues and improvements per older XGS device family

Device specific issues	*Bugs resolved in the past 12 months*	*Improvements added in the past 12 months*
BCM56850 family	348	167
BCM56840_PLUS	69	10
BCM56340 family	73	15
BCM56640 family	148	36
BCM56450 family	226	67
BCM56150 family	28	2
BCM56440 family	61	10
BCM53440 family	44	30

ALERT ON END OF MAINTENANCE RELEASE CYCLE FOR SDK 6.4

The SDK 6.4.11 release is the last official maintenance release under the SDK 6.4 delivery stream. For new designs or software releases, customers should plan on moving to this or future 6.5.x releases.

Section 6: Summary of BCM API changes and enhancements

This section summarizes BCM API changes in this release. Complete documentation will be available in the Network Switching Software Programmer's Guide number 56XX-PG653-R.

For the full list of API support by Broadcom device, please reference the file SDK-6.5.x-Support-Matrix.xls in the sdk-all-6.5.x/RELDOCS directory in the release package.

BCM MODULES

The BCM APIs are classified into API groups called 'modules'. The following table lists the new BCM modules and their string equivalents.

Table 19: BCM Module IDs

BCM Module ID	*String Equivalent*
BCM_MODULE_LB	lb
BCM_MODULE_IPSEC	ipsec

BFD EVENTS

These are the new BFD Event types.

Table 20: BFD Event Types

Event type	*Description*
bcmBFDEventEndpointRemotePollBitSet	A BFD control packet received from remote endpoint with Poll Bit set.
bcmBFDEventEndpointRemoteFinalBitSet	A BFD control packet received from remote endpoint with Final Bit set.
bcmBFDEventEndpointLocalSessionStateAdminDown	A local endpoint has been reported as admindown
bcmBFDEventEndpointLocalSessionStateDown	A local endpoint has been reported as down
bcmBFDEventEndpointLocalSessionStateInit	A local endpoint has been reported as init
bcmBFDEventEndpointLocalSessionStateUp	A local endpoint has been reported as up

CLASS OF SERVICE QUEUE CONFIGURATION

New features that can be controlled on a gport/cosq basis.

Table 21: CoSQ Control Type Values

Value	Description	Arg value
bcmCosqControlIngressPortPGHeadroomLimitBytes	Ingress port PG Headroom limit setting.	number of bytes
bcmCosqControlIngressPortPGResetFloorBytes	Ingress port PG reset floor setting.	number of bytes
bcmCosqControlEgressPoolSharedLimitBytes	Shared limit setting for per chip based egress Service Pool.	number of bytes. Note:This value must be greater than at least 3*MTU size.
bcmCosqControlEgressPoolResumeLimitBytes	Resume limit setting for per chip based egress Service Pool.	number of bytes
bcmCosqControlEgressPoolYellowSharedLimitBytes	Shared limit setting for per chip based egress Service Pool for yellow packets.	number of bytes
bcmCosqControlEgressPoolYellowResumeLimitBytes	Resume limit setting for per chip based egress Service Pool for yellow packets.	number of bytes
bcmCosqControlEgressPoolRedSharedLimitBytes	Shared limit setting for per chip based egress Service Pool for red packets.	number of bytes
bcmCosqControlEgressPoolRedResumeLimitBytes	Resume limit setting for per chip based egress Service Pool for red packets.	number of bytes
bcmCosqControlOCBOnly	configures a queue to be OCB only	value: 0-disable, 1-enable
CosqControlDefaultInvalidQueue	Configures the default queue that traffic will be redirected to in case it's destined for invalid queue	gport should be of type BCM_GPORT_IS_UCAST_QUEUE_GROUP, cosq is the queue offset
bcmCosqControlIngressPortDropTpid1	cosq ingress port drop global TPID 1.	
bcmCosqControlIngressPortDropTpid2	cosq ingress port drop global TPID 2.	
bcmCosqControlIngressPortDropUntaggedPCP	cosq ingress port drop untagged PCP	

The new configure/retrieve color flags.

Table 22: size flags

COSQ_GPORT_SIZE_SRAM	Configure SRAM values.
COSQ_GPORT_SIZE_DRAM_BOUND	Configure DRAM bound values.
COSQ_GPORT_SIZE_RECOVERY_FAILURE	Configure recovery failure values.
COSQ_GPORT_SIZE_RECOVERY	Configure recovery values.
COSQ_GPORT_SIZE_GLOBAL	Configure global values.

New retrieve/set various MMU statistics.

Table 23: bcm_cosq_stat_t

Name	*Description*
bcmCosqStatHOLDropPackets	Per COSQ HOL drop packet count.

bcm_cosq_gport_pkt_size_adjust_set

The purpose of this API is to configure a delta value for a source or enable/disable a certain compensation type.

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_gport_pkt_size_adjust_set(
intunit,
bcm_cosq_pkt_size_adjust_info_tadjust_info,
intdelta);
```

Parameters

`unit`	BCM device number
`adjust_info`	(IN) `bcm_cosq_pkt_size_adjust_info` structure
`delta`	(IN) delta value

Description

The purpose of this API is to configure a delta value for a source or enable/disable a certain compensation type.

```
/* cosq_pkt_size_adjust_info structure */
typedef struct bcm_cosq_pkt_size_adjust_info_s {
    uint32 flags;                       /* flags */
    bcm_gport_t gport;                  /* gport value */
    bcm_cos_queue_t cosq;               /* bcm_cos_queue_t */
    bcm_cosq_pkt_size_adjust_source_info_t source_info; /* adjust source (see
@xref=BCM_COSQ_CONTROL_table)*/
} bcm_cosq_pkt_size_adjust_info_t;

/* cosq_pkt_size_adjust_source_info structure */
typedef struct bcm_cosq_pkt_size_adjust_source_info_s {
    bcm_cosq_pkt_size_adjust_source_t source_type; /* compensation source */
    uint32 source_id; /* command id: 0,1 */
} bcm_cosq_pkt_size_adjust_source_info_t;
```

Table 24: Type of a resource allocated.

bcmCosqPktSizeAdjustSourceScheduler	*source is Scheduler*
bcmCosqPktSizeAdjustSourceCrpsInPP	source is CRPS in PP
bcmCosqPktSizeAdjustSourceCrpsInTM	source is CRPS in TM
bcmCosqPktSizeAdjustSourceStatReportIn	source is Stat Report In
bcmCosqPktSizeAdjustSourceStatReportOut	source is Stat Report out

Returns

```
BCM_E_NONE
BCM_E_XXX
```

---

bcm_cosq_gport_pkt_size_adjust_get

The purpose of this API is to get the delta value configured for a certain compensation type

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_gport_pkt_size_adjust_get(
intunit,
bcm_cosq_pkt_size_adjust_info_tadjust_info,
int *delta);
```

Parameters

`unit`	BCM device number
`adjust_info`	(IN) `bcm_cosq_pkt_size_adjust_info` structure
`delta`	(OUT) received delta value

Description

The purpose of this API is to get the delta value configured for a certain compensation type

Returns

```
BCM_E_NONE
BCM_E_XXX
```

---

bcm_cosq_pkt_size_adjust_delta_map_set

Scheduler final delta mapping API

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_gport_pkt_size_adjust_get(
intunit,
intdelta,
intfinal_delta);
```

Parameters

unit	BCM device number
delta	(IN) delta
final_delta	(IN) final delta

Description

Scheduler final delta mapping API

Returns

BCM_E_NONE
BCM_E_XXX

---

bcm_cosq_pkt_size_adjust_delta_map_get

get the final delta mapping

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_gport_pkt_size_adjust_get(
intunit,
intdelta,
int*final_delta);
```

Parameters

unit	BCM device number
delta	(IN) delta
final_delta	(OUT) get the final delta

Description

get the final delta mapping

Returns

BCM_E_NONE
BCM_E_XXX

---

bcm_cosq_ingress_port_drop_enable_set
bcm_cosq_ingress_port_drop_enable_get

set/get enable/disable cosq ingress port drop

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_ingress_port_drop_enable_set(
        intunit,
        bcm_port_t port,
        uint32 flags,
        int enable);

int bcm_cosq_ingress_port_drop_enable_get(
        intunit,
        bcm_port_t port,
        uint32 flags,
        int *enable);
```

Parameters

unit	BCM device number
bcm_port_t	(IN) port
flags	(IN) flags
enable	(IN)(for "_set") TRUE to enable, FALSE to disable
enable	(OUT)(for "_get") TRUE to enable, FALSE to disable

Description

enable/disable cosq ingress port drop

Returns

BCM_E_NONE
BCM_E_XXX

---

bcm_cosq_ingress_port_drop_threshold_set bcm_cosq_ingress_port_drop_threshold_get

set/get cosq ingress port drop thresholds

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_ingress_port_drop_threshold_set(
        intunit,
        bcm_port_t port,
        uint32 flags,
        bcm_color_t color,
        uint32 value);

int bcm_cosq_ingress_port_drop_threshold_get(
        intunit,
        bcm_port_t port,
```

```
        uint32 flags,
        bcm_color_t color,
        uint32 *value);
```

Parameters

unit	BCM device number
bcm_port_t	(IN) port
flags	(IN) flags
color	(IN) color
value	(IN)(for "_set") the required value
value	(OUT)(for "_get") the required value
	The available flags are BCM_COSQ_INGRESS_PORT_DROP_FLAGS_THRESHOLD_f

Description

get/set cosq ingress port drop thresholds

Returns

BCM_E_NONE
BCM_E_XXX

---

bcm_cosq_ingress_port_drop_map_set
bcm_cosq_ingress_port_drop_map_get

set/get cosq ingress port drop map value

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_ingress_port_drop_map_set(
        intunit,
        bcm_port_t port,
        uint32 flags,
        bcm_cosq_ingress_port_drop_map_t map,
        uint32 key,
        bcm_color_t color);

int bcm_cosq_ingress_port_drop_map_set(
        intunit,
        bcm_port_t port,
        uint32 flags,
        bcm_cosq_ingress_port_drop_map_t map,
        uint32 key,
        bcm_color_t *color);
```

Parameters

unit	BCM device number
bcm_port_t	(IN) port
flags	(IN) flags
map	(IN) map
key	(IN) key
color	(IN)(for "_set") color
color	(OUT)(for "_get") color
	The available flags are BCM_COSQ_INGRESS_PORT_DROP_FLAGS_THRESHOLD_f
	The available values for map are:
	We may use these macros for the key parameter:
	#define BCM_COSQ_INGRESS_PORT_DROP_ETH_MAP_KEY(pcp, dei) (((pcp << 1) | dei) & 0xF) #define BCM_COSQ_INGRESS_PORT_DROP_TM_MAP_KEY (tc, dp) (((tc << 2) | dp) & 0x3F)

Description

set/get cosq ingress port drop map value

Returns

BCM_E_NONE
BCM_E_XXX

PORT EXTENSION DATA TYPES

The `bcm_extender_port_t` structure describes a Port Extender port logically attached to a Controlling Bridge, the following new members are added.

```
typedef struct bcm_extender_port_s {
...
    int inlif_counting_profile;          /* In LIF counting profile */
    int outlif_counting_profile;         /* Out LIF counting profile */
} bcm_extender_port_t;
```

FABRIC

bcm_fabric_module_control_set
bcm_fabric_module_control_get

Per module configuration

Syntax

```
#include <bcm/fabric.h>

int
bcm_fabric_module_control_set(
    int unit,
    uint32 flags,
    bcm_module_t modid, /* module id */
    bcm_fabric_module_control_t control,
    int value);

int
bcm_fabric_module_control_get(
    int unit,
    uint32 flags,
    bcm_module_t modid, /* module id */
    bcm_fabric_module_control_t control,
    int *value);
```

Parameters

`unit`	(IN) Unit number.
`flags`	(IN) Flags
`modid`	(IN) Module ID
`control`	(IN) Control
`value`	(IN) (for "`_set`")
`value`	(OUT) (for "`_get`")

Description

Per module configuration

Table 25: Fabric Module Control Types

enum	*Description*
bcmFabricModuleControlFSMSeqNumberEnable	Fabric Status Message sequence numbers mechanism enable/ disable.

Returns

`int`

FORWARDING FAILOVER PROTECTION

bcm_failover_multi_level_t_init

Initializes 2 level failover data structure.

Syntax

```
#include <bcm/failover.h>
void
bcm_failover_multi_level_t_init(bcm_failover_multi_level_t
*multi_level_failover)
```

Parameters

`multi_level_failover` (IN) Pointer to `bcm_failover_multi_level_t` structure

Description

This API initializes multi level failover data structure to it default values. Currently all defaults values are 0.

Returns

`void`

FCMAP API

```
New status has been added
/* link fault trigger. */
#define BCM_FCMAP_LF_TR_SYS_LNK_FAILURE             0xE

/* link fault reason code. */
#define BCM_FCMAP_LF_RC_SYS_LINK_FAILURE           0xB
```

This API returns the current diagnostic code from the specified FC port. The `bcm_error_t` type returns the `BCM_FCMAP_DIAG_` codes.

```
#define BCM_FCMAP_DIAG_OK                                0x0
#define BCM_FCMAP_DIAG_PORT_INIT                         0x1
#define BCM_FCMAP_DIAG_OPEN_LINK                         0x2
#define BCM_FCMAP_DIAG_LINK_FAILURE                      0x3
#define BCM_FCMAP_DIAG_OLS_RCVD                          0x4
#define BCM_FCMAP_DIAG_NOS_RCVD                          0x5
#define BCM_FCMAP_DIAG_SYNC_LOSS                         0x6
#define BCM_FCMAP_DIAG_BOUCELINK_FROM_ADMIN              0x7
#define BCM_FCMAP_DIAG_CHGSPEED_FROM_ADMIN               0x8
#define BCM_FCMAP_DIAG_DISABLE_FROM_ADMIN                0x9
#define BCM_FCMAP_DIAG_AN_NO_SIGNAL                      0xA
#define BCM_FCMAP_DIAG_AN_TIMEOUT                        0xB
#define BCM_FCMAP_DIAG_PROTO_TIMEOUT                     0xC
#define BCM_FCMAP_DIAG_SYS_LNK_FAILURE                   0xD
```

FIELD PROCESSOR

Table 26: Field Qualifiers

bcmFieldQualifyL2StationMove	***L2 Source port mismatch. L2 Source port mismatch alone cannot confirm the L2StationMove scenario in all cases. For Example, if the MAC entry is static in L2 table, then L2 Station Move doesn't happen even if the L2 Source port mismatch happens. So to avoid false positives for identifying L2StationMove, it is recommended to use bcmFieldQualifyL2StationMove qualifier in concatenation with bcmFieldQualifyL2SrcStatic qualifier. i.e., L2SrcStatic is not true (bcm_field_qualify_L2SrcStatic(unit, entry, 0x0, 0x1)) and L2StationMove(L2 Source port mismatch) is true (bcm_field_qualify_L2SationMove(unit,entry,0x1,0x1)) will identify the actual L2 station move scenario.***
bcmFieldQualifyExternalValue4	External lookup 4 value
bcmFieldQualifyExternalValue5	External lookup 5 value
bcmFieldQualifyExternalHit4	External lookup 4 hit
bcmFieldQualifyExternalHit5	External lookup 5 hit

Table 27: Field Application Type

Type	***Purpose***
bcmFieldAppTypeIp4DoubleCapacity	IPv4 Unicast Routing with double capacity in external tcam..

Table 28: Field Actions for bcm_field_action_add

bcmFieldActionSnoop	***Snoop matched packets (treat them according to the specified snoop profile)***	***Snoop GPort Tag***	***n/a***
bcmFieldActionExternalValue4Set	Set External lookup 4 result	n/a	n/a
bcmFieldActionExternalValue5Set	Set External lookup 5 result	n/a	n/a
bcmFieldActionSwapMacDaSa	Swap Destination and Source Mac addresses	n/a	n/a
bcmFieldActionSwapSourceIPDestIP	Swap Destination and Source IP addresses	n/a	n/a
bcmFieldActionReplaceTtl	TTL is replaced with the value programmed by bcmFieldControlReplacementTtl instead of decrementing	n/a	n/a
bcmFieldActionPacketIsL3SAT	Indicates matched packet is L3SAT packet and calculates offset to the start of the L3SAT PDU	n/a	n/a
bcmFieldActionReplaceSrcMac	Replace L2 field Source Mac	Source Mac	n/a
bcmFieldActionReplaceDstMac	Replace L2 field destination Mac	Destination Mac	n/a

Table 28: Field Actions for bcm_field_action_add

bcmFieldActionSnoop	*Snoop matched packets (treat them according to the specified snoop profile)*	*Snoop GPort Tag*	*n/a*
bcmFieldActionReplaceInnerVlan	Replace L2 field inner Vlan	Inner Vlan	n/a
bcmFieldActionReplaceOuterVlan	Replace L2 field Outer Vlan	Outer Vlan	n/a
bcmFieldActionReplaceInnerVlanPriority	Replace L2 field inner Vlan priority	Vlan priority type flag (BCM_FIELD_REPLACE_VLAN_PRIORITY / BCM_FIELD_REPLACE_VLAN_PRIORITY_USE_INTERNAL_CFI / BCM_FIELD_REPLACE_VLAN_PRIORITY_FROM_INT_PRIO_AND_CNG)	Priority value
bcmFieldActionReplaceOuterVlanPriority	Replace L2 field outer Vlan priority	Vlan priority type flag (BCM_FIELD_REPLACE_VLAN_PRIORITY / BCM_FIELD_REPLACE_VLAN_PRIORITY_USE_INTERNAL_CFI / BCM_FIELD_REPLACE_VLAN_PRIORITY_FROM_INT_PRIO_AND_CNG)	Priority value.
bcmFieldActionStatAndPolicer	Activates both Stat and Policer actions	Stat + Meter Id	n/a
bcmFieldActionSnoopAndTrap	Activates both Snoop and Trap actions	Snoop + Trap GPort	n/a
bcmFieldActionIEEE1588	Setting varoius parameters for 1588 frames	Packet-is-IEEE-1588(1 Bit); IEEE-1588-Update-Time-Stamp(1 Bit) ; IEEE-1588-Command, (2 Bit) ; IEEE-1588-Encapsulation(1 bits) ; IEEE-1588-Header-Offset (7 bits)	n/a

Table 29: Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get

Note that not all values below are available for all devices.
Internal Action
SOC_PPC_FP_ACTION_TYPE_DEST
SOC_PPC_FP_ACTION_TYPE_DP
SOC_PPC_FP_ACTION_TYPE_TC
SOC_PPC_FP_ACTION_TYPE_TRAP
SOC_PPC_FP_ACTION_TYPE_SNP
SOC_PPC_FP_ACTION_TYPE_MIRROR
SOC_PPC_FP_ACTION_TYPE_MIR_DIS
SOC_PPC_FP_ACTION_TYPE_EXC_SRC
SOC_PPC_FP_ACTION_TYPE_IS
SOC_PPC_FP_ACTION_TYPE_METER
SOC_PPC_FP_ACTION_TYPE_COUNTER
SOC_PPC_FP_ACTION_TYPE_STAT

Table 29: Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get

Note that not all values below are available for all devices.
SOC_PPC_FP_ACTION_TYPE_OUTLIF
SOC_PPC_FP_ACTION_TYPE_LAG_LB
SOC_PPC_FP_ACTION_TYPE_ECMP_LB
SOC_PPC_FP_ACTION_TYPE_STACK_RT_HIST
SOC_PPC_FP_ACTION_TYPE_VSQ_PTR
SOC_PPC_FP_ACTION_TYPE_CHANGE_KEY
SOC_PPC_FP_ACTION_TYPE_NOP
SOC_PPC_FP_ACTION_TYPE_EGR_TRAP
SOC_PPC_FP_ACTION_TYPE_EGR_OFP
SOC_PPC_FP_ACTION_TYPE_EGR_TC_DP
SOC_PPC_FP_ACTION_TYPE_EGR_OUTLIF
SOC_PPC_FP_ACTION_TYPE_EEI
SOC_PPC_FP_ACTION_TYPE_IN_PORT
SOC_PPC_FP_ACTION_TYPE_USER_PRIORITY
SOC_PPC_FP_ACTION_TYPE_METER_B
SOC_PPC_FP_ACTION_TYPE_COUNTER_B
SOC_PPC_FP_ACTION_TYPE_DP_METER_COMMAND
SOC_PPC_FP_ACTION_TYPE_SRC_SYST_PORT
SOC_PPC_FP_ACTION_TYPE_FWD_CODE
SOC_PPC_FP_ACTION_TYPE_FWD_OFFSET
SOC_PPC_FP_ACTION_TYPE_BYTES_TO_REMOVE
SOC_PPC_FP_ACTION_TYPE_SYSTEM_HEADER_PROFILE_ID
SOC_PPC_FP_ACTION_TYPE_VSI
SOC_PPC_FP_ACTION_TYPE_ORIENTATION_IS_HUB
SOC_PPC_FP_ACTION_TYPE_VLAN_EDIT_COMMAND
SOC_PPC_FP_ACTION_TYPE_VLAN_EDIT_VID_1
SOC_PPC_FP_ACTION_TYPE_VLAN_EDIT_VID_2
SOC_PPC_FP_ACTION_TYPE_VLAN_EDIT_PCP_DEI
SOC_PPC_FP_ACTION_TYPE_IN_RIF
SOC_PPC_FP_ACTION_TYPE_VRF
SOC_PPC_FP_ACTION_TYPE_IN_TTL
SOC_PPC_FP_ACTION_TYPE_IN_DSCP_EXP
SOC_PPC_FP_ACTION_TYPE_RPF_DESTINATION_VALID
SOC_PPC_FP_ACTION_TYPE_RPF_DESTINATION
SOC_PPC_FP_ACTION_TYPE_INGRESS_LEARN_ENABLE
SOC_PPC_FP_ACTION_TYPE_EGRESS_LEARN_ENABLE
SOC_PPC_FP_ACTION_TYPE_LEARN_FID
SOC_PPC_FP_ACTION_TYPE_LEARN_SA_0_TO_15
SOC_PPC_FP_ACTION_TYPE_LEARN_SA_16_TO_47
SOC_PPC_FP_ACTION_TYPE_LEARN_DATA_0_TO_15
SOC_PPC_FP_ACTION_TYPE_LEARN_DATA_16_TO_39
SOC_PPC_FP_ACTION_TYPE_LEARN_OR_TRANSPLANT
SOC_PPC_FP_ACTION_TYPE_IN_LIF
SOC_PPC_FP_ACTION_TYPE_SEQUENCE_NUMBER_TAG
SOC_PPC_FP_ACTION_TYPE_IGNORE_CP
SOC_PPC_FP_ACTION_TYPE_PPH_TYPE

Table 29: Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get

Note that not all values below are available for all devices.
SOC_PPC_FP_ACTION_TYPE_PACKET_IS_BOOTP_DHCP
SOC_PPC_FP_ACTION_TYPE_UNKNOWN_ADDR
SOC_PPC_FP_ACTION_TYPE_FWD_HDR_ENCAPSULATION
SOC_PPC_FP_ACTION_TYPE_IEEE_1588
SOC_PPC_FP_ACTION_TYPE_OAM
SOC_PPC_FP_ACTION_TYPE_USER_HEADER_1
SOC_PPC_FP_ACTION_TYPE_USER_HEADER_2
SOC_PPC_FP_ACTION_TYPE_INVALID_NEXT
SOC_PPC_FP_ACTION_TYPE_COS_PROFILE
SOC_PPC_FP_ACTION_TYPE_COUNTER_PROFILE
SOC_PPC_FP_ACTION_TYPE_ACE_POINTER
SOC_PPC_FP_ACTION_TYPE_NATIVE_VSI
SOC_PPC_FP_ACTION_TYPE_IN_LIF_PROFILE
SOC_PPC_FP_ACTION_TYPE_FLP_ACTION_0
SOC_PPC_FP_ACTION_TYPE_FLP_ACTION_1
SOC_PPC_FP_ACTION_TYPE_FLP_ACTION_2
SOC_PPC_FP_ACTION_TYPE_FLP_ACTION_3
SOC_PPC_FP_ACTION_TYPE_FLP_LEM_1ST_RESULT_0
SOC_PPC_FP_ACTION_TYPE_FLP_LEM_1ST_RESULT_1
SOC_PPC_FP_ACTION_TYPE_FLP_LEM_2ND_RESULT_0
SOC_PPC_FP_ACTION_TYPE_FLP_LEM_2ND_RESULT_1
SOC_PPC_FP_ACTION_TYPE_SLB_HASH_VALUE
SOC_PPC_FP_ACTION_TYPE_IN_RIF_PROFILE
SOC_PPC_FP_ACTION_TYPE_ACE_TYPE
SOC_PPC_FP_ACTION_TYPE_TRAP_REDUCED

Table 30: Match rule value used by OAM CCM Embedded App to indicate MEP type through bcmFieldActionCopyToCpu Action

BCM_FIELD_OAM_MATCHED_RULE_xxx	*Description*
BCM_FIELD_OAM_MATCHED_RULE_ETHERNET_PORT	Match rule values used by OAM CCM Embedded App to indicate Port MEP type
BCM_FIELD_OAM_MATCHED_RULE_ETHERNET_INNER_VLAN	Match rule values used by OAM CCM Embedded App to indicate Inner vlan MEP type
BCM_FIELD_OAM_MATCHED_RULE_ETHERNET_OUTER_VLAN	Match rule values used by OAM CCM Embedded App to indicate Outer vlan MEP type
BCM_FIELD_OAM_MATCHED_RULE_ETHERNET_INNER_PLUS_OUTER_VLAN	Match rule values used by OAM CCM Embedded App to indicate Inner + Outer vlan MEP type

Table 31: Field Actions for bcm_field_action_config_add

See fp_actions in the guide.
Action
bcmFieldActionStat
bcmFieldActionStat

Table 31: Field Actions for bcm_field_action_config_add

See fp_actions in the guide.
bcmFieldActionStat
bcmFieldActionStat

Table 32: Macros used as param0 for bcmFieldActionReplaceInnerVlanPriority and bcmFieldActionReplaceOuterVlanPriority to define the type of source for Vlan priority.

BCM_FIELD_REPLACE_VLAN_xxx
BCM_FIELD_REPLACE_VLAN_PRIORITY
BCM_FIELD_REPLACE_VLAN_PRIORITY_USE_INTERNAL_CFI
BCM_FIELD_REPLACE_VLAN_PRIORITY_FROM_INT_PRIO_AND_CNG

Table 33: Field Control Values

Name	Purpose
bcmFieldControlRedirectExcludeEtherSrcPort	Exclude the ingress front panel port from FP redirection destination.
bcmFieldControlRedirectExcludeHiGigSrcPort	Exclude the ingress higig port from FP redirection destination.
bcmFieldControlReplacementTtl	To program the TTL value to be used by bcmFieldActionReplaceTtl

Table 34: Range Control Flags

Flags	Description
BCM_FIELD_RANGE_REPLACE	Modify configurations of an existing range checker. Using this flag, the Range checker's range limits (min/max) and Type can be modified dynamically.

```
/*The new memebers to the DATA qualifier object, used for creation of custom
data qualifiers. */
typedef struct bcm_field_data_qualifier_s {
    ...
    bcm_field_qualify_t qualifier;      /* Data qualifier based on a predefined
                                           qualifier. Used when
                                       BCM_FIELD_DATA_QUALIFIER_OFFSET_PREDEFINED
                                           is set. */
    bcm_field_stage_t stage;           /* Data qualifier stage (e.g. when based
                                           on predefined qualifiers). */
    uint8 name[BCM_FIELD_MAX_NAME_LEN]; /* Name for Data Qualifier. */
} bcm_field_data_qualifier_t;

/*
 * The new members added to group configuration structure. Used to create a field
group with
 * specific attributes.
 */
typedef struct bcm_field_group_config_s {
    uint32 flags;                       /* Group create flags
```

```
                                                BCM_FIELD_GROUP_CREATE_XXX. */
    ...
    uint8 name[BCM_FIELD_MAX_NAME_LEN]; /* Name for Group. */
} bcm_field_group_config_t;
```

bcm_field_presel_config_set

Map a presel id to a name

Syntax

```
#include <bcm/field.h>

extern int bcm_field_presel_config_set(
    int unit,
    bcm_field_presel_t presel_id,
    bcm_field_presel_config_t *presel_config);
```

Parameters

unit	BCM device number
presel_id	presel ID
presel_config	presel NAME

Description

Map a presel id to a name

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other error code

bcm_field_presel_config_get

Get name from presel id

Syntax

```
#include <bcm/field.h>

extern int bcm_field_presel_config_get(
    int unit,
    bcm_field_presel_t presel_id,
    bcm_field_presel_config_t *presel_config);
```

Parameters

unit	BCM device number
presel_id	presel ID
presel_config	presel NAME

Description

Get name from presel id

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other error code

---

bcm_field_presel_config_t_init

Initialize Field Presel Config structure

Syntax

```
#include <bcm/field.h>

extern void bcm_field_presel_config_t_init(
    bcm_field_presel_config_t *presel_config);
```

Parameters

presel_config	presel NAME

Description

Initialize Field Presel Config structure

Returns

None.

```
/* Field name. */
typedef struct bcm_field_presel_config_s {
    uint8 name[BCM_FIELD_MAX_NAME_LEN];
} bcm_field_presel_config_t;
```

New field qualifyier api's added.

int bcm_field_qualify_ExternalValue4(int unit, bcm_field_entry_t entry, uint64 data, uint64 mask); int bcm_field_qualify_ExternalValue5(int unit, bcm_field_entry_t entry, uint64 data, uint64 mask); int bcm_field_qualify_ExternalHit4(int unit, bcm_field_entry_t entry, uint8 data, uint8 mask); int bcm_field_qualify_ExternalHit5(int unit, bcm_-

`field_entry_t` entry, `int bcm_field_qualify_ClassId(int` unit, `bcm_field_entry_t` entry, `uint32` data, `uint32` mask);

Get a match criteria for a specific qualifier from a field entry.

```
/*
 * core spesific paramters for a single action key, for a field group
 * entry
 */
typedef struct bcm_field_action_core_config_s {
    uint32 flags;   /* flags for core specific action */
    int param0;     /* action core-specific paramter No.0 */
    int param1;     /* action core-specific paramter No.1 */
    int param2;     /* action core-specific paramter No.2 */
} bcm_field_action_core_config_t;

/*
 * BCM action and the stage on which it applies specified internal
 * action. See bcm_field_internal_to_bcm_action_map()
 */
typedef struct bcm_field_internal_to_bcm_map_s {
    bcm_field_action_t bcm_action;  /* BCM action containing specified internal
                                       action. See
                                     bcm_field_internal_to_bcm_action_map() */
    bcm_field_stage_t bcm_stage;    /* BCM stage related to specified internal
                                       action. See
                                     bcm_field_internal_to_bcm_action_map() */
} bcm_field_internal_to_bcm_map_t;

/*
 * Bitmap for the various options on 'flags' on
 * bcm_field_action_width_set() or bcm_field_action_width_get()
 */
typedef enum bcm_field_action_width_e {
    bcm_default_action_width = 0    /* When this bit is set then ignore 'width'
                                       and set/get action runtime width to its
                                       default value. */
} bcm_field_action_width_t;
```

---

bcm_field_internal_to_bcm_action_map

Given low level action, get the BCM actions and stages which use it.

Syntax

```
#include <bcm/field.h>
int
bcm_field_internal_to_bcm_action_map(
  int unit,
  uint32 flags,
  int    internal_action,
  uint32 bcm_action_size,
  bcm_field_internal_to_bcm_map_t *bcm_action_stage,
  uint32 *bcm_action_size_actual);
```

Parameters

unit	BCM device number
flags	Control flags. Currently not in use.
internal_action	Low level field action identifier. See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72)
bcm_action_size	Number of elements available in bcm_action_stage.
bcm_action_stage	Array of bcm_field_internal_to_bcm_map_t. To be loaded by this procedure
bcm_action_size_actual	Number of elements actually loaded into bcm_action_stage

Description

Get list of all BCM action and stages which use specified internal action.

See Field Actions for bcm_field_action_add (page 71) for the list of BCM actions See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72) for the list of Internal actions

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_UNIT	Input unit is wrong
BCM_E_PARAM	Wrong internal parameter. SW error.
BCM_E_MEMORY	No memory could be allocated to carry out this operation
BCM_E_NOT_FOUND	internal_action not found
BCM_E_RESOURCE	Input bcm_action_size is too small
BCM_E_XXX	Other error code

bcm_field_action_width_set

Set new width (in bits) to a specific action. Width may be required to be the default value.

Syntax

```
#include <bcm/field.h>
int
bcm_field_action_width_set(
  int unit,
  uint32 flags,
  int internal_action,
  uint32 width);
```

Parameters

unit	BCM device number
flags	Control flags. See bcm_field_action_width_t.
internal_action	Low level field action identifier. See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72)
width	Number of bits to assign to internal_action

Description

Set new width (in bits) to a specific action. Width must be smaller than the default value. Also, it is not allowed to change action width if action is in use on any data base. Using data base must first be destroyed.

See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72) for the list of Internal actions

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Specified internal_action could not be found for this device or width is larger than default
SOC_E_INIT	Initializations (specifically 'field') have not been completed yet.
SOC_E_FAIL	Failed to access internal arrays
BCM_E_NOT_FOUND	internal_action not found (same as BCM_E_PARAM)
SOC_E_BUSY	Internal_action is in use on some data base
BCM_E_XXX	Other error code

---

bcm_field_action_width_get

Get current width (in bits) set to a specific action or get default width (in bits) of a specific action

Syntax

```
#include <bcm/field.h>
int
bcm_field_action_width_get(
  int unit,
  uint32 flags,
  int internal_action,
  uint32 *width_p) ;
```

Parameters

unit	BCM device number
flags	Control flags. See bcm_field_action_width_t.
internal_action	Low level field action identifier. See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72)
width_p	Loaded by this procedure by number of bits to assigned to internal_action

Description

Get width (in bits) of specified action. Depending on 'flags', get runtime value or default value. If initializations have not yet been completed, this procedure returns the default value.

See Low level (internal) field Actions for bcm_field_internal_to_bcm_action_map,bcm_field_action_width_set,bcm_field_action_width_get (page 72) for the list of Internal actions

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Specified internal_action could not be found for this device
SOC_E_FAIL	Failed to access internal arrays
BCM_E_XXX	Other error code

IP MULTICAST

Table 35: IPMC Flags

Name	*Purpose*
BCM_IPMC_RAW_ENTRY	Insert raw data payload.

LAYER 2 ADDRESS MANAGEMENT

New BCM Layer2 Flags added

Table 36: BCM Layer 2 Flags

Name	*Purpose*
BCM_L2_SET_ENCAP_VALID	DNX only: indication that encap_id_valid needs to set even when encap_id is not valid. The application is supposed to pass the same flag back again while adding the entry in the L2 table so that the overflow entry can be cleared and the process resumed.

The following structure contains information used for setting or getting the L2 learn messages distribution and configuration. New members were added.

```
typedef struct bcm_l2_learn_msgs_config_s {
    ...
    uint8               priority;     /* L2: traffic class, Range: 0..7 */
    uint8               color;        /* L2: drop precedence, Range: 0..3 */
} bcm_l2_learn_msgs_config_t;
```

When creating a new L2 learn distribution (bcm_l2_addr_distribute_t), use the bcm_l2_addr_distribute_t_init() routine, new flags added.

Elements src_port and src_port_mask also takes gport type format. So the module ID/port or trunk ID can be specified on supported hardware switches. If user gives a mask value larger than the hardware field. The extra bits will simply be truncated to fit the hardware field. New flags were added.

Valid L2 replace flags are described in table L2 Replace Flags () .

bcm_l2_cache_profile_set

creates and writes down to HW the bitmap profile for L2CP traps.

Syntax

```
#include <bcm/l2.h>

int
bcm_l2_cache_profile_set(int unit, int trap_type, uint32 profile_idx, uint32
flags, bcm_l2cp_profile_info_t *l2cp_profile_info);
```

Parameters

unit	(IN) BCM device number
trap_type	(IN) 1 -for trap trap bcmRxTrapL2cpDrop and 2 for bcmRxTrapL2cpPeer
profile_idx	(IN)
flags	(IN) reserved for future use
l2cp_profile_info	(IN) contains number of MACs in the structure and list of MACs to be specified as activating the trap

Description

creates and writes down to HW the bitmap profile for L2CP traps.

```
/* Max number of MACs in the structure below. */
#define BCM_RX_TRAP_L2CP_NOF_ENTRIES    64

/* Set of MACS for L2CP traps profile build. */
typedef struct bcm_l2cp_profile_info_s {
    int nof_entries;
    bcm_mac_t mac_add[BCM_RX_TRAP_L2CP_NOF_ENTRIES];
} bcm_l2cp_profile_info_t;
```

Returns

BCM_E_XXX

---

bcm_l2_cache_profile_get

reads from HW the bitmap profile for L2CP traps.

Syntax

```
#include <bcm/l2.h>

int
bcm_l2_cache_profile_get(int unit, int trap_type, uint32 profile_idx, uint32
flags,  bcm_l2cp_profile_info_t *l2cp_profile_info);
```

Parameters

unit	(IN) BCM device number
trap_type	(IN) 1 -for trap trap bcmRxTrapL2cpDrop and 2 for bcmRxTrapL2cpPeer
profile_idx	(IN)
flags	(IN) reserved for future use
l2cp_profile_info	(OUT) contains number of MACs in the structure and list of MACs specified as activating the trap

Description

reads from HW the bitmap profile for L2CP traps and converts to MAC array.

Returns

```
BCM_E_XXX
```

---

bcm_l2_cache_vpn_to_profile_map_set

links the given VPN to required profile index (from 0 to 3) for L2CP traps.

Syntax

```
#include <bcm/l2.h>

int
bcm_l2_cache_vpn_to_profile_map_set(int unit, uint32 vsi, uint32 profile_idx);
```

Parameters

unit	(IN) BCM device number
vsi	(IN) vsi number
profile_idx	(IN) index of a defined profile for L2CP traps

Description

links the given VPN to required profile index (from 0 to 3) for L2CP traps.

Returns

```
BCM_E_XXX
```

---

bcm_l2_cache_vpn_to_profile_map_get

for the given VPN returns profile index (from 0 to 3) for L2CP traps.

Syntax

```
#include <bcm/l2.h>

int
bcm_l2_cache_vpn_to_profile_map_get(int unit, uint32 vsi, uint32 *profile_idx);
```

Parameters

unit	(IN) BCM device number
vsi	(IN) vsi number
profile_idx	(OUT) index of a defined profile for L2CP traps

Description

for the given VPN returns profile index (from 0 to 3) for L2CP traps.

Returns

```
BCM_E_XXX
```

---

bcm_l2_learn_limit_enable
bcm_l2_learn_limit_disable

Enable/Disable BCM L2 learn limit subsystem.

Syntax

```
#include <bcm/l2.h>
int bcm_l2_learn_limit_enable(int unit);
int bcm_l2_learn_limit_disable(int unit);
```

Parameters

unit	BCM device number

Description

Enable/Disable BCM L2 learn limit subsystem

Returns

```
BCM_E_XXX
```

---

bcm_l2_addr_multi_add

Add mutli L2 address entry to the specified device.

Syntax

```
#include <bcm/l2.h>
int bcm_l2_addr_multi_add(int unit, bcm_l2_addr_t *l2addr, int count);
```

Parameters

unit	BCM device number
l2addr	L2 address which is properly initialized
count	L2 address count

Description

Add multi MAC address to the switch Address Resolution Logic (ARL) port with the given VLAN ID and parameters. Use CMIC_PORT for the port value to associate the entry with the CPU. Use flag of BCM_L2_COPY_TO_CPU to send a copy to the CPU. Use flag of BCM_L2_TRUNK_MEMBER to set trunking (TGID must be passed as well with a valid trunk group ID).

Returns

BCM_E_XXX

---

bcm_l2_addr_multi_delete

Delete multi L2 address entry from the specified device.

Syntax

```
#include <bcm/l2.h>
int bcm_l2_addr_delete(int unit, bcm_l2_addr_t *l2addr, int count);
```

Parameters

l2addr	Pointer to layer2 address type
l2addr	L2 address which is properly initialized
count	L2 address count

Description

Delete the given MAC address with the specified VLAN ID from the device. If users want to know which entries are deleted, they should register callback routines by API bcm_l2_addr_register(). If L2_MOD_FIFO is used, but the thread bcmL2MOD.x is stopped, users can not know which entries are deleted, even though they already had registered callback routines.

Returns

BCM_E_XXX

L2GRE MANAGEMENT

Table 37: MPLS Port Match Criteria

Name	Purpose
BCM_L2GRE_PORT_MATCH_SHARE	Shared logic port

LAYER 3 MANAGEMENT

The `flags` field in may take on values which are the logical OR of one or more of the following new flags.

Table 38: BCM Layer3 Ingress Interface Flags

Name	*Purpose*
BCM_L3_INGRESS_L3_MCAST_L2	Set bridge V4MC forwarding instead of bridge forwarding in case of IPV4 MC with IPMC disable.
BCM_L3_INGRESS_EXT_IPV4_DOUBLE_CAPACITY	Enable L3 routing on IPv4 double capacity packets.

