Software Development Kit
Release Notes
SDK 6.4.8

October 20, 2015

Broadcom
Network Switching

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.4.8.

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.

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-PG648-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-PG818-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 preview status are provided to allow early integration of the customer's application with the SDK APIs that support that device. This software has not been tested on the physical target device and should not be expected to fully function.

Devices in "Supported" status 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.

Table 2: Supported Switch Devices

Family	*Devices*	*Description*
BCM56860	BCM56860 A1	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switch
	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 A1	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switch
BCM56960	BCM56962 B0	24x 100G/48x 40G
	BCM56968 B0	32x HG[106]/64x HG[42] Multilayer Switch
	BCM56963 B0	6x 100G + 48x 25G

Table 3: Preview Switch Devices

Family	*Devices*	*Description*
BCM56260	BCM56460 A0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks
	BCM56461 A0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without iHost support
	BCM56462 A0	24 port 54Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch
	BCM56463 A0	34 Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch
	BCM56465 A0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without external packet buffer
	BCM56466 A0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without iHost support and external packet buffer
	BCM56467 A0	24 port 54Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch without external packet buffer
	BCM56468 A0	34 Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch without external packet buffer
	BCM56260 A0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56261 A0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without iHost support
	BCM56262 A0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56263 A0	16 Gbps Integrated Lower Power Carrier Ethernet Access
	BCM56265 A0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer

Table 3: Preview Switch Devices

Family	Devices	Description
	BCM56266 A0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without iHost support and external packet buffer
	BCM56267 A0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer
	BCM56268 A0	16 Gbps Integrated Lower Power Carrier Ethernet Access without external packet buffer
	BCM56460 B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks
	BCM56461 B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without iHost support
	BCM56462 B0	24 port 54Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch
	BCM56463 B0	34 Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch
	BCM56465 B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without external packet buffer
	BCM56466 B0	24-Port 1GbE Multilayer Switch with Four 10GbE/HiGig Uplinks without iHost support and external packet buffer
	BCM56467 B0	24 port 54Gbps Throughput Integrated Low-Power Carrier Ethernet Access Switch without external packet buffer
	BCM56260 B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56261 B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without iHost support
	BCM56262 B0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch
	BCM56263 B0	16 Gbps Integrated Lower Power Carrier Ethernet Access
	BCM56265 B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer
	BCM56266 B0	52 Gbps Integrated Lower Power Carrier Ethernet Access Switch without iHost support and external packet buffer
	BCM56267 B0	20 Gbps Integrated Lower Power Carrier Ethernet Access Switch without external packet buffer
	BCM56268 B0	16 Gbps Integrated Lower Power Carrier Ethernet Access without external packet buffer
BCM56860	BCM56832 A1	27 x 20GbE - Bring-Up Quality
	BCM56833 A1	32 x 40GbE/104 x 10GbE - Bring-Up Quality
	BCM56836 A1	8 x 100GbE/32 x 10GbE/32 x 40GbE/104 x 10GbE - Bring-Up Quality
	BCM56868 A1	8 x HG127/32 x HG42 - Bring-Up Quality

The PHYs in Supported PHYs (page 5) have been qualified against GA target devices in this release.

Table 4: Supported PHYs

Device	Driver Family	Description
BCM82780 A0	82780	10G Octal SFI-XFI PHY with 10GbE/40GbE
BCM84856 B0	84856	Dual 10GBASE-T Transceiver

Table 5: Preview PHYs

Device	Driver Family	Description
BCM84868 B0	84868	Quad 10GBASE-T Transceiver (Beta) - 1588 Gen 2 not yet supported
BCM82864	82864	40GbE PMA MUX/DEMUX with 100GbE Bypass
BCM82790 A0	82790	100GbE/OTN CAUI-to-10GbE VSR/CR4/SR4/LR4/KR4 Gearbox PHY (1.24MB)
BCM82792 A0	82790	Dual 100G/OTN CAUI-to-100GbE VSR/CR4/SR4/LR4 Gearbox PHY (1.28MB)

Table 5: Preview PHYs

Device	*Driver Family*	*Description*
BCM82796 A0	82790	Dual 100G/Quad 40G VSR/CR4/SR4/LR4 Gearbox PHY (1.31MB)
BCM82209 A0	82209	28 nm Octal 25G Unidirectional NRZ Retimer (813KB)
BCM82109	82109	28 nm Octal 25G/28G Unidirectional Repeater
BCM54292	54292	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver with 1588 Gen 1 Support
BCM54290	54290	Octal SGMII Copper Gigabit Ethernet Transceiver (without 1588 support)
BCM54294	54294	Quad SGMII Copper/Fiber Gigabit Ethernet Transceiver (without 1588 support)
BCM84793	84793	100GbE/OTN 4x25/28G VSR28 to 10x10/11G CAUI Gearbox PHY
BCM82752	82752	10G Dual SFI-XFI PHY with 10GbE IEEE 1588 Support
BCM82758	82758	10G Quad SFI-XFI PHY with 10GbE/40GbE and IEEE 1588 Support
BCM8707	8707	Single-Channel 10GbE SFI-to-XAUI Transceiver with EDC

Section 4: New Features per Device

BCM56860 GENERAL AVAILABILITY (GA) RELEASE

The BCM56860 family of switches is aimed specifically at the cloud DC market, ToR and spine. The switch offers up to 104 x 10GbE, up to 32 x 40 GbE or up to 8 x 100 GbE ports. The faster member of the family provides up 1.28Tbps of Ethernet switching in oversub mode or 960G in line rate mode. This release provides GA support for the BCM56860, BCM56861, BCM56862, BCM56864, BCM56865, BCM56866 and BCM56867 A1 devices.

NEW FEATURES IN THIS RELEASE

FLEXPORT SUPPORT

API: bcm_port_resource_multi_set() with Port Mapping changes

The following port transformations have been verified to be functional in both Line Rate and Oversub modes with the internal SERDES.

Table 6: FlexPort Configurations

Initial Configuration	*FLEXPORT*	*Flexed Configuration*
1x10	<--->	1x1
1x10	<--->	1xHG[11]
1x20	<--->	1x10
1x20	<--->	1xHG[21]
4x10	<--->	2x20
4x10	<--->	1x40
4x10	<--->	1x20+2x10
4x10	<--->	2x10+1x20
4x10	<--->	4xHG[11]
4x10	<--->	2xHG[21]
4x10	<--->	1xHG[42]
4x10	<--->	1xHG[21]+2xHG[11]
4x10	<--->	2xHG[11]+1xHG[21]
4x10	<--->	4x1
2x20	<--->	4x10
2x20	<--->	1x40
2x20	<--->	1x20+2x10
2x20	<--->	2x10+1x20
2x20	<--->	4xHG[11]
2x20	<--->	2xHG[21]
2x20	<--->	1xHG[42]
2x20	<--->	1xHG[21]+2xHG[11]
2x20	<--->	2xHG[11]+1xHG[21]
2x20	<--->	2x10
1x40	<--->	4x10
1x40	<--->	2x20

Table 6: FlexPort Configurations

Initial Configuration	*FLEXPORT*	*Flexed Configuration*
1x40	<--->	1x20+2x10
1x40	<--->	2x10+1x20
1x40	<--->	4xHG[11]
1x40	<--->	2xHG[21]
1x40	<--->	1xHG[42]
1x40	<--->	1xHG[21]+2xHG[11]
1x40	<--->	2xHG[11]+1xHG[21]
1x40	<--->	1x10
1x40	<--->	1x20
4xHG[11]	<--->	4x10
4xHG[11]	<--->	2x20
4xHG[11]	<--->	1x40
4xHG[11]	<--->	1x20+2x10
4xHG[11]	<--->	2x10+1x20
4xHG[11]	<--->	2xHG[21]
4xHG[11]	<--->	1xHG[42]
4xHG[11]	<--->	1xHG[21]+2xHG[11]
4xHG[11]	<--->	2xHG[11]+1xHG[21]
2xHG[21]	<--->	4x10
2xHG[21]	<--->	2x20
2xHG[21]	<--->	1x40
2xHG[21]	<--->	1x20+2x10
2xHG[21]	<--->	2x10+1x20
2xHG[21]	<--->	4xHG[11]
2xHG[21]	<--->	1xHG[42]
2xHG[21]	<--->	1xHG[21]+2xHG[11]
2xHG[21]	<--->	2xHG[11]+1xHG[21]
1xHG[42]	<--->	4x10
1xHG[42]	<--->	2x20
1xHG[42]	<--->	1x10
1xHG[42]	<--->	1x20
1xHG[42]	<--->	1x40
1xHG[42]	<--->	1x20+2x10
1xHG[42]	<--->	2x10+1x20
1xHG[42]	<--->	4xHG[11]
1xHG[42]	<--->	2xHG[21]
1xHG[42]	<--->	1xHG[21]+2xHG[11]
1xHG[42]	<--->	2xHG[11]+1xHG[21]
1x100	<--->	10x10
1x100	<--->	12x10
1x100	<--->	6x20
1x100	<--->	3x40
1x100	<--->	10xHG[11]
1x100	<--->	12xHG[11]
1x100	<--->	6xHG[21]
1x100	<--->	3xHG[42]
1x100	<--->	1xHG[106]
1x100	<--->	1xHG[127]
1xHG[106]	<--->	10x10
1xHG[106]	<--->	12x10
1xHG[106]	<--->	6x20

Table 6: FlexPort Configurations

Initial Configuration	FLEXPORT	Flexed Configuration
1xHG[106]	<--->	3x40
1xHG[106]	<--->	1x100
1xHG[106]	<--->	10xHG[11]
1xHG[106]	<--->	6xHG[21]
1xHG[106]	<--->	3xHG[42]
1xHG[106]	<--->	1xHG[127]
1xHG[127]	<--->	10x10
1xHG[127]	<--->	12x10
1xHG[127]	<--->	6x20
1xHG[127]	<--->	3x40
1xHG[127]	<--->	1x100
1xHG[127]	<--->	12xHG[11]
1xHG[127]	<--->	6xHG[21]
1xHG[127]	<--->	3xHG[42]
1xHG[127]	<--->	1xHG[106]
1x20+2x10	<--->	4x10
1x20+2x10	<--->	2x20
1x20+2x10	<--->	1x40
1x20+2x10	<--->	2x10+1x20
1x20+2x10	<--->	4xHG[11]
1x20+2x10	<--->	2xHG[21]
1x20+2x10	<--->	1xHG[42]
2x10+1x20	<--->	4x10
2x10+1x20	<--->	2x20
2x10+1x20	<--->	1x40
2x10+1x20	<--->	1x20+2x10
2x10+1x20	<--->	4xHG[11]
2x10+1x20	<--->	2xHG[21]
2x10+1x20	<--->	1xHG[42]
1x40+2x20+4x10	<--->	1x100
4x10	<--->	4x1
12x10	<--->	1x100
3x40	<--->	1x100
3x40	<--->	12x10
3x40	<--->	6x20
3xHG[42]	<--->	12xHG[11]
3xHG[42]	<--->	6xHG[21]
3xHG[42]	<--->	1xHG[127]
3xHG[42]	<--->	12x10
3xHG[42]	<--->	6x20
3xHG[42]	<--->	1x100

BCM56860 FAMILY OF DEVICES A1 SUPPORT

This release provides GA support for the BCM56860, BCM56861, BCM56862, BCM56864, BCM56865, BCM56866 and BCM56867 A1 devices.

BCM5686x Switch A1 SKUs:

GA support has been added for Rev A1 of switch class devices BCM56861, BCM56862, BCM56864, BCM56865, BCM56866 and BCM56867.

BCM56868 A1 SKU:

Bring-up support has been added for the Rev A1 of this fabric only device derivative.

BCM5683x A1 SKUs:

Bring-up support has been added for Rev A1 of devices BCM56832, BCM56833 and BCM56836.

RIOT PROVIDER BACKBONE BRIDGING

BCM56860 supports Routing into and out of Provider Backbone Bridging (PBB) tunnels. This feature enables the capability to route the packets into the overlay network. Overlay network is set up with PBB tunnels. Similar to RioT VXLAN feature, the capability to route in the virtual domain (VPN) is also needed. Before this feature was introduced, only L2 switching into the overlay network was supported.

WARMBOOT UPGRADE SUPPORT

This release supports "same to same" Warmboot on BCM56860 devices (CPU rebooting to the same SW version). Warmboot upgrade is supported starting with this release, which means customers will be able to execute Warmboot upgrade from SDK-6.4.8 to the next, future release.

SDK OVERALL FEATURES STATUS FOR BCM56860

All features are supported in this release at GA level of quality.

Table 7: BCM56860 Feature Status

Feature	***Status***
100G Support	GA
ALPM	GA
BCMI2C	GA
COE	GA
Cosq	GA
FCOE	GA
Hierarchical ECMP	GA
HIGIG Proxy	GA
KNET	GA
Legacy Field Processor	GA
Field OAM Enhancements	GA
L2	GA
L2 and IP Multicast	GA
L3	GA
L2GRE	GA

Table 7: BCM56860 Feature Status

Feature	Status
MPLS Enhancements Entropy Label and ECMP over MPLS	GA
SW Linkscan	GA
Mac-in-Mac and SPB	GA
MPLS	GA
Mirror	GA
NAT	GA
OAM	GA
Over Subscription	GA
Over Subscription Buffer Enhancements	GA
Packet Rx/Tx	GA
Port	GA
PROXY	GA
Rate	GA
RioT	GA
Rtag7	GA
Resilient Hash	GA
SER	GA
Switch Control	GA
Stack	GA
STAT	GA
STG	GA
TIME	GA
Tunnel	GA
Trill	GA
Trunk	GA
Port Extender	GA
VLAN	GA
VPLAG	GA
VXLAN	GA
VXLT Drop	GA
Warmboot	GA
Warmboot Upgrade	GA
RioT Provider Backbone Bridging	GA
FlexPort	GA

FEATURES NOT YET SUPPORTED

All features are now supported in this release.

BCM56960 NEW FEATURES

ADDED A CONTROL TO SELECT 160 BIT AND 320 BIT MODE OF GROUP CREATION

API bcmFieldGroupModeIntraSliceDouble mode support added for BCM56960 devices. This allows configuring both 160 bit and 320 bit mode of group creation.

BCM56260 UPDATES

The BCM56260 family of switches is aimed specifically at the cost/power sensitive Carrier Ethernet Access (EAD/NID), Microwave backhaul, Compact PTN and Compact Cell-site-Router markets. The switch offers 12x2.5G + 2x10G SerDes with support for 10M/100M/1000M, 2.5G and 10G-XAUI Ethernet speeds on Front panel Viper SerDes and 10M/100M/1000M, 2.5G, 10G-XAUI and 10G Serial Ethernet speeds on Uplink Eagle SerDes ports. The switch provides 50Gbps of Ethernet switching.

The NEON/VPF feature is enabled in B0 SKUs as default. This release provides preview support for the BCM56260 B0 device in addition to features on the A0.

ADDITIONAL FEATURES SUPPORTED

- E2ECC on HG
- PM Support for MPLS LM/DM Embedded App
- Linkphy: Verify one S1 in S0 in currrent implementation
- Linkphy: Implement one stream per subport in MMU
- Add support for 56263 A0 and 56466 A0
- Support for 56260 B0, 56261 B0, 56262 B0, 56263 B0, 56265 B0, 56266 B0, 56267 B0, 56268 B0
- Support for 56460 B0, 56461 B0, 56462 B0, 56463 B0, 56465 B0, 56466 B0, 56467 B0
- Request to support flexible allocation of `MA_INDEX` table
- Performance Monitoring support for MPLS LM/DM Embedded App
- Added S/w swap option for CMIC register access from SDK running on iProc internal A9 CPU in BE mode (`CMIC_SOFT_BYTE_SWAP` flag), in order to work with Firmware applications which run on internal R5 cores in LE mode.
- Individual BHH fault status retrieve Support has been added.
- Support to display PVT monitor voltage using show voltage command

BCM56263 SUPPORT

Support has been added for this device derivative. This table defines the main differences between this SKU and the base SKU BCM56260.

Table 8: BCM56263 SKU

Features	***BCM56260***	***BCM56263***
IO throughput	52 Gbps	16 Gbps
Frequency	118 Mhz	37 Mhz
Other features	Same	Same

BCM56466 SUPPORT

Support has been added for this device derivative. This table defines the main differences between this SKU and the base SKU BCM56460.