```
@p
 bcm_tunnel_terminator_t contains information used to set up
tunnel terminator parameters.  The description for each below describes
which structure new members are added.

@loc=bcm_tunnel_terminator_t
@code
typedef struct bcm_tunnel_terminator_s {
    ...
    int inlif_counting_profile;                 /* In LIF counting profile */
} bcm_tunnel_terminator_t;

bcm_tunnel_terminator_config_key_t contains information used to set up the
lookup key, part of a tunnel termination lookup.

@loc=bcm_tunnel_terminator_config_key_t

@code
typedef struct bcm_tunnel_terminator_config_key_s {
    bcm_ip_t dip;   /*  DIP for tunnel header match. */
} bcm_tunnel_terminator_config_key_t;
```

`bcm_tunnel_terminator_config_action_t` contains information used to set up the lookup payload, part of a tunnel termination lookup.

```
typedef struct bcm_tunnel_terminator_config_action_s {
    int tunnel_class;   /* Tunnel class id for VTEP match. */
} bcm_tunnel_terminator_config_action_t;

@p
bcm_tunnel_initiator_t contains information used when setting or getting
L3 tunnel initiator parameters.  The description for each below describes
which structure new members are added.

@loc=bcm_tunnel_initiator_t
@code
typedef struct bcm_tunnel_initiator_s {
    ...
```

```
    int                 outlif_counting_profile;  /* Out LIF counting profile */
} bcm_tunnel_initiator_t;
```

LINK BONDING (LB)

LB OVERVIEW

The idea behind link bonding is that a transmitting flow is spread across several slower links (or modems), and later joined again to reform the original stream. Each packet of the stream is sliced into fragments and sent to an available modem of the group (link bonding group). Fragments are wrapped in a valid Ethernet frame and a fragmentation header, and are sent over to the Qumran either by a dedicated port, or by a port that aggregates multiple modems by an external switch.

LB DATA TYPES

- `bcm_lbg_t` is an identifier used to refer to a link bonding group.
- `bcm_modem_t` is an identifier used to refer to a modem.

LB MISC DEFINES

Table 39: LB Miscellaneous defines

Name	*Description*
`BCM_LB_FLAG_GLOBAL`	Global configuration.
`BCM_LB_FLAG_UNMAP`	Unmap operation.

LB DIRECTION INFO

Table 40: LB direction type

Rate Pattern Mode	*Description*
bcmLbDirectionRx	Link bonding in receive side
bcmLbDirectionTx	Link bonding in transmit side
bcmLbDirectionBoth	Link bonding in both receive and transmit side

LB FORMAT INFO

Table 41: LB format type

format type	*Description*
bcmLbFormatTypeBypass	LB bypss
bcmLbFormatTypeNonChannelize	LB non channelize
bcmLbFormatTypechannelize	LB channelize

LB CONTROL INFO

Table 42: LB control type

control type	*Description*
bcmLbControlSegmentationMode	segmentation mode(see `LB segment mode` (page 93))

Table 42: LB control type

control type	*Description*
bcmLbControlPacketCrcEnable	If set then adding ETH CRC32 to the end of the packet
bcmLbControlTimeoutThreshold	A timer is configured(per LBG). if wait time for next segment is expired, segment is assumed as dropped.
bcmLbControlMaxOutOfOrder	Maximum difference allowed between the expected sequence number and heads of the active FIFOs of the group
bcmLbControlMaxBuffer	Maximum number of buffers the LBG can acquire.
bcmLbControlExpectedSequenceNumber	expected sequence number of the LBG

Table 43: LB modem control type

modem control type	*Description*
bcmLbModemControlHeaderCompensation	Header compensation value for modem shaper update.

LB STATUS INFO

Table 44: LB status type

status type	*Description*
bcmLbStatusIsInSync	Indicates the expected sequence number is in sync state
bcmLbStatusIsEmptyFifo	LBG FIFOs empty indication

Table 45: LB modem status type

modem status type	*Description*
bcmLbModemStatusIsEmptyFifo	Empty FIFO Indication

LB STATISTIC INFO

Table 46: LB statistic type

statistic type	*Description*
bcmLbStatsUnicastPkts	LBG packet counter for UC packets
bcmLbStatsMulticastPkts	LBG packet counter for MC packets
bcmLbStatsBroadcastPkts	LBG packet counter for BC packets
bcmLbStatsPkts64Octets	LBG Bytes counter with size up to 64B
bcmLbStatsPkts65to127Octets	LBG Bytes counter with size of 65-127B
bcmLbStatsPkts128to255Octets	LBG Bytes counter with size of 128-255B
bcmLbStatsPkts256to511Octets	LBG Bytes counter with size of 256-511B
bcmLbStatsPkts512to1023Octets	LBG Bytes counter with size of 512-1023B
bcmLbStatsPkts1024to1518Octets	LBG Bytes counter with size of 1024-1518B
bcmLbStatsPkts1519Octets	LBG Bytes counter with size of 1519B and above.
bcmLbStatsPkts	LBG packet counter
bcmLbStatsOctets	LBG bytes counter
bcmLbStatsDiscardFragments	Count the number of fragments dropped
bcmLbStatsErrorPkts	LBG error packets counter

Table 47: LB modem statistic type

modem statistic type	*Description*
bcmLbModemStatsPkts	LBG packet counter
bcmLbModemStatsOctets	LBG bytes counter

LB SEGMENT MODE INFO

Table 48: LB segment mode

segment mode	*Description*
bcmLbSegmentMode128	segment of 128B - fixed
bcmLbSegmentMode192	segment of 192B - fixed
bcmLbSegmentMode192Enhanced	segment of 192B - enhanced schedule

LB RX MODEM MAP INFO

LB Rx Modem Map Index:

```
typedef struct bcm_lb_rx_modem_map_index_s {
    bcm_gport_t port;      /* port associated with the modem */
    bcm_vlan_t vlan;       /* relevant to channelized format */
} bcm_lb_rx_modem_map_index_t;
```

LB Rx Modem Map Result:

```
typedef struct bcm_lb_rx_modem_map_config_s {
    bcm_lb_format_type_t lbg_type;    /* Link bonding format type */
    bcm_modem_t modem_id;             /* Modem ID */
} bcm_lb_rx_modem_map_config_t;
```

LB TX PACKET INFO

LB global packet format:

```
typedef struct bcm_lb_packet_config_s {
    uint16 outer_vlan_tpid;      /* tpid in VLAN-Tag */
    uint16 lbg_tpid;             /* tpid in LB-Ether-Type */
} bcm_lb_packet_config_t;
```

LB modem packet format:

```
typedef struct bcm_lb_modem_packet_config_s {
    bcm_lb_format_type_t pkt_format;      /* Link bonding format type */
   bcm_mac_t dst_mac;                  /* The destination MAC address in segment
header */
    bcm_mac_t src_mac;                    /* The source MAC address in segment
header */
    uint16 vlan;                          /* VLAN tag in segment header */
```

```
} bcm_lb_modem_packet_config_t;
```

LB TC DP MAP INFO

LB Tc Dp Map Index:

```
typedef struct bcm_lb_tc_dp_s {
    uint32 tc;                          /* Traffic class */
    uint32 dp;                          /* Drop Precedence */
} bcm_lb_tc_dp_t;
```

LB Tc Dp Map Result(packet priority):

```
typedef struct bcm_lb_pkt_pri_s {
    uint32 pkt_pri;                     /* Vlan priority in packet */
    uint32 pkt_cfi;                     /* Vlan CFI in packet */
    uint32 pkt_dp;                      /* Drop Precedence in packet */
} bcm_lb_pkt_pri_t;
```

LB LBG SCHEDULE INFO

LB LBG schedule weight:

```
typedef struct bcm_lb_lbg_weight_s {
    bcm_lbg_t lbg_id;                   /* LBG ID */
    int lbg_weight;                     /* LBG weight in LBG scheduler */
} bcm_lb_lbg_weight_t;
```

LB MODEM SHAPER INFO

LB modem shaper:

```
typedef struct bcm_lb_modem_shaper_config_s {
    uint32 enable;                      /* Enable/disable modem shaper */
    uint32 rate;                        /* traffic rate. Units: kbps */
    uint32 max_burst;                   /* traffic burst. Units: byte */
} bcm_lb_modem_shaper_config_t;
```

LB APIS

LB MODEM TO PORT APIS

---

bcm_lb_modem_to_port_map_set
bcm_lb_modem_to_port_map_get

(for "_set")Connect modem to port.
(for "_get")Get info of which port the modem is connectted to.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_to_port_map_set(
    int unit,
    bcm_modem_t modem_id,
    uint32 flags,
    bcm_gport_t port);
int bcm_lb_modem_to_port_map_get(
    int unit,
    bcm_modem_t modem_id,
    uint32 flags,
    bcm_gport_t *port);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
flags	(IN) should be 0
port	(IN/OUT) which port the modem is connectted to.

Description

Get info of which port the modem is connectted to.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set mapping of modem to port with invalid parameters
BCM_E_CONFIG	Operation is not allowed
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB ENABLE LBG APIS

---

bcm_lb_enable_set
bcm_lb_enable_get

(for "_set")enable LBG.
(for "_get")Get info of whether LBG is enabled

Syntax

```
#include <bcm/lb.h>
int bcm_lb_enable_set(
    int unit,
    bcm_lbg_t lbg_id,
    uint32 flags,
    uint32 enable);
```

```
int bcm_lb_enable_get(
    int unit,
    bcm_lbg_t lbg_id,
    uint32 flags,
    uint32 *enable);
```

Parameters

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
flags	(IN) should be 0
bw	(OUT) Pointer to info of whether LBG is enabled

Description

Get info of whether LBG is enabled.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_NOT_FOUND	Attempt to get bandwidth for GTF which does not exist
BCM_E_PARAM	Attempt to get bandwidth with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB MODEM TO LBG APIS

---

bcm_lb_modem_to_lbg_map_set
bcm_lb_modem_to_lbg_map_get

(for "_set")Connect modem to lbg.
(for "_get")Get info of which lbg the modem is connectted to

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_to_lbg_map_set(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lbg_t lbg_id);
int bcm_lb_modem_to_lbg_map_get(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lbg_t *lbg_id);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_UNMAP : disconnect modem to port
lbg_id	(IN) LBG ID

Description

Connect modem to lbg.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to connect modem to lbg with invalid parameters
BCM_E_CONFIG	Operation is not allowed
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB SEQUENCE NUMBER WIDTH APIS

---

bcm_lb_sequence_number_width_set
bcm_lb_sequence_number_width_get

Set/get sequence number width.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_sequence_number_width_set(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    int number_width);
int bcm_lb_sequence_number_width_get(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    int *number_width);
```

Parameters

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
number_width	(IN/OUT) sequence number width

Description

Set sequence number width.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set sequence number width with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB CONTROL APIS

bcm_lb_control_set
bcm_lb_control_get

Set/get configuration at LBG or global level.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_control_set(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_control_t lb_control,
    int arg);
int bcm_lb_control_get(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_control_t lb_control,
    int *arg);
```

Parameters

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
lb_control	(IN) type of operation (see LB control type (page 91))
arg	(IN) (for "_set") Argument whose meaning is dependent on lb_control
arg	(OUT) (for "_get") Argument whose meaning is dependent on lb_control

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

---

bcm_lb_modem_control_set
bcm_lb_modem_control_get

Set/get configuration at modem level.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_control_set(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_modem_control_t modem_control,
    int arg);
int bcm_lb_modem_control_get(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_modem_control_t modem_control,
    int *arg);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
modem_control	(IN) type of operation (see LB modem control type (page 92))
arg	(IN) (for "_set") Argument whose meaning is dependent on modem_control
arg	(OUT) (for "_get") Argument whose meaning is dependent on modem_control

Description

Features that can be controlled on modem basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB STATUS APIS

bcm_lb_status_get

Get Status of LBG or global.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_status_get(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_status_type_t status_type,
    int *value);
```

Parameters

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
status_type	(IN) type of operation (see LB status type (page 92))
value	(OUT) Argument whose meaning is dependent on status_type

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to get status with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

---

bcm_lb_modem_status_get

Get Status of modem.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_status_get(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_modem_status_type_t status_type,
    int *value);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
direction	(IN) which direction is relevant to
flags	(IN) should be 0
status_type	(IN) type of operation (see LB modem status type (page 92))
value	(OUT) Argument whose meaning is dependent on status_type

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to get status with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB STATISTIC APIS

bcm_lb_stat_get

Get statistic of LBG or global.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_stat_get(
    int unit,
    bcm_lbg_t lbg_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_stat_val_t type,
    uint64 *value);
```

Parameters

unit	(IN) BCM device number
lbg_id	(IN) LBG ID
direction	(IN) which direction is relevant to
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
type	(IN) type of operation (see LB statistic type (page 92))
value	(OUT) Argument whose meaning is dependent on status_type

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to get statistic with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

---

bcm_lb_modem_stat_get

Get statistic of modem.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_stat_get(
    int unit,
    bcm_modem_t modem_id,
    bcm_lb_direction_type_t direction,
    uint32 flags,
    bcm_lb_modem_stat_val_t type,
    uint64 *value);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
direction	(IN) which direction is relevant to
flags	(IN) should be 0
type	(IN) type of operation (see LB modem statistic type (page 93))
value	(OUT) Argument whose meaning is dependent on status_type

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to get statistic with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB RX MODEM MAP APIS

bcm_lb_rx_modem_map_index_t_init

Initialize a bcm_lb_rx_modem_map_index_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_rx_modem_map_index_t_init(bcm_lb_rx_modem_map_index_t *map_index);
```

Parameters

map_index	(OUT) Pointer to LB rx modem map index structure to initialize

Description

Initializes a LB rx modem map index structure to default values. This function must be used to initialize any LB rx modem map index structure before passing it to an API function.

bcm_lb_rx_modem_map_config_t_init

Initialize a bcm_lb_rx_modem_map_config_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_rx_modem_map_config_t_init(bcm_lb_rx_modem_map_config_t
```

```
*map_config);
```

Parameters

map_config	(OUT) Pointer to LB rx modem map config structure to initialize

Description

Initializes a LB rx modem map config structure to default values. This function must be used to initialize any LB rx modem map config structure before passing it to an API function.

---

bcm_lb_rx_modem_map_set
bcm_lb_rx_modem_map_get

Set/get mapping of {port+vlan} to modem.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_rx_modem_map_set(
    int unit,
    bcm_lb_rx_modem_map_index_t *map_index,
    uint32 flags,
    bcm_lb_rx_modem_map_config_t *map_config);
int bcm_lb_rx_modem_map_get(
    int unit,
    bcm_lb_rx_modem_map_index_t *map_index,
    uint32 flags,
    bcm_lb_rx_modem_map_config_t *map_config);
```

Parameters

unit	(IN) BCM device number
map_index	(IN) index of mapping traffic to modem
flags	(IN) should be 0
map_config	(IN/OUT) modem info

Description

Set/get mapping of {port+vlan} to modem.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_CONFIG	Operation is not allowed
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB PACKET CONFIG APIS

---

bcm_lb_packet_config_t_init

Initialize a bcm_lb_packet_config_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_packet_config_t_init(bcm_lb_packet_config_t *packet_config);
```

Parameters

packet_config	(OUT) Pointer to LB packet config structure to initialize

Description

Initializes a LB packet config structure to default values. This function must be used to initialize any LB packet config structure before passing it to an API function.

---

bcm_lb_packet_config_set
bcm_lb_packet_config_get

Set/get packet format at global level.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_packet_config_set(
    int unit,
    uint32 flags,
    bcm_lb_packet_config_t *packet_config);
int bcm_lb_packet_config_get(
    int unit,
    uint32 flags,
    bcm_lb_packet_config_t *packet_config);
```

Parameters

unit	(IN) BCM device number
flags	(IN) BCM_LB_FLAG_GLOBAL : global configuration
packet_config	(IN) (for "_set") Pointer to packet config structure
packet_config	(OUT) (for "_get") Pointer to packet config structure

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

---

bcm_lb_modem_packet_config_t_init

Initialize a bcm_lb_modem_packet_config_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_modem_packet_config_t_init(bcm_lb_modem_packet_config_t
*packet_config);
```

Parameters

packet_config	(OUT) Pointer to LB modem packet config structure to initialize

Description

Initializes a LB modem packet config structure to default values. This function must be used to initialize any LB modem packet config structure before passing it to an API function.

---

bcm_lb_modem_packet_config_set
bcm_lb_modem_packet_config_get

Set/get packet format at modem level.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_packet_config_set(
    int unit,
    bcm_modem_t modem_id,
```

```
    uint32 flags,
    bcm_lb_modem_packet_config_t *packet_config);
int bcm_lb_modem_packet_config_get(
    int unit,
    bcm_modem_t modem_id,
    uint32 flags,
    bcm_lb_modem_packet_config_t *packet_config);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
flags	(IN) should be 0
packet_config	(IN) Pointer to modem packet config structure
packet_config	(OUT) Pointer to modem packet config structure

Description

Features that can be controlled on modem basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB PRIORITY MAPPING APIS

bcm_lb_tc_dp_t_init

Initialize a bcm_lb_tc_dp_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_tc_dp_t_init(bcm_lb_tc_dp_t *tc_dp);
```

Parameters

tc_dp	(OUT) Pointer to LB tc dp structure to initialize

Description

Initializes a LB tc dp structure to default values. This function must be used to initialize any LB tc dp structure before passing it to an API function.

bcm_lb_pkt_pri_t_init

Initialize a bcm_lb_pkt_pri_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_pkt_pri_t_init(bcm_lb_pkt_pri_t *pkt_pri);
```

Parameters

pkt_pri	(OUT) Pointer to LB packet pri structure to initialize

Description

Initializes a LB packet pri structure to default values. This function must be used to initialize any LB packet pri structure before passing it to an API function.

bcm_lb_tc_dp_to_priority_config_set bcm_lb_tc_dp_to_priority_config_get

Set/get mapping of {tc+dp} to packet priority.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_tc_dp_to_priority_config_set(
    int unit,
    uint32 flags,
    bcm_lb_tc_dp_t *tc_dp,
    bcm_lb_pkt_pri_t *pkt_pri);
int bcm_lb_tc_dp_to_priority_config_get(
    int unit,
    uint32 flags,
    bcm_lb_tc_dp_t *tc_dp,
    bcm_lb_pkt_pri_t *pkt_pri);
```

Parameters

unit	(IN) BCM device number
flags	(IN) should be 0
tc_dp	(IN) (for "_set") Pointer to tc dp structure
pkt_pri	(IN) (for "_set") Pointer to pkt pri structure
pkt_pri	(OUT) (for "_get") Pointer to pkt pri structure

Description

Features that can be controlled on LBG/global basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB TX SCHED APIS

---

bcm_lb_lbg_weight_t_init

Initialize a bcm_lb_lbg_weight_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_lbg_weight_t_init(bcm_lb_lbg_weight_t *lbg_weights);
```

Parameters

lbg_weights	(OUT) Pointer to LB weight structure to initialize

Description

Initializes a LB weight structure to default values. This function must be used to initialize any LB weight structure before passing it to an API function.

---

bcm_lb_tx_sched_set
bcm_lb_tx_sched_get

Set/get scheduler between lbg.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_tx_sched_set(
    int unit,
    uint32 flags,
    int lbg_count,
    bcm_lb_lbg_weight_t *lbg_weights);
int bcm_lb_tx_sched_get(
    int unit,
    uint32 flags,
    int max_lbg_count,
    bcm_lb_lbg_weight_t *lbg_weights,
    int *lbg_count);
```

Parameters

unit	(IN) BCM device number
flags	(IN) should be 0
lbg_count	(IN) (for "_set") indicates how many members in lbg_weights
lbg_weights	(IN) (for "_set") Pointer to lbg weight array
max_lbg_count	(IN) indicates how many members in lbg_weights
lbg_weights	(OUT) (for "_get") Pointer to lbg weight arry
lbg_count	(OUT) (for "_get") Number of lbg_weights retrieved

Description

Features that can be controlled on LBG basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

LB MODEM SHAPER APIS

---

bcm_lb_modem_shaper_config_t_init

Initialize a bcm_lb_modem_shaper_config_t structure.

Syntax

```
#include <bcm/lb.h>
void bcm_lb_modem_shaper_config_t_init(bcm_lb_modem_shaper_config_t *shaper);
```

Parameters

shaper	(OUT) Pointer to LB modem shaper config structure to initialize

Description

Initializes a LB modem shaper config structure to default values. This function must be used to initialize any LB modem shaper config structure before passing it to an API function.

---

bcm_lb_modem_shaper_set
bcm_lb_modem_shaper_get

Set/get modem shaper.

Syntax

```
#include <bcm/lb.h>
int bcm_lb_modem_shaper_set(
    int unit,
    bcm_modem_t modem_id,
    uint32 flags,
    bcm_lb_modem_shaper_config_t *shaper);
int bcm_lb_modem_shaper_get(
    int unit,
    bcm_modem_t modem_id,
    uint32 flags,
    bcm_lb_modem_shaper_config_t *shaper);
```

Parameters

unit	(IN) BCM device number
modem_id	(IN) Modem ID
flags	(IN) should be 0
shaper	(IN/OUT) Pointer to modem shaper config structure

Description

Features that can be controlled on LBG basis.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_PARAM	Attempt to set/get configuration with invalid parameters
BCM_E_UNAVAIL	current API is not supported on the specified unit
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INTERNAL	Unable to release resource lock / Failed to write memory

MAC-IN-MAC MANAGEMENT

The `bcm_mim_vpn_config_t` structure describes the MAC-in-MAC VPN attributes. It allows the application to configure different broadcast, multicast and unknown unicast groups. New members were added.

```
typedef struct bcm_mim_vpn_config_s {
    ...
    int inlif_counting_profile;     /* In LIF counting profile */
    int outlif_counting_profile;    /* Out LIF counting profile */
   bcm_vlan_protocol_packet_ctrl_t protocol_pkt; /* Protocol packet control */
} bcm_mim_vpn_config_t;

typedef struct bcm_mim_port_s {
    ...
    int inlif_counting_profile;         /* In LIF counting profile */
    int outlif_counting_profile;        /* Out LIF counting profile */
} bcm_mim_port_t;
```

MIRRORING

This bcm_mirror_port_set API function takes a flags parameter that may contain any of the flags defined in BCM Mirroring Flags (page 114) below, new flags added.

Table 49: BCM Mirroring Flags

Name	Description
BCM_MIRROR_PORT_UNTAGGED_ONLY	Enable Ingress mirroring for Untagged traffic only
BCM_MIRROR_PORT_TAGGED_DEFAULT	Enable Ingress mirroring for Tagged traffic with unspecified VID

Table 50: BCM Mirror Destination Flags

Name	Description
BCM_MIRROR_DEST_UPDATE_TUNNEL_SFLOW	Update mirror destination with sflow encapsulation

Table 51:

Name	Description
BCM_MIRROR_INVALID_PROFILE	Mirror profile invalid value used in bcm_mirror_destination_get and bcm_mirror_port_destination_get

bcm_mirror_pkt_header_updates_t represents header updates of mirrored packets

```
typedef struct {
    ...
bcm_mirror_pkt_header_fabric_t header_fabric; /* Fabric header updates of
mirrored packets */
} bcm_mirror_pkt_header_updates_t; /* represents header updates of mirrored
packets */
```

MPLS MANAGEMENT

The bcm_mpls_egress_label_t structure describes the MPLS label header values (label, EXP, TTL) and QOS options. New members were added.

```
typedef struct bcm_mpls_egress_label_s {
    ...
  int outlif_counting_profile;        /* Out LIF counting profile */

} bcm_mpls_egress_label_t;

The bcm_mpls_stat_info_t structure describes the MPLS label stat information
values
(number of entries, index, values) and specify type such as byets or packets.
This is used when
getting/setting counter statistics.

@code
```

```
typedef struct bcm_mpls_stat_info_s {
    bcm_mpls_stat_t  stat; /* To specify In/Out Bytes/Pkts */
    uint32           num_entries; /* Number of Counter Entries */
    uint32           *counter_indexes; /* Pointer to Counter indexes */
    bcm_stat_value_t *counter_values; /* Pointer to counter values */
} bcm_mpls_stat_info_t;
```

Table 52: MPLS Egress Label Flags

Name	Purpose
BCM_MPLS_EGRESS_LABEL_EVPN	Create an EVPN tunnel.

Table 53: MPLS Egress Actions

Name	Purpose
BCM_MPLS_EGRESS_ACTION_SWAP_OR_PUSH	Incoming MPLS label is swapped or pushed according to user and network configuration.

The bcm_mpls_port_t structure describes an MPLS port (or "virtual-port") used for L2 MPLS VPNs (VPWS and VPLS VPNs), new members were added.

```
typedef struct bcm_mpls_port_s {
...
bcm_failover_t ingress_failover_id;         /* Ingress Failover Object
Identifier. */
    bcm_gport_t ingress_failover_port_id;       /* Ingress Failover MPLS Port
Identifier. */
    ...
uint32 egress_class_id;             /* Egress Class id for mpls flow. if
application passes class id here, then
                                           class id will not be picked from
egress_tunnel_if egress object.
                                         else SDK will use class id passed in
egress_tunnel_if egress object.
                                           Duplication of class ids are not
allowed. User cannot pass same class
                                      id in egress object and egress_class_id.
please pass class id in egress
                                           object if user wants to have same
class id. value 0 is reserved */
    int inlif_counting_profile;         /* In LIF counting profile */
} bcm_mpls_port_t;
```

Table 54: MPLS Port Match Criteria

Name	Purpose
BCM_MPLS_PORT_MATCH_SHARE	Multiple match criteria share one MPLS logical port

```
typedef struct bcm_mpls_tunnel_switch_s {
...
int inlif_counting_profile;          /* In LIF counting profile */
} bcm_mpls_tunnel_switch_t;
```

bcm_mpls_tunnel_stat_attach

Attach statistics entity to the MPLS tunnel derived from the given L3 Egress interface

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_stat_attach(
    int      unit,
    bcm_if_t intf_id
    uint32   stat_counter_id);
```

Parameters

`unit`	(IN) Unit number.
`intf_id`	(IN) Interface ID of a egress L3 object
`stat_counter_id`	(IN) Stat Counter ID

Description

This API will attach statistics entity to the given MPLS tunnel derived from the given L3 Egress interface

(Ref `FLEXIBLE_COUNTER_s`)

Returns

`BCM_E_XXX`

bcm_mpls_tunnel_stat_detach

Detach statistics entity to the MPLS tunnel derived from the given L3 Egress interface.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_stat_detach(
    int      unit,
    bcm_if_t intf_id);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.

Description

This API will detach statistics entity to the MPLS tunnel derived from the given L3 Egress interface.

(Ref: FLEXIBLE_COUNTER_s)

Returns

BCM_E_XXX

---

bcm_mpls_tunnel_stat_counter_get

Get counter value for the specified MPLS statistic type and Tunnel interface
derived from the given L3 interface ID

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_stat_counter_get(
    int               unit,
    bcm_if_t          intf_id,
    bcm_mpls_stat_t     stat,
    uint32            num_entries,
    uint32            *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(OUT) Pointer to counter values

Description

This API will retrieve counter values for the specified MPLS statistic type and tunnel interface deroved from the L3 Interface ID.

(Ref: FLEXIBLE_COUNTER_s)

Returns

```
BCM_E_XXX
```

bcm_mpls_tunnel_stat_counter_sync_get

Force an immediate counter update and retrieve the specified counter statistic for a MPLS tunnel

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_stat_counter_sync_get(
    int               unit,
    bcm_if_t          intf_id,
    bcm_mpls_stat_t     stat,
    uint32            num_entries,
    uint32            *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

`unit`	(IN) Unit number.
`intf_id`	(IN) Interface ID of a egress L3 object.
`stat`	(IN) Type of the counter to retrieve that is, ingress/egress byte/packet
`num_entries`	(IN) Number of counter Entries
`counter_indexes`	(IN) Pointer to Counter indexes entries
`counter_values`	(OUT) Pointer to counter values

Description

Similar to `bcm_mpls_tunnel_stat_counter_get()`, value returned is software accumulated counter synced with the hardware counter

Returns

```
BCM_E_XXX
```

bcm_mpls_tunnel_stat_counter_set

Set the counter value for the specified MPLS statistic type and Tunnel interface derived derived from the given L3 Interface ID.

Syntax

```
#include <bcm/mpls.h>
```

```
int
bcm_mpls_tunnel_stat_counter_set(
    int               unit,
    bcm_if_t          intf_id,
    bcm_mpls_stat_t     stat,
    uint32            num_entries,
    uint32            *counter_indexes,
    bcm_stat_value_t *counter_values);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat	(IN) Type of the counter to set that is, ingress/egress byte/packet
num_entries	(IN) Number of counter Entries
counter_indexes	(IN) Pointer to Counter indexes entries
counter_values	(IN) Pointer to counter values

Description

This API will set counter statistic values for the derived MPLS tunnel.

(Ref: FLEXIBLE_COUNTER_s)

Returns

BCM_E_XXX

---

bcm_mpls_tunnel_stat_id_get

Get stat counter ID associated with the MPLS tunnel derived from the given L3 interface ID.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_stat_id_get(
    int           unit,
    bcm_if_t      intf_id,
    bcm_mpls_stat_t stat,
    uint32        *stat_counter_id);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat	(IN) Type of the counter
stat_counter_id	(OUT) stat counter ID

Description

This API will provide stat counter IDs associated with the MPLS tunnel derived from the given L3 Interface id.

(Ref: FLEXIBLE_COUNTER_s)

Returns

BCM_E_XXX

---

bcm_mpls_stat_info_t_init

Initialize the MPLS tunnel Stat info structure.

Syntax

```
#include <bcm/mpls.h>
void bcm_mpls_stat_info_t_init(bcm_mpls_stat_info_t  *stat_info);
```

Parameters

stat_info	Pointer to the struct to be initialized

Description

Initialize the MPLS tunnel Stat info structure.

Returns

None.

---

bcm_mpls_tunnel_label_counter_id_detach

Detach counter entry of the specific MPLS tunnel label
pointed to by the given stat counter id.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_label_counter_id_detach(
    int           unit,
    bcm_if_t      intf_id,
```

```
    uint32          stat_counter_id);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat_counter_id	(IN) stat counter ID

Description

This API will allow detaching counter entry of the specific MPLS tunnel label pointed to by the given stat counter id.

Returns

```
BCM_E_XXX
```

---

bcm_mpls_tunnel_label_counter_id_stat_get

Get the specified counter statistics for the given MPLS Tunnel label and counter id associated with it.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_label_counter_id_stat_get(
    int                     unit,
    bcm_if_t                intf_id,
    uint32                  stat_counter_id,
    bcm_mpls_stat_info_t *stat_info);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat_counter_id	(IN) stat counter ID
stat_info	(INOUT) Mpls Stat Info Structure

Description

This API gets specified counter statistics for the given MPLS Tunnel label and counter id associated with it.

Returns

```
BCM_E_XXX
```

bcm_mpls_tunnel_label_counter_id_stat_set

Set the specified counter statistics for the given MPLS Tunnel label
and counter id associated with it.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_label_counter_id_stat_set(
    int                   unit,
    bcm_if_t              intf_id,
    uint32                stat_counter_id,
    bcm_mpls_stat_info_t *stat_info);
```

Parameters

`unit`	(IN) Unit number.
`intf_id`	(IN) Interface ID of a egress L3 object.
`stat_counter_id`	(IN) stat counter ID
`stat_info`	(IN) Mpls Stat Info Structure

Description

This API sets specified counter statistics for the given MPLS Tunnel label and counter id associated with it.

Returns

```
BCM_E_XXX
```

bcm_mpls_tunnel_label_counter_id_stat_sync_get

Get the specified counter statistics for the given MPLS Tunnel label
and counter id associated with it.
SW accumulated counters are made in sync with the hw count.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_label_counter_id_stat_sync_get(
    int                   unit,
    bcm_if_t              intf_id,
```

```
    uint32                stat_counter_id,
    bcm_mpls_stat_info_t *stat_info);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
stat_counter_id	(IN) stat counter ID
stat_info	(INOUT) Mpls Stat Info Structure

Description

This API gets specified counter statistics for the given MPLS Tunnel label and counter id associated with it. SW accumulated counters are made in sync with the hw count.

Returns

BCM_E_XXX	

bcm_mpls_tunnel_label_counter_id_num_get

Get the array of stat counter ids associated with the given MPLS tunnel interface. It can also provide the number of counter indexes associated with the MPLS tunnel.

Syntax

```
#include <bcm/mpls.h>
int
bcm_mpls_tunnel_label_counter_id_num_get(
    int       unit,
    bcm_if_t  intf_id,
    uint32    *num_counters
    uint32    *stat_counter_id);
```

Parameters

unit	(IN) Unit number.
intf_id	(IN) Interface ID of a egress L3 object.
num_counters	(INOUT) Number of counters attached to the interface id
stat_counter_id	(OUT) Array of stat counter ids associated with the interface id

Returns

BCM_E_XXX	This API gets array of stat counter ids associated with the given MPLS tunnel interface. It can also provide the number of counter indexes associated so provide the number of counter indexes associated.

OAM SD/SF DETECTION OBJECTS

new Data types added.

- `bcm_oam_sd_sf_detection_t` is used to refer to an OAM signal degradation/signal fault object
- `bcm_oam_y_1711_alarm_t` is used to refer to an OAM `y_1711` alarm object

OAM group info structure - new flags added.

Table 55: OAM Group Structure Flag Definitions

BCM_OAM_GROUP_TYPE_BHH	***Indicating that this group belongs to BHH endpoint***
`BCM_OAM_GROUP_FLEXIBLE_MAID_48_BYTE`	fully flexible 48B MAID format

Table 56: OAM Group Structure Flag Definitions

Flag	***Description***
`BCM_OAM_GROUP_TYPE_BHH`	Indicating that this group belongs to BHH endpoint
`BCM_OAM_GROUP_FLEXIBLE_MAID_48_BYTE`	fully flexible 48B MAID format

OAM endpoint info structure. This is equivalent to an 802.1ag Maintenance Endpoint (MEP): new members added.

```
typedef struct bcm_oam_endpoint_info_s {
...
uint32 ccm_tx_update_lm_counter_base_id[BCM_OAM_LM_COUNTER_MAX]; /* Base LM
counter id to be incremented by Tx CCM packets */
    uint32 ccm_tx_update_lm_counter_offset[BCM_OAM_LM_COUNTER_MAX]; /* Offset
to the Base LM counter Id to be incremented by Tx CCM packets */
   uint8 ccm_tx_update_lm_counter_action[BCM_OAM_LM_COUNTER_MAX]; /* LM Counter
action (type bcm_oam_lm_counter_action_t) to be used by TX CCM packets */
    uint8 ccm_tx_update_lm_counter_size; /* Number of LM counters to be
incremented by Tx CCM packets */
    uint32 session_id;                  /* OAM session id for packet processing
                                           in BTE. In FP based OAM - This will
                                           indicate flex counter base ID */
    uint8 session_num_entries;          /* Number of entries that can result in
                                           same session ID. In FP based OAM -
                                           This will indicate number of flex
                                         counter entries corresponding to same
                                           OAM session */
    uint8 lm_count_profile;              /* LM count profile for this endpoint.
                                           It will be 1 or 0 for selecting one
                                         of the two OAM LM count profiles. */
    uint8 mpls_exp;                     /* EXP on which BHH will be running */

} bcm_oam_endpoint_info_t;
```

Table 57: OAM Endpoint2 Structure Flag Definitions

BCM_OAM_ENDPOINT2_MPLS_INGRESS_ONLY	***Endpoint with only an ingress entry. Applicable only for endpoints of type bcmOAMEndpointTypeBHHMPLS, bcmOAMEndpointTypeBHHPWE***
BCM_OAM_ENDPOINT2_UPDATE_COUNTER_ACTION	Use along with BCM_OAM_ENDPOINT_REPLACE to update the just LM counters updated by Tx CCM packets
BCM_OAM_ENDPOINT2_MPLS_EGRESS_ONLY	Endpoint with only an egress entry. Applicable only for endpoints of type bcmOAMEndpointTypeBHHMPLS, bcmOAMEndpointTypeBHHPWE
BCM_OAM_ENDPOINT_FLAGS2_MATCH_SINGLE_TAGGED_OUTER_VLAN	Selection of MEP look-up based on Single outer tagged packets only

Table 58: OAM Event Types - new events

Event type	***Description***
bcmOAMEventGroupMACStatus	Interface or Port Status defect detected in a group
bcmOAMEventGroupCCMxconClear	All xcon have been cleared in a group
bcmOAMEventGroupCCMErrorClear	All endpoints in group have CCM Error cleared
bcmOAMEventGroupRemoteClear	All endpoints in group have remote defect cleared
bcmOAMEventGroupCCMTimeoutClear	All endpoints in group have CCM timeout cleared
bcmOAMEventGroupMACStatusClear	All endpoints in group have MAC Status defect cleared
bcmOAMEventBHHCCMTimeoutClear	Timeout has been cleared on BHH CCM endpoint
bcmOAMEventBhhPmCounterRollover	Counter rollover event for BHH OAM PM processed stats
bcmOAMEventEthLmDmPmCounterRollover	Counter rollover event for ETH LM/DM OAM PM processed stats
bcmOAMEventMplsLmDmPmCounterRollover	Counter rollover event for MPLS LM/DM OAM PM processed stats

Table 59: OAM Control Formats - new Controls added

Control Format Type	***Description***
bcmOamControlSrcMacLsbReserve	Reserve Src-Mac LSB value to be used for OAM endpoints (resource shared with BFD endpoints)
bcmOamControlSrcMacLsbUnReserve	Unreserve Src-Mac LSB value (resource shared with BFD endpoints)
bcmOamControlLmPcpCounterRangeMax	lm_counter_base_ids between Max and Min will count by priority
bcmOamControlLmPcpCounterRangeMin	lm_counter_base_ids between Max and Min will count by priority
bcmOamControlOampPuntPacketIntPri	Bits [1:0] of given value determine the color (DP), bits [4:2] determine the Traffic Class
bcmOamControlOampCCMWeight	Given may be between 0 and 6
bcmOamControlOampSATWeight	Given may be between 0 and 6
bcmOamControlOampResponseWeight	Given may be between 0 and 6
bcmOamControlCount	Always Last. Not a usable value.

```
/* OAM delay object. */
typedef struct bcm_oam_delay_s {
...
uint32 dm_tx_update_lm_counter_base_id [BCM_OAM_LM_COUNTER_MAX]; /* Base LM
counter id to be incremented by Tx DM packets */
   uint32 dm_tx_update_lm_counter_offset[BCM_OAM_LM_COUNTER_MAX]; /* Offset to
the Base LM counter Id to be incremented by Tx DM packets */
```

```
   uint32 dm_tx_update_lm_counter_byte_offset[BCM_OAM_LM_COUNTER_MAX]; /* Byte
offset in the Tx DM packet from which to increment the LM byte counter */
    uint8 dm_tx_update_lm_counter_size; /* Number of LM counters to be
incremented by Tx DM packets */
} bcm_oam_delay_t;



/* PM stats structure */
typedef struct bcm_oam_pm_stats_s {
    ...

 uint32 flags;                        /* Flags related to PM processed stats */
} bcm_oam_pm_stats_t;

/* OAM loopback discovery record. */
typedef struct bcm_oam_loopback_discovery_record_s {
...
    uint16 name;            /* Discovered MEP ID for XGS devices */
} bcm_oam_loopback_discovery_record_t;


/* OAM loopback object. */
typedef struct bcm_oam_loopback_s {
...
uint32 lb_tx_update_lm_counter_base_id [BCM_OAM_LM_COUNTER_MAX]; /* Base LM
counter id to be incremented by Tx LB packets */
   uint32 lb_tx_update_lm_counter_offset[BCM_OAM_LM_COUNTER_MAX]; /* Offset to
the Base LM counter Id to be incremented by Tx LB packets */
    uint8 lb_tx_update_lm_counter_size; /* Number of LM counters to be
incremented by Tx LB packets */
    bcm_cos_t int_pri;                  /* Optional: priority fields on ITMH
                                           header */
    uint8 pkt_pri;                      /* Optional: priority fields on outer
                                            VLAN (PCP + DEI). -1 will ignore this
                                           field */
    uint8 inner_pkt_pri;                /* Optional: priority fields on inner
                                           VLAN (PCP + DEI) , when applicable.
                                           -1 will ignore this field */
} bcm_oam_loopback_t;
```

Table 60: OAM Loopback Flag Definitions

Flag	*Description*
BCM_OAM_LOOPBACK_PERIOD_IN_PPS	Default is milliseconds. flag may be used for packet per second.

```
/* OAM AIS (alarm indication signal) object. */
typedef struct bcm_oam_ais_s {
...
uint8 pkt_pri;                  /* Egress marking for outgoing AIS messages. */
    uint8 drop_eligible;            /* Drop Eligibility for AIS. */
```

```
} bcm_oam_ais_t;

/* OAM TST rx object. */
typedef struct bcm_oam_tst_rx_s {
...
  uint64 rx_count;
    uint64 invalid_tlv_count;          /* Count of invalid packets received
                                       (incorrect TLV) */
    uint64 out_of_sequence;          /* Count of out of sequence packets */
} bcm_oam_tst_rx_t;

/* OAM SD/SF detection object. */
typedef struct bcm_oam_sd_sf_detection_s {
    bcm_oam_endpoint_t mep_id;          /* Endpoint ID of Local MEP. */
    bcm_oam_endpoint_t   rmep_id;       /* Endpoint ID of Remote MEP. */
    uint16 sd_set_threshold;           /* threshold of signal degradation
indication */
   uint16 sf_set_threshold;          /* threshold of signal fault indication */
   uint16 sd_clear_threshold;       /* threshold of signal degradation clear */
    uint16 sf_clear_threshold;         /* threshold of signal fault clear */
    uint8 flags;                     /* flags use to set alert method, supress
and others */
    uint8 sd_indication;               /* signal degradation indication */
    uint8 sf_ indication;              /* signal fault indication */
    uint8 sliding_window_length;      /* size of the sliding window (1-256) */
   uint16 sliding_window_cnt[BCM_OAM_MAX_NUM_SLINDING_WINDOWS];        /* 256
counters of per-period packets(read only)*/
    uint16 sum_cnt;                   /*Sum of the sliding window packets*/
} bcm_oam_sd_sf_detection_t;
```

Table 61: OAM SD/SF detection Flag Definitions

Flag	*Description*
BCM_OAM_SD_SF_FLAGS_ALERT_METHOD	If set, issue event on signal degradation/signal fault set, otherwise issue alert on state change
BCM_OAM_ SD_SF_ FLAGS_ALERT_SUPRESS	Suppress alert event generation

RETRIEVE FAULTS PER ENDPOINT, ASSOCIATED WITH EACH ENDPOINT ID.

```
/* OAM Y.1711 alarm object. */
typedef struct bcm_oam_y_1711_alarm_s {
    bcm_oam_endpoint_t mep_id;   /* Endpoint ID of Local MEP. */
    bcm_oam_endpoint_t  rmep_id;  /* Endpoint ID of Remote MEP. */
    uint8 d_excess_threshold;           /* used for Y.1711 defects*/
    uint8 indicator_low_threshold;     /* Y.1711 indicator control low
threshold*/
    uint8 indicator_high_threshold;    /* Y.1711 indicator control high
threshold*/
    uint8 flags;                              /* flags use to set alert method
and others */
    uint8 d_excess;               /* excess detect */
    uint8 d_mismatch;            /* mismatch detect */
    uint8 d_mismerge;            /* mismerge detect */
} bcm_oam_y_1711_alarm_t;
```

Table 62: OAM Y.1711 alarm Flag Definitions

Flag	*Description*
BCM_OAM_Y_1711_ FLAGS_ALERT_METHOD	If `alert_method` 1, issue event on mismerge, mismatch or excess indication set, otherwise issue event on state change

```
/* OAM endpoint action object. */
typedef struct bcm_oam_endpoint_action_s {
...
    bcm_gport_t destination2;          /* Another destination when required for
                                           example: MIP requires 2 destination
                                           traps, one for each direction */
} bcm_oam_endpoint_action_t;

/* Fault status of each endpoint alongwith endpoint id */
typedef struct bcm_oam_endpoint_fault_s {
    bcm_oam_endpoint_t endpoint_id; /* Endpoint id assocaietd with the faults
                                       bitmap */
    uint32 faults;                 /* Faults bitmap for the endpoint */
} bcm_oam_endpoint_fault_t;
```

---

bcm_oam_endpoint_faults_multi_get

Retrieve all faults for all valid end points up to a maximum of "max_endpoints" for a given protocol

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_faults_multi_get(
        int unit,
        uint32 flags,
```

```
bcm_oam_protocol_type_t endpoint_protocol,
 uint32 max_endpoints,
bcm_oam_endpoint_fault_t *faults,
 uint32 *endpoint_count);
```

Parameters

unit	(IN) BCM device number
flags	(IN) Flags is kept unused as of now. This is for any future enhancement
endpoint_protocol	(IN) Protocol type of the endpoints to retrieve faults
max_endpoints	(IN) Number of max endpoint for the protocol
faults	(OUT) Pointer to faults for all endpoints
endpoint_count	(OUT) Pointer to Number of valid endpoints with faults, filled by get API. This is the number of actual endpoints created.

Description

It takes a buffer pointed by faults with size max_endpoints multiplied by sizeof bcm_oam_endpoint_fault_t. The maximum number of endpoints is also taken. It fills up the faults bitmap and endpoint_id for each endpoint. The number of created and valid endpoints, is provided dereferencing endpoint_count taken as a pointer.

Release Note - Currently this API is supporting on Saber2, Katana, Katana2 and Tr3 endpoints associated with firmware. This API does not support HW-based endpoints in Saber2. So, In the case of ETH-OAM on Saber2, Katana, Katana2 and Tr3, this faults_get API will return as unavailable. This can be implemented if there is any customer requirement or any future chip that needs this support.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Attempt to use incorrect protocol or max_endpoints is zero
BCM_E_INTERNAL	Unable to retrieve data due to switch HW internal error or unlock error
BCM_E_UNAVAIL	Feature is unavaiable for the protocol in current switch HW

APIS FOR MANIPULATING COUNTING PROFILE FOR OAM OPCODES

bcm_oam_opcodes_count_profile_create

Create a Hardware profile entry for setting count profile of OAM opcodes

Syntax

```
#include <bcm/oam.h>
int bcm_oam_opcodes_count_profile_create(int unit,
        uint8 *lm_count_profile);
```

Parameters

`unit`	(IN) BCM device number
`lm_count_profile`	(OUT) `lm_count_profile` id that will be returned by create API and will be used for `count_profile_set`

Description

Create a new LM count profile for opcodes with all opcodes having counting disabled. The id of the profile created is returned in `lm_count_profile` variable.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_TIMEOUT`	Unable to obtain resource lock
`BCM_E_INIT`	Module not initialized
`BCM_E_INTERNAL`	Unable to retrieve data due to switch HW internal error or unlock error
`BCM_E_UNAVAIL`	Feature is unavaiable for the protocol in current switch HW

bcm_oam_opcodes_count_profile_set

Set the counting profile of opcodes for a given Loss count profile id

Syntax

```
#include <bcm/oam.h>
int bcm_oam_opcodes_count_profile_set(
        int unit,
        uint8 lm_count_profile,
        uint8 count_enable,
        bcm_oam_opcodes_t *opcodes_bitmap);
```

Parameters

unit	(IN) BCM device number
lm_count_profile	(IN) OAM lm count profile index
count_enable	(IN) Setting count enable (1) or disable (0)
opcodes_bitmap	(IN) Opcode Bitmap for setting count for opcode packets

Description

Enable/disable counting for the opcodes in the opcodes_bitmap in the profile pointed by lm_count_profile variable.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Attempt to use incorrect lm count profile value
BCM_E_INTERNAL	Unable to retrieve data due to switch HW internal error or unlock error
BCM_E_UNAVAIL	Feature is unavaiable for the protocol in current switch HW

---

bcm_oam_opcodes_count_profile_get

Get the opcodes for a given counting profile for a given Loss count profile id

Syntax

```
#include <bcm/oam.h>
int bcm_oam_opcodes_count_profile_get(
        int unit,
        uint8 lm_count_profile,
        uint8 count_enable,
        bcm_oam_opcodes_t *opcodes_bitmap);
```

Parameters

unit	(IN) BCM device number
lm_count_profile	(IN) OAM lm count profile index to get the opcodes bitmap for a given count_enable information
count_enable	(IN) Given count enable (1) or disable (0)
opcodes_bitmap	(OUT) Opcode Bitmap for the given count_enable information with the given lm_count_profile

Description

Get the opocdes in opcodes_bitmap for given count enable as part of lm_count_profile

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Attempt to use incorrect lm count profile value
BCM_E_INTERNAL	Unable to retrieve data due to switch HW internal error or unlock error
BCM_E_UNAVAIL	Feature is unavaiable for the protocol in current switch HW
BCM_E_NOT_FOUND	lm_count_profile is already deleted

bcm_oam_opcodes_count_profile_delete

Delete the count profile with the given id

Syntax

```
#include <bcm/oam.h>
int oam_opcodes_count_profile_delete(
        int unit,
        uint8 lm_count_profile);
```

Parameters

unit	(IN) BCM device number
lm_count_profile	(IN) Id of lm_count_profile that will be destroyed by this API. The counting for all opocdes will be destroyed in the profile

Description

Delete the count profile with the id lm_count_profile.The counting for all opocdes will be destroyed in the profile.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Attempt to use incorrect lm count profile value
BCM_E_INTERNAL	Unable to retrieve data due to switch HW internal error or unlock error
BCM_E_UNAVAIL	Feature is unavaiable for the protocol in current switch HW
BCM_E_NOT_FOUND	lm_count_profile is already deleted

bcm_oam_sd_sf_detection_t_init

Initialize a bcm_oam_sd_sf_detection_t structure.

Syntax

```
#include <bcm/oam.h>
void bcm_oam_sd_sf_detection_t_init(bcm_oam_sd_sf_detection_t *sd_sf_ptr);
```

Parameters

sd_sf_ptr	(OUT) Pointer to OAM SD/SF detection structure to initialize

Description

Initializes an OAM SD/SF detection structure to default values. This function must be used to initialize any OAM SD/SF detection structure before passing it to an API function.

---

bcm_oam_sd_sf_detection_add

Add an OAM SD/SF detection object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_sd_sf_detection_add(int unit,bcm_oam_sd_sf_detection_t  *sd_sf_ptr);
```

Parameters

unit	(IN) Unit number.
sd_sf_ptr	(IN) The OAM SD/SF detection object to add

Description

Add an OAM SD/SF detection object

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

---

bcm_oam_sd_sf_detection_get

Get an OAM SD/SF detection object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_sd_sf_detection_get(int unit, bcm_oam_sd_sf_detection_t  *sd_sf_ptr);
```

Parameters

`unit`	(IN) Unit number.
`sd_sf_ptr`	(IN) The OAM SD/SF detection object to get

Description

Get an OAM SD/SF detection object

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_INIT`	BCM unit's OAM subsystem not initialized
`BCM_E_NOT_FOUND`	Indicated entry does not exist or is not a SD/SF detection entry
`BCM_E_PARAM`	Invalid parameter or `NULL` pointer parameter
`BCM_E_XXX`	

---

bcm_oam_sd_sf_detection_delete

Delete an OAM SD/SF detection object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_sd_sf_detection_delete(int unit, bcm_oam_sd_sf_detection_t
*sd_sf_ptr);
```

Parameters

`unit`	(IN) Unit number.
`sd_sf_ptr`	(IN) The OAM SD/SF detection object to delete

Description

Delete an OAM SD/SF detection object

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_NOT_FOUND	Indicated entry does not exist or is not a SD/SF detection entry
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

bcm_oam_y_1711_alarm_t_init

Initialize a bcm_oam_y_1711_alarm_t structure.

Syntax

```
#include <bcm/oam.h>
void bcm_oam_ y_1711_alarm_t_init(bcm_oam_y_1711_alarm_t *alarm_ptr);
```

Parameters

alarm_ptr	(OUT) Pointer to OAM y_1711 alarm structure to initialize

Description

Initializes an OAM y_1711 alarm structure to default values. This function must be used to initialize any OAM y_1711 alarm structure before passing it to an API function.

bcm_oam_y_1711_alarm_add

Add an OAM y_1711 alarm object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_y_1711_alarm_add(int unit,bcm_oam_y_1711_alarm_t *alarm_ptr);
```

Parameters

unit	(IN) Unit number.
alarm_ptr	(IN) The OAM y_1711 alarm object to add

Description

Add an OAM y_1711 alarm object

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

---

bcm_oam_y_1711_alarm_get

Get an OAM y_1711 alarm object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_y_1711_alarm_get(int unit, bcm_oam_y_1711_alarm_t *alarm_ptr);
```

Parameters

unit	(IN) Unit number.
alarm_ptr	(IN) The OAM SD/SF detection object to get

Description

Get an OAM y_1711 alarm object

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_NOT_FOUND	Indicated entry does not exist or is not a y_1711 alarm entry
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

---

bcm_oam_y_1711_alarm_delete

Delete an OAM y_1711 alarm object

Syntax

```
#include <bcm/oam.h>
int
bcm_oam_y_1711_alarm_delete(int unit, bcm_oam_y_1711_alarm_t *alarm_ptr);
```

Parameters

unit	(IN) Unit number.
alarm_ptr	(IN) The OAM y_1711 alarm object to delete

Description

Delete an OAM y_1711 alarm object

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_NOT_FOUND	Indicated entry does not exist or is not a y_1711 alarm entry
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

---

bcm_oam_endpoint_egress_intf_egress_attach

Attach an egress interface to a MPLS LSP endpoint

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_egress_intf_egress_attach(int unit,
                                               bcm_oam_endpoint_t endpoint_id,
                                               bcm_if_t egress_intf);
```

Parameters

unit	(IN) BCM device number.
endpoint_id	(IN) ID of the endpoint to which you would like to associate the egress interface
egress_intf	(IN) Egress interface which needs to be associated with the endpoint

Description

This API helps in adding multiple egress interfaces to a single MPLS LSP OAM endpoint endpoint.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Invalid parameter passed
BCM_E_INTERNAL	Unable to release resource lock / Failed to read or write register

bcm_oam_endpoint_egress_intf_egress_attach_get

Get the list of Egress interfaces attached to MPLS LSP endpoint

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_egress_intf_egress_attach_get(int unit,
                                                   bcm_oam_endpoint_t endpoint_id,
                                                   int max_count,
                                                   bcm_if_t *egress_intf,
                                                   int *count);
```

Parameters

unit	(IN) BCM device number.
endpoint_id	(IN) ID of the endpoint for which you would like to get the associated egress interfaces
max_count	(IN) Number of elements in an array which is passed in the egress_intf parameter
egress_intf	(OUT) List of egress interfaces that are associated to the endpoint
count	(OUT) Actual number of egress interfaces which is stored in the egress_intf array

Description

This API is used to get multiple egress interfaces associated with a single MPLS LSP OAM endpoint.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Invalid parameter passed
BCM_E_INTERNAL	Unable to release resource lock / Failed to read or write register

bcm_oam_endpoint_egress_intf_egress_detach

OAM Miscellaneous defines

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_egress_intf_egress_detach(int unit,
                                               bcm_oam_endpoint_t endpoint_id,
                                               bcm_if_t egress_intf);
```

Parameters

unit	(IN) BCM device number.
endpoint_id	(IN) ID of the endpoint from which you would like to dis-associate the egress interface
egress_intf	(IN) Egress interface which needs to be dis-associated from the endpoint

Description

This API helps in removing the association of egress interfaces from the MPLS LSP OAM endpoint.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_TIMEOUT	Unable to obtain resource lock
BCM_E_INIT	Module not initialized
BCM_E_PARAM	Invalid parameter passed
BCM_E_NOT_FOUND	Passed egress interface is not associated with the endpoint
BCM_E_INTERNAL	Unable to release resource lock / Failed to read or write register

Table 63: Miscellaneous OAM defines

Define Name	*Description*
BCM_OAM_MAX_NUM_SLINDING_WINDOWS	The max numbers of sliding windows

bcm_oam_protection_packet_header_get

Get the packet header of the protection packets injected by the OAMP

Syntax

```
#include <bcm/oam.h>
int bcm_oam_protection_packet_header_get(
    int unit,
    bcm_pkt_blk_t *packet_header);
```

Parameters

unit	(IN) Unit number.
packet_header	(OUT) The packet header of the protection packet

Description

Get the packet header of the protection packets injected by the OAMP.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

---

bcm_oam_protection_packet_header_set

Set the packet header on the protection packets injected by the OAMP

Syntax

```
#include <bcm/oam.h>
int bcm_oam_protection_packet_header_set(
    int unit,
    bcm_pkt_blk_t *packet_header);
```

Parameters

unit	(IN) Unit number.
packet_header	(IN) The packet header of the protection packet

Description

Set the packet header on the protection packets injected by the OAMP.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_INIT	BCM unit's OAM subsystem not initialized
BCM_E_PARAM	Invalid parameter or NULL pointer parameter
BCM_E_XXX	

POLICER

Table 64: Policer Flags

Name	Purpose
BCM_POLICER_PKT_ADJ_HEADER_TRUNCATE	Indicates that this Policer needs to be considerate of header truncate adjustment.

Create a policer entry. - new members added.

```
typedef struct bcm_policer_config_s {
...
   uint32 actual_ckbits_sec;         /* Granularity adjusted committed rate in
kbps or packets per second. */
    uint32 actual_ckbits_burst;       /* Granularity adjusted committed burst
size in kbits or number of packets. */
    uint32 actual_pkbits_sec;         /* Granularity adjusted peak rate in kbps
or packets per second. */
    uint32 actual_pkbits_burst;       /* Granularity adjusted peak burst size
in kbits or number of packets. */
} bcm_policer_config_t;
```

NOTE: The variables `actual_ckbits_sec, actual_ckbits_burst, actual_pkbits_sec, actual_pkbits_burst` represent actual hardware values configured based on granularity for the given packet rates. These are read-only and will be updated only after field entry install.

bcm_policer_aggregate_group_create

Allocate a block of policer entries. bcm_policer_set must be called to setup the individual policers.

Syntax

```
#include <bcm/policer.h>
int
bcm_policer_aggregate_group_create(
    int unit,
    bcm_policer_aggregate_group_info_t *info,
    bcm_policer_t *policer_id,
    int *npolicers);
```

Parameters

unit	(IN) Unit number.
info	(IN) Policer aggregate group info.
policer_id	(OUT) Base Policer ID.
npolicers	(IN/OUT) Number of Policers created including base Policer ID.

Description

Allocates the requested number of policers according to the mode, and returns the base policer ID. This API does not set up the policers in the hardware. bcm_policer_set must be called to setup the individual policers. API is supported on Jericho and above.

```
typedef struct bcm_policer_aggregate_group_info_s {
    bcm_policer_group_mode_t mode;    /* Policer mode. */
    bcm_core_t core;          /* Device core on which to allocate the policer */
    uint32 flags;               /* BCM_POLICER_* Flags. */
} bcm_policer_aggregate_group_info_t;
```

Returns

BCM_E_XXX

---

bcm_policer_group_mode_attr_selector_t_init

Initialize a policer group attribute selector structure.

Syntax

```
#include <bcm/policer.h>
void bcm_policer_group_mode_attr_selector_t_init(
         bcm_policer_group_mode_attr_selector_t *pol_group_mode_attr_selector)
```

Parameters

pol_group_mode_attr_selector	Pointer to policer group attribute selector structure to initialize.

Description

Initialize a policer group attribute selector structure.

Returns

None.

Table 65: EXTENDED_PORT_ABILITY_s

BCM_PORT_ABILITY_FEC	*Ability to support FEC*
BCM_PORT_ABILITY_FEC_REQUEST	Ability to request FEC

Table 66: Port Interface Values

BCM_PORT_IF_CR10	*Copper CR10 64B/66B interface.*
BCM_PORT_IF_KR10	Backplane KR10 64B/66B interface.
BCM_PORT_IF_LR10	Fiber LR10 64B/66B interface.
BCM_PORT_IF_LBG	Link bonding interface

PRECISION TIME PROTOCOL

bcm_ptp_clock_local_priority_get

Get PTP Clock local priority .

Syntax

```
#include <bcm/ptp.h>
int
bcm_ptp_clock_local_priority_get(
    int unit,
    bcm_ptp_stack_id_t ptp_id,
    int clock_num,
    uint8 *local_priority);
```

Parameters

unit	(IN) Unit Number
ptp_id	(IN) PTP Stack ID
clock_num	(IN) PTP Clock Number
local_priority	(OUT) PTP clock local priority value

Description

Get PTP Clock local priority value

Returns

BCM_E_NONE	Operation completed successfully

bcm_ptp_clock_local_priority_set

Set PTP Clock local priority .

Syntax

```
#include <bcm/ptp.h>
int
bcm_ptp_clock_local_priority_set(
    int unit,
    bcm_ptp_stack_id_t ptp_id,
    int clock_num,
    uint8 local_priority);
```

Parameters

`unit`	(IN) Unit Number
`ptp_id`	(IN) PTP Stack ID
`clock_num`	(IN) PTP Clock Number
`local_priority`	(IN) PTP clock local priority value

Description

Set PTP Clock local priority value

Returns

`BCM_E_NONE`	Operation completed successfully

QUALITY OF SERVICE

`bcm_qos_map_t` The type `bcm_qos_map_t` is a structure that represents a mapping between packet attributes and internal attributes. New members were added.

```
typedef struct bcm_qos_map_s {
    ...

    int inherited_dscp_exp;     /* inherited DSCP EXP value */
} bcm_qos_map_t;
```

IPSEC

`bcm_ipsec_config_t` Structure used for `bcm_ipsec_config_set` API.

```
typedef struct bcm_ipsec_config_s {
    uint32 num_of_ipsec_channels;

    uint32 num_of_bfd_channels;

} bcm_ipsec_config_t;
```

`bcm_ipsec_tunnel_info_t` Structure used for `bcm_ipsec_tunnel_add` API.

```
typedef struct bcm_ipsec_tunnel_info_s {
    uint32 context_id;          /* VRF ID */
    bcm_ip_addr_t local_ip;     /* IPV4 or IPV6 */
    bcm_ip_addr_t remote_ip;    /* IPV4 or IPV6 */
} bcm_ipsec_tunnel_info_t;
```

---

bcm_ipsec_tunnel_add

Add IPSEC tunnel

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_tunnel_add(int unit, uint32 flags, uint32 *tunnel_id,
bcm_ipsec_tunnel_info_t tunnel_info);
```

Parameters

`unit`	BCM device number
`flags`	Control flags. Currently not in use.
`tunnel_id`	Tunnel ID
`tunnel_info`	`bcm_ipsec_tunnel_info_t` structure.

Description

This API will be used to add an IPSEC Tunnel to the BRCM data plane

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_XXX`	Other errors

---

bcm_ipsec_tunnel_delete

Delete IPSEC tunnel

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_tunnel_delete(int unit, uint32 flags, uint32 tunnel_id);
```

Parameters

unit	BCM device number
flags	Control flags. Currently not in use.
tunnel_id	Tunnel ID

Description

This API will be used to delete an IPSEC Tunnel from BRCM data plane.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other errors

bcm_ipsec_algo_type_t Structure used in bcm_ipsec_sa_info_t for bcm_ipsec_sa_create API.

```
typedef enum bcm_ipsec_algo_type_e {
    bcmAES128;
    bcmAES192;
    bcmAES256;
} bcm_ipsec_algo_type_t;
```

bcm_ipsec_auth_type_t Structure used in bcm_ipsec_sa_info_t for bcm_ipsec_sa_create API.

```
typedef enum bcm_ipsec_auth_type_e {
    bcmSHA1;
    bcmSHA224;
    bcmSHA256;
    bcmAES;
} bcm_ipsec_auth_type_t;
```

bcm_ipsec_direction_t Structure used for in bcm_ipsec_sa_info_t for bcm_ipsec_sa_create API.

```
typedef enum bcm_ipsec_direction_e {
    bcmENCRYPTION;
    bcmDECRYPTION;
} bcm_ipsec_direction_t;
```

bcm_ipsec_sa_info_t Structure used for bcm_ipsec_tunnel_add API.

```
typedef struct bcm_ipsec_sa_info_s {
    uint32 initial_seq_num;
    uint32 mtu;
```

```
    uint8 fragmentation_enable;
    bcm_ip_addr_t dest_addr;            /* IPV4 or IPV6 */
    bcm_ip_addr_t src_addr;             /* IPV4 or IPV6 */
    bcm_ipsec_algo_type_t auth_algo;
    bcm_ipsec_auth_type_t encrypt_algo;
    uint16 auth_key_length;
    uint16 enc_key_length;
    uint16 aes_enc_key_length;
    uint8 encap_mode;
    bcm_ipsec_direction_t sa_direction;
} bcm_ipsec_sa_info_t;
```

bcm_ipsec_sa_create

Add IPSEC SA

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_sa_create(int unit, uint32 flags, uint32 *spi_id,
bcm_ipsec_sa_info_t *sa_info);
```

Parameters

`unit`	BCM device number
`flags`	Control flags. Currently not in use.
`spi_id`	SA ID
`sa_info`	`bcm_ipsec_sa_info_t` structure

Description

This API will be used to add an IPSEC SA to the BRCM data plane

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_XXX`	Other errors

`bcm_ipsec_sa_keys_t` Structure used for `bcm_ipsec_sa_update_key` API.

```
typedef struct bcm_ipsec_sa_keys_s {
    uint8 enc_key[BCM_IPSEC_MAX_ENCR_KEY_LEN];
    uint8 auth_key[BCM_IPSEC_MAX_AUTH_KEY_LEN];
} bcm_ipsec_sa_keys_t;
```

bcm_ipsec_sa_update_key

Add IPSEC SA Keys

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_sa_update_key(int unit, uint32 flags, uint32 spi_id,
bcm_ipsec_sa_keys_t *sa_keys);
```

Parameters

unit	BCM device number
flags	Control flags. Currently not in use.
spi_id	SA ID
sa_keys	bcm_ipsec_sa_keys_t structure

Description

This API will be used to add an IPSEC Encryption and Authentication keys to SA

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other errors

---

bcm_ipsec_sa_delete

Delete IPSEC SA

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_sa_delete(int unit, uint32 flags, uint32 spi_id);
```

Parameters

unit	BCM device number
flags	Control flags. Currently not in use.
spi_id	SA ID

Description

This API will be used to delete an IPSEC SA from BRCM data plane.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other errors

---

bcm_ipsec_sa_to_tunnel_map_set

Map IPSEC SA to tunnel

Syntax

```
#include <bcm/ipsec.h>
int bcm_ipsec_sa_to_tunnel_map_set(int unit, uint32 tunnel_id, uint32 spi_id);
```

Parameters

unit	BCM device number
tunnel_id	Control flags. Currently not in use.
spi_id	SA ID

Description

This API will be used to delete an IPSEC SA from BRCM data plane.

Returns

BCM_E_NONE	Operation completed successfully
BCM_E_XXX	Other errors

RATE LIMITING

RateLimitFlags lists the flags for the rate-limiting functions, new flags added.

SERVICE ACTIVATION TEST (SAT)

SAT CTF Report Event Data:

```
typedef struct bcm_sat_report_event_data_s {
    ...
    bcm_sat_ctf_t ctf_id;       /* SAT CTF id */
} bcm_sat_report_event_data_t;
```

Table 67: SOC Properties

Define	*Property*	*Type*	*Default*	*Description*
spn_RX_POOL_NOF_PKTS	rx_pool_nof_pkts		BCM_RX_POOL_COUNT_DEFAULT	Specifies number of packets for BCM RX buffer allocation
spn_PHY_ALT_DATAPATH_MODE	phy_alt_datapath_mode	Port	0	Set phy datapath mode (default/ alternate). Value 0 => Default Datapath (default) Value 1 => Alternate DataPath
spn_ESM_SERDES_DRIVER_CURRENT	esm_serdes_driver_current		4	ESM serdes lane TX amplitude control, default value is 4
spn_ESM_SERDES_PRE_DRIVER_CURRENT	esm_serdes_pre_driver_current		4	ESM serdes lane TX amplitude control, default value is 4
spn_ESM_SERDES_PRECURSOR_TAP	esm_serdes_precursor_tap		0	ESM serdes lane TX fir control to AFE for ESM and PentaCore only, default value is 0
spn_ESM_SERDES_MAIN_TAP	esm_serdes_main_tap		40	ESM serdes lane TX fir control to AFE for ESM and PentaCore only, default value is 40
spn_ESM_SERDES_POSTCURSOR_TAP	esm_serdes_postcursor_tap		7	ESM serdes lane TX fir control to AFE for ESM and PentaCore only, default value is 7
spn_PBMP_SKIP_DEFAULT_LLS	pbmp_skip_default_lls	String	0	Port bitmap for skipping SDK default LLS tree creation mode. In this mode, the application must create a LLS tree for each port. Note:On BCM56850/ BCM56860, the creation of default LLS trees will be skipped when port bitmap spn_PBMP_SKIP_DEFAULT_LLS is non-zero. User should enable MMU traffic for each port with bcmPortControlMmuTrafficEnable if use this mode.
spn_BCM56346_4X10_2X21	bcm56346_4X10_2x21		0	Enable 7xF.QSGMII + Flex[4x10] + 2xHGd[21] mode for BCM56346
spn_BCM56345_4X10_2X21	bcm56345_4X10_2x21		0	Enable 12xF.QSGMII + Flex[4x10] + 2xHGd[21] mode for BCM56345
spn_BCM56345_2X21	bcm56345_2x21		0	Enable 12xF.QSGMII + 2xHGd[21] mode for BCM56345
spn_PCI_CMCS_NUM	pci_cmcs_num		0	Number of CMC used by the PCI Host

Table 67: SOC Properties

Define	Property	Type	Default	Description
spn_BFD_SESSION_DOWN_EVENT_ON_CREATE	bfd_session_down_event_on_create		0	If this config property is enabled (set to 1), there would be a state change event (State would be down for the endpoint) received after the session is created if we do not receive any packets from the peer within detection_multiplier * 1 seconds. Default value is 0 (disabled)
spn_BFD_USE_ENDPOINT_ID_AS_DISCRIMINATOR	bfd_use_endpoint_id_as_discriminator		0	Enable/Disable the configuration of using endpoint index itself as local discriminator for BFD endpoints.If the config property is enabled (set to 1), any value passed in local_discriminator field in endpoint info will be ignored and endpoint index would be used instead. The default value of the config property is 0 (disabled)
spn_SYSTEM_IS_ARAD_IN_SYSTEM	system_is_arad_in_system		TRUE	Whether there is a Arad or Arad Plus device in the system.
spn_SECONDARY_IS_MULTICAST	secondary_is_multicast		FALSE	Map all incoming multicast traffic to the secondary switch
spn_ITMH_ARAD_MODE_ENABLE	itmh_arad_mode_enable		0	If set ITMH processing works in ARAD mode, itmh_programmable_mode_enable needs to be set to 0 in order for this property to work
spn_COSQ_ADMISSION_PREFERENCE	cosq_admission_preference	String		A configuration of higher preference to addmitance tests or to resource reservation. If admit tests have higher precedence, they will drop packets even if they are with in the guarenteed/ reserved range. The configuration is either for all drop precendences, or for a specific one when its number (0-3) is specified as a postfix. possiable values: ADMIT_OVER_GUARANTEE GUARANTEE_OVER_ADMIT
spn_INGRESS_CONGESTION_MANAGEMENT_STAG_MAX_ID	ingress_congestion_management_stag_max_id		0	The allowed maximum ID of ST-VSQs
spn_INGRESS_CONGESTION_MANAGEMENT_PKT_HEADER_COMPENSATION_ENABLE	ingress_congestion_management_pkt_header_compensation_enable		1	Enable header-compansation
spn_INGRESS_CONGESTION_MANAGEMENT_TM_PORT_MAX_ID	ingress_congestion_management_tm_port_max_id		255	the maximal ID for TM-ports, if 0 then the any-value is legal.
spn_TDM_EGRESS_DP	tdm_egress_dp		0	Select egress drop precedence TDM traffic.
spn_MICRO_BFD_SUPPORT_MODE	micro_bfd_support_mode	String	NONE	Enable support for micro bfd May be set to NONE, IPv4, IPv6 or IPv4_AND_IPv6

Table 67: SOC Properties

Define	Property	Type	Default	Description
spn_ILKN_BURST_MAX	ilkn_burst_max		256	This soc property is used to set ILKN burst max value supported values are 128 and 256
spn_ILKN_BURST_SHORT	ilkn_burst_short		32	This soc property is used to set ILKN burst short value value should be a multiplier of 32 and not bigger than ilkn_burst_max /2
spn_HQOS_MAPPING_ENABLE	hqos_mapping_enable		0	This soc property is used to enable mapping of many system ports to a single destination (modport) Relevant only in DIRECT mapping mode value should be 0 or 1
spn_BHH_BASE_ENDPOINT_ID	bhh_base_endpoint_id		512	This will be used to specify starting endpoint ID for BHH protocol on FP based OAM devices Default is 512
spn_BHH_BASE_GROUP_ID	bhh_base_group_id		256	This will be used to specify starting OAM group ID for BHH protocol on FP based OAM devices Default is 256
spn_PHY_MLD_MAP	phy_mld_map	Port	0xff9876543210	This per-physical port property is used to specify the mode-lane configuration. For BCM56860 based devices, Each nibble-index is the TSC lane number (0..11) and the nibble value is the 100GE port lane number(0..9). A nibble value of "f" indicates the TSC lane is not used. Supported 100GE lane distributions: - TSC 100g 4-4-2 mode: phy_mld_map{x} = 0xff9876543210 - TSC 100g 3-4-3 mode: phy_mld_map{x} = 0xf9876543f210 - TSC 100g 2-4-4 mode: phy_mld_map{x} = 0x98765432ff10
spn_PHY_AN_CORE_NUM	phy_an_core_num	Port	1	This per-physical port property specifies the master core number that is used for autonegotiation. Valid values are 0, 1, 2.
spn_FIELD_SCACHE_SIZE	field_scache_size		0	This will be used to specify warmboot scache size needed for field module Default value is 0
spn_FP_COMPRESSION_ENABLE	fp_compression_enable		1	This will be used to specify compression feature enable/disable for Tomahawk IFP Default value is 1 which means enabled.
spn_BHH_ENCAP_MAX_LENGTH	bhh_encap_max_length		76	The max length in bytes of a BHH PDU's encapsulation Default value is 76 which includes OLP Tx header(34) + L2 header(22) + 4 MPLS labels (16) + ACH header(4).
spn_MPLS_ENCAP_INVALID_VALUE	mpls_encap_invalid_value		0	Configure the default invalid value for egress mpls labels Valid values: 0 - 1048575
spn_MPLS_ENCAPSULATION_ACTION_SWAP_OR_PUSH_ENABLE	mpls_encapsulation_action_swap_or_push_enable		0	Enable MPLS egress encapsulation switch and push actions on same OutLif
spn_VPWS_TAGGED_MODE	vpws_tagged_mode		0	Enable VPWS tagged mode termination

Table 67: SOC Properties

Define	Property	Type	Default	Description
spn_IPMC_L3MCASTL2_MODE	ipmc_l3mcastl2_mode		0	In case of an IPv4 MC packet with IPMC disable allows a per RIF program selection instead of global RIF setting using the bcmSwitchL3McastL2 switch control. 0 = Disable 1 = Enable
spn_EGRESS_MEMBERSHIP_MODE	egress_membership_mode		0	Egress Membership mode. 0: VSI 1: VLAN
spn_VXLAN_TUNNEL_TERM_IN_SEM_VRF_NOF_BITS	vxlan_tunnel_term_in_sem_vrf_nof_bits		12	This soc property is used in vxlan termination according to DIP SIP VRF, using my-vtep-index, enabled using bcm886xx_vxlan_tunnel_lookup_mode = 3. It defines the number of bits available for VRF in IP tunnel termination lookup: my-vtep-index, SIP, VRF lookup. Max value is number of VRF in the device. Note that number of bits for VRFs + number of bits for my-vtep-index must be <= 15.
spn_VXLAN_TUNNEL_TERM_IN_SEM_MY_VTEP_INDEX_NOF_BITS	vxlan_tunnel_term_in_sem_my_vtep_index_nof_bits		3	This soc property is used in vxlan termination according to DIP SIP VRF, using my-vtep-index, enabled using bcm886xx_vxlan_tunnel_lookup_mode = 3. It defines the number of bits available for my-vtep-index in IP tunnel termination lookup: my-vtep-index, SIP, VRF lookup. Max value is 4. Note that number of bits for VRFs + number of bits for my-vtep-index must be <= 15.
spn_MIRROR_STAMP_SYS_ON_DSP_EXT	mirror_stamp_sys_on_dsp_ext		0	This soc property is used to enable stamping of the DSP-Ext field in mirror/snooped packets with the system port DSP. FTMH Extension must be enabled. Default value is 0 which means disabled.
spn_TRUNCATE_DELTA_IN_PP_COUNTER	truncate_delta_in_pp_counter		0	This soc property is used to Enable/Disable truncate (IRPP editing) for counter processor Default value is 0 which means disabled.
spn_RCY_CHANNELIZED_SHARED_CONTEXT_ENABLE	rcy_channelized_shared_context_enable		0	This soc property is used to enable allocating seperate recycling termination context (in pp port) for each channel on a channelized interface which should be recycled
spn_MPLS_EGRESS_LABEL_EXTENDED_ENCAPSULATION_MODE	mpls_egress_label_extended_encapsulation_mode		0	Enable MPLS extended encapsulation.
spn_LINK_BONDING_ENABLE	link_bonding_enable		1	link bonding enable
spn_LB_BUFFER_SIZE	lb_buffer_size		128	The size of data buffer in the OCB for link bonding. Allowed values: 0/2/4/6/8 MB. Default: 8 MBytes.
spn_LB_MODEM_FIFO_SIZE	lb_modem_fifo_size	Port	0	Number of FIFO per Modem
spn_LB_LBG_FIFO_SIZE	lb_lbg_fifo_size	Port	0	Number of FIFO per Link Bonding Group

Table 67: SOC Properties

Define	Property	Type	Default	Description
spn_NIV_ENABLE	niv_enable	String	1	This soc_property is used to enable/disable NIV feature at init time. Example: niv_enable = 1 - NIV will be enabled at init time if the device supports NIV. This is the default value of the config property. niv_enable = 0 - NIV will be disabled at init time if the device supports NIV.
spn_OAM_CCM_MAX_GROUPS	oam_ccm_max_groups		256	This soc property is used to specify number of Max Groups for CCM EApp Default value is 256.
spn_OAM_CCM_MAX_MEPS	oam_ccm_max_meps		256	This soc property is used to specify number of Max MEPs for CCM EApp Default value is 256.
spn_EXT_IP4_DOUBLE_CAPACITY_FWD_TABLE_SIZE	ext_ip4_double_capacity_fwd_table_size			External IPv4 double capacity forward table size
spn_ENHANCED_FIB_SCALE_PREFIX_LENGTH	enhanced_fib_scale_prefix_length		0	This soc_property is used to select a specific route prefix length that can be stored in the LEM. Possible values: 0(disabled)/4/8/12/16/20/24/28.
spn_OAM_ENABLE	oam_enable		0	Enable OAM features CFM and Y1731
spn_BFD_ENABLE	bfd_enable		0	Enable BFD features
spn_DMA_DESC_AGGREGATOR_CHAIN_LENGTH_MAX	dma_desc_aggregator_chain_length_max		0	This soc property is the max number of descriptors in a single chain.
spn_DMA_DESC_AGGREGATOR_BUFF_SIZE_KB	dma_desc_aggregator_buff_size_kb		0	This soc property is the total size of the DMA memory double-buffer.
spn_DMA_DESC_AGGREGATOR_TIMEOUT_USEC	dma_desc_aggregator_timeout_usec		0	This soc property is the timeout between the creation of a new descriptor chain and its commit to HW.
spn_DMA_DESC_AGGREGATOR_ENABLE_SPECIFIC	dma_desc_aggregator_enable_specific		0	This soc property enables descriptor DMA for a specific memory, _MEM suffix.
spn_BHH_CONSOLIDATED_FINAL_EVENT	bhh_consolidated_final_event		0	During occurrence of conflicting events - TimeOut and State change, only final event(Either TimeOut (or) State change) will be communicated to SDK. Other events will not be affected by this.

MULTI-DEVICE STACK CONTROL

Set/get relevant control value per module id

Table 68: bcm_stk_module_control_t

enum	*Description*
bcmStkModuleMHPriorityUse	Assign the priority and color to the MH header and utilize the same for queuing purposes. This will be used for HG Proxy and two-pass loopback applications. For the HG Proxy, the color/priority can be assigned at the Ingress device and consumed at the Egress device for queuing purposes, in this case this control needs to be set at both the devices. For a two-pass loopback solution, the priority/color can be assigned in pass one and used in pass two, in such a case this control will only be set on the local device. This control is only applicable on these XGS devices: bcm5634x, bcm5663x, bcm5644x, bcm5645x, bcm5684x, bcm5664x, bcm5685x, bcm5686x and bcm5696x.

STATISTICS