Table 9: BCM56466 SKU

Features	***BCM56460***	***BCM56466***
IO throughput	68 Gbps	34 Gbps

Table 9: BCM56466 SKU

Features	**BCM56460**	**BCM56466**
Frequency	130 Mhz	75 Mhz
Other features	Same	Same

LEGACY FEATURE STATUS

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

Table 10: BCM56260 Feature Status

Features	**Status**
Auth	GA
BCMI2C	GA
COE	LGA
COSQ	GA
DMVOQ/DPVOQ	GA
Dynamic Egress and Ingress Thresholds	Beta
E2ECC	Beta
Failover	GA
Field Processor	GA
Flex IO	Beta
Flex Stats	GA
Hashing - RTAG7 - Flexible Hashing	GA
HiGig Lite	Beta
Higig Pause	Beta
IPMC	GA
IP Tunnel	GA
L2	GA
L2 Multicast	GA
L3	GA
LED Processor	GA
Linkscan	GA
LinkPHY	Beta
Mirroring	GA
MIM	GA
MMU IPMC Protection	LGA
MPLS/ VPLS/VPWS	GA
OAM (Legacy)	GA
OOBFC	Beta
QoS	GA
Packet Rx/Tx	GA
PFC	Beta
Port	GA
Proxy	GA
Rate Control	GA
SAFC	Beta
SAT Event Callback	Beta
SAT Diagnostics	Beta
SAT Port 5 Hotswap	Beta
Service Meters	LGA
SPB	GA

Table 10: BCM56260 Feature Status

Features	Status
STACK	LGA
STAT and Flex Stats	GA
STG	GA
Switch	GA
Time	GA
TR101/156/200	LGA
Trunk	GA
Tunnel	GA
UDF	Beta
VLAN	GA
Warmboot	LGA

FEATURES NOT YET SUPPORTED

All features are now supported in this release per the above table.

BCM88670 GENERAL AVAILABILITY (GA) RELEASE

This release is for the BCM88370-Family and BCM88670-Family product lines. Compared to 6.4.7, the 6.4.8 release delivers increased quality as well as introducing new features.

The subsequent sections describe the increment in available features compared to 6.4.7, major bug-fixes and known issues.

IMPORTANT NOTES

- Release is aligned with KBP-SDK 1.3.11 for external TCAM and KAPS.
- Scheduled Warmboot is supported

VALIDATED FEATURES

Basic data path, connectivity and Traffic Management features:

- TDM ILKN validated
- ILKN 5 lanes (under certain conditions)

Packet Processing:

- Port speculative parsing for MPLS packets is now supported
- SLM is now supported, SLM/R-PCP (counting by priority) is also avaiable
- SLB as flexible hashing bring-up completed, initial example is provided. In future release (6.4.9, 6.4.10) we plan to complete SLB as flexible hashing validation.
- SSM (L2-based IPMC TCAM) is now supported
- Added a BCM88670 CINT example for AC 1:1 Protection with Egress Protection.
- L2/L3 Reflector is functional
- OAM: SAT (LBM/LBR) is supported.
- BFDv4 compelte validation of BCM88660 compatible features: Protection packets, OAMP errors.
- BFDv4 Micro-BFD (RFC 7130) is now supported
- BFDv6 BCM88660 compatible is now supported

- Encapsulation of Entropy label at BOS and ELI label using PRGE resources.
- External TCAM (NL12 KBP):
- In 6.4.8 release we introduce the support of a single-core with single External TCAM. We support and validated only BCM88660 features (BCM88660 compatible).
- Future releases (6.4.9, 6.4.10) we plan to complete full system validation for 2 KBPs (one on each core). IPV6 ACL is planned on 6.4.10.
- Additional diagnostics for TPIDs: "diag pp tpid"
- Support LIF discard for asymmetric LIFs
- IP Tunnel termination according to DIP, SIP & VRF.
- MPLS Tunnel ingress protection.

MAJOR BUGFIXES

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

Packet Processing:

- OAM MPLS MEP fails wih assymetric LIFs
- VRF default route delete not functional
- Better LIF allocation management in case of last LIF entry in bank
- MPLS-Port get/delete better handling
- Number of RIFs, VRFs, VLAN-Domains are now consistent with BCM88670 capabilities
- `bcm_ipmc_traverse` is now functional
- BFD TX Source-IP extended and 1588 on the same packets didn't encapsulate correctly
- Metering several important bug fixes
- Improved 2-bit orientation for `VLAN_PORT`
- CPU Learning events parsing of protected PWEs, Trunk
- AC 2 bit split horizon bug fix
- `bcm_mpls_port_get` fixed to support all mpls port configurations
- BFD over PWE non-BOS is now functional
- Ethernet-local-policer is now functional `(bcm_rate_bandwidth_*` and `bcm_policer_group_*` APIs)
- VLAN Translation standard mode EVE fixed for VLAN-Domain values above 255.
- Encapsulation diagnostics shows Native Out-RIF as well ("diag pp encap").
- Some major EVPN bugs fixed.
- `bcm_mpls_port_get` and `_delete` apis fixed for `INGRESS/EGRESS_ONLY` and protection cases.

ERRATA

The list below relates to open bugs that are not resolved:

Packet Processing:

- bcmSwitchMeterAdjust is not functional
- OAMP transimission with UDH is not functional
- `bcm_petra_tunnel_initiator_clear` clears ipv6 tunnels partially. Consequently, the operation of clearing and creating tunnel initiator will return an error after all ressources are allocated.
- MPLS PORT: In case of mpls port configured with FEC learning information but with no actual FEC configuration (this might happen in ingress only for example), `_get` and `_delete` apis will fail.
- Performing warmboot before adding any KAPS entries causes the KAPS to malfunction in core 1. This issue is

scheduled to be fixed in KBPSDK 1.3.12.

- Enabling KAPS dynamic tables causes IPv4 UC public tables key construction to malfunction.

FEATURE STATUS

Table 11: Features Status

Features	*Status*
Initialization sequence, Register Access, PCID, Access TR Tests	GA
BCM88660 : Base Legacy Features (L2, L3, STG, MPLS, COSQ,QOS, RX/TX)	GA
FP (PMF)	GA
Ports and Interfaces	
10GE KR/XFI	GA
10GE RXAUI	GA
10GE XAUI	GA
40GE XLGE	GA
40GE XLGE-2	GA
100GE CGE	GA
SGMII	GA
QSGMII	GA
ILKN-12	GA
ILKN-24	GA
Eyescan	GA
Autoneg (CL72/73/93...)	GA
LinkScan	LGA
NL12	Beta
Statistics/Counting	
MIB counters	GA
Statistics Interface	GA
Counter Processor	GA
CPU/Access	
Packet DMA (CPU RX/TX)	GA
RCPU	GA
Fabric	
TDM	GA
Pipes: single, dual, triple	GA
Cell packing	GA
Repeater link	LGA
Reachability enhancements: routing bitmap, links to destination	LGA
Fabric RX/TX FIFO Scheduling	LGA
Fabric RX FC, Drop & RCI	LGA
Link Level FC	GA
Mesh Connectivity	GA
Source Routed Cells	GA
Live card Insertion/removal	GA
Fabric Isolation	GA
Traffic Management and Queuing	
Basic UC/MC	GA
MC double-CUD	GA

Table 11: Features Status

Features	***Status***
Mirroring	GA
Recycling	GA
VSQ	GA
SYSTEM RED	GA
Other Legacy Features (Routing & Tunneling & Leased svcs)	
TRILL	GA
LAG	GA
Tunnel	GA
MIM	GA
FCOE	GA
Cell-packing	GA
VPLS	GA
FEC Management	GA
Vxlan	GA
Vswitch	GA
ECMP	GA
Flow Control	
Inband Link level	GA
Inband PFC	GA
Inband ILKN	GA
OOB ILKN	Beta
HCFC	Beta
SPI	GA
COE	GA
FC Generation (based on VSQ/Ing. Resources)	Beta
New or Enhanced Features	
Policer	GA
Routing	
MPLS Enhancements	GA
Hierarchical-FEC L3	GA
VRRP Enhancements	GA
RIOT/ROO (Unicast)	GA
TDM	LGA
IPV4 (LPM)	GA
IPV6 (LPM)	GA
H-FEC VPLS	GA
MPLS/VPLS Multicast	GA
Routing Improvements	GA
Overlay/ROO/RIOT MC	GA
Routing over Trill	GA
DCB	
ECN	GA
802.1BR	Beta
Miscellaneous	
Interrupts	GA
External PHY (mechanism only, each specific phy/retimer requires dedicated interop	LGA
Dynamic Port Provisioning	GA
HiGig	GA

Table 11: Features Status

Features	*Status*
SER Protection	GA
SyncE	GA
Protection improvements : Coupled-mode OutLIF	GA
EVPN	GA
Ingress Congestion Management	GA
Ingress Shaping	GA
SAT	Beta
OAM & Access	
Ethernet OAM over LAG	GA
OAM (BCM88660 compatible)	GA
BFD (BCM88660 compatible)	GA
OAM/BFD enhancements	LGA

FEATURES NOT YET SUPPORTED

Table 12: Features Not Yet Supported

Feature	*Release with Support*
B0 revision	6.4.9 (BU), 6.4.10 (GA)
NL12 KBP PP	6.4.9 (LGA), 6.4.10 (GA), see more information in Validated features section.
PON (including ARM)	6.4.9 (GA), validation is still in progress
BFD: Remote Multiplier detection	6.4.9
BFD/CCM statistics (v4, v6)	6.4.10
BFD echo	6.4.10
BFD remote	6.4.10
Y.1731 MAID 11B	6.4.10
Aggregate-Ethernet-Policer	6.4.9
OAM DMA support	6.4.9

BCM88770 (FE3600) SUS 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.

NEW EXTERNAL PHY SUPPORT

BCM8485X ADDITIONAL SUPPORT

SDK 6.4.8 provides GA level support for this PHY. Additional features supported in this release:

- Broadcast Microcode Download for BCM56860
- DSC diagnostics on system side of BCM84858
- BCM8485X upgrade to firmware 01.02.10

Limitations:

- 1588 Gen 2 support not available in this release.
- Due to a recent change in PRBS polynomial mapping, please use the following PRBS polynomial number and pattern mapping:
- p=0 for PRBS7
- p=1 for PRBS9
- p=2 for PRBS11
- p=3 for PRBS15
- p=4 for PRBS23
- p=5 for PRBS31
- For example, to test PRBS15:
- phy diag xe1 prbs set u=1 if=sys p=3 /* this means PRBS 15, use BCM8485x pattern mapping */
- phy diag xe1 prbs set u=0 p=1 /* this means PRBS 15 use diag pattern mapping */
- phy diag xe1 prbs get u=0

BCM82780 ADDITIONAL SUPPORT

6.4.8 provides GA level support for this PHY. Additional features supported in this release:

- Support Broadcast Firmware download
- Supports Failover mode on 10 and 40G mode
- Supports EEPROM flashing
- Supports module read/Write through I2C
- Supports GPIO configuration
- Supports EDC (LRM interface) mode on 10G
- Supports 10G ZR
- Supports HG speeds - 40G (KR4), 11G (XFI, SR), 10G (SR, LRM, KR, CR)

Limitations:

- 1588 support not available in this release.
- HG 42 support not available in this release.
- Flexport support is not available in this release.
- Broadcast firmware download for 10G configs, variable `xphy_primary_core_num` should be set.

BCM54292 ADDITIONAL SUPPORT

Additional features supported in this release:

- 10/100/1G speeds
- Full and half duplex
- Autonegotiation
- SyncE support
- EEE support

Limitations:

- 1588 Gen 2 support not available in this release.
- Cannot set 10M speed with autoneg in this release.

BCM82864 SUPPORT

Basic features supported in this release:

Supported Modes:

- 100G (4x25G) Pass through mode (KR4, LR4, SR4, CR4)
- 40G (4x10G) Pass through mode (KR4, LR4, SR4, CR4)
- Independent 10G lane mode (KR, LR, SR, CR)
- 40G (2x20G) Pass through mode (KR4, LR4, SR4, CR4)

Supported APIs:

- `phymod_core_init`
- `phymod_core_identify`
- `phymod_core_info_get`
- `phymod_core_reset_set`
- `phymod_core_firmware_info_get`
- `phymod_core_pll_sequencer_restart`
- `phymod_phy_polarity_set`
- `phymod_phy_polarity_get`
- `phymod_phy_reset_set`
- `phymod_phy_reset_get`
- `phymod_phy_power_set`
- `phymod_phy_power_get`
- `phymod_phy_fec_enable_set`
- `phymod_phy_fec_enable_get`
- `phymod_phy_interface_config_set`
- `phymod_phy_interface_config_get`
- `phymod_phy_loopback_set`
- `phymod_phy_loopback_get`
- `phymod_phy_rx_pmd_locked_get`
- `phymod_phy_link_status_get`
- `phymod_phy_status_dump`
- `phymod_phy_reg_read`
- `phymod_phy_reg_write`
- `phymod_phy_prbs_config_set`
- `phymod_phy_prbs_config_get`
- `phymod_phy_prbs_enable_set`
- `phymod_phy_prbs_enable_get`
- `phymod_phy_prbs_status_get`
- `phymod_phy_diagnostics_get`
- `phymod_phy_eyescan_run`

Limitations:

- In 10G mode interface set on one lane can cause flap on other lanes in the same quad.

BCM82109 SUPPORT

This release adds initial support for BCM82109 with BCM56960.

- `bcm_pm_if_init`
- `bcm_pm_if_cleanup`
- `bcm_pm_if_version_get`
- `bcm_pm_if_reg_value_set` / get
- `bcm_pm_if_rev_id`
- `bcm_pm_if_sigdet_get`
- `bcm_pm_if_power_set` / get
- `bcm_pm_if_reset_set` (Core)
- `bcm_pm_if_tx_lane_control_set` /get
- `bcm_pm_if_cfg_gpio_pin_set/get`
- `bcm_pm_if_static_config_set/get`

BCM82328 ADDITIONAL SUPPORT

- Upgraded firmware to version 0xF

BCM8279X ADDITIONAL SUPPORT:

The BCM82796/792 is dual 100Gbps Gearbox phy that multiplexes and demultiplexes eight 25Gbps channels to/from twenty 10Gbps channels supporting Ethernet and Optical Transport Networking(OTN). The BCM82796/792 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. Additional features supported in this release:

- Dual full-duplex, ten lane-to-four lane Gearbox via IEEE 802.3ba Clause 83PMA
- Dual full-duplex, dual 10 lane-to-four lane Gearbox via ITU OTL4.4
- Optional 40G mode, Quad full-duplex four lane to two lane gearbox
- 40G PMD support includes CR4/SR4/LR4
- Dual full-duplex, four lanes-to-four lanes at 10.3125 Gbps to 11.18 Gbps
- Dual-duplex, four lanes-to-four lanes at 10.3125 Gbps to 11.18 Gbps (XLAUI to XLPPI)
- Driver supports 100G with SR4/LR4/CR4/KR4/CAUI

Functionality Supported:

- Link status
- Speed and Interface configuration
- port enable/disable
- Polarity inversion
- Power set/get
- PRBS
- PLL sequencer restart
- Phy dump
- Rx PMD lock
- remote/digital loopback set/get
- DSC
- Phy dump

- Tx lane control set/get
- Tx set/Get
- Rx set/Get
- Tx Training
- Link monitors
- GPIO config set/get
- GPIO value set/get
- Reg read/write (MDIO/I2C).
- Eyescan
- BER projection
- Autonegotiation
- PHY Reset (Hard/soft)
- Probe

Driver Limitations:

- Chip reset issues reset to all the ports those shares same PHY-ID(MDIO ADDR).
- Only 100G Gearbox mode is supported - 10G/40G is not available in this release.
- Interrupts, HIGIG2, and Flexport not supported in this release.

Section 5: Things to note

This section lists items that require special attention.

ALERT ON UPCOMING END OF MAINTENANCE RELEASE CYCLE FOR SDK 6.4

Broadcom is planning to have the upcoming SDK 6.4.11 version be the last official maintenance release under the SDK 6.4 delivery stream.

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.

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.

The total scache required for IFP in worst case configuration is around 36 MB and total scache requirement is around 40 MB.

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

SPN_PHY_PORT_PRIMARY_AND_OFFSET CONFIG PROPERTY IS REQUIRED FOR NEW PHYS

Setting of the config property `spn_PHY_PORT_PRIMARY_AND_OFFSET` is absolutely required for the following PHYs. BCM54880E BCM54680E BCM54682E BCM54685E BCM54640E BCM542XX

NEW SPINLOCK APIS

In the SDK release v6.3.8/SDK 6.4.1 a new lock mechanism -- spinlock was introduced to satisfy some requirements for protecting small critical sections more efficiently. The spinlock mechanism is applicable to the scenario in which the critical section to be protected only contains simple operations, such as inserting or removing nodes from a linked list, increasing or decreasing shared data. The data structure and interfaces of the spinlock are defined at SAL layer in SDK as follows,

```
typedef struct sal_spinlock_s {
    char spinlock_opaque_type;
} *sal_spinlock_t;
```

```
sal_spinlock_t sal_spinlock_create(char *desc);
int sal_spinlock_destroy(sal_spinlock_t lock);
int sal_spinlock_lock(sal_spinlock_t lock);
int sal_spinlock_unlock(sal_spinlock_t lock);
```

These primitives have been implemented for vxWorks and Linux. The changes are available in src/sal/core/linux/sync.c for Linux kernel mode, src/sal/core/unix/sync.c for Linux user mode and src/sal/core/vxworks/sync.c for vxWorks. Customers who use a different OS will need to make similar implementation in their OS specific SAL layer source files. Contact your Broadcom field support staff for more information.