```
/* Ingress and Egress Statistics Accounting Objects */
typedef enum bcm_stat_object_e {
...
StatObjectIngGport = 38,           /* Ingress Gport Object */
    StatObjectIngVlanXlateSecondLookup = 39, /* Ingress Vlan Translate Second
Lookup
                                                Object */
    StatObjectEgrVlanXlateSecondLookup = 40, /* Egress Vlan Translate Second
Lookup
                                                Object */
    bcmStatObjectIngMplsSwitchSecondLabel = 41 /* Mpls Entry Lookup object for
                                                  the Second Label*/
    bcmStatObjectEgrMplsTunnelSecondLabel = 42 /* Egress Mpls Tunnel Second
                                                  Label Object */
    bcmStatObjectMaxValue = 43         /* This should be the maximum value
                                          defined in this enum */
} bcm_stat_object_t;
```

Table 69: Flex Counter Index Information

Group Mode	*Ingress Counter Index*	*Egress Counter Index*
bcmStatGroupModeTrafficType	0) L2UC PKT , L2DLF PKT, KNOWN L3UC PKT , UNKNOWN L3UC PKT 1) KNOWN L2MC PKT,UNKNOWN L2MC PKT 2) L2BC PKT	0) Unicast, 1) Multicast endtable

bcm_stat_lif_counting_profile_set bcm_stat_lif_counting_profile_get

Configure a LIF-counting-profile - define it's range - and what counter-command collects it.

Syntax

```
#include <bcm/stat.h>
/*
 * Mapping a LIF-couting-profile to a counting-source and
 * LIF-counting-range.
 */
int bcm_stat_lif_counting_profile_set(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_t *lif_counting);
```

Description

```
/* LIF Counting Source. */
typedef struct bcm_stat_lif_counting_source_s {
    bcm_stat_counter_source_type_t type;
```

```
    int command_id;
} bcm_stat_lif_counting_source_t;

/* LIF Counting Configuration. */
typedef struct bcm_stat_lif_counting_s {
    bcm_stat_lif_counting_source_t source;
    bcm_stat_counter_lif_counting_range_t range;
} bcm_stat_lif_counting_t;

#define BCM_STAT_LIF_COUNTING_PROFILE_NONE  -1        /* An indicator for a LIF
                                                         to be uncounted */

#ifndef BCM_HIDE_DISPATCHABLE

/*
 * Mapping a LIF-couting-profile to a counting-source and
 * LIF-counting-range.
 */
extern int bcm_stat_lif_counting_profile_set(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_t *lif_counting);

/*
 * Get a mapping of a LIF-couting-profile to a counting-source and
 * LIF-counting-range.
 */
extern int bcm_stat_lif_counting_profile_get(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_t *lif_counting);
```

---

bcm_stat_lif_counting_stack_level_priority_set bcm_stat_lif_counting_stack_level_priority_get

Set A priority of a LIF in a stack

Syntax

```
#include <bcm/stat.h>
int bcm_stat_lif_counting_stack_level_priority_set(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_source_t *source,
    bcm_stat_counter_lif_stack_id_t *lif_stack_level,
    int priority);
```

Description

```
/*
```

```
 * Set a priority for a LIF-stack-level: in case two LIF from the same
 * range are found on the same packet.
 */
extern int bcm_stat_lif_counting_stack_level_priority_set(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_source_t *source,
    bcm_stat_counter_lif_stack_id_t *lif_stack_level,
    int priority);

/*
 * Get a priority for a LIF-stack-level: in case two LIF from the same
 * range are found on the same packet.
 */
extern int bcm_stat_lif_counting_stack_level_priority_get(
    int unit,
    uint32 flags,
    int lif_counting_profile,
    bcm_stat_lif_counting_source_t *source,
    bcm_stat_counter_lif_stack_id_t *lif_stack_level,
    int *priority);
```

SUBPORT CONFIGURATION

bcm_subport_gport_modport_get

Given a subport gport, returns the corresponding module and port associated with that subport.
This is only applicable on devices BCM5645x, BCM5626x, BCM5646x, BCM5686x, BCM5656x and BCM5676x that support CoE applications like Subtag and LinkPhy.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_gport_modport_get(int unit,
                              bcm_gport_t subport_gport, bcm_module_t *module,
bcm_port_t *port);
```

Parameters

`unit`	(IN) BCM Device number
`subport_gport`	(IN) Subport gport returned when a subport is added to a subport group.
`module`	(OUT) The module associated with the subport
`port`	(OUT) The port associated with the subport

Description

Given a subport gport, returns the corresponding module and port associated with that subport. This gport value is a OUT parameter from the `bcm_subport_port_add` API. The mod,port returned from this API will be used in a stacked system to address the subtended ports from remote devices.

Returns

`BCM_E_XXX`

SWITCH CONTROL - NEW VALUES ADDED

Table 70: Switch Type Values

Value	Description	Arg Value
bcmSwitchHashSetTflowMode	BCM56960 If on, hash selection of b0/b1 is allowed regardless of th_tflow_enabled state.	TRUE/FALSE
bcmSwitchVpnDetachPorts	Set VPN detach mode to either detach ports that are attached to the VSI or ignore them. Default: 1 - Detach attached ports.	TRUE/FALSE
bcmSwitchClassL3L2Marking	Set the In-Lif-Profile->Resolved-In-Lif-Profile map (In-Lif-Profile mapping).	0x0 - 0x3
bcmSwitchL3L2MarkingMode	Set the DP map mode.	0x0 - 0x1
bcmSwitchSTPDisabledFieldBypass	If the ingress port's spanning tree state is in Disabled state, then the FP is disabled when this control is set to TRUE.	TRUE/FALSE
bcmSwitchHashSctpL4Port	Activate ecmp hashing for SCTP	TRUE/FALSE
bcmSwitchGlobalTpidEnable	Enable or Disable global TPIDs(BCM88670 and above only).	TRUE/FALSE
bcmSwitchECMPResilientHashCombine	Enable combination of both regular and resilient hashing for ECMP key (BCM88670_B0 only).	TRUE/FALSE

Table 71: BCM switch flex hash type flags

Name	Purpose
BCM_SWITCH_HASH_FIELD_TYPE_UDF	Field type is UDF chunk.

The bcm_switch_hash_entry_config_t structure describes the hash entry configurations.

```
typedef struct bcm_switch_hash_entry_config_s {
    ...
bcm_udf_id_t udf_id;                /* udf id when field type is udf chunk. */
} bcm_switch_hash_entry_config_t;

typedef enum bcm_switch_pkt_trace_drop_reason_e {
    bcmSwitchPktTraceNoDrop = 0,
    bcmSwitchPktTraceDropReasonInternal = 1,
    bcmSwitchPktTraceDropReasonMplsLabelLookupMiss = 2,
    bcmSwitchPktTraceDropReasonMplsInvalidAction = 3,
    bcmSwitchPktTraceDropReasonMplsInvalidPayload = 4,
    bcmSwitchPktTraceDropReasonMplsTtlCheckFail = 5,
    bcmSwitchPktTraceDropReasonMplsInvalidControlWord = 6,
    bcmSwitchPktTraceDropReasonL2greSipLookupMiss = 7,
    bcmSwitchPktTraceDropReasonL2greVpnLookupMiss = 8,
    bcmSwitchPktTraceDropReasonL2greTunnelError = 9,
```

```
    bcmSwitchPktTraceDropReasonVxlanSipLookupMiss = 10,
    bcmSwitchPktTraceDropReasonVxlanVnidLookupMiss = 11,
    bcmSwitchPktTraceDropReasonVxlanTunnelError = 12,
    bcmSwitchPktTraceDropReasonVlanNotValid = 13,
    bcmSwitchPktTraceDropReasonIngressPortNotInVlanMember = 14,
    bcmSwitchPktTraceDropReasonTpidMismatch = 15,
    bcmSwitchPktTraceDropReasonIpv4ProtocolError = 16,
    bcmSwitchPktTraceDropReasonHigigLoopback = 17,
    bcmSwitchPktTraceDropReasonHigigMirrorOnly = 18,
    bcmSwitchPktTraceDropReasonHigigUnknownHeader = 19,
    bcmSwitchPktTraceDropReasonHigigUnknownOpcode = 20,
    bcmSwitchPktTraceDropReasonLagFailLoopback = 21,
    bcmSwitchPktTraceDropReasonL2SrcEqualL2Dst = 22,
    bcmSwitchPktTraceDropReasonIpv6ProtocolError = 23,
    bcmSwitchPktTraceDropReasonNivVntagPresent = 24,
    bcmSwitchPktTraceDropReasonNivVntagNotPresent = 25,
    bcmSwitchPktTraceDropReasonNivVntagFormat = 26,
    bcmSwitchPktTraceDropReasonTrillErrorFrame = 27,
    bcmSwitchPktTraceDropReasonBpdu = 28,
    bcmSwitchPktTraceDropReasonBadUdpChecksum = 29,
    bcmSwitchPktTraceDropReasonTunnlDecapEcnError = 30,
    bcmSwitchPktTraceDropReasonIpv4HeaderError = 31,
    bcmSwitchPktTraceDropReasonIpv6HeaderError = 32,
    bcmSwitchPktTraceDropReasonParityError = 33,
    bcmSwitchPktTraceDropReasonRpfCheckFail = 34,
    bcmSwitchPktTraceDropReasonCount = 35
} bcm_switch_pkt_trace_drop_reason_t;

typedef struct bcm_switch_pkt_trace_info_s {
    ...
        uint32 pkt_trace_drop_reason[BCM_SWITCH_PKT_TRACE_DROP_REASON_MAX];
...
} bcm_switch_pkt_trace_info_t;
@desc pkt_trace_info_t is not required to be initialized before it is used in
bcm_switch_pkt_trace_info_get. New members were added.
```

Table 72: Flags for bcm_switch_lif_property_set/get - Correlates with bcmLifPropertyEEDBBankPhase

Name	*Description*
BCM_SWITCH_LIF_PROPERTY_EEDB_PHASE_5	EEDB bank phase 5.
BCM_SWITCH_LIF_PROPERTY_EEDB_PHASE_6	EEDB bank phase 6.

bcm_switch_switch_dram_vendor_info_get

Get DRAM vendor info.

Syntax

```
#include <bcm/switch.h>

int bcm_petra_switch_dram_vendor_info_get(
```

```
    int unit,
    bcm_switch_dram_vendor_info_t *info);
```

Parameters

unit	(IN) BCM device number
info	(IN) Vendor Information Struct.

Description

This API returns DRAM vendor info.

Returns

```
BCM_E_xxx
```

```
typedef struct bcm_switch_dram_vendor_info_s {
       uint32 dram_density;
       uint32 fifo_depth;
       uint32 revision_id;
       uint32 manufacture_id;
} bcm_switch_dram_vendor_info_t;
```

---

bcm_switch_thermo_sensor_read

Get switch thermo sensor data

Syntax

```
#include <bcm/switch.h>

int bcm_switch_thermo_sensor_read(
    int unit,
    bcm_switch_thermo_sensor_type_t sensor_type,
    int interface_id,
    bcm_switch_thermo_sensor_t *sensor_data);
```

Parameters

`unit`	(IN) BCM device number
`sensor_type`	(IN) sensor type from enum
`interface_id`	(IN) interface ID, ignored when not relevant
`sensor_data`	(OUT) return data structure.

```
/* Thermo Sensor Struct */
typedef struct bcm_switch_thermo_sensor_s {
    int thermo_sensor_min;     /* thermo sensor minimal read */
    int thermo_sensor_max;     /* thermo sensor maximal read */
    int thermo_sensor_current;  /* thermo sensor current read */
} bcm_switch_thermo_sensor_t;

/* Thermo sensors enum */
typedef enum bcm_switch_thermo_sensor_type_e {
    bcmSwitchThermoSensorInvalid = 0,
    bcmSwitchThermoSensorDram = 1,
    bcmSwitchThermoSensorSpare1 = 2,
    bcmSwitchThermoSensorSpare2 = 3,
    bcmSwitchThermoSensorCount = 4
} bcm_switch_thermo_sensor_type_t;
```

Description

This API returns thermo sensors Data.

Returns

`BCM_E_xxx`

TIME CONFIGURATION

typedef struct `bcm_time_interface_s { ... bcm_time_spec_t ntp_offset; /* NTP Offset */ } bcm_time_interface_t;`

`BCM Time Synchronization Flags` (page 167)

Table 73: BCM Time Synchronization Flags

Name	Description
BCM_TIME_NTP_OFFSET	Get/Set the NTP Offset for Interface

to configure logging for BroadSync the following structure is used.

```
typedef struct bcm_time_bs_log_cfg_s {
    uint32 debug_mask;              /* BCM_BS_LOG_NONE:To disable
BCM_BS_LOG_SYS_ALL:To enable*/
    uint64 udp_log_mask;            /* mask for udp log */
    bcm_mac_t src_mac;              /* source mac address */
    bcm_mac_t dest_mac;             /* destination mac address */
    uint16 tpid;                    /* TPID */
    uint16 vid;                     /* vlan id */
    uint8 ttl;                      /* time-to-live */
    bcm_ip_t src_addr;              /* source ipv4 address */
    bcm_ip_t dest_addr;             /* destination ipv4 address */
    uint16 udp_port;                /* udp port number */
} bcm_time_bs_log_cfg_t;
```

Table 74: BCM BS log configure debug mask values

Name	Description
BCM_BS_LOG_NONE	Disable logs
BCM_BS_LOG_SYS_INFO	BS log level INFO
BCM_BS_LOG_SYS_WARN	BS log level Warning
BCM_BS_LOG_SYS_ERR	BS log level Error
BCM_BS_LOG_SYS_CRIT	BS log level Critical
BCM_BS_LOG_SYS_ALL	Enable all logs

bcm_time_bs_log_configure_set

Configures the BroadSync logging

Syntax

```
#include <bcm/time.h>
extern int bcm_time_bs_log_configure_set(int unit, bcm_time_bs_log_cfg_t
bs_log_cfg);
```

Parameters

`unit`	BCM device number
`bs_log_cfg`	BS log configuration

Description

Set the BroadSync logging configuration

Returns

`BCM_E_NONE`	- Success
`BCM_E_XXX`	- Any other error

---

bcm_time_bs_log_configure_get

Get the BroadSync logging configuration

Syntax

```
#include <bcm/time.h>
extern int bcm_time_bs_log_configure_get(int unit, bcm_time_bs_log_cfg_t
*bs_log_cfg);
```

Parameters

`unit`	BCM device number
`bs_log_cfg`	BS log configuration

Description

Gets the BroadSync logging configuration

Returns

`BCM_E_NONE`	- Success
`BCM_E_XXX`	- Any other error

TRILL TRILL MANAGEMENT

The bcm_trill_port_t structure describes a TRILL virtual port used for creating TRILL Network endpoints.

```
typedef struct bcm_trill_port_s {
    ...
int inlif_counting_profile;          /* In LIF counting profile */
    int outlif_counting_profile;         /* Out LIF counting profile */
} bcm_trill_port_t;
```

TRUNKING (LINK AGGREGATION)

bcm_trunk_spa_to_system_phys_port_map_get

map a system port aggregate (SPA) to its corresponding system physical port.

Syntax

```
#include <bcm/trunk.h>
int bcm_trunk_spa_to_system_phys_port_map_get(
    int                  unit,
    uint32               flags,
    uint32               system_port_aggregate,
    bcm_gport_t*         gport);
```

Parameters

unit	(IN) Unit number.
flags	(IN) relevant flags.
system_port_aggregate	(IN) SPA to get it pysical system port.
gport	(OUT) returned system physical port.

Description

map a system port aggregate (SPA) to its corresponding system physical port.

Returns

BCM_E_NONE	- If managed to map.
BCM_E_PARAM	- Parameter is not invalid.
BCM_E_UNAVAIL	- Feature is not supported on the device

PACKET TRANSMIT AND RECEIVE

Table 75: bcm_pkt_t Structure Description

Field	Type	Description
src_port	int16	The apparent port on which the packet arrived. For RX, this is the port on the device on which the packet was originally received (paired with src_mod).
dest_port	uint16	The port to which the packet was mapped. For RX, if the packet was copied to the CPU, this will be the port to which the packet would normally be sent, paired with dest_mod below. For TX, this controls the value placed in the HiGig header if the packet is transmitted on a HiGig port. The immediate port on which the packet will be transmitted is determined by tx_pbmp, described in the following paragraph. If the packet goes across more than one device, then the dest_port will determine the ultimate egress port along with dest_mod.
dest_port	uint16	The port to which the packet was mapped.

Table 76: Packet flags2 Flags Descriptions

Flag	Description
BCM_PKT_F2_MC_QUEUE	This flag indicates that the host cpu transmit packet to mc queue.

Table 77: DNX Internal Structure Description

Field	Type	Description
trap_qualifier	uint32	Trap Qualifier
trap_id	uint32	Trap Id

The following are new rx trap codes.

Table 78: Rx Trap Codes.

Trap Code	Description
bcmRxTrapReserved	ARP request with unknown host address.
bcmRxTrapOamUpMEP2	Default destination is OAMP
bcmRxTrapOamUpMEP3	
bcmRxTrapOamUpMEP4	To be used by bcm_oam_endpoint_action_set

bcm_petra_rx_trap_type_from_id_get

Get trap type according to trap id.

Syntax

```
#include <bcm/rx.h>

int bcm_petra_rx_trap_type_from_id_get(int unit, int flags, int trap_id,
bcm_rx_trap_t *trap_type);
```

Parameters

`unit`	BCM device number
`flags`	reserved for future use
`trap_id`	Trap id
`trap_type`	Trap type

Description

Get trap type according to trap id.

Returns

`BCM_E_NOT_FOUND`	if trap type not found

The flags used by the VLAN gport API are defined in Table.

Table 79: VLAN gport flags

Flag	*Description*
`BCM_VLAN_GPORT_ADD_EGRESS_STP_DISABLE`	Egress STP state is DISABLE.
`BCM_VLAN_GPORT_ADD_EGRESS_STP_BLOCK`	Egress STP state is BLOCK.
`BCM_VLAN_GPORT_ADD_EGRESS_STP_LEARN`	Egress STP state is LEARN.
`BCM_VLAN_GPORT_ADD_EGRESS_STP_FORWARD`	Egress STP state is FORWARD.

The VLAN action flags are defined in `bcm_vlan_action_set_t BCM_VLAN_ACTION_t`

Table 80: VLAN action flags

Flag	*Value*	*Description*
`BCM_VLAN_ACTION_SET_OUTER_VID_UNTAGGED`	0x0040	Set the packet format after EVE to untagged.

VIRTUAL SWITCHING INSTANCES

bcm_vswitch_flexible_connect_add

Attach given match port to forward destination.

Syntax

```
#include <bcm/vswitch.h>
#include <bcm/port.h>
int
bcm_vswitch_flexible_connect_add(
    int unit,
    uint32 flags,
    bcm_vswitch_flexible_connect_match_t *port_match,
    bcm_vswitch_flexible_connect_fwd_t *port_fwd);

typedef struct bcm_vswitch_flexible_connect_match_s {
    bcm_port_match_t match; /* Match criteria */
    bcm_gport_t match_port; /* Match port */
    bcm_vlan_t match_vlan; /* Outer VLAN ID to match */
    bcm_vlan_t match_inner_vlan; /* Inner VLAN ID to match */
} bcm_vswitch_flexible_connect_match_t;

typedef struct bcm_vswitch_flexible_connect_fwd_s {
    bcm_gport_t forward_port; /* gport to be used as forward destination */
    int forward_encap; /* forward encap id */
} bcm_vswitch_flexible_connect_fwd_t;
```

Parameters

`unit`	(IN) Unit number.
`flags`	(IN) Flags
`port_match`	(IN) Match info
`port_fwd`	(IN) Forward info

Description

Attach given match port to forward destination. You can also use the following flags:

Returns

`BCM_E_XXX`

bcm_vswitch_flexible_connect_get

Return forwarding information for given match peer

Syntax

```
#include <bcm/vswitch.h>
#include <bcm/port.h>
```

```
int
int bcm_vswitch_flexible_connect_get(
    int unit,
    uint32 flags,
    bcm_vswitch_flexible_connect_match_t *port_match,
    bcm_vswitch_flexible_connect_fwd_t *port_fwd);
```

Parameters

unit	(IN) Unit number.
flags	(IN) Flags
port_match	(IN) Match info
port_fwd	(OUT) Forward info

Description

Return forwarding information for given match peer

Returns

BCM_E_XXX

---

bcm_vswitch_flexible_connect_delete

Remove flexible connection.

Syntax

```
#include <bcm/vswitch.h>

int bcm_vswitch_flexible_connect_delete(
    int unit,
    bcm_vswitch_flexible_connect_match_t *port_match);
```

Parameters

unit	(IN) Unit number.
port_match	(IN) Match info

Description

Remove flexible connection.

Returns

BCM_E_XXX

VXLAN MANAGEMENT

Table 81: VXLAN port flags -new flags added

Name	Purpose
BCM_VXLAN_PORT_ENCAP_LOCAL	Enable egress HiGig proxy and encap VXLAN tunnel on local device

Section 7: Test Statistics

HOW TO READ THE DATA

The below tables represent a spread of data gathered per-device, per-suite, per-release. The percentages represent the aggregate rate of failure for that suite when run against all variants of the family of devices.

OVERVIEW

Each suite listed below is indicative of a specific module. Golden refers to a suite of tests that takes representation across multiple modules and serves as a sanity regression. Each suite contains tests of various types, loosely categorized as shown in Table 59: Suite Test Types

Table 82: Suite Test Types

Test Categories	*Description*
Configuration Tests	Tests that verify that each API functions appropriately and can configure the device as expected.
Functionality Tests	Tests that further validate each of the API through functional use often requiring traffic to be run through the system.
Semantic Tests	Tests that ensure that the proper error handling mechanisms are working and users cannot crash the device through the API.

LINUX KERNEL VERSIONS USED IN THIS RELEASE

In SDK 6.4.x, the following Linux kernel versions were used in our development and regression cycles with these main CPUs:

- BCM9XLP208XMC (WRX): 3.10.59
- BCM958625XMC (RSX): 3.6.5
- BCM98548PPCXMC (GTO): 2.6.21.7
- BCM98548PPCXMC (XLR): 3.14

Please refer to the Broadcom Network Switching Software Platform Guide for more details about these CPUs.

TOTAL TESTS

The below data in Table 60: Tests per Module represents the number of unique cases for each release.

Note that although a particular test case will execute for each and every chip, it is only counted once.

Table 83: Tests per Module

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	154	154	154
warmboot	1242	812	666
bfd	63	37	37
bhh	46	43	43
chip	9	9	9
cint	73	60	60
coe	568	568	542
cosq	1670	513	479
custom	7	7	7
ea	108	108	108
eav	19	19	19
extender	49	45	45
fabric	7	7	7
failover	10	8	8
fcoe	37	37	37
field	1404	1339	1252
higigproxy	129	129	129
infra	114	114	114
ipfix	17	17	17
ipmc	116	116	100
l2	337	329	327
l2gre	33	13	13
l3	506	494	454
l3.alpm	476	356	318
link	26	26	26
mim	46	19	19
mirror	173	173	171
misc	20	20	18
mpls	461	362	323
multicast	28	25	25
niv	65	58	58
oam	344	271	263
pkt	44	44	44
port	372	360	343
proxy	37	37	37
ptp	115	115	115
qos	13	13	12
rate	21	21	21
rtag7	43	42	36
rx	25	25	25
ser	157	157	93
stack	117	117	117

Table 83: Tests per Module

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
stat	386	351	241
stg	42	42	42
switch	197	197	194
time	33	33	32
tlvMsg	13	13	13
trill	47	40	40
trunk	223	219	199
tunnel	133	123	91
subport	31	31	31
vlan	239	233	217
vxlan	206	174	172
wlan	17	17	17
Test Suite Total	10868	8692	7980

API TEST RESULTS

Below tables show percentages of failures for corresponding test suites per SDK release.

ALL DEVICES

Note: This section represents aggregate results for all devices in the release.

Table 84: Tests Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.2 %	0.1 %	0.1 %
warmboot	0.1 %	0.5 %	0.3 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.1 %	0.2 %	0.4 %
bcm.cosq	0.1 %	0.2 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.1 %	0.1 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.2 %	0.3 %	0.3 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.5 %	0.5 %	0.6 %
bcm.ipmc	0.0 %	0.3 %	0.1 %
bcm.l2	0.1 %	0.2 %	0.2 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.1 %	0.1 %	0.2 %
bcm.l3.alpm	0.0 %	0.1 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.1 %	0.2 %	0.2 %
bcm.misc	0.0 %	0.1 %	0.3 %
bcm.mpls	0.0 %	0.1 %	0.1 %
bcm.multicast	0.0 %	0.1 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.1 %	0.2 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.5 %	0.6 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.1 %
bcm.rtag7	0.1 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.1 %	0.0 %

Table 84: Tests Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.ser	0.0 %	0.3 %	0.1 %
bcm.stack	0.1 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.1 %	0.1 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.1 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.1 %	0.1 %	0.1 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.1 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.3 %	0.3 %	0.3 %

TRIDENT2

Table 85: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.1 %	0.1 %	0.2 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.1 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.1 %	0.0 %	0.0 %
bcm.l2	0.1 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.2 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %

Table 85: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.1 %	0.0 %	0.0 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.2 %

TRIUMPH3

Table 86: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.2 %	0.0 %	0.1 %
warmboot	0.1 %	0.4 %	0.1 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.1 %	0.3 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %

Table 86: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.infra	0.0 %	0.1 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.2 %	0.3 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.2 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.3 %	0.6 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	1.4 %	1.3 %	1.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.2 %	0.3 %

KATANA2

Table 87: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.0 %	0.1 %	0.0 %
warmboot	0.1 %	0.9 %	0.2 %
bcm.bfd	0.0 %	0.0 %	0.0 %
bcm.bhh	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %

Table 87: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.1 %	0.1 %	0.4 %
bcm.cosq	0.1 %	0.1 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.2 %	0.3 %	0.2 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.1 %	0.1 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.4 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.7 %	0.5 %	1.2 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.2 %	0.2 %

GREYHOUND

Table 88: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.3 %	0.0 %	0.0 %
warmboot	0.0 %	0.1 %	0.2 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.0 %	0.0 %	0.1 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	1.3 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.7 %	0.5 %	0.5 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %

Table 88: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

TOMAHAWK

Table 89: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.1 %	0.1 %	0.0 %
warmboot	0.1 %	0.1 %	0.2 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	0.0 %	0.0 %
bcm.cosq	0.3 %	0.3 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.ea	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.3 %	0.3 %	0.3 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.4 %	0.0 %
bcm.l2	0.0 %	0.2 %	0.2 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.0 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.4 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.1 %	0.0 %	0.0 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %

Table 89: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.2 %	0.2 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.1 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.tlvMsg	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.2 %	0.3 %

TRIDENT2+

Table 90: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.1 %	0.2 %	0.9 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.2 %	0.2 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.2 %	0.0 %
bcm.l2	0.0 %	0.1 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.1 %
bcm.l3.alpm	0.0 %	0.0 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.6 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %

Table 90: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.1 %	0.2 %	0.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.1 %	0.9 %
bcm.vxlan	0.0 %	0.1 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.2 %	0.3 %

SABER2

Table 91: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.1 %	0.2 %	0.2 %
warmboot	0.1 %	1.4 %	0.5 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	0.1 %	0.9 %	0.7 %
bcm.cosq	0.1 %	0.0 %	0.1 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.3 %	0.3 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.1 %	0.1 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.2 %	0.2 %
bcm.link	0.0 %	0.0 %	0.0 %

Table 91: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.6 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.6 %	0.4 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.1 %	0.1 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.2 %	0.5 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.0 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.3 %	0.2 %

HURRICANE3

Table 92: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.0 %	0.0 %	0.7 %
warmboot	0.1 %	1.4 %	N/A
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	N/A	N/A	N/A
bcm.cosq	0.0 %	0.0 %	0.0 %
bcm.custom	0.0 %	0.0 %	0.0 %
bcm.eav	0.0 %	0.0 %	0.0 %
bcm.extender	0.0 %	0.0 %	0.0 %
bcm.fabric	0.0 %	0.0 %	0.0 %
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.0 %	0.0 %	0.2 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %

Table 92: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.ipfix	0.0 %	0.0 %	0.0 %
bcm.ipmc	0.0 %	0.0 %	0.0 %
bcm.l2	0.0 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.0 %	0.2 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.0 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	0.0 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	N/A	N/A	N/A
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.4 %	1.2 %
bcm.proxy	0.0 %	0.0 %	0.0 %
bcm.ptp	0.0 %	0.0 %	0.0 %
bcm.qos	0.0 %	0.0 %	0.0 %
bcm.rate	0.0 %	0.0 %	0.0 %
bcm.rtag7	0.0 %	0.0 %	0.0 %
bcm.rx	0.0 %	0.0 %	0.0 %
bcm.ser	0.0 %	0.0 %	0.0 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.0 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.0 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.5 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.1 %	0.1 %	0.1 %

MAVERICK

Table 93: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
golden	0.0 %	N/A	N/A
bcm.chip	0.0 %	N/A	N/A
bcm.coe	0.0 %	N/A	N/A
bcm.cosq	0.0 %	N/A	N/A
bcm.custom	0.0 %	N/A	N/A
bcm.eav	0.0 %	N/A	N/A
bcm.extender	0.0 %	N/A	N/A

Table 93: Test Results

	sdk-6.5.3	*sdk-6.5.2*	*sdk-6.5.1*
bcm.fabric	0.0 %	N/A	N/A
bcm.failover	0.0 %	N/A	N/A
bcm.fcoe	0.0 %	N/A	N/A
bcm.field	0.0 %	N/A	N/A
bcm.higigproxy	0.0 %	N/A	N/A
bcm.ipfix	0.0 %	N/A	N/A
bcm.ipmc	0.0 %	N/A	N/A
bcm.l2	0.0 %	N/A	N/A
bcm.l2gre	0.0 %	N/A	N/A
bcm.l3	0.0 %	N/A	N/A
bcm.link	0.0 %	N/A	N/A
bcm.mim	0.0 %	N/A	N/A
bcm.mirror	0.0 %	N/A	N/A
bcm.misc	0.0 %	N/A	N/A
bcm.mpls	0.0 %	N/A	N/A
bcm.multicast	0.0 %	N/A	N/A
bcm.niv	0.0 %	N/A	N/A
bcm.oam	0.0 %	N/A	N/A
bcm.pkt	0.0 %	N/A	N/A
bcm.port	0.1 %	N/A	N/A
bcm.proxy	0.0 %	N/A	N/A
bcm.ptp	0.0 %	N/A	N/A
bcm.qos	0.0 %	N/A	N/A
bcm.rate	0.0 %	N/A	N/A
bcm.rtag7	0.0 %	N/A	N/A
bcm.rx	0.0 %	N/A	N/A
bcm.ser	0.0 %	N/A	N/A
bcm.stack	0.0 %	N/A	N/A
bcm.stat	0.0 %	N/A	N/A
bcm.stg	0.0 %	N/A	N/A
bcm.switch	0.0 %	N/A	N/A
bcm.time	0.0 %	N/A	N/A
bcm.trill	0.0 %	N/A	N/A
bcm.trunk	0.0 %	N/A	N/A
bcm.tunnel	0.0 %	N/A	N/A
bcm.subport	0.0 %	N/A	N/A
bcm.vlan	0.0 %	N/A	N/A
bcm.vxlan	0.0 %	N/A	N/A
bcm.wlan	0.0 %	N/A	N/A
Test Suite Total	0.1 %	N/A	N/A

PHY TEST RESULTS

Starting from SDK 6.5.2, we have increased our focus and coverage on testing specific PHY and switch combinations in order to improve our quality in this area. The tables below represent specific results from our interoperability and regression testing for the release. We have been continually working to improve our results and coverage in ongoing releases.

SQA EXTERNAL PHY

Table 94: PHY Suite

Switch Device	*External Phy Device*	*Total Tests*	*% Fail*
56639_A0	phy8747_phy8728__10G	147	2.04%
56846_A0	Warpcore 40G/20G/10G	147	2.04%
56850_A2	phy84856_phy84858_10G	147	4.08%
56960_A0	100GFuria_40GMuxSesto	147	0.68%
56960_A0	10GFuria_10GSesto	147	0.68%
56860_A1	phy84856_phy84858_10G	147	1.36%
56860_A0	10GQuadra28	147	1.36%
56860_A0	40GQuadra28	147	2.04%
56860_A0	100GSesto	147	2.72%
56867_A1	phy82764_10G	147	1.36%
56867_A1	phy82764_40G	147	1.36%
56960_B0	phy82864_100G	147	3.40%
56960_B0	phy82864_100G_alt	147	5.44%
56960_B0	phy82864_40GPt	147	2.72%
56960_B0	phy82864_40GPt_alt	147	5.44%
56960_B0	phy82864_10G	147	4.08%
56960_B0	phy82864_10G_alt	147	4.76%
56960_B0	phy82864_11G	147	3.40%
56960_B0	phy82864_11G_alt	147	4.08%
56960_B0	phy82864_42GPt	147	2.04%
56960_B0	phy82864_42GPt_alt	147	4.08%
56960_B0	phy82864_106G	147	2.72%
56960_B0	phy82864_106G_alt	147	4.08%
56960_B0	phy82864_40G2x20	147	4.76%
56960_B0	phy82864_40G2x20_alt	147	6.12%
56960_B0	phy82864_42G2x20	147	8.84%
56960_B0	phy82864_42G2x20_alt	147	4.76%
56960_B0	phy82864_40GMux	147	4.76%

INTEROP INTERNAL PHY

Table 95: P2P Suite

Switch Device	Port Macro	Total Tests	% Fail
56260_A0	GE_VIPER_12x2P5	36	0.00%
56260_A0	EAGLE_XE_4x10	36	0.00%
56260_A0	GE_VIPER_24x1	39	0.00%
56260_A0	XE_VIPER_6x10	36	0.00%
56260_A0	EAGLE_GE_4x1	12	0.00%
56765_A0	XE_72X10	72	0.00%
56765_A0	XE_TSCE_X1	123	0.00%
56765_A0	XE_16X25	217	7.37%

Table 96: Loopback Suite

Switch Device	Port Macro	Total Tests	% Fail
56260_A0	GE_VIPER_12x2P5	28	0.00%
56260_A0	EAGLE_XE_4x10	22	0.00%
56260_A0	GE_VIPER_24x1	30	0.00%
56260_A0	XE_VIPER_6x10	10	0.00%
56260_A0	EAGLE_GE_4x1	10	0.00%
56765_A0	XE_72X10	11	0.00%
56765_A0	XE_TSCE_X1	44	0.00%
56765_A0	XE_16X25	20	0.00%

STATIC CODE QUALITY ANALYSIS

In 2016 we have upgraded our static analysis code tool to a version with many new checkers and will be working down the backlog of issues. Below shows the current baseline and progress in recent 6.5.x releases:

Table 97:

Line of Business	*New baseline as of 3/1/15*	*Open Issues SDK 6.5.3*	*Open Issues SDK 6.5.2*	*Open Issues SDK 6.5.1*	*Open Issues SDK 6.5.0*
DNX	120	104	61	75	125
XGS	90	86	16	27	23
SBX	40	34	1	0	48
SerDes	35	34	12	45	45
Common	45	42	11	16	46
Total	330	300	101	164	301

Section 8: Service Impacting Defects

A Service Impacting Defect (SID) is any defect (internal or external) that has high potential to severely disrupt network operations in a deployed system. This section lists the SIDs that have been identified since last release in Table 77: Resolved Service Impacting Defects .