WARMBOOT: VALIDATED WARMBOOT UPGRADES.

Following warmboot upgrades have been validated in this release.

Table 13: Validated Warmboot upgrades

Software upgrade	*Supported*
6.4.7 to 6.4.8	Yes
6.3.12 to 6.4.8	Yes

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

- BCM53400/BCM56060
- BCM56150
- BCM56220
- BCM56260
- BCM56340
- BCM56440
- BCM56450
- BCM56640
- BCM56840
- BCM56850
- BCM56960

WARMBOOT SUPPORT FOR EMBEDDED APPLICATIONS

This section describes warmboot support and procedures for associated firmware supported in SDK 6.4.8. Below are the various embedded applications (FW) available and supported with SDK 6.4.8

- 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.*/

COSQ WARMBOOT

The scache file which is used to store software data related to Cosq during level-2 warmboot for BCM56640 and related SKUs has a different data structure between 6.3.x releases and 6.4.x releases. However, the data structure difference cannot be distinguished, because they are marked with the same warmboot version number.

If level-2 warmboot is used to upgrade from release 6.3.3 and later to release 6.4.x, the Cosq module will be restored with incorrect data.

FLEXHASH WARMBOOT

Flex hash warmboot was implemented in SDK-6.4.7. If level-2 warmboot is used to upgrade from release 6.3.12, the Flex hash will be restored with incorrect data.

POTENTIAL WARMBOOT ISSUE IN 6.3.12 TO 6.4.7 ON BCM56450 DEVICE FAMILY

In 6.3.12 and earlier releases, the scheduling node properties were programmed in the wrong node for HiGig and loop-back ports. This was corrected in 6.4.5 and later releases.

If the configuration was done on 6.3.12, it was configured incorrectly. After the Warmboot upgrade is performed and a read is issued, the reading is taken from the correct node and the comparison test fails for mismatched values.

The values don't match and it is causing failure in some of tests which read back and compare the values.

- 6.3.12 -> 6.3.12 -> Passes - value is configured and read from wrong node. Comparison test passes due to wrong reason.
- 6.4.5/6/7-> 6.4.5/6/7 -> Passes -> Value is configured and read from correct node.
- 6.3.12-> 6.4.7 -> Fails -> Value configured on wrong node, but read from right node. Comparison test fails since value does not match.

The fix has to go into 6.3.12. Patch is available for 6.3.x related branches.

It is recommended to patch this in 6.3.12, if scheduling node properties of HG or loop back ports are modified in 6.3.12.

POTENTIAL WARMBOOT ISSUES IN 6.3.12 TO 6.4.7 ON BCM56850 DEVICE FAMILY

In SDK-6.4.6 and SDK-6.3.12 releases, the port index used to store and recover E2ECC configuration during Warmboot synchronization and recovery was incorrect. This issue has been fixed in SDK-6.4.7. As a result, some 6.3.12/6.4.6 -> 6.4.7 warmboot test cases would fail since the port index value doesn't match after the warmboot. To get the case working, customers will require patching the code changes attached in SDK-79646 to SDK-6.4.6 or SDK-6.3.12. Releases starting with SDK-6.4.7 have no such problem.

SDK-6.4.6 and SDK-6.3.12 releases didn't include Warmboot support for the global configuration variable `_soc_td2_mmu_traffic_ctrl` which is used to control whether the traffic to a port should be enabled or not. This would potentially cause traffic to a port to be dropped after system Warmboot-upgrades to SDK-6.4.7. To get the case working, customers will require patching the code changes committed in SDK-77046 to SDK-6.4.6 or SDK-6.3.12. Releases starting with SDK-6.4.7 have no such problem.

In SDK-6.3.12 release, the value of the enum `BCM_PORT_PHY_CONTROL_PARALLEL_DETECTION` was different with that in SDK-6.4.X releases. This would cause the enum value saved in SDK 6.3.12 cannot be interpreted correctly in SDK 6.4.7. To get the case working, customers will require patching the code changes attached in SDK-80136 to SDK-6.3.12. SDK-6.4.X releases have no such problem.

SDK-6.3.12 release didn't include Warmboot support for WRED Time Domain Configuration. This would cause the scache data structure mismatch between SDK-6.3.12 and SDK-6.4.7. To get the case working, customers will require patching the code changes attached in SDK-72889 to SDK-6.3.12. SDK-6.4.X releases have no such problem.

POTENTIAL BCM MODULE DETACH ISSUE IN THE

RELEASES PRIOR TO 6.4.7

In pre 6.4.7 releases, during bcm detach, UDF module is detached even before field module. And during Field detach, there is deference to UDF module elements which got freed earlier. This was causing segmentation fault.

If the segmentation fault issue is seen during `bcm_detach,` patch from SDK-76765 needs to be applied.

Table 14: Potential Impacts

Scenarios	*Devices*
Warmboot Applications need detaching BCM modules	• BCM56240 • BCM56260 • BCM56340 • BCM56440 • BCM56450 • BCM56530 • BCM56545 • BCM56630 • BCM56640 • `BCM56840_PLUS` • BCM56850 • BCM56860 • BCM56960

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 */

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 newer than 56XX-PG648-R. (See section 2 earlier in this document for availability).

BCM FUNDAMENTALS

NEW VNTAG/ETAG DATA TYPES

VNTAG information is added in the Ethernet packet to process packets in a NIV-Enabled Network, and ETAG information is added in the Ethernet packet to process packets in a PE-Enabled Network.

`VNTAG/ETAG types (page 29)` shows the types that are designated for working with VNTAG/ETAG.

Table 15: VNTAG/ETAG types

Type	*Description*
`bcm_vntag_t`	A structure that includes • `direction` -Direction (0-IV to VIS, 1-VIS to IV) • `pointer` -Pointer (0-Unicast, 1-Multicast) • `dst_vif` -Destination Virtual Interface • `loop` -Looped (0-No loop, 1-Loop) • `src_vif` -Source Virtual Interface
`bcm_vntag_action_t`	An enum that includes • `bcmVnTagActionNone` -No action • `bcmVnTagActionChange` -Replace VnTag if already present or add VnTag if not already present • `bcmVnTagActionDelete` -Delete VnTag • `bcmVnTagActionCount` -Always Last, not a usable value
`bcm_etag_t`	A structure that includes • `pcp` -Priority code point • `de` -Discard eligibility • `src _vid-Source` extended port vlan id • `vid` -Extended port vlan id
`bcm_etag_action_t`	An enum that includes • `bcmETagActionNone` -No action • `bcmETagActionChange` -Replace ETag if already present or add ETag if not already present • `bcmETagActionDelete` -Delete ETag • `bcmETagActionCount` -Always Last, not a usable value

BIDIRECTIONAL FORWARDING DETECTION

A new BFD endpoint is added to configure for the micro BFD session.

Table 16: BFD Endpoint Info Structure Flag Definitions

Flag	*Description*
`BCM_BFD_ENDPOINT_MICRO_BFD`	Use to configure micro BFD session.

CLASS OF SERVICE QUEUE CONFIGURATION

New COSQ control is added in this release.

Table 17: CoSQ Control Type Values

Value	*Description*	*Arg value*
`bcmCosqControlE2ECCTransmitEnable`	`Enabling` Transmission of E2ECC Message	`value:` 0-disable, 1-enable

New BCM COSQ APIs are added in this release.

bcm_cosq_port_mapping_multi_set bcm_cosq_port_mapping_multi_get

Get or set the mapping of multiple internal priorities to their corresponding CoS queues.

Syntax

```
#include <bcm/cosq.h>
int bcm_cosq_port_mapping_multi_set(int unit, bcm_port_t port,int count,
      bcm_cos_t *priority_array, bcm_cos_queue_t *cosq_array);
int bcm_cosq_port_mapping_multi_get(int unit, bcm_port_t port, int count,
  bcm_cos_t *priority_array, bcm_cos_queue_t *cosq_array);
```

Parameters

`unit`	BCM device number
`port`	Device or logical port or GPORT ID
`count`	number of elements in priority array and cosq array in case of multi get or set
`priority_array`	array of internal priority to map
`cosq_array`	array of Cos queue where each element of `cosq_array` (cos) corresponds to the respective element of `priority_array` (internal priority)

Description

Configure or retrieve the mapping of multiple internal priority value to their corresponding Cos Queue. The internal priority is usually initialized from the packet priority. For tagged packets the priority is extracted from the PRI field of the tag. For untagged packets, the priority is specified using the API `bcm_port_untagged_priority_set` in API guide 56XX-PG648-R.

On some devices, the mapping tables may be shared among several ports, so setting one port's mappings may affect other ports'.

On some devices, packets directed to the CPU interface port may have their priority assigned independent of this setting. For more information see `bcm_switch_control_set` in API guide 56XX-PG648-R.

On some devices, a port value of -1 or gport type for system configuration will configure all ports.

Some devices support Enhanced Transmission Selection (ETS) which requires to use GPORT ID for the port argument. When the ETS mode is enabled, they configure and retrive the mapping from the internal priority to the offset (which is equal to the value of cosq) of the unicast queues and multicast queues attached to the egress port.

Most devices have 8 internal priorities (0 to 7).

Returns

`BCM_E_NONE`	
`BCM_E_PARAM`	Invalid priority, or device not configured for the number of CoS queues specified.
`BCM_E_XXX`	

PORT EXTENSION MANAGEMENT

Two new fields are added in `bcm_extender_port_t` data structure. of Port Extension Data Types

```
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;
```

FIELD PROCESSOR

The type `bcm_field_qualify_t` is an enumeration of the possible field qualifiers. New qualifiers are added, where the parameters for qualifier bcmFieldQualifyOamEgressMulticastMacHit are defined as following:

Table 18: OAM Multicast hit bits for bcmFieldQualifyOamEgressMulticastMacHit

BCM_FIELD_OAM_MULTICAST_MAC_xxx	***Description***
`BCM_FIELD_OAM_MUTLICAST_MAC_NO_HIT`	Destination MAC address not in OAM Multicast MAC CLASS1 or CLASS 2 Address Range
`BCM_FIELD_OAM_MUTLICAST_MAC_CLASS1_HIT`	Destination MAC address in OAM Multicast MAC CLASS1 Address Range 01-80-C2-00-00-30 to 01-80-C2-00-00-37
`BCM_FIELD_OAM_MUTLICAST_MAC_CLASS2_HIT`	Destination MAC address in OAM Multicast MAC CLASS2 Address Range 01-80-C2-00-00-38 to 01-80-C2-00-00-3f

Table 19: Field Qualifiers

Qualifier	*Purpose*
bcmFieldQualifyTosLower4Bits	To qualify on Lower 4bits of TOS.
bcmFieldQualifyOamEgressClass2Vxlt	To qualify on OAM Class Id assigned for packet based on Second Lookup in EGR_VLAN_XLATE table.
bcmFieldQualifyOamEgressVxltFirstHit	To qualify on First EGR_VXLT lookup hit status.
bcmFieldQualifyOamEgressVxltSecondHit	To qualify on Second EGR_VXLT lookup hit status.
bcmFieldQualifyOamDownMEPLoopbackPacket	Indicates Down-MEP Loopback or Down-SAT Latching Loopback packet.
bcmFieldQualifyOamEgressPortUnicastDstMacHit	To qualify on EGR_PORT MACDA hit status.
bcmFieldQualifyOamSat	Pseudo qualifier for OAM/SAT.
bcmFieldQualifyIngressDropEthernetOamControl	To qualify on Drop reasons for Ethernet Oam Control Packets based on Mep Type.
bcmFieldQualifyIngressDropEthernetOamData	To qualify on Drop reasons for Ethernet Oam Data Packets based on Mep Type.
bcmFieldQualifyIngressDropMplsOamControl	To qualify on Drop reasons for Mpls Oam Control Packets based on Mep Type.
bcmFieldQualifyIngressDropMplsOamData	To qualify on Drop reasons for Mpls Oam Data Packets based on Mep Type.
bcmFieldQualifyEgressClassVxlan	To qualify on Class Id assigned by Egress Vxlan
bcmFieldQualifyOamEgressEtherType	To qualify on OAM/SAT ethernet type. Allowed values are 0x8902 and 0x88b7.
bcmFieldQualifyOamEgressMulticastMacHit	To qualify on Multicast OAM Dest Mac Address Range Hit status. Allowed values BCM_FIELD_OAM_MULTICAST_MAC_XXX.
bcmFieldQualifyTrunkMemberSourceModuleId	To qualify Source module ID if packet receives on a trunk port locally.
bcmFieldQualifyOamEgressClassSrcMac	To qualify on SourceMac Class ID assigned for packet based on Hit in EGR_OAM_SAT_MISC_MAC_SA_0.15 regsisters

The new Field Qualify Drop reasons related to OAM Data or Control packets based on MEP type are introduced in this release.

bcm_field_oam_drop_mep_type_t elements allow the user to Qualify on Drop reasons related to OAM Data or Control packets based on MEP type. The allowed formats are enumerated in bcm_field_oam_drop_mep_type_t.

Table 20: New Field Qualify OAM MEP type Format

Formats	*Purpose*
bcmFieldOamDropInvalid	Invalid.
bcmFieldOamDropPortDownMepControl	Drop Reasons for Port Down MEP control packets.
bcmFieldOamDropPortDownMepData	Drop Reasons for Port Down MEP data packets.
bcmFieldOamDropInnerVlanDownMepControl	Drop Reasons for Inner Vlan Down MEP control packets.
bcmFieldOamDropInnerVlanDownMepData	Drop Reasons for Inner Vlan Down MEP data packets.
bcmFieldOamDropOuterVlanDownMepControl	Drop Reasons for Outer Vlan Down MEP control packets.
bcmFieldOamDropOuterVlanDownMepData	Drop Reasons for Outer Vlan Down MEP data packets.
bcmFieldOamDropInnerPlusOuterVlanDownMepControl	Drop Reasons for Inner + Outer Vlan Down MEP control packets.
bcmFieldOamDropInnerPlusOuterVlanDownMepData	Drop Reasons for Inner + Outer Vlan Down MEP data packets.
bcmFieldOamDropInnerVlanUpMepData	Drop Reasons for Inner Vlan Up MEP data packets.
bcmFieldOamDropOuterVlanUpMepData	Drop Reasons for Outer Vlan Up MEP data packets.

Table 20: New Field Qualify OAM MEP type Format

Formats	***Purpose***
bcmFieldOamDropInnerPlusOuterVlanUpMepData	Drop Reasons for Inner + Outer Vlan Up MEP data packets.
bcmFieldOamDropSectionPortMepControl	Drop Reasons for Port Section MEP control packets.
bcmFieldOamDropSectionPortMepData	Drop Reasons for Port Section MEP data packets.
bcmFieldOamDropSectionInnerVlanMepControl	Drop Reasons for Inner Vlan Section MEP control packets.
bcmFieldOamDropSectionInnerVlanMepData	Drop Reasons for Inner Vlan Section MEP data packets.
bcmFieldOamDropSectionOuterVlanMepControl	Drop Reasons for Outer Vlan Section MEP control packets.
bcmFieldOamDropSectionOuterVlanMepData	Drop Reasons for Outer Vlan Section MEP data packets.
bcmFieldOamDropSectionInnerPlusOuterVlanMepControl	Drop Reasons for Inner + Outer Vlan Section MEP control packets.
bcmFieldOamDropSectionInnerPlusOuterVlanMepData	Drop Reasons for Inner + Outer Vlan Section MEP data packets.
bcmFieldOamDropLSPMepControl	Drop Reasons for LSP MEP control packets.
bcmFieldOamDropLSPMepData	Drop Reasons for LSP MEP data packets.
bcmFieldOamDropPwMepControl	Drop Reasons for PsuedoWire MEP control packets.
bcmFieldOamDropPwMepData	Drop Reasons for PsuedoWire MEP data packets.

New field actions are added in this release as listed.

Table 21: New Field Actions

Action	***Description***	***param0***	***param1***
bcmFieldActionLoopbackOamResponse	Generates OAM Loopback response. LBR generation means swap DA/SA and change opcode to LBR opcode and send packet back to the SGPP that it was received from.	n/a	n/a
bcmFieldActionLoopbackSatResponse	Generates SAT Loopback response. Latching loopback means swap DA/SA and send packet back to the SGPP that it was received from.	n/a	n/a
bcmFieldActionOamStatObjectSessionId	Adds LM counter index to OLP header to uniquely identify OAM session.	Flex Stat Object `(bcm_stat_object_t)`	n/a
bcmFieldActionServicePriIntCosQNew	To assign new Unicast Queue Number for all packets based on the Gport and Classifier ID and Service Base Queue Number. Gport is created using `bcm_cosq_gport_XXX` API's and Classifier ID from `bcm_cosq_classifier_create/` `bcm_cosq_classifier_mapping_multi_set` API's.	Classifier ID	l2 UC queue Gport

Table 21: New Field Actions

Action	Description	param0	param1
bcmFieldActionRpServicePrioIntCosQNew	To assign new Unicast Queue Number for Red packets based on the Gport and Classifier ID and Service Base Queue Number. Gport is created using `bcm_cosq_gport_XXX` API's and Classifier ID from `bcm_cosq_classifier_create/` `bcm_cosq_classifier_mapping_multi_set` API's.	Classifier ID	l2 UC queue Gport
bcmFieldActionIntYpServicePrioIntCosQNew	To assign new Unicast Queue Number for Yellow packets based on the Gport and Classifier ID and Service Base Queue Number. Gport is created using `bcm_cosq_gport_XXX` API's and Classifier ID from `bcm_cosq_classifier_create/` `bcm_cosq_classifier_mapping_multi_set` API's.	Classifier ID	l2 UC queue Gport
bcmFieldActionIntGpServicePrioIntCosQNew	To assign new Unicast Queue Number for Green packets based on the Gport and Classifier ID and Service Base Queue Number. Gport is created using `bcm_cosq_gport_XXX` API's and Classifier ID from `bcm_cosq_classifier_create/` `bcm_cosq_classifier_mapping_multi_set` API's.	Classifier ID	l2 UC queue Gport
bcmFieldActionProtectionSwitchingDrop	To drop the data packets on working or protected path in VFP. The drop packets can be qualified in IFP using the IFP qualifier "bcmFieldQualifyRecoverableDrop" and the drop action can be cancelled by "bcmFieldActionRecoverableDropCancel"	n/a	n/a
bcmFieldActionCount	Always last; not a usable action	n/a	n/a

New Field Control type is added for device specific configuration parameter, which are specified in terms of a control setting `bcm_field_control_t.` It is set or read via `bcm_field_control_set` API calls.

Table 22: New Field Control Values

Name	*Purpose*
bcmFieldControlOverrideStageLookupPhb	If set, internal priority assigned by L2/L3/DSCP lookups overrides internal priority assigned by VFP

New Field Qualify OLP header types are added.

Table 23: New Field Qualify OLP header type Format

Formats	*Purpose*
bcmFieldOlpHeaderTypeEthOamCcm,	Ethernet Oam Down MEP Ccm packet.
bcmFieldOlpHeaderTypeEthOamLmDm,	Ethernet Oam Down MEP LM/DM packet.
bcmFieldOlpHeaderTypeEthOamOthers,	Ethernet Oam Down MEP Other opcode packet.
bcmFieldOlpHeaderTypeEthOamUpMepCcm,	Ethernet Oam Up MEP Ccm packet.
bcmFieldOlpHeaderTypeEthOamUpMepLm,	Ethernet Oam Up MEP Lm packet.
bcmFieldOlpHeaderTypeEthOamUpMepDm,	Ethernet Oam Up MEP Dm packet.
bcmFieldOlpHeaderTypeEthOamUpMepOthers,	Ethernet Oam Up MEP Other opcode packet.
bcmFieldOlpHeaderTypeUpSat,	Up Service Activation Test packet.

New supported field group creation mode is listed as below.

`BCM_FIELD_ENTRY_INVALID` is defined for indicating the invalid or free entry of Field Processor.

New field APIs are introduced in this release.

bcm_field_qualify_XXX

Add a qualification to a field entry

Syntax

```
#include <bcm/field.h>
int bcm_field_qualify_TosLower4Bits(int unit, bcm_field_entry_t entry,
    uint8 data, uint8 mask);
int bcm_field_qualify_OamEgressClass2Vxlt(int unit, bcm_field_entry_t entry,
  uint32 data, uint32 mask);
int bcm_field_qualify_OamEgressVxltFirstHit(int unit, bcm_field_entry_t entry,
    uint8 data, uint8 mask);
int bcm_field_qualify_OamEgressVxltSecondHit(int unit, bcm_field_entry_t entry,
     uint8 data, uint8 mask);
int bcm_field_qualify_OamDownMEPLoopbackPacket(int unit, bcm_field_entry_t
entry,
       uint8 data, uint8 mask);
int bcm_field_qualify_OamEgressPortUnicastDstMacHit(int unit,
bcm_field_entry_t entry,
    uint8 data, uint8 mask);
int bcm_field_qualify_IngressDropEthernetOamControl(int unit,
bcm_field_entry_t entry,
    bcm_field_oam_drop_mep_type_t mep_type);
int bcm_field_qualify_IngressDropEthernetOamData(int unit, bcm_field_entry_t
entry,
 bcm_field_oam_drop_mep_type_t mep_type);
int bcm_field_qualify_IngressDropMplsOamControl(int unit, bcm_field_entry_t
entry,
bcm_field_oam_drop_mep_type_t mep_type);
int bcm_field_qualify_IngressDropMplsOamData(int unit, bcm_field_entry_t entry,
     bcm_field_oam_drop_mep_type_t mep_type);
int bcm_field_qualify_EgressClassVxlan(int unit, bcm_field_entry_t entry,
       uint16 data, uint16 mask);
int bcm_field_qualify_OamEgressEtherType(int unit, bcm_field_entry_t entry,
 uint16 data);
int bcm_field_qualify_OamEgressMulticastMacHit(int unit, bcm_field_entry_t
entry,
       uint8 data);
int bcm_esw_field_qualify_TrunkMemberSourceModuleId(int unit,
bcm_field_entry_t entry,
    int modid, int modid_mask);
int bcm_field_qualify_OamEgressClassSrcMac(int unit, bcm_field_entry_t entry,
                                          uint32 data, uint32 mask);

para=unit     BCM nevice number
```

Parameters

`entry`	Field entry ID
`data`	Data to match against
`mask`	Mask to choose which bits of data to match against

Description

Adds a qualification to a filter entry. Each qualification added makes the entry more specific and match fewer possible packets.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_INIT`	BCM unit not initialized
`BCM_E_NOT_FOUND`	Entry ID not found
`BCM_E_PARAM`	Qualifier not in group's Qset
`BCM_E_XXX`	Other error code

bcm_field_qualify_XXX_get

Get a qualification match criteria from a field entry

Syntax

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

```
int bcm_field_qualify_TosLower4Bits_get(int unit, bcm_field_entry_t entry,
uint8 *data, uint8 *mask);
int bcm_field_qualify_OamEgressClass2Vxlt_get(int unit, bcm_field_entry_t
entry,
      uint32 *data, uint32 *mask);
int bcm_field_qualify_OamEgressVxltFirstHit_get(int unit, bcm_field_entry_t
entry,
uint8 *data, uint8 *mask);
int bcm_field_qualify_OamEgressVxltSecondHit_get(int unit, bcm_field_entry_t
entry,
 uint8 *data, uint8 *mask);
int bcm_field_qualify_OamDownMEPLoopbackPacket_get(int unit, bcm_field_entry_t
entry,
   uint8 *data, uint8 *mask);
int bcm_field_qualify_OamEgressPortUnicastDstMacHit_get(int unit,
bcm_field_entry_t entry,
uint8 *data, uint8 *mask);
int bcm_field_qualify_IngressDropEthernetOamControl_get(int unit,
bcm_field_entry_t entry,
    bcm_field_oam_drop_mep_type_t *mep_type);
int bcm_field_qualify_IngressDropEthernetOamData_get(int unit,
bcm_field_entry_t entry,
 bcm_field_oam_drop_mep_type_t *mep_type);
int bcm_field_qualify_IngressDropMplsOamControl_get(int unit,
bcm_field_entry_t entry,
bcm_field_oam_drop_mep_type_t *mep_type);
int bcm_field_qualify_IngressDropMplsOamData_get(int unit, bcm_field_entry_t
entry,
     bcm_field_oam_drop_mep_type_t *mep_type);
int bcm_field_qualify_EgressClassVxlan_get(int unit, bcm_field_entry_t entry,
   uint16 *data, uint16 *mask);
int bcm_field_qualify_OamEgressEtherType_get(int unit, bcm_field_entry_t entry,
 uint16 *data);
int bcm_field_qualify_OamEgressMulticastMacHit_get(int unit, bcm_field_entry_t
entry,
       uint8 *data);
int bcm_esw_field_qualify_TrunkMemberSourceModuleId_get(int unit,
bcm_field_entry_t entry,
int *modid, int *modid_mask);
int bcm_field_qualify_OamEgressClassSrcMac_get(int unit, bcm_field_entry_t
entry,
                                             uint32 *data, uint32 *mask);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`data`	Data to match against
`mask`	Mask to choose which bits of data to match against

Description

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

Returns

`BCM_E_XXX`

KERNEL NETWORK (KNET) CONFIGURATION

New criteria of RX packet filter for KNET is added.

Table 24: New BCM KNET Rx Packet Filter Match Criteria

Name	***Purpose***
`BCM_KNET_FILTER_M_ERROR`	Match error bit

LAYER 2 ADDRESS MANAGEMENT

New L2 learning event flag is added for CPU DMA distributer.

Table 25: New L2 Learning events flag

Name	***Purpose***
`BCM_L2_ADDR_DIST_SET_CPU_DMA_DISTRIBUTER`	Send events to the CPU DMA distributer

LAYER 3 MANAGEMENT

New flag for Layer 3 Flags2 is added for L2 egress object creation.

Table 26: BCM Layer 3 Flags2

Name	***Purpose***
`BCM_L3_FLAGS2_UNDERLAY`	When L3 egress objects are created with this flag, underlay next hop could be created on a virtual port.

Table 26: BCM Layer 3 Flags2

Name	Purpose
`BCM_L3_FLAGS2_SRC_DST_MAC_SWAP`	Swap source/destination MAC for outgoing packets, for IFP application only.

New fields are added in L3 egress data structure `bcm_l3_egress_t` in this release for VNTAG and ETAG.

```
/*
 * L3 Egress Structure
 * Description of an L3 forwarding destination.
 */
typedef struct bcm_l3_egress_s {
    ...
    bcm_vntag_t         vntag;                    /* VNTAG. */
    bcm_vntag_action_t  vntag_action;             /* VNTAG action. */
    bcm_etag_t          etag;                     /* ETAG. */
    bcm_etag_action_t   etag_action;              /* ETAG action. */
} bcm_l3_egress_t;
```

New field `l3a_expected_src_gport` is added in `bcm_l3_route_t` data structure for L3 Multicast group expected source port and trunk

```
typedef struct bcm_l3_route_s {
    ...
        /* L3 Multicast group expected source port/trunk */
    bcm_gport_t l3a_expected_src_gport;
} bcm_l3_route_t;
```

The new flag for L3 ingress interface flag is added, where it may take on values which are the logical OR of one or more of the defined flags.

Table 27: New BCM Layer3 Ingress Interface Flags

Name	Purpose
`BCM_L3_INGRESS_SRC_DST_NAT_REALM_ID`	To set the nat realm id to both source nat id and destionation nat id.

New tunnel flag is added.

Table 28: New BCM Tunnel Flag

Name	Purpose
`BCM_TUNNEL_TERM_USE_OUTER_PCP`	Use outer header PCP.

New fields are added in `bcm_tunnel_terminator_t` and `bcm_tunnel_initiator_t` for L3 tunnel terminator and initiator parameters respectively.

```
typedef struct bcm_tunnel_terminator_s {
    ...
```

```
        int inlif_counting_profile;                 /* In LIF counting profile */
    } bcm_tunnel_terminator_t;

    typedef struct bcm_tunnel_initiator_s {
        ...
       int                outlif_counting_profile;  /* Out LIF counting profile */
    } bcm_tunnel_initiator_t;
```

MAC-IN-MAC MANAGEMENT

New criteria of MiM port match for VPN is added.

Table 29: MiM port Match Criteria

Criteria	*Description*
BCM_MIM_PORT_MATCH_SHARE	Multiple matches share one MiM port.

New fields of the bcm_mim_vpn_config_t and bcm_mim_port_t structures that describes the MAC-in-MAC VPN attributes are added. It allows the application to configure different broadcast, multicast and unknown unicast groups.

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

New flag is added for the MAC-in-MAC port flags.

Table 30: New MAC-in-MAC Port Flag

Name	*Purpose*
BCM_MIM_PORT_FAILOVER_DROP	Be enabled to drop data for 1+1 protection.

MPLS MANAGEMENT

New flag BCM_MPLS_EGRESS_LABEL_ENTROPY_INDICATION_ENABLE is added for MPLS Egress label as well as the new field in bcm_mpls_egress_label_t data structure that describes the MPLS label header values (label, EXP, TTL) and QOS options.

Table 31: MPLS Egress Label Flags

Name	Purpose
BCM_MPLS_EGRESS_LABEL_ENTROPY_INDICATION_ENABLE	Add Entropy Indication label for outgoing Tunnel-label

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

New flag in the extension of MPLS port flags is added:

Table 32: New MPLS Port Flags(2)

Name	Purpose
BCM_MPLS_PORT2_PROXY_MODE	Act as proxy mode, and do packet modifications for HG output port on VPWS stacking mode

Two new fields are added in the bcm_mpls_port_t structure that describes an MPLS port (or "virtual-port") used for L2 MPLS VPNs (VPWS and VPLS VPNs):

```
typedef struct bcm_mpls_port_s {
    ...
    uint32 class_id;                    /* Class ID */
    int inlif_counting_profile;         /* In LIF counting profile */
} bcm_mpls_port_t;
```

A new field inlif_counting_profile is added in bcm_mpls_tunnel_switch_t data structure for bcm_mpls_tunnel_switch_add and bcm_mpls_tunnel_switch_create APIs.

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

OPERATIONS, ADMINISTRATION, AND MAINTENANCE

New OAM endpoint action flag is added that is used with bcm_oam_endpoint_action_t data structure.

Table 33: New OAM Endpoint action flags

Flag	Description
BCM_OAM_ENDPOINT_ACTION_DISABLE	Reset the endpoint action(s)

New OAM loopback flag BCM_OAM_LOOPBACK_PERIOD_IN_KPPS is added in this release.

Table 34: OAM Loopback Flag Definitions

Flag	Description
BCM_OAM_LOOPBACK_PERIOD_IN_KPPS	Default is milliseconds. flag may be used for kilo packet per second.
BCM_OAM_LOOPBACK_TX_ENABLE	Enable Loopback Packet Transmission. From 6.4.8, 6.5.2 onwards it is mandatory to set this flag while calling the API bcm_oam_loopback_add() to start LBM transmission.

The data types of rx_count and invalid_tlv_count in OAM TST rx objects are extended to 64 bits.

```
/* 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) */
    ...
} bcm_oam_tst_rx_t;
```

New field far_loss is added in the data structure bcm_oam_pm_stats_t.