Table 98: Resolved Service Impacting Defects

Number	*Chips*	*Affected versions*	*Errata Synopsis*	*Details*
SDK-89279	`56850_A2, 56860_A1`	6.4.4, 6.4.5, 6.4.6, 6.4.7, 6.4.8, 6.4.9	Customers running BCM56850 or BCM56860 with SDK release 6.4.4 through 6.4.9 have a risk of seeing ALPM failures and traffic drops in their system.	This issue was introduced by an improvement to the 'raw bucket split' introduced in SDK 6.4.4 and having L3RouteCache set to 1. ALPM IPv6 did not invalidate corresponding memory's cache for old bucket. Both ALPM IPv4 and IPv6 did not invalidate corresponding memory's cache for new bucket. Solution: Set L3RouteCache to 0, thus disabling ALPM memory caching or apply patch for SDK-89279.
SDK-91799	All devices	6.4.9	Potential memory leak with regard to the features that use `internalshr_htbl_destroyfunction`	All customers and all devices running Switch SDK might have a risk of encountering table a memory leak. The memory leak appears `whenevershr_htbl_destroy` was used. Memory leak was observed in DNX OAM/BFD interrupt context. This issue may also occur in BCM56450 OAM, BCM56260 OAM/SAT, BCM56640 OAM, BCM88030 L2/allocator/MPLS/OAM, and BCM88020 MPLS/OAM/L2. Fix of `shr_htb_destroy()`: Memory freeing performed with a while loop, instead of the for loop which didn't free all of the buffers

Section 9: Resolved Issues for 6.5.3

The following issues are resolved in version 6.5.3 of the SDK.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
PHY-1720	915319	NA	Not to disable super-isolate while initialization. Let the hardware-strapping determines the super-isolate mode. Except 54292 & 54291.
PHY-2030	984291	56860_A1	fixed the interface from xgmii to xlaiu for 40G
PHY-2049	989717	56850_A0 56850_A1 56850_A2	Added command handler support for Jumbo frame.
PHY-2060	992147	84848_A0	Avoided Config 13 strap mismatch with config propery phy_ext_rom_boot value
PHY-2068	987260	56860_A0 56860_A1	Retry count has been increaded to check the fw enable satus
PHY-2086	995131	AllChips	TX DIsable on line side when digital loopack
PHY-2092	997956	56850_A0 56850_A1 56850_A2	New FW version 0x29 relased to fix this issue
PHY-2093	998450	56850_A0 56850_A1 56850_A2	Enabled the Internal PHY only if the previous state is enable.
PHY-2153	1007500	56860_A0 56860_A1	PRBS Err count registers updated for KOI as 0xD0DA and 0xD0DB
PHY-2156	1007774	56860_A0 56860_A1	KOI Driver now supports PRBS API calls.
PHY-2172	1012223	56548_A0	Added support to CL37 autoneg. To enable CL37 user need to enable auto negotiation on 1G SR mode.
PHY-2188	992572	56860_A0 56860_A1	Code will go into master and SDK_6_4_11_BRANCH as specified in the planned-version.
SDK-48159	361224	56224_B0 AllChips	In earlier releases, L2 multicast lookup using L2_USER_ENTRYm is not supported in SDK. It has been supported in this release.
SDK-48834		56334_A0 56334_B0 AllChips	1. with config variable sw_timestamp_fifo_enable=0 2. counter thread doesn't read the timestamp fifo 3. API directly access the timestamp fifo. (there will be no delay) 4. **** for link down event, fifo needs to be flushed. ****
SDK-70346	884193	88660_A0	Disabling only one of table DMA or SLAM DMA is not support. They need to both be enabled (the default) or both be disabled. Added a startup error for unsupported configurations. Table DMA is disabled by: table_dma_enable=0 SLAM DMA is disabled by: tslam_dma_enable=0
SDK-71589	900749	AllChips	Fixed MII timeout errors during SOC initialization on MSI-enabled non-iProc XGS devices.
SDK-72073	893101	88670_A0	Strict preamble was on for SGMII ports on PM4x25, which is diffrent from PM4x10 and from specification. Fixed.
SDK-72969	907274	56640_A0	In earlier releases VPLS replace operation spent too long time in BCM56640. In this release, the performance for VPLS replace operation has been improved.
SDK-73323	910638	88650_B1 88660_A0	Fixed RFCS and RERPKT error on RXAUI port when traffic is 10G or almost 10G, and packet size is 1518 byte.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-73686		88470_A0 88670_A0	MPLS: Additional labels encapsulation is now supported through new apis: To enable the feature set soc property mpls_egress_label_extended_encapsulation_mode to 1. To add MPLS tunnel using LL entry: 1) bcm_l3_egress_create Add flag Flag BCM_L3_EGRESS_WIDE to configure wide entry 2) bcm_port_control_set Set class bcmPortControlMPLSEncapsulateAdditionalLabel 3) Configure Labels C3 and D1 by calling bcm_port_wide_data_set Gport is the LL Flag BCM_PORT_WIDE_DATA_EGRESS Data is 20bit MPLS label To add MPLS tunnel using existing mpls tunnel: 1) bcm_mpls_tunnel_initiator_set Set flag BCM_MPLS_EGRESS_LABEL_WIDE to indicate wide entry 2) bcm_port_control_set Set class bcmPortControlMPLSEncapsulateAdditionalLabel 3) Configure Labels C3 and D1 by calling bcm_port_wide_data_set Gport is the mpls tunnel Flag BCM_PORT_WIDE_DATA_EGRESS Data is 20bit MPLS label For example see cint cint_ip_route_tunnel_segment_routing.c
SDK-75174	903355	88660_A0	For BCM88650 and BCM88660 the meter id will be only 14 bits since 15 bit overlap in HW with out register Note that bit 15 is allways set hence the values of possible meter id are:16383 - 32767
SDK-77850	934604	88660_A0 88670_A0	api full description and usage can be found in grog
SDK-78368	934268	88650_B1 88660_A0	For fabric counters counter thread touch the counters with no protection. When the counters are read using API the counters thread and or stat APIs might return wrong counter value. Fixed.
SDK-78588		88660_A0 88670_A0	System Headers: The FHEI system header carries Ingress VLAN Edit (IVE) information or in case of IVE Command 0 selection (NOP action), it can pass important information such as In-LIF filtering resolution at the egress and allow Up-MEP OAM functionality. In Standard VLAN Translation mode (bcm886xx_vlan_translate_mode=0), this information wasn't passed for IVE command 0. The FHEI size was fixed for this scenario and 3 bytes of information now pass, as in the case of Advanced VLAN Translation mode.
SDK-79726	946978	56260_A0 56450_B0 56460	bcm_qos_map_destroy was clearing up the entire EGR_MPLS_PRI_MAPPING table when called with map_id 15. This has been fixed by setting egr_mpls_bitmap correctly in bcm_qos_map_create.
SDK-80014	940185	56440_B0 56450_B1	In the SDK API call for bcm_cosq_stat_get with L2 gport, stats for all L2 queues for the port were summed up before returning to the user. This was fixed to return stats only for the requested queue.
SDK-80152	949709	56960_A0	Now udf offset entries can be created up to 8 entries per pipe (total 32 entries). In addition, Wb upgrade from 1.2 to 1.3 (in both global and perpipe mode) is allowed (1.2 did not have per pipe udf offset configuration but only per pipe udf tcam configurations)
SDK-80663	950370	88670_A0	Interface to bcm_petra_mirror_port_destination_add() has been extended so that, on BCM88670, PMF egress mirroring is activated and assigned strength for 'copy' and for 'forward'
SDK-80941		56547_A0	Fixed the right SVP Gports qualifiers offsets during qualifiers init for BCM56340.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-80991	947340	88670_A0 88670_B0 AllChips	Fixed presentation of range qualifier in bcm_field_group_dump
SDK-81138	911055	88670_A0 88670_B0	SDK initialization with no ports on core 1 is now possible.
SDK-81331	949490	88660_A0	Restore local fault/remote fault generation when RX LOS is enabled but link does not go up successfully.
SDK-81367	954428	88670_A0 88670_B0	diag field Table [db=<table id>] [start=<entry id>] [num=<number>] Table show facility. Using without parameters will present all tables with qualifiers and actions. Use db in order to present specific table. Use start and num parameters if the table has a lot of entries
SDK-81579		88950_a0	-
SDK-81601	957484	88670_A0 88670_B0	Snoop packets with OutLIF stamping are now supported. So far even in case of correct snoop outlif action settings, the device didn't stamp it. The issue has been addressed by device enabling outlif stamp into FTMH.
SDK-81643		88660_A0	The 10M speed isn't returned as supported in the MAC?the advertisement of 10M is masked away and never written to the PHY. Add the 10M supported in Xmac_drv.c
SDK-81670	938722	88650_B1 88660_A0	packets in certain sizes may cause faulty ECC errors in the first 2k packets sent. this fix prevents those errors from occurring
SDK-81937	959373	88660_A0	CRPS: case of ING_PP, FULL COLOR - problem was found while reading a specific counter set which is being split between 2 engines. issue fixed
SDK-81988	958474	AllChips	Support PCI hot plug based on single BDE design.The PCI hot swap allows hot insert and remove the device in the platform where hardware supports the PCI hot plug. Upgrading BDE kernel modules is required to enable this feature.
SDK-82045	959568	56260_A0 56460_A0	PFC support for pre-allocated Subscriber queue(extended queues).
SDK-82240	959235	56260_A0 56260_B0	Customer can use bcmPortControl1588P2PDelay port control for configuring link delay.
SDK-82662	963076	56860_A0	Made changes in the SDK by adding support to the stat group bcmStatObjectIngFieldStageIngress for TD2+.
SDK-82951	965831	88670_A0	Using cascaded entries does not require any special custom flags now.
SDK-83015	966003	AllChips	implementation of L0 backpressure for CMIC ports.
SDK-83206		88670_A0	In earlier releases the callback for interrupts was skipped by wrong judgement on BCM88675. In this release, the judgement has been corrected on BCM88675.
SDK-83233	968261	88650_B1 88660_A0 88670_A0	TXRX: When setting SOC tm_port_header_type_out to CPU, internal dune header will be inserted before packet data. In previous release, 1) internal dune header was treated as part of packet, which cause decoder in packetwatcher work wrongly. 2) packet total length was gotten from internal dune header, which is not correct. In this release, 1) decoder skip internal dune header when parsing packet. 2) packet totol length is gotten from DCB.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-83339	967368	88660_A0 88670_A0 88670_B0	Extender: Control Bridge E-Channel registration check added. The following APIs were added to support E-Channel registration: 1. bcm_extender_forward_add – Register an e-channel 2. bcm_extender_forward_delete – Delete an e-channel registry 3. bcm_extender_forward_delete_all – Delete all entries 4. bcm_extender_forward_get- Retrieve an e-channel registry 5. bcm_extender_forward_t_init - Init bcm_extender_forward_t structure. bcm_extender_forward_t relevant fields: 1. flags - must be BCM_EXTENDER_FORWARD_UPSTREAM_ONLY 2. name_space - vlan domain 3. extended_port_vid - E-channel Id The feature requires adding E-Channel registration to port extender configuration. Example is available in cint_port_extender_cb_uc.c
SDK-83495	967170	AllChips	In previous releases, will access uC internal registers even though uC has been shut down in soc_uc_iproc_reset(). This will hang the system. In this release, will block accessing to uC internal registers when the uC has been shut down.
SDK-83501	970353	56640_B0	In previous releases, Jumbo frames isn't completely supported by BCM54280. In this release, this issue has been addressed.
SDK-83850	972446	88670_A0	In earlier releases interrupt names did not have interface id for BCM88670. In this release, added interface id to interrupt names for BCM88670.
SDK-83933	963451	56450_A0 56450_B0 56450_B1	race condition on the queue disable register access leads to the issue. Single non re-entry function to program the queue disable register fixed the issue.
SDK-83938		88670_A0	When systems are over-subscribed, packets might be corrupted or lost due to XLMAC TX_PKT_UNDERFLOW. In that cases, increasing TX_START_TX_THRESHOLD_QMLF_N field of the NBIH_TX_QMLF_CONFIG and NBIL_TX_QMLF_CONFIG registers will solve the problem, with the cost of some packet delay. For that, a custom feature was added to control the TX_START_TX_THRESHOLD_QMLF_N field of the NBIH_TX_QMLF_CONFIG and NBIL_TX_QMLF_CONFIG registers. Details about the custom feature: Name: custom_feature_mac_fifo_start_tx_thrs Default value: 4 Functionality: The value from custom_feature_mac_fifo_start_tx_thrs is written to TX_START_TX_THRESHOLD_QMLF_N field of all NBIH_TX_QMLF_CONFIG and NBIL_TX_QMLF_CONFIG registers!
SDK-83982	940778	88660_A0	The default behavior for DNX FAP devices, is that ingress resource (buffer descriptors) reservation take precedence over discard decisions (WRED, tail drop. FADT) only for DP 0 (green). This means that for DP 1-3 discards were still applied for packets in the reserved/guaranteed resource range. A new soc property is added to control the precedence of guarantee over admit was added. It can be used without a DP to specify all DPs, or with a specific DP: cosq_admission_preference={ADMIT_OVER_GUARANTEE,ADMIT_OVER_GUARANTEE} cosq_admission_preference_[0-3]={ADMIT_OVER_GUARANTEE,ADMIT_OVER_GUARANTEE}

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-84054	972956	88670_A0	Changed magic number to a macro that is getting the number of bit needed to be masked, according to device (compatible to BCM88670). updated the nof bits of the regarding actions to the Jericho UM values.
SDK-84156		AllChips	SDK-84156 Improve debug information of config add new command "config dump" for debugging config.bcm with better layout
SDK-84197	970765	AllChips	In previous releases, MMU limit settings at Ingress Port PG/ Egress Pool levels wasn't supported. This has been implemented in this release by adding cosq control types bcmCosqControlIngressPoolLimitBytes, bcmCosqControlIngressPortPGHeadroomLimitBytes, bcmCosqControlIngressPortPGResetFloorBytes, bcmCosqControlEgressPoolSharedLimitBytes, bcmCosqControlEgressPoolResumeLimitBytes, bcmCosqControlEgressPoolYellowSharedLimitBytes, bcmCosqControlEgressPoolYellowResumeLimitBytes, bcmCosqControlEgressPoolRedSharedLimitBytes, bcmCosqControlEgressPoolRedResumeLimitBytes.
SDK-84234	970867	88670_A0	Added support for XLAUI 48G interface.
SDK-84246	973779	AllChips	In the this release, a new feature was added to KNET. With the feature, customers can bring down/up base device to pause/ restore traffic. This feature is enabled by default in NAPI mode. A new module parameter "basedev_suspend" can be used to enable this feature in non-NAPI mode.
SDK-84267	973697	56850_A0 56850_A1 56850_A2	Added the support to set individual lane firmware mode
SDK-84270	974457	56960_A0 56960_B0	Added the config to allow CI73 at HG42G on BCM56960 without PLL change
SDK-84271	974389	56850_A2	In the previous releases, there was a race condition within virtual devices operations which would cause kernel crash. This issue has been fixed in this release.
SDK-84274	972927	AllChips	In previous releases ,the LR&LR4 interface was not supported in TSCMOD driver. This feature has been implemented in this release.
SDK-84347	973744	AllChips	Added the support of BRCM Hg2 CODEC on BCM56960.
SDK-84432	973747	56850_A0 56850_A1 56850_A2 AllChips	In previous releases, when called command "l3 l3table show", there was no hw index information printed. This has been improved by adding a new column "H/W Index" in l3table to show hw index.
SDK-84474	973890	56547_A0	In earlier releases remote trunk member could not be deleted in BCM56547. In this release, SDK supports deleting remote trunk member in BCM56547.
SDK-84494	957458	56850_A0 56850_A1 56850_A2	In previous releases, when a vxlan port was created with BCM_PORT_MATCH_PORT based on a trunk, SOURCE_TRUNK_MAP was programmed incorrectly when adding/deleing member to/from trunk. In this release, this issue has been addressed through calling bcm_port_match_add/delete.
SDK-84496	975653	56960_A0 56960_B0	added the support of XLAUI for 40G and LR4 into TH
SDK-84522	972938	88670_A0 88670_B0	Payload mask length was improperly defined for Jericho devices. Mask that should have ignored the whole payload left the two MSBs of the payload unmasked. Due to the wrong masking some entries were filtered out by the flash machine. Mask length was changed to fit the whole payload.
SDK-84536	973014	88670_A0 88670_B0	Dynamically adding and removing QSGMII port fixed.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-84557	975289	56860_A0	Doesn't reserve HSP MMU Ports on a pipe if there are no flex ports requiring a HSP MMU Port on that pipe.
SDK-84600	975898	88670_A0	'bcmFieldQualifyOutPort' does not have a 'core' bit and may, therefore, not be used on BCM88670 and up since pp_port is ambiguous. Use bcmFieldQualifyDstPort instead
SDK-84655	976124	56860_A0 56860_A1	In previous releases, config properties couldn't be parsed when a port was (re)probe/attached. This has been fixed adding a process of updating port configurations when a port is (re)probe/attached.
SDK-84784	850934	56260_A0	In earlier releases, the viper driver did not support Hardware LOS solution. In this release, the SOC property "serdes_rx_los " has been implemented in BCM56260. This will enable "signal_detect_en" and "invert_signal_detect" in register 0x8300.
SDK-84796	977472	84793_A0	The code has a incorrect assignment to control variable instead of rv.., the code is fixed.
SDK-84937	977514	56960_B0	previously, tomahawk's rtag7_hash_sel was initialized to use port_based_hash and trident2's rtag7_hash_sel was initialized to use flow_based_hash. With this change, tomahawk's rtag7_hash_sel is initialized to use flow_based_hash and rtag7_flow_based_hash table is populated to support flow based hash.
SDK-84945	966743	53288_A0	API bcm_trunk_set doesn't clear load balancing key of previous setting if existed. This was fixed.
SDK-84948	978355	56860_A0	In previous releases, PORTMOD didn't surport external PHY preemphasis and driver current setting in config.bcm. If added "serdes_preemphasis" and "serdes_driver_current" in config file, the properties always set to the most external PHY, no matter external PHY or internal PHY. In this release, the config properties "serdes_preemphasis" and "serdes_driver_current" is for internal PHY. "phy_preemphasis" and "phy_driver_current" is for external PHY.
SDK-84955	977289	56960_A0 56960_B0 AllChips	In previous releases, bcmSwitchObjectL3HostV4Max and bcmSwitchObjectL3HostV6Max were used for maximum number of IPv4 or IPv6 routes possible in L3 host table. They have been changed to represent maximum number of L3 Hosts possible that can be used for IPv4 and IPv6 routes respectively, which is for the consistency with bcmSwitchObjectL3HostV4Used and bcmSwitchObjectL3HostV6Used.
SDK-85056		56860_A0 56860_A1	In previous releases, a subport could not be added into a given multicast group in TD2plus. In this release, this issue has been addressed.
SDK-85113	973115	56450_A0 56450_B0 56450_B1	NONUCAST_TRUNK_BLOCK_MASK table was not being programmed correctly when a local port was added to a trunk group followed by a remote port. This issue has been fixed.
SDK-85133	979396	56960_A0 56960_B0	Added 40G XLAUI support on BCM56960.
SDK-85216		88670_A0 88670_B0	ILKN quad memory resources were not verified when dynamically adding a new non-ILKN port. Now, these resources are verified when dynamically adding non-ILKN port.
SDK-85254	979515	56850_A0 56850_A1 56850_A2 AllChips	In the previous release, the API bcmSwitchObjectIpmcHeadTableFree might not get the correct free entires of the table REPL_HEAD. In this release, this issue has been addressed.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*

SDK-85283	979660	56260_A0 56460_A0	Implementation of faults status of multiple endpoints.
			This is the implementation of bcm_sb2_oam_endpoint_faults_multi_get API to get the fault status of all BHH endpoints associated with firmware running on BTE.
			Please enable OAM_FAULTS_MULTI_GET feature in feature_list in Make.local to enable this API implementation.
			Usage: ======
			int ep_count; int sizof_struct; int unit = 0; uint32 flags = 0; uint32 endpoint_count = 0; uint32 bhh_num_sessions = 256; uint32 max_ep_no_bhh = (bhh_num_sessions) ? bhh_num_sessions : 128; /* Take valid maximun nuber of endpoints from applicatuion specific max number of sessions soc property if present or take the defualt value provided in SDK */ bcm_oam_endpoint_fault_t faults[256]; /* It is not used as malloc as cint has some problem with sizeof operator. So, allocating statically. * User can maloc it withn their aplication. */
			bcm_oam_protocol_type_t endpoint_protocol = bcmOamProtocolBhh;
			print bcm_oam_endpoint_faults_multi_get(unit, flags, endpoint_protocol, max_ep_no_bhh, faults, &endpoint_count);
			printf("Number of valid endpoints with faults sent by get API %u", endpoint_count);
			for (ep_count = 0; ep_count < endpoint_count; ep_count++) { print faults[ep_count]; }

SDK-85338	978366	56850_A0 56850_A1 56850_A2 AllChips	In the previous release, a deadlock issue could happen between L3 and trunk module. In this release, the deadlock issue is resovled.
SDK-85353		56850_A0	In previous releases, some resources were not freed at device detaching. This has been addressed by releasing resources properly when detaching device.
SDK-85355	980505	56545_B0	In previous releases, the packet with size more than 64 did not trigger IBOD WAR recovery sequence. That caused packets miss-forwarding. In this release, runt threshold was used to check runt length.
SDK-85377	980247	56450_A0 56450_B0 56450_B1	Fixed KNET local tx in pipeline bypass mode for Katana2 and Saber2.
SDK-85454	981222	88670_A0	Added support for bcmRxTrapEgTxMtuFilter, bcmRxTrapEgTxDiscardFrameTypeFilter and bcmRxTrapEgTxSplitHorizonFilter traps
SDK-85492	980674	AllChips	In previous releases, the legacy TSCE/TCSF drivers didn't support to configure the interface on the internal PHY in the presence of an external PHY. In this release, this function was added to support.
SDK-85501	979817	56440_A0 56440_A1 56440_B0	Multiple field action entries for action bcmFieldActionFabricQueue with same cosq index and profile index are allowed to be installed without throwing error.
SDK-85503	979464	AllChips	change the default tx main taps from 0x66 to 0x64
SDK-85624	982442	56450_A0	SP<->WERR dynamic scheduler conver had a WAR for KT. Issue fixed in KT2 Hardware and not required. Removed the changes related to WAR.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-85656	979473	56960_A0 56960_B0 56961_B0 56962_B0 56963_B0 56965_A0 56966_A0 56967_A0 56968_B0	In previous releases, only provided limited information when BIST failed. In this release, add failure index and test case index to test results.
SDK-85683	982549	56340_A0 56850_A0 56860_A0	In previous release, soc_features_bcm56634_a0 is disabled improperly for BCM56850, BCM56860 and BCM56340. Now we enable this feature for those chips.
SDK-85695	974626	AllChips	There are two copy of some tables. One copy locates PIPE_X, and another locates PIPE_Y. Command "dump" can read each copy of those tables. In previous releases, there is no document to state this function of "dump". Now, this function of "dump" is explained for users.
SDK-85770	979673	88660_A0 AllChips	The feature is redesigned. New PAPI is added for now.
SDK-85779	980456	AllChips	The previous versions don't use proper function to construct tunnel decapsulation information. Now add a new function to resolve this issue.
SDK-85782		56850_A0 56850_A1 56850_A2 56960_A0 56960_B0	In previous releases, under a multiple threads access scenario, an UFT table HASH MOVE would lead to a system crash. This has been fixed.
SDK-85790	982811	56860_A0	Changing back the programming of the CMIC LED Data ram block based on the physical port number, instead of the logical port number.
SDK-85791	983258	56850_A0	In previous releases, the related L3 entries were traversed by s-channel operation one by one to restore the software status during warmboot, which resulted in a large execution time. In this release, this has been fixed by leveraging DMA for bulk operations.
SDK-85831	982578	AllChips	There will be no wrong updates in other pipes for EFP stats , when user actually tries to update EFP stats in a particular pipe.
SDK-85833	973390	56460_A0	Use VP as EGR_MP_GROUP key for PW and MEPID for LSP based BHH endpoint.
SDK-85881	979732	56440_A0 56440_A1 56440_B0 56441_B0 56442_B0 56443_B0 56444_A1 56445_A0 56445_A1 56445_B0 56446_B0 56447_B0 56448_B0 56449_B0	on SDK 6.5.1, bcm_port_class_set() API fails with gport as an argument. This was due to incorrect argument being passed to the internal functions. This is fixed now.
SDK-85896	982701	56960_A0	In previous release, if only configure physical port number less than 9, SDK will crash as BUS error; In this release, the BUS error has been fixed.
SDK-85965	983831	56850_A0	Update TDM and SDK (by Mahesh) to better support parallel initialization for BCM56850.
SDK-85968	979993	AllChips	In previous releases, frames which exceeded MTU size were dropped but still increased "snmpIfInUcastPkts" and "snmpIfHCInUcastPkts". In this release, the issue has been fixed.
SDK-85977	983488	56960_A0	In previous release,bcmPortControlMmuTrafficEnable type are not implemented through port control.In this release, these are implemented.
SDK-86006	979672	56460_B0	Program timestamp byte adjustment registers to minimize the timestamp jitter.
SDK-86081	980993	56640_A0	In previous release, some alloced memories can not be release as expected by the process of repeated init->detach

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-86121	984629	56060_A0	In previous release, there is a limit to configure cosq mapping. In this release, release this limit.
SDK-86134	986949	56860_A0	In previous releases, after an L3 egress object was added into an ECMP group whose max_paths was changed dynamically, it could not be destroyed when it did not belong to any ECMP group. In this release, this issue has been addressed.
SDK-86141	984194	AllChips	Diag shell command to display minimum cache size required for Field Module for any FP configuration for Tomahawk Device is provided to customers. Also provided soc_property to be used to set the scache_size for field module. The right procedure to use the diag shell command and the config_property. 1) During development, after all FP configuration is done , please use diag shell command "fp show wbcachesize" This prints the amount of cache needed for field module say "x bytes". 2) Set the config property 'field_scache_size' to a value > = x bytes. config add field_scache_size=x config save 3) Reboot the system for config property to take effect. exit ./bcm.user In case of upgrades, for example 6.5.1 to 6.5.2 : 1) Steps on 6.5.1 is same as given above. 2) During development, take the 6.5.2 code, apply all FP configuration and find the scache size for field using diag shell command "fp show wbcachesize" , say y bytes. 3) After warmboot sync is done on 6.5.1, reset the config_property to the new value as per seen on 6.5.2 branch code. config add field_scache_size=y config save 4) Do warmboot, the field scache allocation will be updated during warmboot to the new value as per changed in config property. export SOC_BOOT_FLAGS=0x200000 ./bcm.user
SDK-86217	979290	56440_B0	stat types supported issue fixed
SDK-86224	983657	AllChips	In previous release,below type are not implemented through cosq control. bcmCosqControlIngressPool, bcmCosqControlIngressPortPGSharedLimitBytes, bcmCosqControlIngressPortPGMinLimitBytes, bcmCosqControlIngressPortPoolMaxLimitBytes, bcmCosqControlIngressPortPoolMinLimitBytes, bcmCosqControlIngressPortPGSharedDynamicEnable, In this release, these are implemented.
SDK-86255	979973	56850_A0 56850_A1 56850_A2	In previous releases, the initialization of SDK might fail if the image was built without the macro INCLUDE_L3. In this release, this issue has been addressed.
SDK-86270	984526	AllChips	There is no ENTRY_FIELD field in the FP_GLOBAL_MASK_TCAMm and FP_GM_FIELDSm table views from H/W perspective. The SDK S/W uses overlay_tcam_bitmap to indicate which view a given entry should be interpreted. Currently this bitmap is not saved as bookkeeping information, resulting in cache and vmap information which are recovered from H/W entry during warmboot for FP_GM_FIELDS table are junk information. Thus the memscan will detect ser error during scanning this table.The overlay_tcam_bitmap should be saved as bookkeeping information in Level2 warmboot. And SDK should clear the cache and vmap information of invalid FP_GM_FIELDS entries based on the overlay_tcam_bitap during warmboot.
SDK-86327		56850_A0 56850_A1 56850_A2	In previous releases, the warmboot support for the directed flexible mirror feature was not implemented. In this release, the warmboot for this feature has been implemented.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-86331		56850_A0	In previous releases, VPLAG for MIM based VPs was not supported. This function has been supported in this release.
SDK-86337	988076	56960_A0 56960_B0	In earlier releases, packet visibility was showing incorrect ECMP result if the ECMP index was greater than 4095, this has been fixed.
SDK-86345	988605	56960_A0 56960_B0	By default, all the preselector entries has the same priority. If a priority of higher TCAM index entry is modified to higher priority value, the entry is not moved up. The issue is fixed now.
SDK-86356	988449	56260_A0 56460	Support to map the port to external buffer dynamically is added for BCM5626x and BCM5646x devices.
SDK-86362	987387	56340_A0	In previous releases, BCM56340 LED mapping is incorrect, this problem is derived from SDK-67002, now the code is reverted to correct logic.
SDK-86371	988186	56450_A0 56450_B0 56450_B1	Problem: Counter for unknown L2 unicast packets is not enabled in the bcmStatGroupModeTrafficType group_mode. Solution: Enable the counter for unknown L2 unicast packets for the bcmStatGroupModeTrafficType group_mode.
SDK-86375	988255	56260_A0 56460	Handle and configure subport tpid when inner vlan option is used as well.
SDK-86376	988268	56441_B0	On SDK-6.3.11 Release the L3_DEFIP/ L3_DEFIP_PAIR_128 table sizes are not properly configured. This has been addressed.
SDK-86451	987403	56640_A0 56640_A1 56640_B0 56641_A0 56642_A0 56643_A0 56643_A1 56643_B0 56644_A0 56644_A1 56644_B0 56645_A0 56648_A0 56648_B0 56649_A0 56649_B0	When the first API call to add a node to the LLS tree is called, all queues are disabled. Later, when a queue is added to the lls tree, it was enabled individually. The issue was seen because HSP queues were also disabled upon the first add API call. This was fixed by not disabling the HSP queues when the first bcm_cosq_gport_add API is called.
SDK-86453	987959	56960_A0 56960_B0	For BCM56960 IFP, group QSET update after group is created is supported .
SDK-86483	988807	56960_A0	Corrected the counter_count and meter_count values in fp group status.
SDK-86505	961561	56960_A0 56960_B0	added the support of use bcm api to retrieve the eye margin estimate
SDK-86507	961561	56960_A0 56960_B0	Added new control set to get eye values for horizontal left and right, and vertical upper and lower
SDK-86557	988218	AllChips	Fp show is updated to display Entry free per group information as part of Primary slice information.
SDK-86586	983710	AllChips	In previous releases, four packet types (known_l2_gre_pkt/known_vxlan_pkt/ known_fcoe_pkt/unknown_fcoe_pkt) were ignored when creating a stat group mode with packet attribute equals to " bcmStatGroupModeAttrPktType " and packet attributes value equals to "BCM_STAT_GROUP_MODE_ATTR_ALL_VALUES". As a result, a group mode different with "bcmStatGroupModeSingle" was created. In this release, the issue has been fixed by including all packet types for "BCM_STAT_GROUP_MODE_ATTR_ALL_VALUES".
SDK-86602	984380	56960_A0 56960_B0	Fixed a crash in flex counter module - When retrieving attribute selectors from custom mode data structure, incorrect pointers were accessed since the selectors are stored in physical mode when the mode is not shared.
SDK-86622	976175	88660_A0	RXAUI at 21G fail to TX traffic. Fixed.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-86638	988922	56640_A0 56640_A1 56640_B0 56850_A0 56850_A1 56850_A2	In previous releases, the ingress_if was not retrieved when getting mpls tunnel switch info. The root cause is that l3 ingress mode was not checked. In this release, the ingress_if is retrieved when getting mpls tunnel switch info.
SDK-86697	990607	56850_A0 56850_A1 56850_A2 56851_A0 56851_A1 56851_A2 56851P_A1 56851P_A2 56852_A0 56852_A1 56852_A2 56853_A0 56853_A1 56853_A2 56854_A0 56854_A2 56854_B0 56855_A0 56855_A2	In previous releases, SDK would check the speed during enabling Cut Through on a certain port, and would return error when the port was in oversubscription mode and the actual speed was lower than 40Gbps. This check has been removed in this release since HW logic will check the speeds of source port and destination port to decide whether a dedicated packet is eligible to Cut Through.
SDK-86708		56850_A0 56850_A1 56850_A2	In the previous release, the CLI command "l3 l3table show" would display no entries after the warmboot when the L3 host table was used in embeded view. In this release, this has been addressed.
SDK-86709		56960_A0 56960_B0	In previous releases, the port detach operation didn't clear the reference counter of core. In this release, this issue has been addressed.
SDK-86713	990246	56850_A0 56850_A1 56850_A2 56860_A0 56860_A1 56960_A0 56960_B0	Fixed the bug that bcmSwitchObjectL3RouteV6Routes128bMax and bcmSwitchObjectL3RouteV4RoutesMax always return 65536 and 393216 irrespective of number of pivots available.
SDK-86770	988349	AllChips	In previous releases, flex hashing was not able to work correctly on BCM56960 devices. This has been fixed in this release.
SDK-86852	991387	AllChips	In previous releases, if customers use warmboot mode to start SDK, l2_mod_fifo sbus dma may not be stopped, and relative registers are not cleared, then l2_mod_fifo can't handle L2 events normally. Now, l2_mod_fifo sbus dma is reset during procedure of Warmboot, so its function can be recovered.
SDK-86854	991664	56040_A0 56041_A0 56042_A0 56044_B0 56045_B0 56046_B0 56047_A0 56048_A0 56049_A0	In 6.4.8 release, BCM INIT would stop and return error if there is no scache file configured. This has been fixed.
SDK-86857	987498	56640_A0	In previous releases, if a MPLS entry was attached with a flex counter then the API bcm_mpls_tunnel_switch_add() was called with the flag BCM_MPLS_SWITCH_REPLACE to update it, a new entry would be added instead of updating it. This has been fixed.
SDK-86882	991588	56450_B1	In this release, serdes_asymmetric_speed_mode has been set to 1 for 1G port on BCM56450 devices.
SDK-86905	992252	56340_A0 56340M_A0 56850_A0 56850_A1 56850_A2	In previous releases, bcm_port_priority_group_mapping_set / bcm_port_priority_group_mapping_get was not implemented on Trident2. This has been addressed by implementing these APIs on Trident2.
SDK-86906	992253	56340_A0 56340M_A0 56850_A0 56850_A1 56850_A2	In previous releases, per PG PFC transmit control was not implemented on Trident2. This has been addressed by implementing APIs: bcm_port_priority_group_config_set / bcm_port_priority_group_config_set on Trident2.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-86944	988934	56450_A0 56450_B0 56450_B1 AllChips	Firmware application (Broadsync) re-configuration is take care after firmware re-load
SDK-86972	990578	AllChips	In previous releases, for MIM application entries might be inserted successfully, even thought the return value was BCM_E_EXISTS. In this release, this has been fixed.
SDK-87023		56960_A0 56960_B0	fixed a bug for the 40G XLAUI speed set
SDK-87036	992120	53415_A0 53416_A0	To avoid wrong ext-phy driver been hooked up and wrong phy setting been applied to incorrect ext-phy once user miss the assignment on phy address for a specific port.
SDK-87191	991224	56260_A0 56260_B0 56460	In few scenarios, OAM APIs returned without releasing a mutex. Such areas have been identified and mutex has been released.
SDK-87230	992215	56850_A0	In previous releases, two issues could cause port flapping at warmboot time. One is that a status flex-port needs was not saved for warmboot. The other one is relevant to HW Linkscan Fast mode which caused PHY init sequence could not be finished. The two issues have been fixed in this release.
SDK-87237	993420	56640_B0 56643_B0	In previous releases, if users set L2 aging time for multiple times, all the dynamic entries in the L2 table would be aged immediately. This has been fixed.
SDK-87253	993932	56960_A0	Fixed the memory leak issue while creating a FP group in Tomahawk.
SDK-87256	990246	56960_A0 56960_B0	Fixed the bug that bcmSwitchObjectL3RouteV6Routes128bMax and bcmSwitchObjectL3RouteV4RoutesMax always return 65536 and 393216 irrespective of number of pivots available.
SDK-87275	993932	56960_A0	In previous releases, there was possibility that memory was allocated but not freed when calling API bcm_esw_stat_group_mode_id_create and bcm_esw_stat_group_mode_id_config_create. In this release, these memory leak issue has been fixed by checking return value to free allocated memory.
SDK-87293	993444	56304_B1 56334_A0 56445_A0	Removed a spurious check if endpoint configuration had rx configuration before deleting old rx configuration.
SDK-87308	992819	56860_A0 56860_A1	In previous releases, HSP command would display incorrect MC queue index. This has been addressed by calculating MC queue index correctly.
SDK-87310	992819	56860_A0 56860_A1	In previous releases, MC queue 8 and 9 cannot be successfully attached when HSP port is in MC_GROUP_MODE. This has been addressed by updating parameter validation process.
SDK-87335	993607	56960_A0 56960_B0	In previous releases, bcm_pkt_t.src_port was int8, which was not able to fully support Tomahawk, as Tomahawk has more than 128 ports. This has been fixed by expanding bcm_pkt_t.src_port and bcm_pkt_t.dest_port to 16 bits.
SDK-87348	994473	56860_A1	In previous releases, broadcast of firmware with different PHY types on the same MDIO bus was not supported. It is supported in this release.
SDK-87385	994750	AllChips	Updated BCM56850 qualifier Init routine to set correct secondary selectors for SGLP qualifiers.
SDK-87386	993974	53346_A0 56150_A0	Fix assertion failure while customized physical to logical port mapping table is not defined for every port in config file.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-87395		56800_A0 5680_A0 56820_A0 56820_B0 56830_A0 56830_A1 56830_A2 56832_A1 56833_A1 56834_A2 56836_A1 56840_A0 56841_A3 56841_B0 56842_A0 56843_B0 56844_A0 56845_A2 56845_B0 56846_A0	Add below information in users guide. If bcm_stat_sync_get was continuously called for a counter, the corresponding rate calculated with CLI "show counter" was incorrect. Customers are responsible for calculating if they want precise rate.
SDK-87413	992893	56960_A0 56960_B0	Issue & Fix ========= 1. Existing method of recovering EFP key's based on HW slice mode and entry key info will not recovery if the entries are created with only default key. Need to sync and recover EFP key info for warmboot. 2. In CINT script not all entries are installed. During WB recovery, FP will recover only installed entries. After CINT script change, see that all entries are getting recovered properly 3. One of the EFP selcode combination was not setting slice mode as per regfile encodings. Latter this combination of key was found to be an invalid combination. Removed the invalid combination of key selection. 4. 0: bcm_esw_modules_deinit: Warning: Deinitializing field returned -14: During exit clean, stat module will be cleaned up before field and stat get will return failure. This needs to be handled in field. 5. BCM56960 EFP group and entry sync was not getting triggered in some scenario's (when configuration is double wide group followed by 2 single wide mode groups). Fixed to handle such scenario's. 6. When entry id is more than 12bit, customer needs to use "bcmSwitchStableSaveLongIds" switch control. 7. Slice pbmp update during recovery was not proper for per-pipe mode. Loopback port was getting added to pbmp. Corrected the same. 8. In case of double wide mode, secondary slice pbmp not was getting updated during warmboot recovery. Corrected the same. 9. In auto-expanded slice, entry key match type is not computed on based entry key type and slice mode, instead its always set to 0. Corrected the same. 10. For ActionDrop recovery, take patch from SDK-84442
SDK-87423		88660_A0	In the previous release, routing over ac functions don't support in PON application. In this release, this issue has been addressed by adding routing over ac function in PON application when set soc property custom_feature_router_over_ac as 1.
SDK-87429	993344	56450_B0 56450_B1	The support to create VP and VLAN translation based on VLAN range for GPONs was missing in earlier releases for BCM5645x family of devices. This support has been added.
SDK-87439	993596	56340_A0 56340M_A0	In previous releases, if "lls_num_l2uc" in the config file was set to a value grater than 8, bcm_cosq_gport_mapping_set() would return BCM_E_PARAM on configuring queues with higher index nubmer.
SDK-87443	995034	56540_A0 56540_B0	In previous releases, only the base policer in a cascaded policer group can be used to attach, which was not flexible in some BandWidth sharing applications. In this relaese, all the policers in a cascaded policer group can be used to attach.
SDK-87499	992001	88670_A0 88670_B0	Extender: Added support for L3 forwarding from/to cascaded port (for jericho and above) Example is available in cint_route_over_extender.c

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-87502	989710	88670_A0	CRPS performance issue: when calling bcm_cosq_gport_statistic_multi_get(), the driver pull all CRPS DB from the SW state and at the end of the procedure, it store it back to the SW state. This operation takes a lot of time. In order to short the time of the bcm_cosq_gport_statistic_multi_get(), the driver will pull from SW state (and store back) only the parameters that required for this procedure.
SDK-87544	992466	88670_A0	Fix padding length issue: 4 extra bytes appended to CPU Tx packets
SDK-87667	984478	88202_A0	The code will be checked in to SDK_6_4_BRANCH and master branch.
SDK-87679	994782	56850_A0	In previous releases, the API bcm_l2_addr_delete would wrongly delete the MY_STATION_TCAM entries added by the API bcm_l2_station_add after the warm boot. In this release, this issue has been addressed.
SDK-87758	996078	56340_A0	In previous releases, when customers built their own LLS tree and bcm_cosq_control_set() was called with type=bcmCosqControlEgressPoolLimitEnable, the API would return BCM_E_PARAM. This has been fixed.
SDK-87772	950925	56840_A0 56850_A0 56850_A1	When ser parity error occurs on the memories which cannot be accessed by cpu, sdk miss reporting an event to application. Now in this release, sdk will report an event when this case happens.
SDK-87779	990087	56850_A0 56850_A1 56850_A2	In previous releases, shared cell count for all service pools were updated when modifing per queue guaranteed cell. This has been addressed by updating only the associated service pool.
SDK-87795	995212	56860_A0 56860_A1	A new field EGR_QOS_PROFILE_INDEX was added to table EGR_PORT to provide index for accessing EGR_PRI_CNG_MAP table on Trident2+. This enhancement, however, was not supported by API bcm_qos_port_map_set in previous releases. This has been fixed in this release.
SDK-87825	994488	84858_A0	added the support inside the portmod to invoke external PHy driver cable_diag function
SDK-87843	997205	56840_A0	In previous releases, if the system booted through warmboot mode, L2 MOD FIFO was not stopped, and corresponding registers were not cleared. This would cause bcm_l2_addr_regsiter() callback not to be called. This issue has been addressed in this release.
SDK-87924	990578	AllChips	In previous releases, for MIM application entries might be inserted successfully, even thought the return value was BCM_E_EXISTS. In this release, this has been fixed.
SDK-87925	997676	56260_A0 56260_B0 56460	Added checks to unlock oam control lock mutex when returning for failures.
SDK-87932	994401	56460_A0	Following are the fixes for this Jira 1. Lock both the h/w semaphores on the host(SDK), before resetting any of the cores uC1/uC2. Implemented the timeout mechanism for acquiring these semaphores so as to avoid deadlock 2. Before actually resetting the PTP-core, ensure that the "in_use" flag is set, to ensure that PTP is configured and SDK knows the PTP-core 3. Disable Timesync interrupts as done in SDK-88764 for mhost reset of uC1
SDK-87942	997790	56850_A0 56850_A1 56850_A2	In the previous release, MPLS_EXP could be not retrieved in the API bcm_qos_map_multi_get. In this release, the issue has been fixed.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-87990	994351	56850_A0 56850_A1 56850_A2	In the previous release, memory free was not handled rightly which would cause memory leak. In this release, memory free is handled rightly.
SDK-88002		56850_A0 56850_A1 56850_A2	In previous releases, TDM could be in wrong configuration when parallel initiation was done on multiple units. This has been addressed by preventing multiple units to share the same TDM data.
SDK-88013	996417	56850_A0 56960_A0	In previous releases, the API bcm_l2_addr_delete_by_vlan_gport_multi could not delete the mac address if the virtual port was greater than 0x7ff. In this release, this issue has been addressed.
SDK-88045	996574	56960_A0 56960_B0	Issue & Fix ========== 1. CopyToCpu params are not synced and recoverd for IFP. Added sync and recovery for the same. 2. Update PBMP for VFP group based on stage mode 3. For BCM56960 IFP, action resolution id is of no importance thats the reason, sync is done for it. And it will recovered as -1. Looks like customer is doing memset of group structure instead of calling bcm_field_group_config_t_init. It's recommended that customer call bcm_field_group_config_t_init instead of doing memset.
SDK-88057	996882	56634_A0 56850_A0 56960_A0	In the previous releases, incorrect initializing SER log mutex caused assertion issue during warmboot. This issue has been fixed in this release.
SDK-88104	997669	56860_A1	In previous releases, MPLS_ACTION field of VLAN_XLATE could only be set as L2_SVP or L3_IIF for VPWS flow. It was not supported to set both. This improvement has been implemented in this release.
SDK-88110	996410	56440_A0 56440_A1 56440_B0 AllChips	Avoid deleting the cosq map entry when the entry is still being shared by multiple bcmFieldActionFabricQueue action entries.
SDK-88113	978909	AllChips	Fix BCM56850 TDM to prevent the board from crash when single 20G mode is configured in Vxworks
SDK-88131	996751	56840_A0 56846_A1	In previous releases, bcm_cosq_cpu_cosq_enable_set/get would fail after warmboot. This has been addressed by allocating _bcm_td_cosq_cpu_cosq_config in warmboot initialization.
SDK-88139	997569	56960_A0 56960_B0	When invoking _bcm_pfc_deadlock_control_set() in _bcm_pfc_deadlock_init function, change the first parameter from 0 to unit variable.
SDK-88196	998599	56840_A0 56846_A0 56846_A1	In previous releases, the call to the API bcm_trunk_member_add might cause a crash if too many members were added into a trunk. This issue has been addressed in this release.
SDK-88248	999739	56850_A0	In previous releases, the L2 station entry might not be gotten correctly by calling the API bcm_l2_station_get after warmboot since the relationship between the software index and the hardware index was lost. In this release, this has been fixed by storing the relationships into scache.
SDK-88257		56842_A0	In previous releases, stack topology could be looped when SAFC was enabled. This has been addressed by adding only modport derived from the SAFC algorithm for unit carries only SAFC traffic.
SDK-88263	999892	56540_A0 56540_B0 56640_A0 56640_B0	In previous releases, access the register CMIC_MMU_COSLC_COUNT_DATA from diagnostic shell would cause system crash. But actually it is deprecated. This has been fixed by making it not available from users.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-88273	998858	56850_A0	In previous releases, traffic might be stopped when both shaper rate and bucket size been reduced greatly. This has been addressed by reseting metering before applying new metering configuration.
SDK-88316	997786	56860_A0	To enable the 1G AN on the PHY84328, the PHY should be programmed at 1G forced speed first, then AN should be enabled. PHY84328 supports only cl37 (no SGMII AN) at 1G speed.
SDK-88325	999633	56860_A0	In previous releases, when cut-through was globally enabled, VXLAN encapsulation for multicast packets didn't work if network port had only one receiver. In this release, the issue had been fixed by disabling cut-through on IPMC group specified for VXLAN traffic.
SDK-88338	992728	88660_A0	api bcm_l2_cache_get did not update bellow flags even if they were set: BCM_L2_CACHE_TRUNK BCM_L2_CACHE_SUBTYPE Now the flags updated in case they are set.
SDK-88357	994401	56460_A0	Following are the changes/fixes for this issue 1. Proper API sequence along with code changes are done to close UC Msg thread on Host before resetting UC core. Patch for this is submitted in this Jira 2. Also depends on the fix provided in SDK-88764
SDK-88373	996035	56960_A0 56960_B0	SDK creates group with minimum possible mode(80/160/320/480) when hints are provided for qualifiers in groups QSET.
SDK-88395	991918	88660_A0	bcm_field_qualify_SrcPort_get will not return error, so this field can be used for ACL
SDK-88400	900764	56634_A0 56634_B0 56636_B0	In previous releases, the SD_TAG properties cannot be based on the VPN for MIM application. This feature has been supported from this release through the flag BCM_MIM_VPN_MATCH_SERVICE_VLAN_TPID.
SDK-88428	1000121	56960_A0	Increments and decrements of hw_ent_count of slice is balanced so that it will always carry the right number of entries installed in hardware and cause no MISS in FP_TCAM.
SDK-88465	1001091	56850_A0	In previous releases, the call to the API bcm_l3_intf_delete would lead to a system crash if the L3 egress intf was greater than the max value which chips supported. This has been fixed in this release.
SDK-88505	999763	56860_A0	In previous releases, the dual mode would not work correctly in BCM56860. This has been addressed in this release.
SDK-88506	999081	56334_A0	To support "HashDestination" CLI command for Enduro to predicit RTAG7 ECMP hash destination.
SDK-88564	1001479	AllChips	Issue ==== In entry move, memory allocation for array of pointer was done assuming size of pointer as 4, As pointer size is 8 bytes in 64 bit systems, this was causing assert while memory free . Fix === Used sizeof() instead hardcoded value of 4
SDK-88573	996606	88670_A0 88670_B0	Creating OAM endpoint time improved. Likewise bcm_oam_endpoint_action_set. bcm_oam_endpoint_create() has been measured to be 5 times faster.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*

SDK-88581	985906	88670_A0 88670_B0	In order to disable aging for a given VSI do the following: 1. Use a free aging profile. (2 in the example) 2. Set aging_cycles=0
			Example: bcm_vlan_control_vlan_t vsi_control;
			bcm_vlan_control_vlan_t_init(&vsi_control);
			vsi_control.aging_cycles = 0;
			vsi_control.entropy_id = 2;
			vsi_control.forwarding_vlan = vsi;
			vsi_control.unknown_unicast_group = vsi;
			vsi_control.unknown_multicast_group = vsi;
			vsi_control.broadcast_group = vsi;
			bcm_vlan_control_vlan_set(unit,vsi,vsi_control);

SDK-88604	999957	AllChips	In previous releases, bcm_l2_traverse() would cause high latency to L2 table operation in other threads. This has been fixed.
SDK-88614	1002100	88660_A0	tr 50 changed to skip write only memories.
SDK-88624	1000669	56340_A0	In previous releases, MMU limit settings at Ingress Port PG/ Egress Pool levels wasn't supported. This has been implemented in this release by adding cosq control types bcmCosqControlIngressPoolLimitBytes, bcmCosqControlIngressPortPGHeadroomLimitBytes, bcmCosqControlIngressPortPGResetFloorBytes, bcmCosqControlEgressPoolSharedLimitBytes, bcmCosqControlEgressPoolResumeLimitBytes, bcmCosqControlEgressPoolYellowSharedLimitBytes, bcmCosqControlEgressPoolYellowResumeLimitBytes, bcmCosqControlEgressPoolRedSharedLimitBytes, bcmCosqControlEgressPoolRedResumeLimitBytes.
SDK-88654	1001506	56860_A1	In previous releases, the default entry 0 of the hardware table ING_VLAN_TAG_ATCTION_PROFILE would be deleted wrongly if the API bcm_vlan_translate_action_add was called multiple times. In this release, this has been fixed by avoiding the deletion of the default entry.
SDK-88676	974010	88660_A0	Default application initialize linkscan for some port types. This jira update CAUI ports to be in the default initialed ports.
SDK-88713	993663	88670_A0 88670_B0 88950_a0	Configuring "serdes_firmware_mode" explicitly might mistakenly override other SerDes UCODE configuration.Fixed However, It is recommended to use dedicated control for the required property and not to use SerDes firmware mode control. For example: SerDes DFE on should be enable by: "bcm_port_phy_control_set(unit, port, BCM_PORT_PHY_CONTROL_FIRMWARE_DFE_ENABLE, value):" And not by: "bcm_port_phy_control_set(unit, port, BCM_PORT_PHY_CONTROL_FIRMWARE_MODE, value):"
SDK-88751	1001589	56846_A0 56846_A1	Issue is verified by the customer and it is fixed.
SDK-88764	987879	56450_B0	Fix a possible hang when iProc is reset while running firmware
SDK-88778	1001112	56685_A0 56685_B0	In previous releases, the interrupt and handler of oam event was binded with parity error. Applying a dedicated handler to control the oam events.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-88783	1002625	56850_A0 56850_A1 56850_A2 56851_A0 56851_A1 56851_A2 56851P_A1 56851P_A2 56852_A0 56852_A1 56852_A2 56853_A0 56853_A1 56853_A2 56854_A0 56854_A2 56854_B0 56855_A0 56855_A2	In previous releases, port could stop egressing if lower WRED thresholds with traffic on the fly. This has been addressed by removing WRED refresh operations when changing drop profile.
SDK-88873	1001095	88670_A0	change _unit variables to unit to avoid compilation issues
SDK-88908	996575	88660_A0	Bug fix: Parsing of frwrd_type field for mirror packet.
SDK-88954	1003118	56850_A0	In previous releases, there would be redundant entries of l3_iif_profile table if the profiles were shuffled by calling the API bcm_vlan_control_vlan_set after warmboot. In this release, this has been fixed by correcting the software state of the profile table.
SDK-88967	1003430	88670_B0	added the support of FEC cl91 set/get function into bcm api.
SDK-89031	1002996	56846_A1 56850_A2	In the previous release, VFI table was not read by DMA operation. In this release, it is read by using DMA operation.
SDK-89044	1000564	88670_A0	When CAUI is working with LP DFE mode bcm_port_speed_get API and ps ("PortStatus") command will fail. fixed
SDK-89057	1006501	AllChips	fix TDM function to resolve the vmap allocation error
SDK-89062	1004312	56840_A0	Issue ==== During reinstall, hw index is always assumed as changed and was being deleted during the clearup. Fix === 1. Added fix to cleanup only when the new hw index is different from that of programmed. 2. Ignore resetting next hop table for internal backup entry.
SDK-89065	1004473	56450_A0 56450_B0 56450_B1	In earlier releases the source port information (pkt->src_port) of a RCPU ToCPU packet, when received from a trunk group, was not preserved. The source port (pkt->src_port) information of RCPU ToCPU packet is now preserved when it is received from a trunk group.
SDK-89072	1002647	56450_A0 56450_B0	fixed s/w lpm logic to send the v4-64 bit entry to unpaired tcams if there is still some space.
SDK-89075	1003509	56840_A0	In previous releases, Next-Hop replication type was not handled correctly in _bcm_tr2_repl_list_compare function and led to inefficient use of the table. This has been fixed in this release.
SDK-89078	1004136	AllChips	The logic used for recovery of cosq map was wrong. fixed the traversing the child nodes on reinit.
SDK-89079	997816	88670_A0	Confusion between logic port and TM port when getting tdm packet port was fixed.
SDK-89095	969810	88670_A0	In earlier releases, the tables IQMT_PDM_X were not cleared at init time. In this release, the tables IQMT_PDM_X have been cleared at init time.
SDK-89127	1000525	AllChips	In earlier releases the flag BCM_QOS_MAP_L2_VLAN_PCP was missed when getting qos egrss pcp vlan flags. In this release, the flag BCM_QOS_MAP_L2_VLAN_PCP has been added.
SDK-89144	999992	88670_A0	dynamic port multiple cycles of add and remove ports is fixed.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-89152	1005248	56860_A0 56860_A1	Two new events given below are added to receive notifications on receiving bfd packet with poll/final bit. bcmBFDEventEndpointRemotePollBitSet, /* Received a packet from remote with Poll Bit set */ bcmBFDEventEndpointRemoteFinalBitSet, /* Received a packet from remote with Final Bit set */ A new config property "bfd_session_down_event_on_create" has been added to enable the behavior of sending state change (down) notification after the session is created.
SDK-89161	1004926	56060_A0	Code-fix will go into master as specified in the planned version.
SDK-89196	1004414	56960_A0 56960_B0	If the external PHY is in repeater mode, applies AN flag on internal PHY.
SDK-89209	1005145	AllChips	Impact Chip: TOMAHAWK Impact Feature: LAG Failover Impact Module: Failover In previous release, LINK_STATUS memory is not reflecting the port status correctly on interface of FalconCore on TOMAHAWK. In this release, LINK_STATUS memory is reflecting the port status correctly on interface of FalconCore on TOMAHAWK.
SDK-89224	1005160	56850_A0 56850_A1 56850_A2	In previous releases, an egress ECMP object referenced by a host routing entry could be deleted after the warm boot, which was incorrect. In this release, this issue has been addressed.
SDK-89237	1004741	56960_A0 56960_B0	In Tomahawk, when InPorts is present in Groups QSET in Global Mode, SDK pushes entries using per pipe views. i.e. SDK writes entries to IFP_TCAM_WIDE_PIPE0, IFP_TCAM_WIDE_PIPE1, IFP_TCAM_WIDE_PIPE2, IFP_TCAM_WIDE_PIPE3 (4 S-Channel Commands)instead of writing to IFP_TCAM_WIDE(Single S Channel Command).So there will be small time period where there is a mismatch b/w entries in all pipes, which will cause a leak when ports from different pipes are involved in the test. So Before entry is removed , Disabling that entry in all pipes so that traffic will not be matched differently in different pipes.
SDK-89261	1005946	56340_A0	Updated initialization of EFP keys in the order such that all L2 keys goes into a single slice which would allow to qualify for all types of packets. Order of keys now is KEY4, KEY 1, KEY1_NO_V4,KEY2,KEY3.
SDK-89279	1002156	56850_A0 56850_A1 56850_A2	In earlier releases, when bcmSwitchL3RouteCache was enabled in ALPM mode, l3 routing table could be corrupted due to cache inconsistency. This issue has been resolved.
SDK-89287	1004317	56460_B0	Problem: The PLL reset needs to be done after LDO_CTRL is modified for both A0 and B0 devices of Saber2 family. Solution: During soc initialization, when LDO_CTRL is modified, the PLL reset is also triggered.
SDK-89294	1005982	56850_A0 56850_A1 56850_A2	In the previous releases, early TX/RX sequence used at warmboot time did not cover KNET initialization. This caused customer filters got overridden. The issue has been fixed in this release.
SDK-89312	1003606	88660_A0	Added a program which supports Up tx packets on LAG port.
SDK-89316	1005338	56960_A0 56960_B0	In previous releases, DMA buffer space of all counters was not protected by lock named "COUNTER_LOCK", so there was critical data reentry. Now, the DMA buffer space is protected by "COUNTER_LOCK" properly.
SDK-89327	1004665	AllChips	In previous releases, there was a miss of a lock in the internal function _bcm_xgs3_tbl_add, which might lead to the incorrect NHI reference counts. In this release, this issue has been addressed.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-89328	1006503	88670_A0 88670_B0	ERSPAN Mirroring modified to coinside with the standard - so that in the GRE header, the K flag is reset and the S flag is set.
SDK-89331	1006528	AllChips	We now check port limit on device init such that if a pipe has 100G ports on it, the maximum number of ports it can support is 44.
SDK-89332	1006118	56850_A0 56850_A1 56850_A2	In previous releases, there was deadlock between LC_LOCK and TRUNK_LOCK. Now this deadlock have been removed.
SDK-89391		88670_A0	In external TCAM, new capability to extended the capability for IPv4 packets. when this feaure enable it is possible to add up to 2 million entries. for this case lookup is done with smaller key (VRF 8bit, DIP) and RPF lookup is not available. To add entries use BCM_L3_FLAGS2_SCALE_ROUTE. To create L3 interface use BCM_L3_INGRESS_EXT_IPV4_DOUBLE_CAPACITY.
SDK-89408		88670_A0	Fixed a bug that caused the wrong DRAM ingress buffer type to be selected for ingress multicast groups with many replications. In systems that use DRAM, packets replicated in ingress to over 8 replications (ingress multicast replications and up to two extra replications if the packet is snooped or/and mirrored) might have caused wrong system behavior, including system failure requiring a reset. This should not have affected systems with no DRAM, or where ingress multicast groups with more than six replications were not created.
SDK-89409	1005606	88670_A0	Dynamically adding logical port with id above the id of the max existing logical port at power-up - was causing a memory leak. Issue is now fixed and counters database is now allocated according to the maximum ports allowed. Moreover, connecting port with logical id beyond the maximum allowed (dynamically or statically) will lead to a BCM error.
SDK-89414		88670_A0 88670_B0	bcm_multicast_delete fixed for 2 CUD case.
SDK-89443	1004266	56060_A0	When bcm_stat_sync_get() API is used, the counter value is fetched from the hardware and the software copy of the counter s also updated, so that the count accumulation done by counter thread does not go out of sync. For 32-bit counter registers, the update to the software copy of the count was being done for wrong indexes causing wrap arounds to be detected. The above problem will be seen only in conjunction with counter thread running at very short interval and calling bcm_atat_sync_get() API in another thread context.
SDK-89496		AllChips	Logic added to check the array size before writing to it.
SDK-89508	1007521	56850_A0 56850_A1 56850_A2	In previous releases, the operation of clearing START_BY_START error status in the register MEM_FAIL_INT_STAT on BCM56850 was missed in parity error handler function. This issue has been addressed in this release.
SDK-89515	1003217	53400_A0	bcm_time_capture_get() called with TIME_CAPTURE_MODE_IMMEDIATE will return the timestamp data
SDK-89519	1007400	56860_A0 56860_A1 56864_A1	In previous releases, configuring bcmSwitchMeterAdjust and bcmSwitchShaperAdjust for front panel ports would not take effect in BCM56864. This has been addressed by disabling the soc_feature_egr_lport_tab_profile check in the APIs.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-89522	1007452	56340_A0 56340M_A0	In previous release,the valid range of port_class is 0-63 in function bcm_vlan_translate_egress_action_add . In this release,the valid range of port_class is 0-127 in function bcm_vlan_translate_egress_action_add .
SDK-89541	1007523	88670_B0	move all the logics for the interface get from glue code to phymod api
SDK-89559	997811	AllChips	increased soc property validation checked from 2 levels of '_' (underscore) into 3 levels of '_'. so that following soc property usage for example : ext_ram_dq_swap_dram2_byte0_bit0 will not result as unrecognized soc property warning while following soc property is defined: ext_ram_dq_swap
SDK-89564	1006002	AllChips	With this fix in place the rpc routines would send correct payload length to function bcm_rpc_setup().
SDK-89590	989748	88470_A0 88660_A0 88670_A0	Create HiGiG port with bcm_port_add always fail, fixed.
SDK-89624	1007597	56450_A0 56450_B0 56450_B1	For BCM5645x family of devices FCS error is observed on XE port under certain conditions if the port was mapped to external memory on the fly at run time. This issue has been fixed in this release.
SDK-89632	1002863	AllChips	Failure for foreign master data set entry lookup print is changed from ERROR to VERBOSE log.
SDK-89663	1007804	56450_A0 56450_B0 56450_B1	As part of this jira, a fix is added to lookup ipmc repl tables correctly for matching interface bitmaps.
SDK-89706	1008027	56260_A0 56260_B0	Action NewClassId doesnt require NHI and hence doesnot conflict with Action RedirectEgrNextHop. Removed the incorrect conflict checks placed on the same.
SDK-89712	1007985	56850_A0 56850_A1 56850_A2	When setting a trunk group with no members, the TRUNK_GROUP table will be empty, resulting in the psc field couldn't be recovered from this table when performing warmboot operation. Now in this fix, the psc field of trunk info is saved in scache and it can be recovered from scache during warmboot.
SDK-89715	1007772	56850_A0 56850_A1 56850_A2	In previous releases, SDK added a new semaphore to handle BST thread exiting during re-init, which would increase COSQ module init time if the thread was in a wait state. This has been fixed in this release by waking up the thread to speed its exiting.
SDK-89720	1008377	56860_A0 56860_A1	In previous releases, Trill Multicast egress packets couldn't be decapsulated on HG ports. This has been addressed by disabling HG2 mode on Loopback port.
SDK-89731	990804	AllChips	update _tx_dv_free() function
SDK-89737	1005911	88670_A0 88670_B0	In the previous release, the 1588 timestanmping mode on QSGMII port was 48-bit by default. In this release, this mode is set to 32-bit.
SDK-89758	1006088	AllChips	In previous releases, routine bcm_rpc_setup might result in memory write overflow. So "ASSERT" had been added to this routine, when there was memory write overflow in the routine, SDK would be stopped.
SDK-89941	1009295	56460_A0 56460_B0	Port, Vlan and Level parameters are not required in oam endpoint add CLI for REMOTE endpoints. But this was enforced in the CLI. This has been fixed.
SDK-89942	1009233	56260_A0 56260_B0	Calling bcm_oam_opcode_group_set() was creating a gap between default oam opcode control profile and opcode group mapping. Fixed the issue by refreshing default oam opcode control profile based on new opcode group mapping.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-89978	1009133	56850_A0 56850_A1 56850_A2	In previous releases, after synchronizing MMU traffic flag of one port to WarmBoot cache, there was a chance that the port link status could be changed from UP to DOWN. As a result, the MMU traffic flag was changed, but the new flag was not saved to WarmBoot cache. So when system Warm booted next time, SDK would use old value of the flag which could result in traffic lost. This issue have been solved now.
SDK-89995	1005660	56850_A0 56850_A1 56850_A2	When recovering mirror destination id for mirror ing/egr MTP, sdk will create mirror destination using dest_id getting from scache for every mirror session as long as this destination doesn't already exist. But when there are two mirror sessions which share the same destination port, the dest_id of second session couldn't be recovered correctly. In this release, SDK stores the flags of every mirror destination in scache when performing sync operation, and recovers it from scache during warmboot. For the session with BCM_MIRROR_DEST_REPLACE flag, SDK skips calling _bcm_esw_mirror_destination_find and then creates mirror destination with correct flags.
SDK-90009	1009707	56850_A0 56850_A1 56850_A2	In previous releases, when _bcm_trident_trunk_fp_get() was called, the system might crash because LAG DLB feature was not checked and invalid memory DLB_LAG_MEMBER_ATTRIBUTEm was accessed. This has been addressed by adding checking for LAG DLB feature.
SDK-90020	1008887	56260_A0	An index allocated from BHH pool was not released when returned with failure. This caused running out of BHH pool indexes after multiple failures. This has been fixed by releasing the BHH pool indexes when endpoint create fails.
SDK-90055	1005103	88670_A0 88670_B0	VSWITCH: Flexible cross connect is now available with the following api: int bcm_vswitch_flexible_connect_add(int unit, uint32 flags, bcm_vswitch_flexible_connect_match_t *port_match, bcm_vswitch_flexible_connect_fwd_t *port_fwd); port_match is match information, port_fwd is forward information. The api supports match of port = inlif and BCM_PORT_MATCH_PORT_VLAN_STACKED or BCM_PORT_MATCH_PORT_VLAN For this configuration soc property vpws_tagged_mode should be set to 1. Example can be found in cint_vswitch_vpws_tagged.c
SDK-90056		88670_A0	Single Core can be set to ANY FAP ID (and not only even)
SDK-90070	1004651	88670_A0	When working with ILKN, ref_clock of 125MHz and data-rate of 11500, there were two issues: Functional issue with 4x10 port leading to wrong SerDes rate and diag issue in 4x25 port, leading to 0.5 factor in SerDes display (SerDes rate was OK). Both issue were fixed and correct rate is displayed for ILKN with ref_clock of 125MHz and data rate of 11500, both PML and PMH ports.
SDK-90072	1010005	56850_A0 56850_A1 56850_A2	In previous releases, bcm_switch_pkt_info_hash_get could not get correct destination gport when SOC was in dual modid mode. This has been addressed by using proper port info to construct gport.
SDK-90073	1009403	88670_A0	setting different speeds to QSGMII ports resides on the same lane is now supported.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-90080	1009568	56850_A0 56850_A1 56850_A2	In previous releases, L3_IIF_PROFILE.TRUST_DSCP_PTR was not retained with the original value and L3_IIF.L3_IIF_PROFILE_INDEX would be modified when bcm_vlan_control_vlan_set was invoked. This has been addressed in this release by correctly getting trust-qos-map-id in _bcm_tr_l3_ingress_interface_get().
SDK-90092		56850_A1	In previous releases, bcm_vxlan_port_add didn't check the egress object that was created without BCM_L3_VXLAN_ONLY flag. In this release, bcm_vxlan_port_add will return a BCM_E_CONFIG error if the passed in egress object is created without BCM_L3_VXLAN_ONLY flag.
SDK-90111	999796	56960_A0 56960_B0 56960_B1	Since mode id is released twice, we could make a code change to release mode id for custom mode and then release mode id for regular mode only if it is not custom mode.
SDK-90122	1010094	56860_A0 56860_A1	The register TOP_L1_RCVD_CLK_BKUP_CTRLr.STAGE0_MODEf and STAGE1_MODEf was programmed by incorrect value.
SDK-90127	953951	56150_A0 56151_A0	In previous release parity bit in some MMU memories was generated when parity was disabled in bcm56150 family. Fix the issue by avoid generating the parity bit when parity is disabled.
SDK-90132	1003575	56960_A0 56960_B0 56960_B1	In previous release,usage of bcmPortControlMmuTrafficEnable & bcmPortControlMmuDrain sequence for down ports will set link bit in LED_DATA_RAM. In this release,it has been fixed.
SDK-90136	1010658	AllChips	In previous releases, the parity error in the register PG_SHARED_LIMIT_CELL on BCM56846 couldn't be handled properly, and would cause system crash. This issue has been addressed in this release.
SDK-90151		88670_A0	Changed wrong "DIAG oam" comment in "Diag buffers" command.
SDK-90218	1010399	56854_A0 56854_A2 56854_B0	In previous releases, port control flag bcmPortControlLLFCTransmit was not explained in detail, i.e. for StrataXGS devices, LLFC flags are used to control SAFC other than 802.3x PAUSE, while PFC flags instead are used for PAUSE configuration. This has been addressed by adding extra notes for these flags.
SDK-90271	970603	88670_A0 88670_B0	QSGMII CRC errors were seen sometimes on random links is fixed
SDK-90272	1010782	56850_A0 56850_A1 56850_A2 56860_A0 56860_A1	The PVTMON8 register has been enabled back in the TD2/ TD2+ code to display the temperature as it was ignored previously due to the wrong temperature readings shown by it.
SDK-90281	1011271	56840_A0	In this release, invalid mem check has been added in SER correction to prevent device from crash.
SDK-90294	969667	88670_A0 88670_B0	Injecting packets through PWE outlif (directed to PWE network) can be counted using soc property custom_feature_eg_pwe_counting. CCM packets should be injected with PWE label. Second PWE label which will be generated, will be removed by a PRGE program. Note: In this case MPLS tunnel cannot be encapsulated in the same EEDB entry with the PWE. For LMMs they have to add an OAM-TS and the outlif should be the LSP outlif (or whichever outlif comes after the PWE).
SDK-90297	1011195	56260_A0 56260_B0	There was missing cleanup in case bcm_oam_endpoint_create() was called with invalid parameter. Now SDK is modified to carry out proper cleanup in case there is a failure in bcm_oam_endpoint_create()

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-90328	1010022	88670_A0	assert while calling bcm_cosq_port_mapping_set, because trying to configure IRR_DESTINATION_TABLE per core, but the table is common for both cores. Solution: configure the table only once, using the relevant MACRO
SDK-90389	1011916	88670_A0 88670_B0	88675: Fixed an issue where creating a symmetric global lif and then a non symmetric global lif in the same 4k range would make it impossible to delete the non symmetric lif.
SDK-90392	1011831	56440_A0 56440_A1 56440_B0	The field entry dump command is modified to not remove the loopback ports in entry port bitmap mask.
SDK-90458		AllChips	bcmOAMEventBHHCCMTimeoutClear event will be issued when timeout is cleared on an BHH endpoint.
SDK-90460		88670_A0	Fixed presentation of PMF_DATA qualifier in diag field last
SDK-90479	1010074	56850_A0 56850_A1 56850_A2	In previous releases, the API bcm_l3_host_travers would stop getting the routing information once the reference count of an egress object in use was set to zero incorrectly. In this release, this issue has been addressed.
SDK-90489	1008068	88660_A0 88670_A0	In earlier releases SER handler for the memory IRR_MCDB was not right. In this release, SER handler has been corrected.
SDK-90491	1012311	88670_A0	In early release, SDK only supports priorities and queues 1:1 mapping. This has been extended to support N:1 mapping.
SDK-90492	1012578	56450_A0 56450_B0 56450_B1	In earlier releases, build issue is seen in non-posix OS due to presence of signal API in DDR test. This has been fixed in this release.
SDK-90497	1012570	56450_A0 56450_B0 56450_B1	Wrong device type check function was used to identify the PCI device when using the "dump pci" cli command . Correct the PCI device identity check with bus type get function in this release.
SDK-90502	996365	88670_A0 88670_B0	soft reset for single core mode is fixed.
SDK-90520	1012439	88670_A0 88670_B0	AC2PWE P2P: When setting AC2PWE Cross-connect where PWE destination is OutLIF and FEC (MPLS-tunnel.EEI(OutLIF)) , get/update returned error for example when creating Endpoint on the AC.
SDK-90530	1013040	56850_A0 56850_A1 56850_A2	In previous releases, the warm boot of multicast module took too much time. In this release, the time has been reduced.
SDK-90537		56960_A0	The JIRA enables Falcon parallel detection for 1G KX of CL73.
SDK-90544		88670_A0	In the previous release, ERSPAN could not encapsulate with IP tunnel, which was incorrect. In this release, this issue has been addressed by getting IP tunnel correctly.
SDK-90566	1012499	56850_A0 56850_A1 56850_A2	In the previous releases, extender gport was not available for egress VLAN translation. It has been added in this release.
SDK-90599	1012045	56850_A0 56850_A1 56850_A2	Used the correct index style for obtaining the HW index for a given port/cosq.
SDK-90607	1013571	56850_A0 56850_A1 56850_A2	The internal function of getting the ecmp member count was not implemented correctly. When the ecmp group only had one member, SDK missed to handle this scenario. The issue has been fixed in this release.
SDK-90612	998428	88670_A0 88670_B0	modify the description of itmh.base.jer_dest_info
SDK-90613	1013439	56450_A0 56450_B0 56450_B1	The duplicated sal_abort function declaration has been removed from $SDK/include/sal/appl/sal.h. The declaration is now only in $SDK/include/sal/core/libc.h.
SDK-90676	1013226	88670_B0	25G interface set error leads to this error. Modify furia_line_int_map to correct this error. new line interface (phymodInterfaceSR) has been added to match phymodInterfaceXFI

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-90710	988545	88670_A0 88670_B0	In L2, when using the API bcm_l2_replace() with the flag BCM_L2_REPLACE_MIGRATE_PORT, the destination was not properly updated in the matching MAC table entries. This flag support is now corrected and when used, destination is updated as expected.
SDK-90718	1013987	56450_A0 56450_B0 56450_B1	Fixed hard coded unit parameter for soc_property_get () calls in soc files
SDK-90741	1013393	56340_A0	BCM56345 will use a specified array for flex port interface, and this will not share with other chips.