```
/* PM stats structure */
typedef struct bcm_oam_pm_stats_s {
    ...
    uint32 far_loss;                    /* Total far loss at far-end */
    ...
} bcm_oam_pm_stats_t;
```

Fault flags are defined for OAM BHH endpoint/group in this release.

Table 35: OAM BHH endpoint/group fault flags Definitions

Fault Flag	Description
BCM_OAM_BHH_FAULT_CCM_TIMEOUT	A BHH CCM time out alarm
BCM_OAM_BHH_FAULT_CCM_RDI	A BHH CCM time out alarm
BCM_OAM_BHH_FAULT_CCM_UNKNOWN_MEG_LEVEL	A BHH CCM with unknown MEG level alarm
BCM_OAM_BHH_FAULT_CCM_UNKNOWN_MEG_ID	A BHH CCM with unknown MEG ID alarm
BCM_OAM_BHH_FAULT_CCM_UNKNOWN_MEP_ID	A BHH CCM with unknown MEP ID alarm
BCM_OAM_BHH_FAULT_CCM_UNKNOWN_PERIOD	A BHH CCM with unknown period alarm
BCM_OAM_BHH_FAULT_CCM_UNKNOWN_PRIORITY	A BHH CCM with unknown priority alarm

NEW OAM OPCODE GROUP TABLE SET/GET OBJECTS

```
/* Types of OAM protocols */
typedef enum bcm_oam_protocol_type_e {
    bcmOamProtocolEth = 0,      /* ETH-OAM Protocol */
    bcmOamProtocolBhh = 1,      /* BHH-OAM Protocol */
    bcmOamProtocolMplsLmDm = 2, /* MPLS LM/DM Protocol */
```

```
        bcmOamProtocolMax = 3       /* Max No. of Protcols.Not an usable
                       value.Always at the end */
    } bcm_oam_protocol_type_t;

    /* Types of OAM protocols */
    typedef enum bcm_oam_opcode_e {
        bcmOamOpcodeCCM = 1,        /* Opcode for CCM */
        bcmOamOpcodeLBR = 2,        /* Opcode for LBR */
        bcmOamOpcodeLBM = 3,        /* Opcode for LBM */
        bcmOamOpcodeLTR = 4,        /* Opcode for LTR */
        bcmOamOpcodeLTM = 5,        /* Opcode for LTM */
        bcmOamOpcodeAIS = 33,       /* Opcode for AIS */
        bcmOamOpcodeLCK = 35,       /* Opcode for LCK */
        bcmOamOpcodeTST = 37,       /* Opcode for TST */
        bcmOamOpcodeLinearAPS = 39, /* Opcode for Linear APS */
        bcmOamOpcodeRingAPS = 40,   /* Opcode for Ring APS */
        bcmOamOpcodeMCC = 41,       /* Opcode for MCC */
        bcmOamOpcodeLMR = 42,       /* Opcode for LMR */
        bcmOamOpcodeLMM = 43,       /* Opcode for LMM */
        bcmOamOpcode1DM = 45,       /* Opcode for 1DM */
        bcmOamOpcodeDMR = 46,       /* Opcode for DMR */
        bcmOamOpcodeDMM = 47,       /* Opcode for DMM */
        bcmOamOpcodeEXR = 48,       /* Opcode for EXR */
        bcmOamOpcodeEXM = 49,       /* Opcode for EXM */
        bcmOamOpcodeVSR = 50,       /* Opcode for VSR */
        bcmOamOpcodeVSM = 51,       /* Opcode for VSM */
        bcmOamOpcodeCSF = 52,       /* Opcode for MCC */
        bcmOamOpcode1SL = 53,       /* Opcode for 1SL */
        bcmOamOpcodeSLR = 54,       /* Opcode for SLR */
        bcmOamOpcodeSLM = 55,       /* Opcode for 1DM */
       bcmOamOpcodeMax = 256       /* Max. Opcodes.Not an usable value.Always
last */
    } bcm_oam_opcode_t;

    /* Struture to set the Opcodes bitmap */
    typedef struct bcm_oam_opcodes_s {
       SHR_BITDCL opcode_bmp[_SHR_BITDCLSIZE(256)]; /* Opcodes bitmap vector */
    } bcm_oam_opcodes_t;
```

Table 36: OAM Opcodes Bitmap Macros

Macro	***Description***
`BCM_OAM_OPCODES_BITMAP_SET(_opcodes_bitmap, _opcode_)`	Set the specified opcode in the bitmap of opocdes structure
`BCM_OAM_OPCODES_BITMAP_GET(_opcodes_bitmap, _opcode_)`	Get the status of the specified opcode in the bitmap of opocdes structure
`BCM_OAM_OPCODES_BITMAP_CLEAR(_opcodes_bitmap, _opcode_)`	Clear the specified opcode in the bitmap of opocdes structure
`BCM_OAM_OPCODES_BITMAP_SET_ALL(_opcodes_bitmap)`	set all the opcodes in the bitmap of opocdes structure
`BCM_OAM_OPCODES_BITMAP_CLEAR_ALL(_opcodes_bitmap)`	Clear all the opcodes in the bitmap of opcodes structure

bcm_oam_opcode_group_set

Set the Group for the specified opcodes of the Protocols

Syntax

```
#include <bcm/oam.h>
extern int bcm_oam_opcode_group_set(
int unit,
bcm_oam_protocol_type_t protocol,
bcm_oam_opcodes_t opcodes,
uint8 opcode_group);
```

Parameters

`unit`	BCM device number
`protocol`	(IN) Protocol for which opcode group need to be set
`opcodes`	(IN) Opcodes for which opcode group need to be set
`opcode_group`	(IN) Opcode group value need to be set for the specified opcodes. Range of Opcode Group value is (0-15).

Description

It sets the user specified opcode group for the given opcodes under a protocol. Note that `opcode_bmp` needs to be set with opcode types which are defined in `bcm_oam_mpls_lm_dm_opcode_type_t` for Protocol type = bcmOamProtoMplsLmDm.

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 opcode group value or protocol
`BCM_E_INTERNAL`	Unable to release resource lock

bcm_oam_opcode_group_get

Get the opcode group for each opcode of each protocol

Syntax

```
#include <bcm/oam.h>
extern int bcm_oam_opcode_group_get(
int unit,
bcm_oam_protocol_type_t protocol,
bcm_oam_opcode_t opcode,
uint8 *opcode_group);
```

Parameters

`unit`	BCM device number
`protocol`	(IN) Protocol for which opcode group to get
`opcode`	(IN) Opcode for which opcode group need to be set
`opcode_group`	(OUT) Pointer to the returned opcode group value

Description

It gives back the opcode group value for the specified protocol and single opcode.

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 opcode or protocol
`BCM_E_INTERNAL`	Unable to release resource lock

New OAM APIs that are introduced in this release.

bcm_oam_endpoint_gport_egress_attach

Attach an egress MPLS gport to a MPLS LSP endpoint

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_gport_egress_attach(int unit,
 bcm_oam_endpoint_t endpoint,
 bcm_gport_t gport);
```

Parameters

`unit`	(IN) BCM device number.
`endpoint`	(IN) ID of the endpoint to which you would like to associate the egress MPLS gport
`gport`	(IN) MPLS gport whose egress interface needs to be associated to the endpoint

Description

This API helps in adding multiple MPLS pseudowire egress interfaces to be 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_EXISTS	Gport already attached
BCM_E_INTERNAL	Unable to release resource lock / Failed to read or write register

bcm_oam_endpoint_gport_egress_attach_get

Get the list of MPLS gports attached to a MPLS LSP endpoint

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_gport_egress_attach_get(int unit,
 bcm_oam_endpoint_t endpoint,
 int max_count,
 bcm_gport_t *gport,
 int *count);
```

Parameters

unit	(IN) BCM device number.
endpoint	(IN) ID of the endpoint for which you would like to get the associated MPLS gports
max_count	(IN) Size of array of gports passed in the gport parameter
gport	(OUT) List of MPLS gports whose egress interfaces are associated to the endpoint
count	(OUT) Actual number of MPLS gports which is stored in the gport array

Description

This API is used to get multiple MPLS pseudowire 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_gport_egress_detach

Detach an egress MPLS gport from a MPLS LSP endpoint

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_gport_egress_detach(int unit,
 bcm_oam_endpoint_t endpoint,
 bcm_gport_t gport);
```

Parameters

`unit`	(IN) BCM device number.
`endpoint`	(IN) ID of the endpoint from which you would like to dis-associate the egress MPLS gport
`gport`	(IN) MPLS gport whose egress interface needs to be dis-associated from the endpoint

Description

This API helps in removing the association of MPLS pseudowire egress interfaces with 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 gport is not associated with the endpoint
`BCM_E_INTERNAL`	Unable to release resource lock / Failed to read or write register

POLICER

New policer APIs are added with the required data types.

bcm_policer_color_decision_get

Get an entry from the Policer Color Decision table

Syntax

```
#include <bcm/policer.h>
int bcm_policer_color_decision_get(
    int unit,
    bcm_policer_color_decision_t *policer_color_decision);
```

Parameters

unit	(IN) Unit number
bcm_policer_color_decision_t	(IN) Policer Color Decision entry

Description

The Policer Color Decision table maps the result of two policers and the status of each of their respective 'other color' to result Policer Color and determines which bucket will be updated on each meter. Returns {policer0_update_bucket, policer1_update_bucket, policer_color} for given: {policer0_color, policer0_other_bucket_has_credits, policer1_color, policer1_other_bucket_has_credits}

```
    typedef enum bcm_policer_color_e {
        bcmPolicerColorGreen = 0,   /* Green color */
        bcmPolicerColorInvalid = 1, /* Invalid color */
        bcmPolicerColorYellow = 2,  /* Yellow color */
        bcmPolicerColorRed = 3      /* Red color */
    } bcm_policer_color_t;

    typedef struct bcm_policer_color_decision_s {
        uint32 flags;                       /* color decision flags */
        bcm_policer_color_t policer0_color; /* policer-0 color decision */
      uint8 policer0_other_bucket_has_credits; /* if policer0_color is yellow,
indicate
                           if green bucket has credits,
                           otherwise if yellow bucket has
                           credits */
        bcm_policer_color_t policer1_color; /* policer-1 color decision */
      uint8 policer1_other_bucket_has_credits; /* if policer1_color is yellow,
indicate
                           if green bucket has credits,
                           otherwise if yellow bucket has
                           credits */
       bcm_policer_color_t policer0_update_bucket; /* the policer-0 bucket to
update */
       bcm_policer_color_t policer1_update_bucket; /* the policer-1 bucket to
update */
        bcm_color_t policer_color;          /* policer color decision */
    } bcm_policer_color_decision_t;
```

Returns

BCM_E_XXX

bcm_policer_color_decision_set

Set an entry in the Policer Color Decision table

Syntax

```
#include <bcm/policer.h>
int bcm_policer_color_decision_set(
    int unit,
    bcm_policer_color_decision_t *policer_color_decision);
```

Parameters

unit	(IN) Unit number
bcm_policer_color_decision_t	(IN) Policer Color Decision entry

Description

The Policer Color Decision table maps the result of two policers and the status of each of their respective 'other color' to result Policer Color and determines which bucket will be updated on each meter. Sets {policer0_update_bucket, policer1_update_bucket, policer_color} for given: {policer0_color, policer0_other_bucket_has_credits, policer1_color, policer1_other_bucket_has_credits}

Returns

BCM_E_XXX

bcm_policer_color_resolution_get

Get an entry from the Policer Color Resolution table

Syntax

```
#include <bcm/policer.h>
int bcm_policer_color_resolution_get(
    int unit,
    bcm_policer_color_resolution_t *policer_color_resolution);
```

Parameters

unit	(IN) Unit number
bcm_policer_color_resolution_t	(IN) Policer Color Resolution entry

Description

The Policer Color Resolution table maps the policer-action command, the incoming color, the ethernet policer color (BCM88670 only) and the policer color to ingress color (which is set on the FTMH) and egress color (which is forwarded to the IQM). Returns {ingress_color, egress_color} for given: {policer_action, incoming_color, ethernet_policer_color, policer_color}

```
    typedef struct bcm_policer_color_resolution_s {
        uint32 flags;                   /* color resolution flags */
        uint32 policer_action;          /* policer command */
        bcm_color_t incoming_color;     /* incoming color */
        bcm_color_t ethernet_policer_color; /* ethernet policer color */
        bcm_color_t policer_color;          /* policer color */
        bcm_color_t ingress_color;          /* ingress color */
        bcm_color_t egress_color;           /* egress color */
    } bcm_policer_color_resolution_t;
```

Returns

BCM_E_XXX

bcm_policer_color_resolution_set

Set an entry in the Policer Color Resolution table

Syntax

```
#include <bcm/policer.h>
int bcm_policer_color_resolution_set(
    int unit,
    bcm_policer_color_resolution_t *policer_color_resolution);
```

Parameters

unit	(IN) Unit number
bcm_policer_color_resolution_t	(IN) Policer Color Resolution entry

Description

The Policer Color Resolution table maps the policer-action command, the incoming color, the ethernet policer color (BCM88670 only) and the policer color to ingress color (which is set on the FTMH) and egress color (which is forwarded to the IQM). Sets {ingress_color, egress_color} for given: {policer_action, incoming_color, ethernet_policer_color, policer_color}

Returns

BCM_E_XXX

PORT CONFIGURATION

New field control is added in bcm_port_config_t data structure for bcm_port_config_get and bcm_port_config_set APIs.

Table 37: bcm_port_config_t

Field	*BCM Type*	*Description*
control	bcm_pbmp_t	Bitmap or logical port list of all hot swap controlling ports

The defines of new port extended abilities are introduced in this release.

Table 38: New EXTENDED_PORT_ABILITY_s

Ability	Description
BCM_PORT_ABILITY_FEC	Ability to support FEC
BCM_PORT_ABILITY_FEC_REQUEST	Ability to request FEC

New port interfaces are added in this release for the APIs bcm_port_interface_get and bcm_port_interface_set.

Table 39: New Port Interface Values

BCM_PORT_IF_LR2	Fiber 2x10G 64B/66B interface.
BCM_PORT_IF_LRM	Fiber multimode 64B/66B interface.
BCM_PORT_IF_XLPPI	40G parallel physical interface.

New port class types are added as shown in below table for bcm_port_class_set and bcm_port_class_get APIs.

Table 40: New bcm_port_class_t

Class type	Description
bcmPortClassFieldIngressSystemPort	Class for virtual port instance of SGLP in IFP IPBM key
bcmPortClassFieldIngressSourceGport	Class for virtual port instance of SVP in IFP IPBM key
bcmPortClassFieldIngressDevicePort	Class for virtual port instance of device port in IFP IPBM key

New port control types are added in this release for bcm_port_control_set and bcm_port_control_get APIs.

Table 41: bcm_port_control_t

Control type	Description
bcmPortControlSampleIngressEnable	Enable/Disable ingress port-based sflow.
bcmPortControlSampleEgressEnable	Enable/Disable egress port-based sflow.
bcmPortControlSampleIngressRate	Set Ingress sFlow sample rate.
bcmPortControlSampleEgressRate	Set Egress sFlow sample rate.
PortControlMplsSpeculativeParse	Enable/Disable the speculative parsing of headers over MPLS. Valid values : True/False.

The new port phy configuration types are shown in New bcm_port_phy_control_t (page 52) for bcm_port_phy_control_set and bcm_port_phy_control_get APIs.

Table 42: New bcm_port_phy_control_t

New Port PHY control type	Description
BCM_PORT_PHY_CONTROL_AUTONEG_ALLOW_PLL_CHANGE	value of 1 allow PLL to change during AN, otherwise no pll change is allowed .

New port match criteria is added for Generic Port Match APIs.

The DNX family of switches supports asymmetrical discard mode configuration. To set the discard mode for ingress or egress only, the mode for bcm_port_discard_get and bcm_port_discard_set APIs can be ored with BCM_PORT_DISCARD_MODE_INGRESS_ONLY or BCM_PORT_DISCARD_MODE_EGRESS_ONLY respectively.

New port APIs are introduced with the required data types.

bcm_port_nif_priority_set
bcm_port_nif_priority_get

Set/Get port NIF priority.

Syntax

```
#include <bcm/port.h>
/* Set the NIF Priority between Ports. */
extern int bcm_port_nif_priority_set(
    int unit,
    bcm_gport_t local_port,
    uint32 flags,
    bcm_port_nif_prio_t *priority,
    bcm_pbmp_t *affected_ports);

/* Get the NIF Priority of a Port. */
extern int bcm_port_nif_priority_get(
    int unit,
    bcm_gport_t local_port,
    uint32 flags,
    bcm_port_nif_prio_t *priority,
    bcm_pbmp_t *affected_ports);
```

Parameters

unit	(IN) Unit number.
local_port	(IN) local port to setget NIF prio.
flags	(IN) flags for function.
priority	(INOUT) priority to set.
affected_ports	(IN) other ports that were affected by this configuration.

Description

Set/Get port NIF priority. configuration is per port, however changing one port may have an effect on others, which will be reflacted in affected ports.

The structure bcm_port_nif_prio_t is used to describte the attributes of the Priority.

```
    typedef struct bcm_port_nif_prio_s {
        int priority_level; /* NIF Priority level to set/get */
    } bcm_port_nif_prio_t;
```

Returns

BCM_E_XXX	Error occurred

PRECISION TIME PROTOCOL

New PTP APIs are introduced in this release.

bcm_ptp_signaling_arbiter_register

Register an arbiter for handling PTP signaling logic

Syntax

```
#include <bcm/ptp.h>
int
bcm_ptp_signaling_arbiter_register(
    int unit,
    bcm_ptp_stack_id_t ptp_id,
    int clock_num,
    bcm_ptp_signaling_arbiter_t arb,
    void *user_data);
```

Parameters

unit	(IN) Unit Number
ptp_id	(IN) PTP Stack ID
clock_num	(IN) PTP Clock Number
arb	(IN) The arbiter function to be called
user_data	(IN) Pointer to user data to supply in the callback

Description

Register an arbiter callback for handling PTP signaling logic. This function will be called whenever a signaling message is received, with the parsed fields in the bcm_ptp_cb_signaling_arbiter_msg_t struct. The arbiter function can change these fields as desired (to update the grant duration, for example). The arbiter then can return bcmPTPCallbackReject, to drop the signaling message, or bcmPTPCallbackAccept, to accept it. Note that a signaling grant response with a grant duration of 0 is the PTP mechanim to inform the slave that the grant was not accepted.

Returns

BCM_E_NONE	Operation completed successfully

bcm_ptp_signaling_arbiter_unregister

Unregister an arbiter for handling PTP signaling logic

Syntax

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

Parameters

unit	(IN) Unit Number
ptp_id	(IN) PTP Stack ID
clock_num	(IN) PTP Clock Number

Description

Unregister the current arbiter callback for handling PTP signaling logic.

Returns

BCM_E_NONE	Operation completed successfully

SERVICE ACTIVATION TEST (SAT)

New field last_packet_seq_number is added in the data structure of SAT CTF Stat Info:

```
    typedef struct bcm_sat_ctf_stat_s {
        ...
        uint64 last_packet_seq_number;      /* Last received packet sequence
number */
    } bcm_sat_ctf_stat_t;
```

New SAT APIs are introduced in this release.

bcm_sat_gtf_packet_edit_t_init

Initialize a bcm_sat_gtf_packet_edit_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_gtf_packet_edit_t_init(bcm_sat_gtf_packet_edit_t *packet_edit);
```

Parameters

`packet_edit`	(OUT) Pointer to SAT GTF packet edit structure to initialize

Description

Initializes a SAT GTF packet edit structure to default values.

bcm_sat_endpoint_info_t_init

Initialize a bcm_sat_endpoint_info_t structure.

Syntax

```
#include <bcm/sat.h>
void bcm_sat_endpoint_info_t_init(bcm_sat_endpoint_info_t *endpoint_info);
```

Parameters

`endpoint_info`	(OUT) Pointer to SAT endpoint info structure to initialize

Description

Initializes a SAT endpoint info structure to default values.

STATISTICS

New statistics accounting object is added for Ingress Gport Object.

Table 43: New statistics accounting object

Name	*Description*
bcmStatObjectIngGport	Ingress Gport Object

New data types for LIF counting are added as following.

```
/* 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;
```

New statistics APIs are added.

bcm_stat_counter_filter_set

set counter source to filter in or out certain filter criteria.

Syntax

```
#include <bcm/stat.h>
int bcm_stat_counter_filter_set(
      int unit,
      bcm_stat_counter_source_t source,
      bcm_stat_counter_filter_t* filter_array,
      int filter_count,
      int is_active);
```

Parameters

`unit`	(IN) Unit number
`source`	(IN) engine source to filter (relevant fields are `command_id` and `engine_source`)
`filter_array`	(IN) filters to set
`filter_count`	(IN) members in array
`is_active`	(IN) if NOT set filter in received filters criteria instead of filter out

Description

Function is used to filter in and out certain filters criteria for the CRPS.

```
    /*
     * different types of filters to be filtered in or out by the counter
proccesor.
     */
    typedef enum bcm_stat_counter_filter_e {
      bcmStatCounterTotalPDsThresholdViolated, /* Total PDs Threshold Violated
- for Egress Receive engine source */
       bcmStatCounterTotalPDsUcPoolSizeThresholdViolated, /* Total PDs Uc Pool
Size Threshold Violated - for Egress Receive engine source */
      bcmStatCounterPerPortUcPDsThresholdViolated, /* Per Port Uc PDs Threshold
Violated - for Egress Receive engine source */
        bcmStatCounterPerQueueUcPDsThresholdViolated, /* Per Queue Uc PDs
Threshold Violated - for Egress Receive engine source */
      bcmStatCounterPerPortUcDBsThresholdViolated, /* Per Port Uc DBs Threshold
Violated - for Egress Receive engine source */
        bcmStatCounterPerQueueUcDBsThresholdViolated, /* Per Queue Uc DBs
Threshold Violated - for Egress Receive engine source */
        bcmStatCounterPerQueueDisableBit,   /* Stat Counter Per Queue Disable
Bit - for Egress Receive engine source */
       bcmStatCounterTotalPDsMcPoolSizeThresholdViolated, /* Total PDs Mc Pool
Size Threshold Violated - for Egress Receive engine source */
        bcmStatCounterPerInterfacePDsThreholdViolated, /* Per Interface PDs
Threhold Violated - for Egress Receive engine source */
        bcmStatCounterMcSPThresholdViolated, /* Mc SP Threshold Violated - for
Egress Receive engine source */
       bcmStatCounterPerMcTCThresholdViolated, /* Per Mc TC Threshold Violated
- for Egress Receive engine source */
      bcmStatCounterMcPDsPerPortThresholdViolated, /* Mc PDs Per Port Threshold
Violated - for Egress Receive engine source */
        bcmStatCounterMcPDsPerQueueThresholdViolated, /* Mc PDs Per Queue
Threshold Violated - for Egress Receive engine source */
        bcmStatCounterMcPerPortSizeThresholdViolated, /* Mc Per Port Size
Threshold Violated - for Egress Receive engine source */
        bcmStatCounterMcPerQueueSizeThresholdViolated, /* Mc Per Queue Size
Threshold Violated - for Egress Receive engine source */
        bcmStatCounterGlobalRejectDiscards, /* Global Reject Discards - for
Ingress engine source */
      bcmStatCounterDramRejectDiscards,   /* Dram Reject Discards - for Ingress
engine source */
        bcmStatCounterVoqTailDropDiscards,  /* Voq Tail Drop Discards - for
Ingress engine source */
        bcmStatCounterVoqStatisticsDiscards, /* Voq Statistics Discards - for
Ingress engine source */
        bcmStatCounterVsqTailDropDiscards,  /* Vsq Tail Drop Discards - for
Ingress engine source */
        bcmStatCounterVsqStatisticsDiscards, /* Vsq Statistics Discards - for
Ingress engine source */
        bcmStatCounterQueueNotValidDiscard, /* Queue Not Valid Discard - for
Ingress engine source */
        bcmStatCounterOtherDiscards,        /* Other Discards - for Ingress
engine source */
        bcmStatCounterDropReasonCount       /* Number of Drop Reasons  should
always be last entry */
    } bcm_stat_counter_filter_t;

   #define BCM_STAT_COUNTER_TM_COMMAND 0xFF       /* flag for dedicated engine
using TM command */
```

Returns

```
BCM_E_XXX
```

bcm_stat_counter_filter_get

determine which filter criteria a certain counter source is filtering out.

Syntax

```
#include <bcm/stat.h>
int bcm_stat_counter_filter_get(
      int unit,
      bcm_stat_counter_source_t source,
      int filter_max_count,
      bcm_stat_counter_filter_t* filter_array,
      int* filter_count);
```

Parameters

`unit`	(IN) Unit number
`source`	(IN) engine source to filter (relevant fields are `command_id` and `engine_source`)
`filter_max_count`	(IN) size of array - to prevent overflow
`filter_array`	(OUT) returned filters
`filter_count`	(OUT) nof filters

Description

determine which filter criteria a certain counter source is filtering out.

Returns

```
BCM_E_XXX
```

bcm_stat_counter_filter_is_active_get

determine if counter is filtering out certain filter criterion.

Syntax

```
#include <bcm/stat.h>
int bcm_stat_counter_filter_is_active_get(
      int unit,
      bcm_stat_counter_source_t source,
      bcm_stat_counter_filter_t filter,
      int* is_active);
```

Parameters

`unit`	(IN) Unit number
`source`	(IN) engine source to filter (relevant fields are `command_id` and `engine_source`)
`filter`	(IN) Defines the filter to check
`is_active`	(OUT) return if the given filter criterion is filtered out

Description

determine if counter is filtering out certain filter criterion.

Returns

BCM_E_XXX

SUBPORT CONFIGURATION

New fields in bcm_subport_group_config_t with the new subport group flag queues are added for bcm_subport_group_create API.

Table 44: New Subport Group Flag

Name	*Purpose*
BCM_SUBPORT_GROUP_QUEUE	If set, use subport group queues; Else, use egress port DiffServ queues.

```
typedef struct bcm_subport_group_config_s {
        ...
    bcm_gport_t queue_base; /* Subport group base queue */
   int       qos_map_id; /* QoS map identifier which defines priorities 1-1
                   mapping to queue offsets (0 ~ 7). */
} bcm_subport_group_config_t;
```

SWITCH CONTROL

New switch controls are added in this release for bcm_switch_control_xxx() APIs.

Table 45: New Switch Type Values

Value	*Description*	*Arg Value*
bcmSwitchFlexIP6ExtHdr	Set and get the programmable value for customer defined extension header on ingress.	0-255, 8 bit number
bcmSwitchFlexIP6ExtHdrEgress	Set and get the programmable value for customer defined extension header on egress.	0-255, 8 bit number
bcmSwitchHashUseFlowSelMplsEcmp	Enable flow based hash selection for MPLS ECMP.	TRUE/FALSE
bcmSwitchV6L3SrcDstLocalLinkDropCancel	Do not drop IPv6. It will override drop control bcmSwitchV6L3LocalLinkDrop if source and destination are local link.	TRUE/FALSE
bcmSwitchL3ClassIdForL2Enable	Enable L3 Lookup to get classids based on DIP & SIP for L2 Switched packets.	TRUE/FALSE
bcmSwitchDropSobmhOnLinkDown	Enable/Disable SOBMH packet transmit blocking on link down egress port.	TRUE/FALSE

New switch objects are added in this release for `bcm_switch_object_count_get` related APIs.

Table 46: New switch objects for bcm_switch_object_t

bcmSwitchObjectPFCDeadlockCosMax	***Get the Max COS supported for PFC Deadlock detection and recovery.***
bcmSwitchObjectL3HostV4Max	Maximum number of IPv4 routes possible in L3 host table.
bcmSwitchObjectL3HostV6Max	Maximum number of IPv6 routes possible in L3 host table.
bcmSwitchObjectL3RouteV4RoutesMinGuaranteed	Guaranteed number of IPv4 routes in ALPM mode.
bcmSwitchObjectL3RouteV6Routes64bMinGuaranteed	Guaranteed number of IPv6 64bits routes in ALPM mode.
bcmSwitchObjectL3RouteV6Routes128bMinGuaranteed	Guaranteed number of IPv6 128bits routes in ALPM mode.
bcmSwitchObjectL3EgressMax	Maximum number of L3 Egress objects possible.
bcmSwitchObjectIpmcV4Used	Statistics of used IPv4 Multicast entries.
bcmSwitchObjectIpmcV6Used	Statistics of used IPv6 Multicast entries.
bcmSwitchObjectIpmcV4Max	Maximum number of IPv4 Multicast entries possible.
bcmSwitchObjectIpmcV6Max	Maximum number of IPv6 Multicast entries possible.

TIME CONFIGURATION

The new flag field in the structure is defined and the logical OR of the all other flags listed in `BCM_Time_Synchronization_Flags_x`

Table 47: New BCM Time Synchronization Flags

Name	***Description***
`BCM_TIME_NTP_OFFSET`	To set/get the ntp offset parameter for Interface

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

TRILL MANAGEMENT

New TRILL port flag `BCM_TRILL_MULTICAST_REPLACE_DYNAMIC` is added.

Table 48: New TRILL port flag

Name	***Purpose***
`BCM_TRILL_MULTICAST_REPLACE_DYNAMIC`	Replace an existing dynamic L2 entry with TRILL entry, when table is full

New fields in `bcm_trill_port_t` data structure are added.

```
typedef struct bcm_trill_port_s {
    ...
```

```
        /* Switch split horizon network group ID this TRILL port belongs to*/
      bcm_switch_network_group_t network_group_id;
      int inlif_counting_profile;          /* In LIF counting profile */
      int outlif_counting_profile;         /* Out LIF counting profile */
   } bcm_trill_port_t;
```

PACKET TRANSMIT AND RECEIVE

New OAM packet types are added in this release.

Table 49: New OAM Packet type Descriptions

Name	***Description***
bcmPktRxOamTypeEthOamUpMepCcm	Ethernet OAM Up MEP CCM packet.
bcmPktRxOamTypeEthOamUpMepLm	Ethernet OAM Up MEP LM packet.
bcmPktRxOamTypeEthOamUpMepDm	Ethernet OAM Up MEP DM CCM packet.
bcmPktRxOamTypeEthOamUpMepOther	Ethernet OAM Up MEP other opcode packet.
bcmPktRxOamTypeUpSat	Up Service Activation Test packet.

BCM TYPES

New macros are designated for `field_stat` collction.

Table 50: Macros for Analyzing/Constructing forward encap ids

BCM_FIELD_STAT_ID_SET Macros	***Creates a stat_id encoding of counter engine ID and counter ID***
`BCM_FIELD_STAT_ID_COUNTER_GET` Macros	Returns counter engine ID from `stat_id`
`BCM_FIELD_STAT_ID_PROCESSOR_GET` Macros	Returns counter ID from `stat_id`

VLAN MANAGEMENT

New VLAN control flag is added for `bcm_vlan_control_vlan_set` and `bcm_vlan_control_vlan_get` APIs.

Table 51: New VLAN Control Flags

VLAN Control Flags	***Meaning***
`BCM_VLAN_SRC_DST_NAT_REALM_ID`	To set the nat realm id to both source nat id and destionation nat id.

New fields are added in `bcm_vlan_port_t` data structure where the parameters of a Layer 2 Logical Port are passed to `bcm_vlan_port_create` API in the form of `bcm_vlan_port_t` structure, which is defined as follows:

```
    typedef struct bcm_vlan_port_s {
        ...
        uint8 pkt_pri;                      /* Service tag priority */
        uint8 pkt_cfi;                      /* Service tag cfi */
        unit16 egress_service_tpid;         /* Service VLAN TPID Value */
      bcm_vlan_t egress_vlan;           /* Egress Outer VLAN or SD-TAG VLAN ID. */
        ...
        int inlif_counting_profile;         /* In LIF counting profile */
        int outlif_counting_profile;        /* Out LIF counting profile */
    } bcm_vlan_port_t;
```

New VLAN VPN APIs with the required data types are added.

Table 52: VLAN VPN flags

BCM_VLAN_VPN_ELINE	***Create ELINE VPN***
BCM_VLAN_VPN_ELAN	Create ELAN VPN
BCM_VLAN_VPN_WITH_ID	Create VPN with specified ID

```
    typedef struct bcm_vlan_vpn_config_s {
          unit32 flags;
          bcm_vpn_t vpn;
          bcm_multicast_t broadcast_group;
          bcm_multicast_t unknown_unicast_group;
          bcm_multicast_t unknown_multicast_group;
      } bcm_vlan_vpn_config_t;
```

Table 53: VLAN port flags

BCM_VLAN_PORT_SERVICE_TAGGED	***Service tag mode***
BCM_VLAN_PORT_SERVICE_VLAN_ADD	Add SD-tag
BCM_VLAN_PORT_SERVICE_VLAN_REPLACE	Replace VLAN not TPID
BCM_VLAN_PORT_SERVICE_VLAN_DELETE	Delete VLAN Tag
BCM_VLAN_PORT_SERVICE_VLAN_TPID_REPLACE	Replace VLAN and TPID
BCM_VLAN_PORT_SERVICE_VLAN_PRI_TPID_REPLACE	Replace VLAN, VLAN-PRI, VLAN-CFI and TPID
BCM_VLAN_PORT_SERVICE_VLAN_PRI_REPLACE	Replace VLAN, VLAN-PRI and VLAN-CFI
BCM_VLAN_PORT_SERVICE_PRI_REPLACE	Replace VLAN-PRI and VLAN-CFI
BCM_VLAN_PORT_SERVICE_TPID_REPLACE	Replace TPID only

bcm_vlan_vpn_config_t_init

Initialize VPN config structure.

Syntax

```
#include <bcm/vlan.h>
void bcm_vlan_vpn_config_t_init(bcm_vlan_vpn_config_t *info);
```

Parameters

`info`	(IN) Pointer to the struct to be initialized

Description

Initialize VPN config structure.

Returns

None.

bcm_vlan_vpn_create

Create or update VPN.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_vpn_create(int unit, bcm_vlan_vpn_config_t *info);
```

Parameters

`unit`	(IN) BCM device number
`info`	(IN) VPN info

Description

Create or update VPN.

Returns

```
BCM_E_XXX
```

bcm_vlan_vpn_destory
bcm_vlan_vpn_destory_all

Destroy VPN.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_vpn_destory(int unit, bcm_vpn_t vpn);
int bcm_vlan_vpn_destory_all(int unit);
```

Parameters

`unit`	(IN) BCM device number
`vpn`	(IN) VPN ID

Description

Destroy VPN.

Returns

```
BCM_E_XXX
```

bcm_vlan_vpn_get

Get VPN information.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_vpn_get(int unit, bcm_vpn_t vpn, bcm_vlan_vpn_config_t *info);
```

Parameters

`unit`	(IN) BCM device number
`vpn`	(IN) VPN ID
`info`	(OUT)VPN info

Description

Get VPN information.

Returns

```
BCM_E_XXX
```

bcm_vlan_vpn_traverse

Traverse all valid VPN entries and call the supplied callback routine.

Syntax

```
#include <bcm/vlan.h>
int bcm_vlan_vpn_traverse(int unit, bcm_vlan_vpn_traverse_cb cb, void
*user_date);
```

Parameters

`unit`	(IN) BCM device number
`cb`	(IN) User callback function, called once per VPN entry
`info`	(IN) Cookie

Description

Traverse all valid VPN entry and call the supplied callback routine. The callback function is defined as following:

```
    typedef int (*bcm_vlan_vpn_traverse_cb)(int unit, bcm_vlan_vpn_config_t
*info, void *user_data);
```

Returns

```
BCM_E_XXX
```

VXLAN MANAGEMENT

New VXLAN port flag is added in this release.