SDK-90765	1010634	56960_A0 56960_B0 56960_B1	In previous releases, setting bcmCosqControlEgressUCQueueSharedLimitBytes and bcmCosqControlEgressUCQueueMinLimitBytes to zero didn't disable the queue. This has been fixed.
			bcm_cosq_control_set(0, gport, 3, bcmCosqControlEgressUCQueueSharedLimitBytes, 0);
			bcm_cosq_control_set(0, gport, 3, bcmCosqControlEgressUCQueueMinLimitBytes, 0);

SDK-90770	1014368	88670_A0 88670_B0	MPLS PORT: failover_id and network_group_id replace operation for pw created with ingress_only flag is now supported.
SDK-90808	1012945	88660_A0	OAM: when adding an RMEP in the OAMP while traffic is running spurious timeouts may occur.
SDK-90820	1008304	88670_B0	In the previous release, auto-neg can not work well on 100G tscf port and 40G tsce port. In this release, auto-neg work well on both 100G tscf port and 40G tsce port.
SDK-90830	1011600	88670_A0	In L3, when setting the SOC property "bcm886xx_l3_ingress_urpf_enable=1", a bit was allocated in the inLIF profile for uRPF handling (as in BCM88660). This behavior is correct only for BCM88660, because in BCM88670, the inRIF profile controls the uRPF behavior. All redundant LIF configurations related to uRPF have been removed for BCM88670.
SDK-90843	1014201	56850_A0	The NEW_INT_PRI field of EGR_IM_MTP_INDEX table isn't configured with right value when adding mirror source port after performing warmboot. The root cause is the element int_pri of mirror destination doesn't get recovered during warmboot. Now in this version, this issue is fixed.
SDK-90844	1014681	56850_A0 56850_A1 56850_A2	In previous releases, when the feature of directed flexible mirroring was enabled, the reference counters for ingress and egress MTP index wouldn't be updated correctly. This issue has been fixed in this release.
SDK-90879	1007353	88670_A0 88670_B0	In L3, when calling bcm_l3_host_traverse(), only up to 129 entries were handled by the callback function, even if more hosts reside in the host table. This is fixed so that all configured hosts will be handled when calling bcm_l3_host_traverse().
SDK-90887	1006296	56960_A0 56960_B0 56960_B1	In previous release ,the API bcm_cosq_gport_discard didn't apply to the device-wide discard settings. In this release,it has been fixed.
SDK-90911	1004680	88670_A0	In L3, enhanced_fib_scale_prefix_length=4/8/12/16/20/24/28 now allows a specific route prefix to be stored in the LEM. This requires single DB KAPS setting and replaces the IPv4UC program without RPF. To add UC routes to the LEM use bcm_l3_host_add with BCM_L3_FLAGS2_SCALE_ROUTE, to add UC routes to the LPM use bcm_l3_route_add with BCM_L3_FLAGS2_SCALE_ROUTE.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-90952	1015238	56450_A0 56450_B0 56450_B1	In earlier releases, packets are blocked in queue while enabling/ disabling external memory dynamically for a port. This has been fixed in this release. Due to HW limitation, all ports in a block will be affected if we want to change memory mapping dynamically for a port in that block.
SDK-90957	1014971	56850_A0 56850_A1 56850_A2	In the previous releases, one rack condition would cause both of SKB and BCM API Tx stuck. This issue has been fixed in this release.
SDK-90975	1015450	56960_A0 56960_B0 56960_B1	In previous releases, creating an L3 egress object with MPLS port was not supported. This has been implemented in this release.
SDK-90987	1015596	56850_A0 56850_A1 56850_A2	Parameters to TimeStampToCpu and its coloured variants are not properly read from the correct fields after warmboot. Fixed the same.
SDK-91003	1015512	56850_A0 56850_A1 56850_A2	In previous releases, warmboot function of BST module was not implemented. This has been added in this release.
SDK-91008	1013968	88660_A0	default for use_trunk_as_ingress_mc_destination soc property was changes in the following way: for BCM88650 the default is 0 for BCM88660 and BCM88670 the default is 1 this JIRA cause problems with stacking due to new default in BCM88670 and BCM88660. to prevent problems with stacking, use previous default - use_trunk_as_ingress_mc_destination=0 this default change meaning is that adding trunk destinations to mc group must be done in the new way (add trunk id gport to mc group) in relevant devices (Jericho and Arad Plus)
SDK-91011		56850_A0 56850_A1 56850_A2	In warmboot recovery, calling right group entry recovery method for EFP expanded slices and updated internal variable initializations
SDK-91024	1015667	56850_A0 56850_A1 56850_A2	The port number validation for qualifier bcmFieldQualifyDstPort and bcmFieldQualifySrcPort in a multi module system is set to appropriate value based on the chip.
SDK-91037	1013249	56860_A0 56860_A1	VLAN tagged packet drop issue should not be seen now as the HG port is been added properly the existing VLANs.
SDK-91072	1017510	AllChips	None
SDK-91078	1015424	56450_A0 56450_B0 56450_B1	The datatypes were included from <sys/types.h> through <stdlib.h>. To conform with C99 standard, the datatype declarations are included from <stdint.h>
SDK-91084	1014607	56340_A0	In previous release, IDRIVER value was cleared if configuring IPREDRIVER and POST2_COEFF. In this release, this issue has been addressed.
SDK-91086	1014330	88670_A0	Ports which are marked as reserved aren't being considered when allocating RCY channels after init
SDK-91124	1013840	88660_A0	Fix issue with 100G ports counters showing bad values after port disable-enable.
SDK-91138	1016489	56260_A0 56260_B0	Addressed the issues in bhh pool index free during endpoint create failure cases
SDK-91148	1016552	56850_A0 56850_A1 56850_A2	Syncing and recovering RedirectIpmc Action Type to set the parameters to right type after warmboot
SDK-91174	1015571	56850_A0 56850_A1 56850_A2	In previous releases, soc_mem_alpm_lookup could cause infinite loop and if there was a parity error in pipe_y then the function _soc_mem_alpm_ser_correction was unable to correct it. Now this has been fixed.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-91178	1012350	56850_A0 56850_A1 56850_A2	Before SDK_6_4_8 releases, the initial index of led data RAM was "1". However, in SDK_6_4_8 release, the initial index was changed as "0". So there was mismatch in led port order remap. Now the initial index is reset as "1".
SDK-91212	1016461	56450_A0 56450_B0 56450_B1	Non sequential units were not being parsed correctly in the diag shell, this is now fixed.
SDK-91214	1011941	88670_A0 88670_B0	Removing or adding dynamically a port will cause for MC packet to be dropped due to wrong CGM configuration. Fixed
SDK-91216	1002182	88670_A0	In KBP , to reduce the initialization time, the flag KBP_DEVICE_PROP_INST_LATENCY was removed for BCM88670 devices. This flag was added since the advanced instructions were sometimes causing ROP error due to flow control. The property disables the advanced instructions which will increase the update rate. this property is not needed in BCM88670.
SDK-91222	1015642	88670_A0	DNX configures autoneg mode in soc/dpp level (jer_nif.c). Autoneg configuration was added to portmod, which overrun DNX configuration. AUTONEG_CONFIG_SKIP flag added to portmod_port_add_info_t to enable skipping autoneg configuration in portmod. This flag is set on DNX (jer_nif.c)
SDK-91231	1014185	88670_A0	EVPN: No LSP tunnel configuration between PEs is now supported. Note: in this case PMF egress PE pre-selection cannot be done according to TT Termination
SDK-91348	1015662	88650_B1 88660_A0 88670_A0	DNX: Fixed an issue where writing to ESEM without a SW shadow may return success even if the writing was unsucessful.
SDK-91384	1017345	88660_A0	in IPMC, when the API bcm_ipmc_add() was called without setting the flag BCM_IPMC_REPLACE, "Entry not found" message was printed, as the new entry did not exist. Now the API bcm_ipmc_add() prints the error only in case BCM_IPMC_REPLACE is set. Also, the API bcm_ipmc_find() does not print an error message in case an entry is not found, but returns "entry not found" code value for further use. In API bcm_ipmc_remove() an error is printed if the entry does not exist.
SDK-91385	1016558	56850_A0 56850_A1 56850_A2	In previous releases, the DMA memory space used for packet sending was not freed by "bcmTX" in time, which resulted in DV allocation failures in the next attempt of sending packets by other threads. Now the pending DMA memory space is freed in time.
SDK-91395	1015021	88470_A0 88650_B1 88660_A0 88670_A0 88670_B0	In the previous release, SAV TT program used invalid key constructions and prefix value. In this release, this issue has been addressed by using correct key constructions and prefix value for DHCP IPv4 SAV TT program.
SDK-91411	1016304	56260_A0 56260_B0	Issue: The queues mapped to disabled ports are in enable state after flex operation. This is caused by the race condition between linkscan and port enable function during linkup sequence of flex operation. Solution: The link up sequence of flex operation is modified to enable the linkscan first and then enable/disable port based on default config.
SDK-91416		88670_A0	OTMH can now support 18bit CUD/Outlif values.
SDK-91419	1017128	88670_A0 88670_B0	in IPv4 forwarding, When working in External Tcam mode new forwarding lookup will be enabled in the LEM. This lookup will be with the highest priority compared to External TCAM. To manage entries in the LEM use flag BCM_L3_HOST_LOCAL.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-91421	1017467	AllChips	Calling the API `bcm_rx_init()` fails on device 56224, starting the SDK Release 6.5.1. This is due calling an internal routine, which is not supported on BCM56224. Added a check to prevent calling the internal routine on unsupported devices.
SDK-91426	1017726	`88670_B0`	In previous release, auto-neg does not work well on 1G eagle port with cl37 mode. In this release, auto-neg does work well on 1G eagle port with cl37 mode.
SDK-91430	1017562	`56850_A0 56850_A1 56850_A2`	In earlier releases, enabling L3RouteCache after several ipv4 or ipv6 routes added will cause unnecessary SER events and dirty cache which may result in alpm sanity command failure. This has been resolved.
SDK-91431	1012177	AllChips	`bcm_field_qualify_L3Ingress` API accepted only a 12bit inRif value, while the in-RIF range was extended to 15 bits in BCM88670. This qualifier was adjusted to accommodate the full range.
SDK-91432	1017957	`56860_A0 56860_A1`	Moved the `ing_attr` pointer into the ingress mode part and adding similar attribute `egr_attr` and setting the total counter values for the egress mode.
SDK-91446	1016916	`88670_A0 88670_B0`	In FCoE, delete and get LPM route (using flag `BCM_FCOE_DOMAIN_ROUTE)` failed due to internal error. Issue fixed and now it is possible to get and delete LPM entries.
SDK-91447	1013585	`88660_A0`	OAM: The API `bcm_oam_delay_get()` returns incorrect nanoseconds in `delay_min, delay_max,` delay fields. Actual nanoseconds value is 4 times greater than the returned value.
SDK-91457	1015571	`56850_A0 56850_A1 56850_A2`	.
SDK-91505		`56640_A0`	In previous releases, when all `L3_DEFIP` Tcams were paired in LPM scaling mode, the default routing entry cannot be set correctly. This has been fixed in this release.
SDK-91506		`88670_A0`	Minor bug fix. The following API may have been malfunctioning regarding order of entries on data base if flag `BCM_FIELD_GROUP_CREATE_INSERTION_ORDER_LOOSE` has been used: `bcm_field_group_config_create()`. Note that there was no effect on `bcm_field_group_create_id()`
SDK-91518	1018043	`56960_A0 56960_B0 56960_B1`	the sentence of #if 0 is added for TD2P DEV branch by Wilson, maybe in order to see some intermittent error with BCMSIM, then merged to master branch, It should be moved.
SDK-91553	1017189	`56460_A0 56460_B0`	1) Incorrect modid and port was used to construct gport in case of CoE subport . 2) Also subport port gport was not constructed during warmboot. The above issues have been fixed.
SDK-91564	1018613	`56460_A0 56460_B0`	In KT2/SB2 EFP Byte counter is 36 bits and IFP Byte counter is 35 bits. Used the corresponding counter update for updating the SW counter as per the stage.
SDK-91584		`88670_A0`	bcmFabricLinkRxFifoLLFC default for Jericho is set to 80 instead of 120.
SDK-91598	1017914	`56340_A0`	In previous releases, the `_soc_hx4_age_timer_get` API could return enable=1 and `age_seconds=0`. This issue has been fixed in this release.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-91657	1019274	56960_A0 56960_B0 56960_B1	Issue: IFP entries are used in BCM56960 and L2 entries are used in BCM56860 to assign session id to non-zero discriminator BFD packets. This needs to be removed as per customer's requirements. Fix: This is fixed by adding a new config property called "bfd_use_endpoint_id_as_discriminator". When this config property is configured (enabled by setting to 1), the local_discriminator value in the endpoint_info during bfd endpoint create will be ignored and discriminator is arrived at by adding a +1 to the session/endpoint_id. Hence there exists a direct linear mapping between session id and discriminator and those IFP/L2 entries are not required. The +1 is done to avoid zero endpoint_id/session_id giving out zero discriminator.
SDK-91680	1017339	AllChips	Phase holdover mode support is indicated as part of ptp servo config command
SDK-91751	1015676	56850_A0 56850_A1 56850_A2	For legacy flex stat modes, counter attribute information is not saved / retrieved from scache after warm boot. This causes new modes to be created for every user request, after warm boot, due to which the system runs out of HW counter modes (0-3).
SDK-91759	1019317	88670_B0	Sergey created a new label for BCM88670 B0 and I updated the mcmrelease wit the correct BS_PLL_CFG_RESET_B<173> and BS_PLL_CFG_POST_RESET_B<174> in ECI_BS_PLL_CONFIG for BCM88670 B0.
SDK-91761	1016153	88670_A0	The diagnostics command "diag dbal lp " caused a SDK crash. The issue happens only when the packet was using LPM lookup. This issue caused by incorrect use of signal value. This issue is fixed now.
SDK-91795	1018792	56860_A0 56860_A1	User provided TX PARAMS will be programmed for the port instead of default value during the interface set.
SDK-91796	1019327	56460_A0 56460_B0	Problem: pinfo->vlan_prot_ptr and vp entry is not recovered correctly after warmboot. Solution: Vp entry and Vlan protocol data index (vlan_prot_ptr) are storied in scache and retrieve back during warm boot.
SDK-91799	1018239	AllChips	All customers and all devices running Switch SDK within the "Affected Versions" have a risk of encountering table a memory leak. The memory leak appears whenever shr_htbl_destroy was used. Memory leak was observed in DNX OAM/BFD interrupt context. This issue may also occur in BCM56450 OAM. BCM56260 OAM and SAT, BCM56640 OAM, BCM88030 l2, BCM88030 allocator, BCM88030 mpls, BCM88030 oam, BCM88020 mpls, BCM88020 oam, BCM88020 l2.
SDK-91824	1019661	56440_A0 56440_A1 56440_B0	In KT EFP Byte counter is 36 bits and IFP Byte counter is 35 bits. Used the corresponding counter update for updating the SW counter based on the stage.
SDK-91852	1017672	AllChips	Problem: If we try to configure both AN(true) and Speed or duplex of a port via a single call of bcm_port_selective_set, the API returns success but the AN status set back to false. This is because as part of speed or duplex config the AN gets disabled. Solution: In case both AN (true) and Speed setting together then forced speed we allow only if AN disabled. If AN enabled for speed configure return param error. In case both AN (true) and duplex setting together then AN we enable and allow duplex to get auto-detected. For duplex configure return param error.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-91858	1016677	88670_A0 88670_B0	In L3, API bcm_l3_route_add() returned the error code BCM_E_INTERNAL (-1) for all error cases. Now BCM_E_FULL error code is returned in case the route DB has reached its limit.
SDK-91886	1017589	56850_A0 56850_A1 56850_A2	Fixed an issue that DPC of all attached units will be shutdown while only shutting down an specific unit.
SDK-91894	1019881	88670_A0	In earlier releases PPDB internal memory was not initialized. In this release, PPDB internal memory has been initialized.
SDK-91936	1019593	56860_A0 56860_A1	In this release, improved the time performance of API bcm_mpls_port_add() with flags BCM_MPLS_PORT_REPLACE and BCM_MPLS_PORT_EGRESS_TUNNEL specified for non failover port.
SDK-91939	1020205	88670_A0 88670_B0	Tunnel according DIP+SIP+vrf supported together with tst2 custom feature
SDK-91955	1019524	56860_A0 56860_A1	MY_MODID is the global parameter for the device. But it resides in PORT_TABm. When flexport is executed, the new entry will be created in PORT_TABm. As part of flexport sequence, current MY_MODID needs to be set to PORT_TABm.MY_MODIDf.
SDK-92025	1020140	88660_A0	OAM: destroying/updating OAM server on server side may fail.
SDK-92029	1011827	88670_A0 88670_B0	CINT example for Routing over AC (cint_route_over_ac.c) was fixed: 1. EVE is configured for tag_format_class_id 0 (since L2 header is terminated). 2. ecmp_intf is used (instead of fec_id) in l3_host_create API (Due to LEM entry size limitation).
SDK-92052	1009057	AllChips	Fix to handle PTP master's port re-configuration with new IP address.
SDK-92057	1020287	88670_A0	Now when using the bcm_l3_intf_get the Forwarding MTU will be updated if it was set (the mtu_advanced_mode SOC is mandatory for this)
SDK-92065	1019467	88670_A0 88670_B0	check resilient_hash_enable soc in init, if its enabled on non supported device an error is thrown
SDK-92068	1020449	56640_A0 56640_A1 56640_B0	Issue ==== During warmboot recover, control pool info related to slice was not getting updated properly. In TR3, when a stat is create, one of the counter pool is reserved for that slice. In the given configuration after 2 group with entries are created. Pool id 0 was reserved for entries in slice 0 or group 14 & FP_POLICY_TABLE.ipipe0[0]: COUNTER_SET=0x54000,> // COUNTER_POOL_NUMBER=0 & COUNTER_POOL_INDEX=0 Pool id 1 was reserved for entries in slice 1 or group 20 FP_POLICY_TABLE.ipipe0[512]: COUNTER_SET=0x54200,COUNTER_POOL_NUMBER=1, > // COUNTER_POOL_NUMBER=1 & COUNTER_POOL_INDEX=0 During warmboot recovery, while recovering entries, SW link between counter pool and slice idx was not update properly, due to this SW info says all the counter are free. When a new entry is created for slice 1 or group 20, and stat is attached for it. SW tries to allocated pool id 0 for slice 1. Ideally it should have allocated from pool id 1 as it was reserved for slice 1. Fix == Updated slice id mapping in counter pool during stat recover.
SDK-92081	1014514	88660_A0	CRPS issue: time of bcm_field_stat_multi_get API is too long. the reason is that there are many reading from SW DB. Solution: reduce the number of times reading parameters from SW DB.

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-92086	1018861	88650_A0 88660_A0 88670_A0	When using more than 10 unique OAM groups, The transmitted CCMs of endpoints (belongs to the 10+ groups) had corrupted MAID (MEG ID) value. This bug is fixed.
SDK-92120	1020402	56960_A0 56960_B0 56960_B1	bcm_field_stat_id_get will not fail even if some entries the group have no stats attached to it.
SDK-92122	1020604	56850_A0 56850_A1 56850_A2	In the previous release, Endpoint Queuing was not supported for uc queue assignment per PW in VPWS. In this release, Endpoint Queuing is supported for uc queue assignment per PW in VPWS.
SDK-92123	1019280	88660_A0	This feature is working now.
SDK-92134	1020706	56224_B0	On Raven, while syncing L3 module for warmboot, there was no check to confirm if L3 feature is enabled or not. This is being fixed.
SDK-92136	1015995	56450_A0 56450_B0 56450_B1	bcmPortControl1588P2PDelay option can be used for setting the LINK_DELAY and retrieving the set value on all platforms where EGR_1588_LINK_DELAY or EGR_1588_LINK_DELAY_64 register available
SDK-92138	1016995	56620_B0	Added fix to generate SOC_SWITCH_EVENT_PARITY_ERROR event is detected
SDK-92139	1019827	53342_A0 53343_A0 53344_A0 53346_A0	The parity might not be disabled during memory test for 56150 family when the parity_enable=0 was configured in the previous release. Fix the parity config checking to enable/ disable the parity properly.
SDK-92160	1019858	56860_A0 56860_A1	An error in soc_td2p_mmu_flexport_allocate_deallocate_ports() where physical ports were being indexed from 0-127 rather than 1-128, which caused MMU Ports to be improperly marked as free (physical port 128's MMU Port was never marked as in use because of this off by one). Thus, two ports were being assigned the same MMU Port and therefore the same queues.
SDK-92183	1020413	56860_A0 56860_A1	We are setting the PGW_LED_CHAIN_OUT_CONFIG register for all the PGW blocks irrespective of whether it is active or not.
SDK-92203	1020441	56860_A0 56860_A1	The fix for this defect updates subsidiary ports get functionality for TD2+ so that correct port bitmap is returned to the calling application. Based on the bitmap returned the application can make decisions about flexing of ports within a port block/port macro.
SDK-92228	1005248	56565_A0	Two new events given below are added to receive notifications on receiving bfd packet with poll/final bit. bcmBFDEventEndpointRemotePollBitSet, /* Received a packet from remote with Poll Bit set */ bcmBFDEventEndpointRemoteFinalBitSet, /* Received a packet from remote with Final Bit set */ A new config property "bfd_session_down_event_on_create" has been added to enable the behavior of sending state change (down) notification after the session is created.
SDK-92234	778524	56565_A0	Added support for 64bit statistics counters for BFD endpoints.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-92249	937135	56260_A0 56460	Issue: When one among bcmOAMActionFwd/UcFwd/McFwd/ LowMdlFwd/MyStationMissFwd actions is configured, the internal software pointers for the DGLP are updated. Upon second time configuration of one of those actions, these software pointers are checked and if they are set, in that case, we were not setting appropriate values for FWD actions in OAM_OPCODE_CONTROL_PROFILE setting function. Fix: Fixed by adding appropriate values for FWD actions within those checks for software pointers.
SDK-92274	1020846	56450_A0 56450_B0 56450_B1	The CPU HG port 42 needs to be set to HG2 mode in KT2.
SDK-92280	1021062	88670_A0	MPLS: If the result of MPLS label lookup in the forwarding stage is FEC+Outlif, the following configuration should be used in bcm_mpls_tunnel_switch struct: egress_if - outlif value encoded as BCM interface of type OutLif port - FEC value encoded as BCM gport of type forwarding port
SDK-92310	1021225	56340_A0	Threshold settings for SC and QM queues (queue number 8 and 9) were blocked in the SDK. These have now been enabled.
SDK-92345	1021506	56334_A0 56334_B0	Issue: Wrongly calculating the trunk member due to that failed to initialize the mirror module after disabling the trunk members. Solution: Retrieving the right active members during the mirror initialization.
SDK-92356	1020879	88660_A0	"g * debug" will cause bus error at the device unless "phy_1588_dpll_frequency_lock=1" SoC property is defined.Fixed.
SDK-92364	1017720	88670_A0 88670_B0	Configuration of VLAN mc ports with bcm_multicast_set fixed.
SDK-92367	1019668	56860_A0 56860_A1	Added the support for the PHASE_INTERP on td2+
SDK-92463	1017019	88670_A0	Full range of MC ID may be used up to defined in the system
SDK-92492	1021749	88670_A0	In the previous release, the API bcm_port_timesync_config_set can't disable 1588 mac function on port. In this release, this issue has been addressed by setting the value of field 'IEEE_1588_MAC_ENABLEf' to 0.
SDK-92538	1021178	88950_a0	Added the faulty registers to the ignore list and it is working now.
SDK-92568	1010148	88670_A0	In earlier releases the table PPDB_A_OEMA_KEYT_PLDT_H was not initialized on BCM88670. In this release, this table has been initialized.
SDK-92625	1021903	88950_a0	In earlier releases there was no recover action when initiate FE3200 RTP ECC error. In this release, added recover action for FE3200 RTP ECC.
SDK-92676	1022834	AllChips	IBOD WAR is being executed on the INVALID ports as well. VALID ports check fix the issue.
SDK-92681		88670_A0	Wrong way to compare the error code. Extract the error code from the return value
SDK-92697	1022680	88670_A0 88670_B0	When a fid is disconnected from a learn profile and it was the last fid using the profile, this profile is cleaned. In case the same fid immediately gets the same profile in _bcm_dpp_am_template_l2_learn_profile_exchange, this profile should not be cleared. Now the learn profile is not cleared in this case
SDK-92736	1021717	56860_A0 56860_A1	flexport sequence will check if the system is in ets mode. is ets mode is enabled, i.e the customer is building its own lls hierarchy, it will skip building default lls for the newly inited ports.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-92752	1019451	AllChips	In the previous release, PON tunnel ID is limited to 2016 for all ports, which could not met all customers. In this release, this issue has been addressed by adding a SOC configuration to select tunnel ID allocation mode.
SDK-92760	1019010	56063	L2_ENTRYf is not supported for Ranger2 device, instead it has L2_ENTRY_DATA field. Due to which, SDK throws assertion while executing the L2 Mod FIFO process. Fix: Validating the field before configuring it.
SDK-92767	1021988	88670_A0 88670_B0	Deallocation and creating new extended inLIFs failed in some cases due to incorrect deallocation scheme. This was fixed.
SDK-92782	1023464	88670_A0 88670_B0	In the previous release, aging configuration was updated incorrectly. In this release, this issue has been addressed by not updating the aging configuration when bcm_vlan_control_vlan_set() is called without aging profile and configuration.
SDK-92798	1014937	56860_A0 56860_A1	After BFD session is up, the application needs to wait till the poll bit is cleared before initiating a parameter change. Else, the firmware would return a BCM_E_BUSY error. The issue of endpoint_get returning the changed parameter after endpoint_create failure has been fixed by copying the old endpoint config back in failure cases.
SDK-92809	1023573	56547_A0 56548_A0	Issue ===== Eventhough, EGR_IP_TUNNEL & EGR_IP_TUNNEL_IPV6 maps to the same memory, cache for those tables are maintained differently. When a ip4/ipv6 tunnel is cleared, corresponding cache was not marked as invalid, so any ready for the table will fetch from cache instead of HW. Tunnel clear API always relies on the ENTRY_TYPE in EGR_IP_TUNNEL table to determine if its V4 or V6 view. Since cache was cleared for EGR_IP_TUNNEL when ipv4 tunnel was cleared, ENTRY_TYPE was read as zero and due to this API was not able to determine this as ipv6 tunnel. Due to this ipv6 tunnel was not getting cleared. Fix === Added check to clear cache valid bit when tunnel is cleared from either of the views.
SDK-92842	997252	56640_A0 56640_A1 56640_B0	Added SDK changes to configure CLASSID in EGR_L3_NEXT_HOP for customer port also
SDK-92869		88670_A0 88670_B0	In previous releases, the first route interface profile was configured improperly. This has been resolved.
SDK-92873	1016424	88660_A0 88670_A0	FEC: Redundant error messages were removed by initalizing FEC table to drop destination.
SDK-92875	1022739	88660_A0	When SOC property bcm886xx_ipv6_tunnel_enable was set to '0', VXLAN or any IP tunnel type was not terminated properly, due to wrong processing in the tunnel termination stage. The tunnel termination ucode has been changed to support correct handling when bcm886xx_ipv6_tunnel_enable is set to '0' or '1'.
SDK-92891	1025033	56850_A0 56850_A1 56850_A2	In previous releases, parity error in the memory MMU_WRED_UC_QUEUE_TOTAL_COUNT_X_PIPE would not be corrected. This issue has been addressed in this release.
SDK-92901	1023462	88670_A0 88670_B0	When setting itmh_programmable_mode_enable=0 and itmh_arad_mode_enable=1 on BCM88670, Arad-format ITMH header is now parsed properly.
SDK-92908	1025087	88670_A0 88670_B0	remove the logic ,so that removing operation on a protected FEC entry, it will just remove this entry simply.
SDK-92935		88670_A0 88670_B0	Removed the DPP devices from the supported devices of the test.
SDK-93006	1021778	56260_A0 56260_B0	Internal registers file updated

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-93012	1022525	88650_B1	Actions setup for Arad is now operational (Could fail depending on random value on stack). Could fail in bcm_field_group_action_set()
SDK-93080	1025402	56850_A0 56850_A1 56850_A2	In previous releases, the L3_DEFIP_PAIR_128_DATA_ONLY view used two srams, if there was an error in second sram, this error can be detected but cannot be recovered. Now this issue has been fixed.
SDK-93095	1023648	88660_A0	In previous releases, cosq templates might run out in some API sequences, this has been resolved.
SDK-93156	1007209	56260_A0	Implemented the couner rollover flag and event for OAM PM in BHH. This will make customer aware of any true counter rollover or board reboot.
SDK-93172	1003534	AllChips	"ptp time get" and "mcsts 16" full time are derived from same ptp local time (SYS) which is basically TS1 time. Firmware keeps TS1 time in 64bit format . However TS Events from TS FIFO comes in 48bit format. "mcsts 16"'s full time calculation is modified to address wrap around of 48bit TS Events.
SDK-93219	1025051	88650_B1 88660_A0	In earlier releases the table SCH_SHAPER_DESCRIPTOR_MEMORY_STATIC__SHDS was not protected. In this release, this table has been protected by parity.
SDK-93253		88670_A0	When setting ilkn_tdm_dedicated_queueing=1 soc property device fails during init. Fixed
SDK-93335	1026819	56260_A0 56260_B0 56460 56460_A0 56460_B0	Setting GREEN_TO_PID to 1 by default in POLICY Table is limited to IFP and Exact Match Stages. But it also applies to EFP on some chips. So By default setting GREEN_To_PID to 1 for EFP policy table as well.So that redirect actions will be applicable to all colored packets.
SDK-93388	1010786	56450_A0 56450_B0 56450_B1	DFC need to be reset before flush.
SDK-93393	1027305	AllChips	In previous releases, it was wrong to configure one flexible port as SOC_TH_PORT_MODE_TRI_023 or SOC_TH_PORT_MODE_TRI_012 modes when SDK had configuration "oversubscribe_mode=1". Now, when SDK has "oversubscribe_mode=1", every flexible port can't configure as "tri-ports" mode in TH.
SDK-93468		88670_A0	In earlier releases the table OAMP_MEM_20000 was cached for the device BCM88670 . In this release, the table OAMP_MEM_20000 has changed to cached.
SDK-93469		88670_A0	In earlier releases the severity of the table MRPS_MCDB_PRFSEL was not right. In this release, the severity of this table has been corrected.
SDK-93498	1021591	56960_A0 56960_B0 56960_B1	bcm_field_group_oper_mode_set call for IFP will not fail if soc_feature_field_exact_match_support is not supported on the device.
SDK-93579	1028046	88670_A0 88670_B0	for update operation, the destination field will be generated from hardware lif table.
SDK-93678	1027498	56860_A0 56860_A1	In previous releases, default value of num_ipv6_lpm_128b_entries is always 2048 regardless of value of ipv6_lpm_128b_enable, this required customer to explicitly configure num_ipv6_lpm_128b_entries=0 in config.bcm while ipv6_lpm_128b_enable is set to 0. In this release, default value of num_ipv6_lpm_128b_entries is changed to 0 while num_ipv6_lpm_128b_entries=0.
SDK-93684	1019317	88670_A0	Attached the new label provided by the Asic team and made the changes for the registers.

Table 99:

Number	CSP	Chips	Release Notes For 6.5.3
SDK-93685		88670_A0	IPS_DEQ_COMMAND_CREDIT_ADJUST was not set properly, so in case of full rate traffic drops may occur. Fixed
SDK-93689		88670_A0	In earlier releases exact match tables SER action was not support. In this release, exact match tables do ECC SER action.
SDK-93749	1028338	56860_A0 56860_A1	ASF_HS_FIFO_THRESHOLD for Trident2+ isn't set to the arch recommended value (13). This has now been updated to the recommended value indicated in the MMU settings spread sheet for Trident2+.
SDK-93760	1027974	56860_A0 56860_A1	Release will go into master branch,
SDK-93836	1029007	88660_A0	In earlier releases the cached tables were not converted when process ECC SER issue on BCM88660. In this release, the cached tables has been converted.
SDK-93854	1025369	56640_A0 56640_A1 56640_B0	"bs conf master" command from diag shell with PTP firmware can be used to turn on Broadsync with PTP.
SDK-93968	1028558	56440_B0 56445_A0 56445_A1 56445_B0	MPLS__USE_DVP_IN_EVXLT_KEYf should not be set for SD_TG view.
SDK-94051		88950_a0	In earlier releases the wide memory does not need any special process on FE3200. In this release, the wide memory process has been corrected.
SDK-94052		88950_a0	In earlier releases hard reset handler was not supported on FE3200. In this release, hard reset handler has been supported.
SDK-94158		88750_A0 88950_a0	In earlier releases SER operation of the table RTP_SLSCT was not right. In this release, SER operation of the table RTP_SLSCT added special handing.
SDK-94331	1030153	88670_A0	MPLS PORT: Api bcm_mpls_port_add BCM_MPLS_PORT2_LEARN_ENCAP might fail on validity check when calling with replace flag
SDK-94345	1028146	56060_A0 56150_A0 56160_A0	Fix the issue of multiple registers with the same switch control type is not set correctly for bcmSwitchMeterAdjust.
SDK-94701	1031826	88670_A0 88670_B0	TDM port is added twice, might cause memory override. Fixed
SDK-94707		88470_A0 88670_A0	MPLS: Additional labels encapsulation is now supported through new apis: To enable the feature set soc property mpls_egress_label_extended_encapsulation_mode to 1. To add MPLS tunnel using LL entry: 1) bcm_l3_egress_create Add flag Flag BCM_L3_EGRESS_WIDE to configure wide entry 2) bcm_port_control_set Set class bcmPortControlMPLSEncapsulateAdditionalLabel 3) Configure Labels C3 and D1 by calling bcm_port_wide_data_set Gport is the LL Flag BCM_PORT_WIDE_DATA_EGRESS Data is 20bit MPLS label To add MPLS tunnel using existing mpls tunnel: 1) bcm_mpls_tunnel_initiator_set Set flag BCM_MPLS_EGRESS_LABEL_WIDE to indicate wide entry 2) bcm_port_control_set Set class bcmPortControlMPLSEncapsulateAdditionalLabel 3) Configure Labels C3 and D1 by calling bcm_port_wide_data_set Gport is the mpls tunnel Flag BCM_PORT_WIDE_DATA_EGRESS Data is 20bit MPLS label For example see cint cint_ip_route_tunnel_segment_routing.c
SDK-94773	1033104	88660_A0	In earlier releases SER action for software reset was not supported in BCM88650. In this release, SER action for software reset has been supported in BCM88650.
SDK-94909		88670_A0	Removing dynamically ILKN port causes EGQ to stuck. Fixed

Table 99:

Number	*CSP*	*Chips*	*Release Notes For 6.5.3*
SDK-95033	1032471	88650_A0 88660_A0 88670_A0	Fix a bug where Init could potentially crash during warm reboot on 64-bit systems.

Section 10: Unresolved Issues for 6.5.3

The following open Urgent priority issues are unresolved in version 6.5.3 of the SDK.

Table 100:

Number	*CSP*	*Chips*
SDK-67114	863828	56850_A0 56850_A1 56850_A2
SDK-68785		88660_A0
SDK-81696	957889	56960_A0
SDK-84364	966003	AllChips
SDK-84580	971485	88660_A0
SDK-87026	992730	56960_A0 56960_B0
SDK-88872	1003096	56860_A0 56860_A1
SDK-89832	1006412	56860_A0 56860_A1
SDK-89879	1008577	AllChips
SDK-90885	1008608	56340_A0
SDK-92245	1009233	56260_A0 56260_B0
SDK-92294	968786	84741_C0 84756_A0 84756_C0
SDK-93549	1005565	88660_A0
SDK-94115	1028719	56260_A0 56260_B0
SDK-94162		88670_B0
SDK-94211	1023560	56450_A0 56450_B0 56450_B1
SDK-94883	972956	88670_A0
SDK-95119		88470_A0
SDK-95132	1031344	88650_A0 88650_B0 88650_B1 88660_A0

Section 11: Device and Platform Support

The section describes all devices, platforms, and operating systems that are supported by this release.

SWITCH DEVICES

Table 101: Switch Devices

Family	Devices	Description
BCM5389	BCM5389 A0	8-Port GbE Switch with Integrated Serdes
	BCM5389 A1	8-Port GbE Switch with Integrated Serdes
BCM5396	BCM5396 A0	16-Port GbE Switch with Integrated Serdes
BCM53010	BCM53010 A0	5-Port Gigabit Ethernet Managed Switch integrated with single core ARM Cortex-A9 processor
	BCM53010 A2	
	BCM53011 A0	5-Port Gigabit Ethernet Managed Switch integrated with dual cores ARM Cortex-A9 processor
	BCM53011 A2	
	BCM53012 A0	5-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53012 A2	
BCM53018	BCM53017 A0	2-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53018 A0	5-Port Gigabit Ethernet Managed Switch with one RGMII I/F integrated with dual cores ARM Cortex-A9 processor
	BCM53019 A0	5-Port Gigabit Ethernet Managed Switch integrated with dual cores ARM Cortex-A9 processor
BCM53020	BCM58522 A0	5-Port Gigabit Ethernet Managed Switch integrated with 2 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58525 A0	5-Port Gigabit Ethernet Managed Switch integrated with 2 PHYs, SGMII I/F, ARM Cortex-A9 processor and macsec cores
	BCM58622 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58623 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, ARM Cortex-A9 processor and macsec cores
	BCM58625 A0	8-Port Gigabit Ethernet Managed Switch integrated with 5 PHYs, SGMII I/F, ARM Cortex-A9 processor and macsec cores
BCM53101	BCM53101 A0	5-Port Fast Ethernet Managed Switch + 1 Fast Ethernet WAN port
	BCM53101 B0	
BCM53115	BCM53115 A0	5-Port GbE Managed Switch + 1 Gigabit WAN port with integrated serdes
	BCM53115 A1	
	BCM53115 B0	
	BCM53115 B1	
	BCM53115 C0	
BCM53118	BCM53118 A0	8-Port Gigabit Ethernet Switch
	BCM53118 B0	
	BCM53118 B1	
BCM53125	BCM53125 A0	5-Port Gigabit Ethernet Switch with 1 Gigabit WAN port and 8051 processor
	BCM53125 B0	
BCM53128	BCM53128 A0	8-Port Gigabit Ethernet Switch with embedded 8051 processor
	BCM53128 B0	
BCM53242	BCM53242 A0	Managed Switch with 24 FE Ports + 2 GbE Interface
	BCM53242 B0	

Table 101: Switch Devices

Family	Devices	Description
	BCM53242 B1	
	BCM53262 A0	Managed Switch with 24 FE Ports + 4 GbE Interface
	BCM53262 B0	
	BCM53262 B1	
BCM53280	BCM53282 A0	8-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53282 B0	
	BCM53282 B1	
	BCM53282 B2	
	BCM53283 A0	16-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53283 B0	
	BCM53283 B1	
	BCM53283 B2	
	BCM53284 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53284 B0	
	BCM53284 B1	
	BCM53284 B2	
	BCM53286 A0	24-Port Fast Ethernet + 4-Port Gigabit Ethernet Multilayer Switch
	BCM53286 B0	
	BCM53286 B1	
	BCM53286 B2	
	BCM53288 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch with one 2.5GbE Uplink Port
	BCM53288 B0	
	BCM53288 B1	
	BCM53288 B2	
BCM53300	BCM53300 A0	Managed 24-port L2 Switch
	BCM53300 A1	
	BCM53301 A0	Managed 16-port L2 Switch
	BCM53301 A1	
	BCM53302 A0	Managed 24-port L2 Switch
	BCM53302 A1	
BCM53310	BCM53312 A0	BCM53312 Integrated Multilayer Switch and CPU
	BCM53312 B0	
	BCM53313 A0	BCM53313 Integrated Multilayer Switch and CPU
	BCM53313 B0	
	BCM53314 A0	BCM53314 Integrated Multilayer Switch and CPU
	BCM53314 B0	
BCM53320	BCM53322 A0	BCM53322 Integrated Multilayer Switch and CPU
	BCM53323 A0	BCM53323 Integrated Multilayer Switch and CPU
	BCM53324 A0	BCM53324 Integrated Multilayer Switch and CPU
BCM53400	BCM53402 A0	8 x 1G/2.5G/5G/10G
BCM53400	BCM53405 A0	16-port 10GbE Multilayer Ethernet Switch

Table 101: Switch Devices

Family	Devices	Description
	BCM53406 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Multilayer Ethernet Switch
	BCM53412 A0	8 x 1G/2.5G/5G/10G
	BCM53415 A0	16-port 10GbE Multilayer Ethernet Switch with integrated CPU
	BCM53416 A0	12-port 10GbE plus 8-port 2.5GbE and 4-port 5GbE/2.5GbE Ethernet Switch with integrated CPU
	BCM53422 A0	8 x 1G + 2 x 1G/2.5G/5G/10G
	BCM53424 A0	4 x QSGMII + 8 x 1G + 4 x 1G/2.5G/5G/10G (option1)
	BCM53424 A0	4 x QSGMII + 8 x 1G + 2 x 10G + 2 x HiGigDuo[13] (option2)
	BCM53424 A0	2 x QSGMII + 16x1G + 4 x 1G/2.5G/5G/10G (option3)
	BCM53426 A0	20 x 1G + 4 x 1G/2.5G/5G/10G
	BCM53454 A0	20 x 1G/2.5G + 4 x 1G/2.5G/5G/10G
	BCM53455 A0	20 x 1G/2.5G + 4 x 1G/2.5G/5G/10G with embedded ARM A9 processor
	BCM53456 A0	4 x QSGMII + 8 x 1G/2.5G + 4 x 1G/2.5G/5G/10G (option1)
	BCM53456 A0	4 x QSGMII + 8 x 1G/2.5G + 2 x 10G + 2 x HiGigDuo[13] (option2)
	BCM53456 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G (option3)
	BCM53457 A0	4 x QSGMII + 8 x 1G/2.5G + 4 x 1G/2.5G/5G/10G with ARM A9 (option1)
	BCM53457 A0	4 x QSGMII + 8 x 1G/2.5G + 2 x 10G + 2 x HiGigDuo[13] with ARM A9 (option2)
	BCM53457 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G with ARM A9 (option3)
BCM53600	BCM53602 A0	8-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53603 A0	16-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53604 A0	24-Port Fast Ethernet + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
	BCM53606 A0	24-Port FE with S3MII interface + 3-Port Gigabit Ethernet Switch with one 1/2G-EPON ONU MAC/SerDes and embedded 600MHz MIPS32 74K processor
BCM89500	BCM89500 A0	4-Port Integrated Dedicated BRPHY + 3-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89500 B0	
BCM89500	BCM89501 A0	4-Port Integrated Dedicated BRPHY + 1-Port Integrated Dual-Mode BRPHY + 2-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89501 B0	
BCM89500	BCM89200 A0	1-Port Integrated Dedicated BRPHY + 1-Port Integrated Dual-Mode BRPHY + 2-Port Gigabit Ethernet Switch with embedded ARM processor
	BCM89200 B0	
BCM53710	BCM53714 A0	BCM56714 Integrated Multilayer Switch and CPU
	BCM53714 A1	
	BCM53714 A2	
	BCM53716 A0	BCM56716 Integrated Multilayer Switch and CPU
	BCM53716 A1	
	BCM53716 A2	
	BCM53718 A0	BCM56718 Integrated Multilayer Switch and CPU
	BCM53718 A1	
	BCM53718 A2	
BCM53720	BCM53724 A0	Managed 24-port L2 Switch with Integrated CPU

Table 101: Switch Devices

Family	Devices	Description
	BCM53724 B0	
	BCM53726 A0	Managed 24-port L2 Switch with Integrated CPU
	BCM53726 B0	
BCM5670	BCM5676 A0	4-Port, 96-Gbps Switch Fabric
	BCM5676 A1	
BCM56020	BCM56024 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56024 B0	
	BCM56025 A0	24-Port Integrated L2 Switch and CPU
	BCM56025 B0	
	BCM56026 A0	24-Port Integrated L2 Switch and CPU
	BCM56026 B0	
BCM56060	BCM56060 A0	16-port 10GbE Multilayer Ethernet Switch with integrated CPU
BCM56060	BCM56062 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G (option1)
	BCM56062 A0	2 x QSGMII + 8 x 1G/2.5G + 2 x XAUI + 4 x 1G/2.5G/5G/10G (option2)
	BCM56062 A0	2 x QSGMII + 16 x 1G/2.5G + 4 x 1G/2.5G/5G/10G (option3)
	BCM56063 A0	16-port 1GbE (QSGMII) plus 4-port 1G/2.5G/5G/10G (XFI) Multilayer Switch with integrated CPU
	BCM56064 A0	24-port GbE (4xQSGMII, 8xSGMII) plus 4-port 10GbE Multilayer Managed Switch with HiGi Uplinks and integrated CPU
	BCM56064 A0	4 x QSGMII, 8 x 1GbE, 4 x 10GbE (option1)
	BCM56064 A0	4 x QSGMII, 8 x 1GbE, 2 x 10Gbe, 2 x HiGigDuo[13] (option2)
	BCM56065 A0	12 x 1G/2.5G/5G/10G + 12 x 1G/2.5G
BCM56130	BCM56132 A0	24-Port Fast Ethernet Multilayer Switch with Two 10-GbE/HiGig2 and Two 1G/2.5Gb Uplink Ports
	BCM56132 B0	
	BCM56132 B1	
	BCM56134 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb Uplink Ports
	BCM56134 B0	
	BCM56134 B1	
BCM56150	BCM56150 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM56151 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU (without PHYs)
	BCM56152 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks Managed Switch, integrated CPU and 16 copper PHYs
	BCM53342 A0	8-port GbE Multilayer WebSmart Switch with Integrated CPU and Copper PHYs
	BCM53343 A0	16-port GbE plus 4-port GbE uplinks Multilayer WebSmart Switch with Integrated CPU and 16 Copper PHYs
	BCM53344 A0	24-port GbE plus 2-port GbE and 2-port 1GbE/13GbE uplinks WebSmart Switch, integrated CPU and 16 copper PHYs
	BCM53346 A0	24-port GbE Multilayer WebSmart Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
	BCM53347 A0	24-port GbE Multilayer WebSmart Switch with 6xQSGMII + 4x1/10G
	BCM53393 A0	14-port GbE Multilayer Embedded Switch with integrated CPU (without PHY)
	BCM53394 A0	10-port GbE Multilayer Embedded Switch with 4-port 10 GbE uplinks, integrated CPU (without PHY)
BCM56220	BCM56224 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+