Table 54: New VXLAN port flag

Name	***Purpose***
`BCM_VXLAN_PORT_SHARE`	Create shared multicast VXLAN port which could be shared by multiple next hops

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 follows:

Table 55:

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.

NOTE

The below data is not meant to be a precise indication of quality but instead serves as a guideline for improvements release-over-release. Additionally, although some cells show 0% failures, this does not necessarily mean the feature is supported in the device - tests are run to validate the appropriate SDK support even for unsupported features on older devices to ensure graceful handling of all API.

Finally, some devices have fewer columns listed if they were introduced recently.

TOTAL TESTS

The below data represents the number of unique cases for each release.

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

Table 56:

	sdk-6.4.8	***sdk-6.4.7***	***sdk-6.4.6***
golden	153	154	154

Table 56:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
warmboot	712	580	392
auth	17	17	17
bfd	37	37	37
bhh	43	43	43
chip	9	9	9
cint	68	59	60
coe	568	543	543
cosq	511	480	456
custom	7	7	7
ea	108	108	108
eav	19	19	19
extender	45	45	33
fabric	7	7	7
failover	8	8	8
fcoe	37	37	37
field	1291	1210	971
higigproxy	129	129	129
infra	114	114	114
ipfix	17	17	17
ipmc	111	99	66
l2	325	325	253
l2gre	13	13	13
l3	460	453	351
l3.alpm	356	318	258
link	26	26	26
mim	19	19	19
mirror	171	171	154
misc	17	17	17
mpls	324	235	201
multicast	25	24	24
niv	58	58	41
oam	271	263	187
pkt	44	44	44
port	354	326	308
proxy	37	37	37
ptp	115	115	115
qos	13	12	12
rate	21	21	21
rtag7	42	36	36
rx	25	25	24
ser	93	93	69
stack	117	117	117
stat	326	241	216
stg	42	42	42
switch	194	193	189
time	29	16	16
tlvMsg	13	13	13
trill	40	40	40
trunk	206	199	199
tunnel	123	91	65
subport	31	31	30

Table 56:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
vlan	222	217	217
vxlan	155	155	152
wlan	17	17	17
Test Suite Total	8335	7725	6750

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 57:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.2 %	0.2 %	0.4 %
warmboot	0.2 %	0.1 %	0.5 %
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.bfd	0.0 %	0.0 %	0.2 %
bcm.bhh	0.0 %	0.0 %	0.5 %
bcm.chip	0.3 %	0.0 %	0.4 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.1 %	0.3 %	0.7 %
bcm.cosq	0.2 %	0.2 %	0.2 %
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 %	0.1 %
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	0.3 %	0.3 %	0.5 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.3 %	0.0 %	0.6 %
bcm.ipmc	0.2 %	0.0 %	0.0 %
bcm.l2	0.2 %	0.1 %	0.3 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.1 %	0.2 %	0.3 %
bcm.l3.alpm	0.2 %	0.1 %	0.0 %
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.3 %	0.2 %	0.5 %
bcm.misc	0.2 %	0.0 %	0.2 %
bcm.mpls	0.1 %	0.1 %	0.5 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.1 %	0.0 %	0.0 %

Table 57:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
bcm.oam	0.1 %	0.1 %	0.4 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.6 %	0.8 %
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	0.1 %	0.0 %	0.2 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.2 %	0.2 %	0.3 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.1 %	0.1 %	0.2 %
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	0.3 %	0.0 %	0.5 %
bcm.vlan	0.0 %	0.0 %	0.4 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.3 %	0.3 %	0.4 %

TRIDENT2

Table 58:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.1 %	0.0 %	0.2 %
bcm.auth	0.0 %	0.0 %	0.0 %
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 %	0.2 %	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	0.2 %	0.3 %	0.9 %
bcm.higigproxy	0.0 %	0.0 %	0.0 %
bcm.infra	0.0 %	0.0 %	0.0 %
bcm.ipfix	0.0 %	0.0 %	0.0 %

Table 58:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
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 %	0.3 %
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 %	0.5 %
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	0.3 %	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 %	1.4 %
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 %	0.1 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.2 %	0.2 %	0.3 %

TRIUMPH3

Table 59:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.1 %	0.1 %	0.0 %
warmboot	0.1 %	0.0 %	0.2 %
bcm.auth	0.0 %	0.0 %	0.0 %
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 59:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	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 %	1.0 %
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 %	0.8 %
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	0.1 %	0.4 %	0.4 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.2 %	0.2 %	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	0.4 %	0.4 %	0.4 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.1 %	0.3 %	1.2 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	1.5 %	1.5 %	1.6 %
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 %	0.0 %
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 %	0.3 %
bcm.stat	0.0 %	0.0 %	0.8 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.2 %	0.6 %
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	0.3 %	0.3 %	0.4 %

KATANA2

Table 60:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.1 %	0.1 %	0.1 %
warmboot	0.3 %	0.3 %	0.9 %
bcm.auth	0.0 %	0.0 %	0.0 %
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 %	0.3 %	0.4 %
bcm.cosq	0.3 %	0.3 %	0.3 %
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 %	0.2 %	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 %	0.0 %	0.0 %
bcm.l2	0.1 %	0.2 %	0.4 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.2 %	0.3 %
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	0.1 %	0.9 %	0.9 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.0 %	0.0 %	0.3 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.8 %	0.9 %	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 %	0.7 %	1.4 %
bcm.stack	0.0 %	0.0 %	0.0 %
bcm.stat	0.0 %	0.0 %	0.1 %
bcm.stg	0.0 %	0.0 %	0.0 %
bcm.switch	0.0 %	0.0 %	0.0 %
bcm.time	0.0 %	0.0 %	0.0 %

Table 60:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
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 %

GREYHOUND

Table 61:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.3 %	0.0 %	0.0 %
warmboot	0.1 %	0.0 %	0.2 %
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cint	0.0 %	0.0 %	0.0 %
bcm.coe	0.0 %	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	0.2 %	0.2 %	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 %	0.0 %	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	0.0 %	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 %	0.0 %	0.0 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.6 %	0.5 %	0.5 %
bcm.proxy	0.0 %	0.0 %	0.0 %

Table 61:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
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 %	0.5 %
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 %	0.1 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.0 %	0.0 %	0.0 %
bcm.vlan	0.0 %	0.0 %	0.7 %
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 62:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.1 %	0.0 %	0.0 %
warmboot	0.1 %	0.3 %	N/A
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.bfd	0.0 %	0.0 %	N/A
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	0.2 %	0.1 %	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	0.5 %	0.6 %	2.4 %
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	0.3 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %

Table 62:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
bcm.l3	0.0 %	0.3 %	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 %	0.0 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	1.0 %
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 %	0.9 %
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	0.2 %	0.0 %	0.2 %
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	0.1 %	0.0 %	0.7 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.3 %	0.3 %	0.5 %

TRIDENT2+

Table 63:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.0 %	0.0 %	0.0 %
warmboot	0.1 %	0.0 %	1.4 %
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.cosq	0.0 %	0.1 %	0.2 %
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 %

Table 63:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
bcm.failover	0.0 %	0.0 %	0.0 %
bcm.fcoe	0.0 %	0.0 %	0.0 %
bcm.field	0.1 %	0.5 %	0.9 %
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 %	0.0 %	0.0 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.0 %	0.1 %	0.9 %
bcm.l3.alpm	0.0 %	0.0 %	N/A
bcm.link	0.0 %	0.0 %	0.0 %
bcm.mim	0.0 %	0.0 %	0.0 %
bcm.mirror	0.0 %	0.6 %	0.6 %
bcm.misc	0.0 %	0.0 %	0.0 %
bcm.mpls	0.0 %	0.0 %	1.0 %
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	0.3 %	0.6 %	1.3 %
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 %	0.7 %
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.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 %	0.9 %	0.9 %
bcm.vxlan	0.0 %	0.0 %	0.0 %
bcm.wlan	0.0 %	0.0 %	0.0 %
Test Suite Total	0.3 %	0.3 %	0.4 %

SABER2

Table 64:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
golden	0.4 %	0.4 %	0.4 %
warmboot	0.5 %	0.5 %	0.6 %

Table 64:

	sdk-6.4.8	*sdk-6.4.7*	*sdk-6.4.6*
bcm.auth	0.0 %	0.0 %	0.0 %
bcm.chip	0.0 %	0.0 %	0.0 %
bcm.coe	1.1 %	1.1 %	1.4 %
bcm.cosq	0.0 %	0.0 %	0.7 %
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	0.3 %	0.2 %	0.4 %
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	0.1 %	0.2 %	0.2 %
bcm.l2gre	0.0 %	0.0 %	0.0 %
bcm.l3	0.2 %	0.6 %	1.1 %
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	0.3 %	0.4 %	1.2 %
bcm.multicast	0.0 %	0.0 %	0.0 %
bcm.niv	0.0 %	0.0 %	0.0 %
bcm.oam	0.3 %	0.4 %	0.8 %
bcm.pkt	0.0 %	0.0 %	0.0 %
bcm.port	0.5 %	0.5 %	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 %	0.7 %
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	0.2 %	0.5 %	0.5 %
bcm.time	0.0 %	0.0 %	0.0 %
bcm.trill	0.0 %	0.0 %	0.0 %
bcm.trunk	0.5 %	0.0 %	0.0 %
bcm.tunnel	0.0 %	0.0 %	0.0 %
bcm.subport	0.8 %	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	0.3 %	0.3 %	0.5 %

PHY TEST RESULTS

In SDK-6.4.8 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. The DNE column represents tests that have not been executed in the release. Please note that this is a starting point for this activity and we are continually working to improve our results in future releases.

SQA EXTERNAL PHY

Table 65: PHY Suite

Switch Device	*External Phy Device*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56639_A0	BCM8728/8747_10G	92	56	35	1	1%
56846_A0	Warpcore_40G/20G/10G	92	54	36	2	2%
56640_B0	Enzo_100G	92	22	58	12	13%
56850_A2	Koi_10G	92	42	44	6	7%
56960_A0	40GFuria_40GSesto	92	38	50	4	4%
56960_A0	100GFuria	92	39	50	3	3%
56960_A0	10GSesto	92	36	49	7	8%
56960_A0	100GFuria_40GMuxSesto	92	22	67	3	3%
56960_A0	10GFuria_10GSesto	92	20	66	6	7%
56860_A0	Koi_10G	92	24	65	3	3%
56860_A0	40GQuadra28	92	32	49	11	12%
56860_A0	100GSesto	92	25	63	4	4%

Table 66: PHY1588 Suite

Switch Device	*External Phy Device*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56142_A0	BCM54295/54296_1G	17	17	0	0	0%

INTEROP EXTERNAL PHY

Table 67: P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56860_A1	XE_QD28_10G	72	60	12	0	0%
56860_A1	XE_QD28_40G	72	60	12	0	0%
56860_A1	CE_SESTO	52	31	14	7	13%

Table 68: Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56860_A1	XE_QD28_10G	11	9	0	2	18%
56860_A1	XE_QD28_40G	11	10	0	1	9%
56860_A1	CE_SESTO	11	8	0	3	27%

INTEROP INTERNAL PHY

Table 69: P2P Suite

Switch Device	*Port Macro*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56860_A0	XE_96X10_8X40	54	42	12	0	0%
56860_A0	XE_32X40	90	66	24	0	0%
56860_A0	CE_8X100_244	67	42	24	1	1%
56860_A0	CE_8X100_343_IEEE	62	36	24	2	3%

Table 70: Loopback Suite

Switch Device	*Port Macro*	*Total Tests*	*Pass*	*DNE*	*Fail*	*% Fail*
56860_A0	XE_96X10_8X40	11	10	0	1	9%
56860_A0	XE_32X40	22	16	0	6	27%
56860_A0	CE_8X100_244	11	8	2	1	9%
56860_A0	CE_8X100_343_IEEE	11	9	2	0	0%

STATIC CODE ANALYSIS

In early 2015 we upgraded our static analysis code tool to a version with many new checkers and have been steadily working down the backlog of issues. Below shows the progress by recent releases this year:

Table 71:

	Initial Reported Issues	*New baseline as of 1/31/15*	*Open Issues SDK 6.4.5*	*Open Issues SDK 6.4.6*	*Open Issues SDK 6.4.7*	*Open Issues SDK 6.4.8*
DNX	664	450	295	168	113	108
XGS	271	500	229	116	9	11
SBX	600	650	156	77	23	23
SerDes	147	100	59	49	58	23

Table 71:

	Initial Reported Issues	New baseline as of 1/31/15	Open Issues SDK 6.4.5	Open Issues SDK 6.4.6	Open Issues SDK 6.4.7	Open Issues SDK 6.4.8
Common	2827	400	155	133	29	24
Total	4509	2100	894	543	232	189

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.

Table 72: Resolved Service Impacting Defects

Number	Chips	Affected versions	Errata Synopsis	Details
SDK-82005	AllChips	6.5.1, 6.4.6	Possibility of SDK Tx API transmission hang when using KNET under multi-core CPU SMP circumstances	All customers and all devices running Switch SDK within the sdk-6.4.6 using KNET have a risk of encountering SDK Tx API transmission hang. The issue demonstrates itself readily in systems using BCM KNET and running SDK in multi-core CPU SMP environment. In BCM KNET implementation, a flag is utilized to indicate whether current packet streams are belonging to virtual network interfaces or BCM TX API. This flag was not protected properly in multi-core CPU SMP environment. There existed a race condition in which packet transmission from virtual network interface would conflict with BCM Tx API. SDK Tx API transmission will hang once the conflict happens.
SDK-78906	AllChips	6.5.0, 6.4.6, 6.4.5, 6.4.4, 6.4.3	Possibility of table corruption when running in multi-threaded environment	All customers and all devices running Switch SDK within the "Affected Versions" have a risk of encountering table corruption. The corruption occurs inside switch memory, not the host CPU memory, and typically happens in memories with many updates, such as L2 and L3. The issue demonstrates itself readily in systems running SDK in multi-threated environment. The issue was caused by an improper use of a "static" variable inside one of the `soc_mem_write()` helper functions.
SDK-75905	`56850_A2`	6.4.5	Possibility of internal software objects corruption after a great deal of routes adding/ deleting.	When scaling LPM is enabled `(scaling_lpm_enable=1)`, any customer who is using any SDK release through SDK-6.4.6 has a risk of random LPM internal software management corruption and further infinite loop. The issue demonstrates itself after a great deal of ipv4/ipv6 routes adding/deleting. The devices which are a subject of this vulnerability include BCM56850, BCM56640 and subsequent devices which have this scaling LPM feature.

Section 9: Resolved Issues for 6.4.8

The following issues are resolved in version 6.4.8 of the SDK.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-7988		`88020_A0`	Microcode version 1.2 (g2p2) is no longer supported for BCM88020 devices
SDK-49439		`56850_A1`	Add Qos warmboot support for BCM56850.
SDK-56616 SDK-53918		`56640_B0`	In previous release, DMVOQ feature hadn't been supported on BCM56640. In this release,DMVOQ feature has been supported on BCM56640.
SDK-59702	807246	`88670_A0`	Added LIF-Based-Counting suppport for BCM88670. When the counter-pointer if a LIF ID on the packet, we have the following problem: 1. There is a local-to-global-LIF-mapping. This mapping cannot directly controlled. 2. For LIF-based-counting is the counter pointer is the local-LIF. To define the range of counted LIF for ingress/egress we define two LIF-counting-profiles, for each profile we define: 1. A source and a command-ID to be counted. 2. A range of Local-LIFs to count from. When creating a LIF a field with indicating LIF-counting-profile can have one value from no-counting-profile, counting-profile0|1for egress counting-profile2|3 for ingress. That gives us control from what range a local LIF will be allocated from (the range of the profile), for each counting source (command-ID per ingress/egress). If two, or more, LIFs from the same counting source (command-ID per ingress/egress) appear on a packet only one can be send to the counter engine, For each counting source (command-ID per ingress/egress) and each LIF-stack-level, we define a priority. The LIF in the LIF-stack-level with the highest priority that is in a Usage can be found in the user manual.
SDK-60084		`88670_A0`	1. full support of local policers (per-port Eth. policer) on BCM88670. usage is identical to the BCM88660 usage: `bcm_rate_bandwidth_set/get` 2. aggregate policers - In the existing API, core indication is missing. the policers allocation is duplicated for both cores. So, for example: In case of creating a policer with 100 Kb limit, 2 policers of 100Kb limit will allocated, one of each core. New API with core indication will be added in further releases.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-62986		56526_B0 56845_B0	bcmFieldQualifyInterfaceClassVPort qualifier support is added for BCM56520/BCM56630, BCM56330 and BCM56840/BCM56842 devices
SDK-63042		88660_A0 88670_A0	OAM: Fixed bugs in OAMP server implementation.
SDK-63558 SDK-63650		88670_A0	In the previous release, BCM88670 can't support OAM's Loopback and TST function base on SAT.In this release,user can use the API "bcm_ oam_loopback_add/get/ delete(),bcm _oam_tst_rx_ add/get/ delete(),bcm _oam_tst_rx_ add/get/ delete()" to use Loopback and TST.
SDK-64421		88650_A0 88650_B0 88650_B1 88650ACP_A0 88660_A0 88670_A0	Cosq: Diagnose commands are added to display flow control related information per NIF port or Calendar port for TX and RX directions. Usage of diagnose commands can be gotten by typing "FC ?" on BCM Shell.
SDK-66379		56457_B0	In previous releases external memory related registers were incorrectly addressed for BCM56457 SKU for performing sanity tests. This was causing segmentation fault. This issue has been fixed.
SDK-68878		56540_A0 56540_B0	Issue ==== For qualifier get API's which did not have mask signature, data was returned without checking if the mask is non-zero. Fix == Added check to return error when hw mask is zero and return data only when hw mask is non-zero.
SDK-69122 SDK-73367		88670_A0	KBP support added
SDK-69300		56640_A0 56640_A1 56640_B0 All	In previous releases, VPLS packets encapsulated with IPv4 payload always flooded to access ports on PE. In this release this bug has been resolved by adding a new rule in ESM_PKT_TYPE_IDm.
SDK-69328		88650_B1 88660_A0 88670_A0	bcm_rx_show - For ARAD and Jericho device shows more information:Channel info and Queue info.
SDK-69361		88670_A0	ECC 1 bit correction for BCM88670 and above. include wide memory. does not include XOR memories. the fix for XOR memories must handle by shadowing.
SDK-70121		88670_A0	Support OAM counting by priority. Applies only to SLM/R
SDK-70126		88670_A0	In the previous release, BFD IPV6 single hop uses FLP program. In this release, it uses OAM classifier replacing FLP.
SDK-70588		88202_A0	Added code to fix select the source of the ref_clk and also program the CXX refin / refout appropriately..