Table 101: Switch Devices

Family	Devices	Description
	BCM56224 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56225 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56225 B0	24 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56226 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56226 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56227 A0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56227 B0	16 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56228 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56228 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	BCM56229 A0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
	BCM56229 B0	8 GbE + 4 x 1 Gb/2.5 Gb, L2+
BCM56230	BCM56230 B1	12-Port GbE Multilayer Switch
	BCM56231 B1	6-Port GbE Multilayer Switch
BCM56260	BCM56260 A0	50 Gbps Integrated Lower Power Carrier Ethernet Access Switch
BCM56320	BCM56320 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56320 B0	
	BCM56320 B1	
	BCM56321 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56321 B0	
	BCM56321 B1	
BCM56330	BCM56331 A0	24-Port GbE Multilayer Switch with Four 2.5GbE Uplink Ports
	BCM56331 B0	
	BCM56331 B1	
	BCM56333 A0	16-Port GbE Multilayer Switch
	BCM56333 B0	
	BCM56333 B1	
	BCM56334 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56334 B0	
	BCM56334 B1	
	BCM56338 A0	8-Port GbE Multilayer Switch with two 10-GbE/HiGig2 Uplink Ports
	BCM56338 B0	
	BCM56338 B1	
BCM56340	BCM56040 A0	1xF.QSGMII + 3xF.HG[42] + 1GE
	BCM56041 A0	BCM56040 device, meant for embedded connectivity supports 1Ge (port 49), 2 X GE (iPROC), Flex 4x10G, 3 X 4 X 10G
	BCM56042 A0	12x2.5GE/1GE + 12x2.5GE/1GE + 1GE
	BCM56340 A0	12xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE, 12xF.QSGMII + 4xSGMII + 2xXFI + 2xHGd[21] + 1GE
	BCM56342 A0	7xF.QSGMII + Flex[4x10] + 2xHG[21] + 1GE
	BCM56344 A0	10xF.QSGMII + 3xFlex[4x10] + 1GE
BCM56340	BCM56547 A0	10xF.QSGMII + 3xF.HG[42] + 1GE, 12xF.QSGMII + 2xF.HG[42] + 1GE, 12xF.QSGMII + F.HG[42] + 2xHG[42] + 1GE
BCM56340	BCM56548 A0	7xF.QSGMII + 3xF.HG[42] + 1GE
BCM56440	BCM56440 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports

Table 101: Switch Devices

Family	Devices	Description
	BCM56440 B0	
	BCM56441 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56441 B0	
	BCM56442 A0	16-Port GbE Multilayer Switch
	BCM56442 B0	
	BCM56443 A0	8-Port 2.5GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56443 B0	
	BCM56445 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports pin compatible with BCM56334
	BCM56445 B0	
	BCM56446 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports pin compatible with BCM56338
	BCM56447 A0	16-Port GbE Multilayer Switch pin compatible with BCM56333
	BCM56447 B0	
	BCM56448 A0	24-Port GbE Multilayer Switch with Four 1GbE/ One 2.5G Uplink ports
	BCM56448 B0	
BCM56450	BCM56248L B0	11xGE + 8x2.5G
BCM56450	BCM56450 A0	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager
BCM56450	BCM56450 B0	BCM56450 Access 1 x XAUI + 8 x GE without L3 routing and MPLS features
BCM56450	BCM55450 B0	BCM56450 Access-8 FX + 2 F-HG
BCM56450	BCM56450 B1	24-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking, integrated CPU and Traffic Manager
BCM56450	BCM56452 B0	24xGE + 4xF.XAUI
BCM56450	BCM56454 B0	8xGE + 2 x F.XAUI
	BCM56455 A0	2 x 20GE (G.INT) + 2 x HG13
	BCM56455 B0	2 x 20GE (G.INT) + 2 x HG13
BCM56450	BCM56456 A0	24x GE + 4x F.XAUI
BCM56450	BCM56456 B0	24x GE + 4x F.XAUI
	BCM56456 B0	1 x XAUI + 8 x GE
BCM56456	BCM56456 B0	9xFXAUI + 1 x XAUI + 1x2.5GbE
BCM56456	BCM56458 B0	8xGE + 2xF.XAUI
BCM56460	BCM56460 A0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplink
	BCM56461 A0	24 x GE + 4 x HGs11
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	
	BCM56500 B0	
	BCM56500 B1	
	BCM56500 B2	
	BCM56501 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56501 A1	
	BCM56501 B0	
	BCM56501 B1	
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports

Table 101: Switch Devices

Family	Devices	Description
	BCM56502 A1	
	BCM56502 B0	
	BCM56502 B1	
	BCM56502 B2	
	BCM56503 A0	24-Port GbE Multilayer Switch with Three 10-GbE/HiGig+ Ports
	BCM56503 A1	
	BCM56503 B0	
	BCM56503 B1	
	BCM56503 B2	
	BCM56504 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig+ Ports
	BCM56504 A1	
	BCM56504 B0	
	BCM56504 B1	
	BCM56504 B2	
	BCM56505 A0	24-Port GbE Layer 2 Switch
	BCM56505 A1	
	BCM56505 B0	
	BCM56505 B1	
	BCM56505 B2	
	BCM56506 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56506 A1	
	BCM56506 B0	
	BCM56506 B1	
	BCM56506 B2	
	BCM56507 A0	24-Port GbE Layer 2 Switch with Two 10-GbE/HiGig+ Ports
	BCM56507 A1	
	BCM56507 B0	
	BCM56507 B1	
	BCM56507 B2	
	BCM56508 A0	24-Port GbE Layer 2 Switch with Three 10-GbE/HiGig+ Ports
	BCM56508 A1	
	BCM56508 B0	
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	
	BCM56509 B1	
	BCM56509 B2	
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch

Table 101: Switch Devices

Family	*Devices*	*Description*
	BCM56512 A0	24-Port GbE Multilayer Switch With Two 10-GbE/HiGig+ Ports
	BCM56513 A0	24-Port GbE Multilayer Switch With Three 10-GbE/HiGig+ Ports
	BCM56514 A0	24-Port GbE Multilayer Switch With Four 10-GbE/HiGig+ Ports
BCM56520	BCM56520 A0	24-Port GbE Multilayer Switch
	BCM56520 B0	
	BCM56522 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig2 Uplink Ports
	BCM56522 B0	
	BCM56524 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56524 B0	
	BCM56526 A0	28-Port GbE Multilayer Switch with Six 10-GbE/HiGig2 Uplink Ports
	BCM56526 B0	
BCM56530	BCM56534 B0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56538 B0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
BCM56540	BCM56540 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56540 B0	
	BCM56541 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56541 B0	
	BCM56542 A1	28xGE + 2xF.XAUI/2x10GE + 2xF.HG[42] + 2xF.HG[21] + 1GE, 28xGE + 8xGE/8x2.5GE + 2xHG[42] + 2xHG[21] + 1GE Multilayer Ethernet Switch
	BCM56544 A1	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
	BCM56544 B0	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
	BCM56545 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56545 B0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE, 24xGE + 4xXAUI + 2xXFI + 2xHG[12] + 1GE Multilayer Ethernet Switch
	BCM56546 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56546 B0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
BCM56580	BCM56580 A0	16 x 2.5 GbE + 4 x 10 GbE Ethernet Multilayer Switch
BCM56620	BCM56620 A0	
	BCM56620 A1	
	BCM56620 B0	
	BCM56620 B1	
	BCM56620 B2	
	BCM56624 A0	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56624 A1	
	BCM56624 B0	
	BCM56624 B1	
	BCM56624 B2	

Table 101: Switch Devices

Family	*Devices*	*Description*
	BCM56626 A0	25 port 1-GbE Multilayer Ethernet Switch with 6 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56626 A1	
	BCM56626 B0	
	BCM56626 B1	
	BCM56626 B2	
	BCM56628 A0	8 port 10-GbE/HiGig2 Multilayer Ethernet Switch with External Table Expansion
	BCM56628 A1	
	BCM56628 B0	
	BCM56628 B1	
	BCM56628 B2	
	BCM56629 B0	25 port 1-GbE Multilayer Ethernet Switch with 8 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56629 B1	
	BCM56629 B2	
BCM56630	BCM56630 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56630 B0	
	BCM56634 A0	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56634 B0	
	BCM56636 A0	24-Port GbE + 2-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56636 B0	
	BCM56638 A0	4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56638 B0	
	BCM56639 A0	24-Port GbE + 4-Port 10-GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56639 B0	
BCM56640	BCM56044	BCM56044 SKU - 100G + 3xF.HG[42] + 1GE
BCM56640	BCM56045 B0	3xF.40GE + 3xF.HG[42] + 1GE
	BCM56046 B0	3xF.40GE + 2xF.HG[42] + 1GE
	BCM56640 A1	1x100GE + 1xHG[127], 1x100GE + 4xHG[32], 1x100GE + 8xHGd[16], 3xF.HG[42] + 1xHG[127], 3xF.HG[42] + 4xHG[32], 3xF.HG[42] + 8xHGd[16], 3xF.HG[42] + 3xF.HG[42] Multilayer Ethernet Switch
	BCM56640 B0	
	BCM56643 A1	48xGE + 4xXFI + 4xHG[42] + 1GE Multilayer Ethernet Switch
	BCM56643 B0	
	BCM56644 A1	48xGE + 2xHG[25] + 2xHG[25] + 1GE Multilayer Ethernet Switch
	BCM56644 B0	
	BCM56648 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56648 B0	
	BCM56649 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56649 B0	
BCM56640	BCM56545K	48-port GE switch + 4x10GE + 4xHG[42] / 40GE
BCM56640	BCM56546K	28-port GE switch + 4x10GE + 4xHG[42] / 40GE

Table 101: Switch Devices

Family	Devices	Description
BCM56680	BCM56680 A0	25 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56680 A1	
	BCM56680 B0	
	BCM56680 B1	
	BCM56684 A0	24 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
	BCM56684 A1	
	BCM56684 B0	
	BCM56684 B1	
BCM56685	BCM56685 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56685 B0	
	BCM56689 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56689 B0	
BCM56700	BCM56700 A0	16-Port, 192-Gbps Lossless Switch Fabric
	BCM56701 A0	12-Port, 144-Gbps Lossless Switch Fabric
BCM56720	BCM56720 A0	16 Port, 16-Gbps HiGig2 Switch Fabric
	BCM56721 A0	12 Port, 16-Gbps HiGig2 Switch Fabric
BCM56725	BCM56725 A0	8 Port, 20-Gbps + 4 Port, 16-Gbps HiGig2 Switch Fabric
BCM56740	BCM56743 A0	480 Gbps Switch fabric
	BCM56743 A1	
	BCM56743 A2	
	BCM56743 A3	
	BCM56743 A4	
	BCM56743 B0	
	BCM56743 B1	
	BCM56745 A0	640 Gbps Switch fabric
	BCM56745 A1	
	BCM56745 A2	
	BCM56745 A3	
	BCM56745 A4	
	BCM56745 B0	
	BCM56745 B1	
BCM56740 PLUS	BCM56744 A0	480 Gbps Switch fabric
	BCM56744 A1	
	BCM56746 A0	640 Gbps Switch fabric
	BCM56746 A1	
BCM56820	BCM56820 A0	24 x 10-GbE + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56820 B0	
	BCM56821 A0	12 x 10-GbE + 8 x HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56821 B0	
	BCM56822 A0	12 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56822 B0	

Table 101: Switch Devices

Family	*Devices*	*Description*
	BCM56823 A0	8 x 10-GbE + 4 x 20-Gbps HiGig2 + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56823 B0	
	BCM56825 B0	16 x 10-GbE + 8 x 20-Gbps HiGig2 + 1 x 1-GbE Multilayer Ethernet Switch
BCM56740	BCM56743 A0	480 Gbps Switch fabric
	BCM56743 A1	
	BCM56743 A2	
	BCM56743 A3	
	BCM56743 A4	
	BCM56743 B0	
	BCM56743 B1	
	BCM56745 A0	640 Gbps Switch fabric
	BCM56745 A1	
	BCM56745 A2	
	BCM56745 A3	
	BCM56745 A4	
	BCM56745 B0	
	BCM56745 B1	
BCM56740 PLUS	BCM56744 A0	480 Gbps Switch fabric
	BCM56744 A1	
	BCM56746 A0	640 Gbps Switch fabric
	BCM56746 A1	
BCM56840	BCM56841 A0	320 Gbps Ethernet Multilayer Switch
	BCM56841 A1	
	BCM56841 A2	
	BCM56841 A3	
	BCM56841 A4	
	BCM56841 B0	
	BCM56841 B1	
	BCM56843 A0	480 Gbps Ethernet Multilayer Switch
	BCM56843 A1	
	BCM56843 A2	
	BCM56843 A3	
	BCM56843 A4	
	BCM56843 B0	
	BCM56843 B1	
	BCM56845 A0	640 Gbps Ethernet Multilayer Switch
	BCM56845 A1	
	BCM56845 A2	
	BCM56845 A3	
	BCM56845 A4	
	BCM56845 B0	

Table 101: Switch Devices

Family	*Devices*	*Description*
	BCM56845 B1	
BCM56840 PLUS	BCM56842 A0	320 Gbps Ethernet Multilayer Switch
	BCM56842 A1	
	BCM56844 A0	480 Gbps Ethernet Multilayer Switch
	BCM56844 A1	
	BCM56846 A0	640 Gbps Ethernet Multilayer Switch
	BCM56846 A1	
BCM56846	BCM56831	24-port 10GE switch with 40GE support for embedded applications
BCM56846	BCM56835C	(64 x 10 GbE) + (4 x 1 GbE)
BCM56846	BCM56847	(64 x 10 GbE) + (4 x 1 GbE)
BCM56846	BCM56849	(56 x 1GbE/2.5GbE) + (8 x 10GbE)
BCM56850	BCM56751P A1	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM56850	BCM56751P A2	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM56850	BCM56830 A1	960Gbps Ethernet Switch
BCM56850	BCM56830 A2	960Gbps Ethernet Switch
BCM56850	BCM56834	High density 10G and 40G switch for embedded applications
BCM56850	BCM56838	72/320G Devices with 1.25/3.125/6.25G Serdes and 4 SFIs
BCM56850	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56852 A2	100x10G, 960Gbps Multilayer Switch
	BCM56854 A1	
BCM56851	BCM56751 A2	1.28Tbps I/O, 960Gbps Core Ethernet Switch Fabric
BCM56960	BCM56960 A0	32x100 GbE/64x40GbE/128x10 GbE Multilayer Switch
	BCM56961 A0	64x40G port configuration
	BCM56960 B0	32x100 GbE/64x40GbE/128x10 GbE Multilayer Switch
BCM88732	BCM88732 B2	Eight-Port 10 GbE or 2-Port 40 GbE MAC Aggregation Switch with 80 Gbps Uplink Capacity
BCM88020	BCM88020 A0	XGS Core (XCore/SBX) Fully Programmable Carrier Packet Processor with 24 GbE Ports, 2 10GbE Ports and 2 SPI Interfaces
	BCM88020 A1	
	BCM88020 A2	
BCM88025	BCM88025 A0	XGS Core (XCore/SBX) Fully Programmable Carrier Packet Processor with 24 GbE Ports, 2 10GbE Ports and 2 SPI Interfaces
BCM88030	BCM88030 A0	XGS Core (XCore/SBX) Scalable Switching 100 Gbps Fully Programmable Carrier Packet Processor
BCM88130	BCM88130 A0	XGS Core (XCore/SBX) 630 Gbps Bandwidth Manager and Switching Engine
	BCM88130 A1	
BME-3200	BME-3200 A0	XGS Core (XCore/SBX) Fabric Bandwidth Manager with 32 SCI control ports and up to 40 SFI data ports
	BME-3200 B0	
QE-2000	QE-2000 A1	XGS Core (XCore/SBX) Fabric Queuing Engine with 49 SPI 4.2 subports
	QE-2000 A2	
	QE-2000 A3	
	QE-2000 A4	
BCM88230	BCM88230 A0	XGS Core (XCore/SBX) Fabric Queuing Engine with Integrated Traffic Management with 4 HiGig2 ports, 50Gbps
	BCM88230 B0	
	BCM88235 A0	XGS Core (XCore/SBX) Fabric Queuing Engine with Integrated Traffic Management with 4 HiGig2 ports, 80Gbps

Table 101: Switch Devices

Family	*Devices*	*Description*
	BCM88235 B0	
	BCM88231 A0	XGS Core (XCore/SBX) Traffic Manager with 4 HiGig2 ports, 50Gbps
	BCM88231 B0	
	BCM88236 A0	XGS Core (XCore/SBX) Traffic Manager with 4 HiGig2 ports, 80Gbps
	BCM88236 B0	
BCM56930	BCM56931 A0	XGS pass-through and standalone Traffic Manager, 4 HiGig2 ports, 50Gbps
	BCM56931 B0	
	BCM56936 A0	XGS pass-through and standalone Traffic Manager, 4 HiGig2 ports, 80Gbps
	BCM56936 B0	
BCM88640	BCM88640 A0	DNX 100G Flexible Packet Processor with Integrated Traffic Management
	BCM88640 B0	
BCM88650	BCM88650 A0	DNX 200G Flexible Packet Processor with Integrated Traffic Management
	BCM88650 B0	
	BCM88650 B1	200 GBps DNX Traffic Manager and Packet Processor
BCM88660	BCM88660 A0	DNX 200G Flexible Packet Processor with Integrated Traffic Management
BCM88750	BCM88750 A0	DNX 1600 GBps Switch Fabric
	BCM88750 B0	
BCM56860	BCM56860 A0	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switch
	BCM56860 A1	
	BCM56861 A1	32 x 40GbE/104 x 10GbE Multilayer Switch
	BCM56862 A1	96 x 10Gbe/24 x 40GbE/8 x 100 GbE Multilayer Switch
	BCM56864 A1	72 x 10GbE/18 x 40GbE Multilayer Switch
	BCM56865 A1	72 x 10GbE/18 x 40GbE/4 x 100GbE Multilayer Switch
	BCM56866 A1	104 x 10GbE/32 x 40GbE/8 x 100GbE Multilayer Switch
	BCM56867 A0	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switch
	BCM56867 A1	
BCM56960	BCM56962 B0	24x 100G/48x 40G
	BCM56968 B0	32x HG[106]/64x HG[42] Multilayer Switch
	BCM56963 B0	6x 100G + 48x 25G

PHYS

Table 102: PHYs

Device	Driver Family	Quality Level
BCM5218	522x	10/100Base-TX/FX Octal-PHY(tm) Transceiver
BCM5220	522x	10/100BASE-TX/FX Mini-F(tm) Transceiver
BCM5221	522x	10/100BASE-TX/FX Mini-F(tm) Transceiver
BCM5226	522x	10/100 BASE- TX/FX Hex-PHY(tm) Transceiver
BCM5228	522x	10/100BASE-TX/FX Octal-F(tm) Transceiver
BCM5238	522x	10/100BASE-TX OCTAL-f(tm) Transceiver
BCM5248	522x	10/100BASE-TX Octal-F(tm) Transceiver
BCM52681E A1	54680	Octal 10/100 Ethernet Transceiver
BCM5401	5401	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5402	5402	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5404	5404	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5424	5424	Quad 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5434	5424	Quad 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5411	5411	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5421	5421S	10/100/1000BASE-T Gigabit Copper Transceiver
BCM5421S	5421S	10/100/1000BASE-T Gigabit Copper Transceiver with SerDes
BCM5461	5464	10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5464	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5464R	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5464S	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5464SR	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5466	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5466R	5464	Quad-Port 10/100/1000BASE-T Gigabit Copper Transceiver
BCM5466S	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5466SR	5464	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM5482	5482	Dual-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM5488	5464	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54210SE B0	54210	Single Copper/Fiber Gigabit Ethernet Transceiver. 1588 not yet supported
BCM54220SE B0	54220	Dual Copper/Fiber Gigabit Ethernet Transceiver. 1588 not yet supported
BCM54240 C0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54240 C1	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280 A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280 C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54280 C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282 A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282 C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54282 C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54285 C0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54285 C1	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54290 A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54292 A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54294 A0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (1588 feature is Bring-up)
BCM54295SE B0	54295	Octal Copper/Quad Fiber Gigabit Ethernet Transceiver -1588 not yet supported
BCM54296SE B0	54296	Quad Copper/Fiber Gigabit Ethernet Transceiver - 1588 not yet supported
BCM54340 B0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)

Table 102: PHYs

Device	Driver Family	Quality Level
BCM54340 C0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54340 C1	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380 B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380 C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380 C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382 B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382 C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382 C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385 B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385 C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385 C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54616 A0	54616	Single-Chip 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54640	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E A1	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E B0	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54680 A0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54682E A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54682E B0	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54684 D0	54684	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54684E B0	54682	10/100/1000 Octal (65nm) QSGMII-Copper/Fiber(2) with EEE
BCM54685	54682	Octal QSGMII to 10/100/1000BaseT or Fiber Ethernet Transceiver
BCM54685E A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with Copper/Fiber Media Interface
BCM54810 A0	54880	BroadR-Reach Single-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880 A0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880 B0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880E A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880E B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54881 B0	54880	Octal 10/100Base/Tx Ethernet BroadReach Transceiver
BCM54942 A0	84728	Quad-Channel 10GbE XAUI-to-XFI PHY. Firmware version 0124
BCM54980 B2	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980 C0	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980 C1	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM8040 A2	8040	Eight-Channel Multirate 1-Gbps - 3.2-Gbps Retimer/Switch
BCM8073 A0	8072	Dual-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version d502.

Table 102: PHYs

Device	Driver Family	Quality Level
BCM8074 A0	8072	Quad-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version 010C.
BCM8704	8703	Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface
BCM8705	8705	Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with WIS Layer and XAUI Interface
BCM8725	8705	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with WIS Layer and XAUI Interface
BCM8726 A0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface
BCM8726 B1	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI(TM) Interface. Firmware version 0x0127
BCM8727 B0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface. Firmware version 0406.
BCM8727 C0	8706	Dual Serial 10-Gigabit Ethernet/Fibre Channel Transceiver with XAUI Interface. Firmware version 050D.
BCM84727 A0	84728	Dual SFI to XAUI with 1588 (Firmware version 0x124. Bring-up)
BCM8728 A0	8706	Dual-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511. (Bring-up)
BCM8742	8706	Quad-Channel 10-GbE SFI-to-XAUI(TM) Transceiver. Firmware version 0511.
BCM8747 A0	8706	Quad-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511.
BCM8750 A0	8750	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC
BCM8752 A0	8750	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC
BCM8754 A0	8750	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version 0411.
BCM8481 B0	8481	10GBASE-T Transceiver (Firmware version B0 02.10)
BCM8481 C0	8481	10GBASE-T Transceiver (Firmware version C0 02.13)
BCM84164	BCM84740	Quad 10GBASE-KR-to-XFI or 40GBASE-KR4-to-XLAUI Transceiver Firmware version 0x128
BCM84168	BCM84740	Octal 10GBASE-KR-to-XFI or Dual 40GBASE-KR4-to-XLAUI Transceiver Firmware version 0x128
BCM84318 A0	84740	10.3 Gbps Octal Port CDR/Retimer with EDC. Firmware version D007
BCM82072	82072	CAUI4-TO-KR4 NRZ BACKPLANE PHY/OCTAL 25G BACKPLANE RETIMER.
BCM82073 A1	82073	CAUI4-to-KR4 NRZ Backplane PHY/Octal 25G Backplane Retimer with 2x2 Crosspoint
BCM82322 B0	82328	1x 100G/3x 40G/12x 10G Ethernet/FC PHY. Firmware version D
BCM82328 A0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 9 "(Bring-up)
BCM82328 B0	82328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D
BCM82328F B0	82328	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY
BCM84328 A0	84328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D026
BCM84328 B0	84328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D026
BCM84333 B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84334 B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84336 B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69 (Bring-up) (Needs additional software component)
BCM84793 A0	84793	100GbE/OTN 4x25/28G VSR28 to 10x10/11G CAUI Gearbox PHY. Firmware version 0xD009 (Bring-up - Mode-1 and Mode-3)
BCM84812 A0	8481	Dual 10GBASE-T Transceiver. Firmware version 2.13
BCM84821 A0	8481	10GBASE-T Transceiver. Firmware version 2.13 (Bring-up)
BCM84822 A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823 A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823 B0	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84823 B1	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02

Table 102: PHYs

Device	Driver Family	Quality Level
BCM84833 B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)
BCM84834 B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)
BCM84836 B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.69(Driver support for IEEE 1588 features are Bring-up)
BCM84844 A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84846 A0	8481	Dual 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84848 A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.07.11(Driver support is Bring-up)
BCM84728 A0	84728	Dual-Channel 10 GbE SFI-to-XAUI LAN/WAN PHY with 1588. Firmware version 0124 (Driver support for IEEE 1588 features is Bring-up)
BCM84729 A0	84729	Dual-Channel SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are Bring-up)
BCM84740 A0	84740	40 GbE PPI-to-XLAUI PHY with EDC. Firmware version D106.
BCM84741 B0	84756	40GbE XLPPI-to-XLAUI/Quad 10G with IEEE MACsec/1588 Firmware version 0x0128 [Bring-up]
BCM84747 A0	84728	Quad SFI to XAUI with 1588 (Firmware version 0x124. Bring-up)
BCM84748 A0	84728	Quad SFI to XAUI with WAN/1588 (Firmware version 0x124. Bring-up)
BCM84749 A0	84749	Quad SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are Bring-up)
BCM84752 A0	84740	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105. (Bring-up)
BCM84753 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105.
BCM84754 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105.
BCM84756 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105. (Needs additional software component)
BCM84756 B0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128(Needs additional software component)
BCM84756 C0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128(Needs additional software component) [Bring-up]
BCM84758	84740	10GbE Quad SFI-XFI PHY with IEEE 1588 Firmware version 0x128
BCM84759 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105.
BCM84759 C0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version 0x0128. (Bring-up)
BCM84780 A0	84740	Octal-Channel 10 GbE SFI-to-XFI PHY with 1588. Firmware version 0x128 (Bring-up)
BCM84784 A0	84740	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 0x125 (Bring-up)
BCM84764 A0	84728	Quad SFI to RXAUI with 1588 (Firmware version 0x124. Bring-up)
BCM84064 A0	84740	Quad 10G-KR-to-XFI or 40G-KR4-to-XLAUI Transceiver. Firmware version 0108.
BCM84074 A0	84728	Quad KR to XAUI (Firmware version 0x124. Bring-up)
BCM82381 A2	82381	Dual 100GbE CAUI4-to-CAUI4 PHY/Octal 25G Retimer
BCM82764 A2	82764	Quad 40 GbE PMA MUX/DEMUX
BCM84858 A0	84858	Quad 10GBASE-T Transceiver. Firmware version 1.01.04 - Beta
BCM84858 B0	84858	Quad 10GBASE-T Transceiver. Firmware version 1.02.02 - Beta
BCM84856 B0	84856	Dual 10GBASE-T Transceiver. Firmware version 1.02.02 - Beta
BCM82792	84856	Dual 100G/OTN CAUI-to-100GbE VSR/CR4/SR4/LR4 Gearbox PHY
BCM82780	84856	10G Octal SFI-XFI PHY with 10GbE and IEEE 1588 Support

Section 12: Compatibility

BROADCOM TASK ENGINES (BTE) FIRMWARE COMPATIBILITY MATRIX

The following table Table 82: SDK Firmware Compatibility Matrix shows compatibility between different versions of SDK and Firmware releases. Please refer to the appropriate Network Switching SDK Firmware release notes publication (56XX0_88XX0_FW-RNxxx-R) for the indicated version below.

Table 103: SDK Firmware Compatibility Matrix

	SDK-6.5.3	*SDK-6.4.11*	*SDK-6.4.10*	*SDK-6.4.9*	*SDK-6.4.8*	*SDK-6.4.7*	*SDK-6.4.6*	*SDK-6.4.5*
4.3.0	BCM56760							
4.2.7		BCM56230 BCM56260 BCM56460 BCM56445 BCM56450	BCM56260 BCM56460 BCM56445 BCM56450					
4.2.6				BCM56260 BCM56450 BCM56640 BCM56860 BCM88370 BCM88650 BCM88670				
4.2.5					BCM56450 BCM56260 BCM56860 BCM88660 BCM88375 BCM88675			
4.2.4						BCM56260 BCM56960 BCM53903		
4.2.3							BCM88650 BCM88370 BCM88670	
4.2.2							BCM53400 BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM88030 BCM56260 BCM56340	BCM88650
4.2.0								BCM53400 BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM56260

BMACSEC SDK COMPATIBILITY MATRIX

Table 104: BMACSEC SDK Compatibility Matrix

Switch SDK Release	*BMACSEC SDK Release*
6.5.0	4.14
6.5.1	4.14
6.5.2	4.15
6.5.3	4.15

PHY FIRMWARE COMPATIBILITY MATRIX

The following table Table 84: PHY Firmware Compatibility Matrix identifies changes in PHY firmware for newer PHY devices.

Table 105: PHY Firmware Compatibility Matrix

PHY Core	*6.5.0 Firmware Versions*	*6.5.1 Firmware Versions*	*6.5.2 Firmware Versions*	*6.5.3 Firmware Versions*
BCM84861	00.00.09	00.00.10	00.00.10	00.00.10
BCM84864	00.00.09	00.00.10	00.00.10	00.00.10
BCM84868	00.00.09	00.00.10	00.00.10	00.00.10
BCM84858	01.02.02	01.02.06	01.02.10	01.03.02
BCM84856	01.02.02	01.02.06	01.02.10	01.02.10
BCM84744	0xD105(A0)/ 0x0128(B0/C0)	0xD105(A0)/ 0x0128 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)
BCM84757	0xD105(A0)/ 0x0128(B0/C0)	0xD105(A0)/ 0x0128 (B0/C0)	0x0132 (B0/C0)	0x0132 (B0/C0)
BCM84328	R027	R027	R027	R029
BCM82322/28 B1	0xD	0xF	0xF	0xF
BCM82780	0x22	0x22	0x23	0x23
BCM82752	0x22	0x22	0x23	0x23
BCM82758	0x22	0x22	0x23	0x23
BCM82764	D00B	D00F	D011	D017
BCM82790	D00B	D00F	D011	D017
BCM82792	D00B	D00F	D011	D017
BCM82796	D00B	D00F	D011	D017
BCM82381	D00E	D011	D011	D012
BCM82209	D00E	D011	D011	D012
BCM82073	D00E	D011	D011	D012
Eagle	D10F_03	D10F_03	D10F_05	D10F_OD
Falcon	D10A_02	D10A_06	D10A_07	D10B_07

SDK EXTERNALLY LICENSED SOFTWARE COMPONENTS

SDK contains a number of third-party externally licensed software components. This appendix contains information regarding these components, the license for each of these components, and where these components are used in SDK.

Table 106: EXTERNALLY LICENSED SOFTWARE COMPONENTS

Component	*Origin*	*Location in source tree*	*License terms and conditions*
EDITLINE	/afs/athena.mit.edu/contrib/sipb/src/editline	src/sal/appl/editline	See (EDITLINE License terms and conditions) (page 255)
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c	See (ED Editor License terms and conditions) (page 257)
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]	See (CINT parser license terms and conditions) (page 257)
BIGDIGITS	David Ireland, copyright (c) 2001-11 by D.I. Management Services Pty Limited <www.di-mgt.com.au>	src/soc/dpp/SAND/Utils/sand_u64.c	See (BIGDIGITS license terms and conditions) (page 258)
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]	See (APIMODE parser license terms and conditions) (page 259)
VxWorks	Wind River Systems, Inc.	systems/vxworks	See (Wind River Systems license terms and conditions) (page 259)

EDITLINE LICENSE TERMS AND CONDITIONS

```
This package was obtained from the following location, and was modified
for purposes of inclusion into the SOC diagnostics shell.

Removed files:
        MANIFEST Make.os9 Makefile os9.h sysos9.c testit.c unix.h

Added files:
        sysvxworks.c Makefile

Changed functionality:
        Merged unix.h into editline.h
        M-P and M-N now behave like tcsh.
        list_history(count) routine displays history
        Commented out completion
        Changed rl_complete and rl_list_possib into caller-settable
                global functions
        Don't ring bell on TAB if word is already complete

--------------------------------------------------------------------------------

Index of /afs/athena.mit.edu/contrib/sipb/src/editline

[     ] Name                    Last modified     Size  Description
  ------------------------------------------------------------------------
```

```
 [DIR]  Parent Directory       11-May-99 03:40      -
 [    ]  MANIFEST              07-Jul-97 11:20     1k
 [    ]  Make.os9              07-Jul-97 11:20     1k
 [    ]  Makefile              01-Sep-97 00:34     2k
 [    ]  complete.c            07-Jul-97 11:20     4k
 [    ]  editline.3            07-Jul-97 11:20     5k
 [    ]  editline.c            07-Jul-97 11:20    25k
 [    ]  editline.h            07-Jul-97 11:20     2k
 [    ]  os9.h                 07-Jul-97 11:20     1k
 [    ]  sysos9.c              07-Jul-97 11:20     1k
 [    ]  sysunix.c             07-Jul-97 11:20     3k
 [    ]  testit.c              07-Jul-97 11:20     1k
 [    ]  unix.h                07-Jul-97 11:20     1k

  --------------------------------------------------------------------------

$Revision: 1.7 $

This is a line-editing library.  It can be linked into almost any
program to provide command-line editing and recall.

It is call-compatible with the FSF readline library, but it is a
fraction of the size (and offers fewer features).  It does not use
standard I/O.  It is distributed under a "C News-like" copyright.

Configuration is done in the Makefile.  Type "make testit" to get
a small slow shell for testing.

This contains some changes since the posting to comp.sources.misc:
    -   Bugfix for completion on absolute pathnames.
    -   Better handling of M-n versus showing raw 8bit chars.
    -   Better signal handling.
    -   Now supports termios/termio/sgttyb ioctl's.
    -   Add M-m command to toggle how 8bit data is displayed.
The following changes, made since the last public release, come from
J.G. Vons <vons@cesar.crbca1.sinet.slb.com>:
    -   History-searching no longer redraws the line wrong
    -   Added ESC-ESC as synonym for ESC-?
    -   SIGQUIT (normally ^) now sends a signal, not indicating EOF.
    -   Fixed some typo's and unclear wording in the manpage.
    -   Fixed completion when all entries shared a common prefix.
    -   Fixed some meta-char line-redrawing bugs.

Enjoy,
        Rich $alz
        <rsalz@osf.org>

 Copyright 1992,1993 Simmule Turner and Rich Salz.  All rights reserved.

 This software is not subject to any license of the American Telephone
 and Telegraph Company or of the Regents of the University of California.

 Permission is granted to anyone to use this software for any purpose on
 any computer system, and to alter it and redistribute it freely, subject
```

```
to the following restrictions:
1. The authors are not responsible for the consequences of use of this
   software, no matter how awful, even if they arise from flaws in it.
2. The origin of this software must not be misrepresented, either by
   explicit claim or by omission.  Since few users ever read sources,
   credits must appear in the documentation.
3. Altered versions must be plainly marked as such, and must not be
   misrepresented as being the original software.  Since few users
   ever read sources, credits must appear in the documentation.
4. This notice may not be removed or altered.
```

ED EDITOR LICENSE TERMS AND CONDITIONS

```
 ed - standard editor
 ^^
   Authors: Brian Beattie, Kees Bot, and others

Copyright 1987 Brian Beattie Rights Reserved.
Permission to copy or distribute granted under the following conditions:
1). No charge may be made other than reasonable charges for reproduction.
2). This notice must remain intact.
3). No further restrictions may be added.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 TurboC mods and cleanup 8/17/88 RAMontante.
 Further information (posting headers, etc.) at end of file.
_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
Modification log:
25Aug92  (W.Metzenthen) Changed malloc() call to calloc() in makebitmap()
         to remove bugs under Linux. Changed a few '^' to the correct '~'.
         General tidying. Recognize Linux via the __linux__ symbol.
         Main change based upon suggestion by Wolfgang Thiel.
07Sep99  Changed large amounts of stuff to simplify  --Curt McDowell
```

CINT PARSER LICENSE TERMS AND CONDITIONS

```
The C code for the CINT parser was generated by using GNU Bison parser
generator from the file cint_grammar.y   CINT is an optional diagnostic
tool that can be included in your system  by adding CINT to the
FEATURE_LIST in SDK compilation flags.

Removed files:
        None

Added files:
        None

Changed functionality:
        None

------------------------------------------------------------------------------
/* A Bison parser, made by GNU Bison 2.4.1.  */
```

```
/* Skeleton implementation for Bison's Yacc-like parsers in C

      Copyright (C) 1984, 1989, 1990, 2000, 2001, 2002, 2003, 2004, 2005, 2006
   Free Software Foundation, Inc.

   This program is free software: you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation, either version 3 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program.  If not, see <http://www.gnu.org/licenses/>.  */

/* As a special exception, you may create a larger work that contains
   part or all of the Bison parser skeleton and distribute that work
   under terms of your choice, so long as that work isn't itself a
   parser generator using the skeleton or a modified version thereof
   as a parser skeleton.  Alternatively, if you modify or redistribute
   the parser skeleton itself, you may (at your option) remove this
   special exception, which will cause the skeleton and the resulting
   Bison output files to be licensed under the GNU General Public
   License without this special exception.

   This special exception was added by the Free Software Foundation in
   version 2.2 of Bison.  */

/* C LALR(1) parser skeleton written by Richard Stallman, by
   simplifying the original so-called "semantic" parser.  */
```

BIGDIGITS LICENSE TERMS AND CONDITIONS

```
Contains BIGDIGITS multiple-precision arithmetic code originally
written by David Ireland, copyright (c) 2001-11 by D.I. Management
Services Pty Limited <www.di-mgt.com.au>, and is used with
permission.

David Ireland and DI Management Services Pty Limited make no
representations concerning either the merchantability of this software
or the suitability of this software for any particular purpose. It is
provided "as is" without express or implied warranty of any kind. Our
liability will be limited exclusively to the refund of the money you
paid us for the software, namely nothing. By using the software you
expressly agree to such a waiver. If you do not agree to the terms, do
not use the software.
```

APIMODE PARSER LICENSE TERMS AND CONDITIONS

The C code for the APIMODE parser was generated by using GNU Bison parser generator from the file api_grammar.y APIMODE is an optional diagnostics shell interface that can be included in your system by adding APIMDOE to the FEATURE_LIST in SDK compilation flags.

See (CINT parser license terms and conditions) (page 257) for the Bison licence.

WIND RIVER SYSTEMS LICENSE TERMS AND CONDITIONS

See WRS_LICENSE.pdf contained in each systems/vxworks subdirectory.