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-70737		All	Action ColorIndependent indicates whether the Redirection/L3SwChangeL2Fields actions are color independent or must be applied only to Green Packets. 0x0: Apply these two actions only to Green Packets 0x1: Apply these two actions to all packets. This action is not tracked as part of the Entrys actions list and is used only through Entry Flags. When we have any Dependent actions like bcmFieldActionSrcMacNew, these internally set the entry flag to 1(All packets). Hence, configuration of ColorIndependent to 0, is overridden by the dependent actions setting internally the flags. This should not be the behaviour when user explicitly adds a ColorIndependent to 0 => Only green Packets need to be modified. Fixed the same by setting ColorIndependent flag in the entry by default and using the bcmFieldActionColorIndpendent to set to 0(only Green Packets). By this, all entries in IFP will have the `GREEN_TO_PID` set to 1, meaning it applies for all red/yellow/green packets. When action colorindependent is set to 0 specifically on an entry, we set the `GREEN_TO_PID` field to 0 in the Policy table making the entry specific to green packets.
SDK-71762 SDK-83589		`88670_A0`	Added LEDs SW driver support for 88670 device. Added LEDs example program that activates Negev chassis front panel LEDs ($SDK/tools/led/example/sdk88670.asm). Notes: - 88670 LEDs does not work in QSGMII mode (SDK-83946). - LEDs of four 10G ports of Negev chassis front panel are not activated yet (only 40G, 100G Negev chassis ports LEDs are currently working) (SDK-83948) - To Activate LEDs, need to enable linkscan (linkscan is disabled by default). - Negev chassis front panel LEDs activation will require FPGA version 0x10 or above (to check FPGA version of a Negev chassis run `'cpu_i2c` read 0x44 0x40 0x0' from BCM shell)
SDK-72576		`88670_A0`	For accelerated BFD endpoints of type MPLS-TP, on the BFD frames injected from the OAMP, the MPLS label incorrectly had the BOS bit set. This was incorrect because the GAL label was the BOS label.
SDK-72968		`56860_A0 88950_a0`	Require a way diable the DFE taps. Commited fix enables freeze/unfreeze of DFE taps using with `bcm_port_phy_control_set.`
SDK-73718		`88660_A0`	The previous `bcm_l2_learn_limit_set/get()` didn't support the channelized port configuration. Currently, this issue was fixed which`bcm_l2_learn_limit_set/get()` support the chnanelized port configuration.
SDK-73942		`56960_A0`	Problem: Not able to create pipe mode group using bshell commands. Solution: Added the support for creating pipe mode groups using bshell commands.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-74234		All	In previous releases, the bookkeeping data of all the modules would be committed into persistent storage. To improve the performance, in this release only the changed part of scache would be updated into the persistent storage.
SDK-74519		`56440_A0` `56440_A1` `56440_B0`	In the earlier version, support to enable inner tag color selection for 56440 & 56450 with `BCM_COLOR_INNER_CFI` was absent and it was returning support unavailable. In the current version, the support is added.
SDK-75296		`56850_A2` `56860_A0`	In previous releases, the SDK would not properly handle the scenario of replacing labeled egress objects with `BCM_MPLS_LABEL_INVALID.` In this release, the SDK properly replaces labeled egress objects with `BCM_MPLS_LABEL_INVALID.`
SDK-75342		`88670_A0`	12.5G XFI speed is added for tscf
SDK-75602		`88202_A0` All	Add PortMod statistics API support
SDK-75804		`88670_A0`	`bcm_switch_control_set(0,bcmSwitchL3HeaderErrToCpu,` 1) failed in BCM88670 due to a field name change. It is now possible to use the function.
SDK-75842		`88660_A0` `88670_A0`	Added support for a flexible 1-pass reflector using field processor. To use this, the following soc property must be set: `custom_feature_rfc2544_reflector_mac_swap_1pass=1` This loads an egress program that performs a DA/SA swap. It's the responsibility of the user to redirect packets back to the source port and select this egress program for the packet by stamping a specific code on the system headers. An example of such settings can be found in `cint_field_reflector_1pass.c`
SDK-76129		`88670_A0`	new API, relevant only for customers who need this new API. should allow to filter certain criteria for the CRPS to count/not count. documentation is currently available only as an html reference guide to BCM API. UM or AN for the CRPS will be updated with the information in another JIRA.
SDK-76259		`88670_A0`	this is not a critical fix, just a formality since the bits which are not relevant are masked anyways.
SDK-76329		`88650_B1` `88660_A0` `88670_A0`	MPLS PORT: `cint_vswitch_cross_connect_p2p_directional.c` was added as example to P2P directional cross connect configuration.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-76334		56450_B0 56450_B1	Getting a soc property "mpls_oam_num_sessions" from user and this soc property will indicate whether the user wants to use old mechanism of allocating 8 HW entries per BHH endpoint or use new mechanism of 1 HW entry per BHH endpoint. This soc property will also provide the number of BHH sessions that can run totally on FIRMWARE, HOST CPU and any other customer OLP. Note that this variable cannot be used in configs across warm upgrade from older versions. It needs to be used from newer versions only.
SDK-76440		56460_A0	In the previous release, there was no way to get the sequence number of the last received packet. In this release, this issue has been addressed by adding new API (last_packet_seq_number) to get sequence number of the last received packet.
SDK-76461		88670_A0	1. ARM KAPS FW location: $sdk/src/soc/kbp/r5/kaps_arm_fw.hex 2. ARM KAPS FW load by bcm shell: kbp kaps_arm file=kaps_arm_fw.hex 3. Enable the ARM KAPS thread: kbp kaps_arm_thread enable_dma_thread=1 print_status=1
SDK-76525		88650_A0	In the previous release, there will be failure when running eyescan on the port in which there is lane number more than 9. In this release, there will be failure only if the lane number is greater than the actual number of lanes.
SDK-76704		88660_A0 88670_A0 88670_B0	BFD single hop packets will be demultiplexed through the the LEM based on the Your-Discriminator. The LEMs payload will include the OAM-ID as well as the expected remote Detect Multiplier. A PMF program should then be written that does the following: 1. Extract the OAM-ID from the LEM payload and set it on the trap-code-qualifier. 2. Extract the remote detect multiplier from the packet and from the LEM payload. If the values are not equal the packet should be snooped to the CPU. The LEM payload will be structured in the following way: {0 (2b) ,remote_detect_mult[0:3] (4b) ,OAM-ID (13) ,remote_detect_mult[4:7] (4b) , 0 (1b) ,destination (19b)} Note that this PMF program has not yet been written. The expected remote detect multiplier can be set in the API bcm_bfd_endpoint_create() in the field remote_detect_mult. This behavior may be enabled by setting the soc property bfd_ipv4_single_hop_extended=2
SDK-76882		88670_A0	For BCM88670, provide additional tunnel termination mode: DIP+SIP+VRF, in addition to already existing modes: dip, sip and DIP+SIP. Bidir feature is not compatible with this mode. See cint example: cint_ip_tunnel_term.c

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-77103		56224_A0 56224_B0	ISSUE: L2 table entry move during dual-hash environment was not working. It was timing out. ROOTCAUSE: Generic table operations for insert/delete was used in two places causing the issue. SOLUTION: As BCM56224 does not support generic table operations, device specific functions are used in place of generic table operation functions.
SDK-77129		88670_A0	Added a support to dynamically add or remove interlaken port.
SDK-77342		All	Fixed build errors due to C++ reserved word "template" in *.h files.
SDK-77446		88670_A0	Fixed TR 41 failure when CPU ports exist.
SDK-77543		88670_A0	FID is 15 bits in BCM88670 and the parsing used 14 bits (like BCM88650). Now the parsing is using the correct number of bits for each device
SDK-77594		88670_A0	MIB counters (thread, APIs, and show c) fixed to support QSGMII port.
SDK-77595		88670_A0	"diag nif" fixed to display QSGMII serdes rate correcttly.
SDK-77604		88660_A0 88670_A0 All	Add new port header type for outbound-mirroring - MIRROR_RAW. This header type will be used as RCY header type on outbound mirroring instead of ETH to bypass any unwanted ETH filters for outbound mirror packets
SDK-77724		88670_A0	init sem structure to zero to prevent deadlock on some cpu's
SDK-77840		56960_A0	This release fixed a lag failover issue when one of the trunk port links down
SDK-77970		56960_A0	Problem: The SDK does not allow creation of two UDF objects on different pipes with single pkt format object. In pipeLocalMode, bcm_udf_pkt_format_info_t associates with single pipe only, thus the sw object cannot associate with more than two pipes. However, in previous release, it did not return proper error code when it attempts to create two udf objects on different pipes with single pkt format object. Solution: Return BCM_E_PARAM when SDK attempts to create two UDF objects on different pipes with single pkt format object.
SDK-78018		88670_A0	crush was caused by invalid configuration - there were insufficient dbuffers for configuration. added check for different dbuff categories before writing values to regs, and exit with informative errors in-case of violations
SDK-78044		56850_A0 56850_A1 56850_A2	When there is no link, command handler squelch TX. TSC TX squelch is cleaned up when the external PHY links up.
SDK-78121		All	PTP signaling arbiter registration function has been added to API.
SDK-78135		All	In previous releases, the exit code of bcm.user.proxy was likely to be abnormal, and it could lead to system hang. Now the sequence of system exit is redesigned, and this issue is fixed.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-78166		88660_A0 88670_A0	MPLS PORT: _bcm_dpp_l2_fec_fill_gport does not return failover_id for secondary fec.
SDK-78295		88670_A0	Native-outrif value added to diag pp encap
SDK-78428		56260_A0	Problem: BHH MEP creation fails on BCM56260 device. Solution: Make the SDK software OAM MEP look up table size same as Hardware MEP look up table.
SDK-78462		56850_A0 56850_A1 56850_A2 All	In previous releases, the API bcm_l3_intf_create could not update the NAT realm id to the hardware destination nat id. This has been addressed by proposing a new flag BCM_L3_SRC_DST_NAT_REALM_ID in the API handle this case.
SDK-78565		56960_A0	In previous releases, VC swap label hash info was not correctly recovered during warmboot, caused failure of egress object deletion. In this release, the issue has been fixed by adjusting hash index to recovery correct VC swap label hash info.
SDK-78675		56440_A0 56440_A1 56440_B0 All	Issue ==== During entry move incorrect EFP HW counter fields were being used for writing to new entry. Due to this an assert was observed. Fix == Added fix to update appropriate HW counter fields based entry stage.
SDK-78806		56960_A0	An attempt to perform a Warmboot of SDK configured to operate a BCM56960 device in Low Latency mode ("switch_bypass_mode=2"), would result in a segmentation fault in bcm_port_init(). This has been resolved in the latest release.
SDK-78808		88650_A0 88660_A0 88670_A0	In Advanced VLAN Translation mode, the Egress packet structure is classified according the incoming TPID values by calling bcm_port_tpid_class_set() with the flag BCM_PORT_TPID_CLASS_EGRESS_ONLY. This configuration was applied only to the Egress classification when configuring discard using the BCM_PORT_TPID_CLASS_DISCARD flag, but classification to a tag-format was done on both the Ingress and the Egress. An Egress or Ingress operation now classifies only on the required side.
SDK-78823		56643_A1	In earlier releases for A0&A1 chip, when L1 node had more than one child(L2 node), change the child(L2 node) schedule mode between WRR and SP,it was not supported. In this release,it has been fixed for A0&A1 chip.
SDK-78827 SDK-83586		All	This JIRA provides support for synchronous clock recovery from port on BCM56260.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-78847		`56260_A0` 56460	Issue: Passive demultiplexing was being done for Maintenance intermediate points. This should not be done. Fix: The flag `BCM_OAM_ENDPOINT_INTERMEDIATE` is handled now and passive demultiplexing is avoided for Maintenance intermediate points.
SDK-78852		`56842_A0 56846_A1`	In previous releases, `bcm_cosq_cpu_cosq_enable_set` might not work in some cases. This has been addressed by synchronizing the hardware status in time to prevent potential API malfunction.
SDK-78886 SDK-80144		`56850_A0 56850_A1 56850_A2`	In previous releases, the API `bcm_l3_egress_ecmp_create` would return `BCM_E_FULL` when the API bcmSwitchObjectEcmpCurrent indicated there was resource left, which was incorrect. This has been addressed.
SDK-78941		`56260_A0` 56460	Problem: 1. The BHH, MPLS LM/DM uController was sending invalid packet when configured to send untagged packet. 2. There was no support for double tagged packet in BHH & MPLS LM/DM Solution: 1. Fixed the packet construction for untagged packet. 2. Added support for double tagged BHH & MPLS LM/DM
SDK-78954		`56860_A0`	In previous releases, BCM56860 did not support RioT feature for MIM. It has been added in this release. This feature includes the followings: 1. VLAN membership check and STG checks. 2. Overlay next hop support on MiM. 3. Access shared VP for MiM.
SDK-78956		`88670_A0 88670_B0`	In the previous release, Higig port index was supported only smaller than 32. In this release, this issue has been addressed by extending PTC from 5 bits to 8 bits and support index from 0 to 255.
SDK-78975		`56640_A0`	In previous releases, when a `'bcm_trill_multicast_entry_add'` was attempted and the L2 table was full, it was not able to insert Trill entries. In this release, a dynamic replacement mechanism has been added for `bcm_trill_multicast_entry_add.`
SDK-79062		`56260_A0` 56460	Added support to configure BHH endpoint on trunk port member. Application has to provide the trunk port member index on which BHH should be configured in `"trunk_index"` member of `bcm_oam_endpoint_info_t.`
SDK-79072		`56450_B0`	SER handler for RMEP table was accessing illegal memory which caused kernel crash. Fixed the SER handler to access legitimate memory only.
SDK-79116		`56840_A0`	In previous releases, when port disable/enable was toggled quickly, the linkdown interrupt was sometimes missed. In this release this has been fixed.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-79127		56260_A0 56260_B0 56262_A0 56262_B0 56265_A0 56265_B0 56267_A0 56267_B0 56268_A0 56268_B0 56460 56460_A0 56460_B0 56461_A0 56461_B0 56463_A0 56463_B0 56465_A0 56465_B0 56468_A0 56468_B0	Issue : The EFIFO configurations were not supporting 2.5G speed and the bandwidth configuration for the new ports created after hotswap were not configured. Solution: The EFIFO configuration during initialization and post hotswap is fixed. The bandwidth configuration for new ports created after hotswap is added.
SDK-79140		56450_B0	Provided SOC-Property to configure the polling interval for debug buffers
SDK-79174		56960_A0	Operation mode was set after UDF software control is initialized. If mode was changed from local to global, this can cause a crash while UDF software control is initialized. A fix has been provided in this release so that operation mode is set before initialization.
SDK-79217		88660_A0	MPLS PORT: replacing protected PWE outlif with MPLS tunnel label on same entry was resulting in deletion of the MPLS tunnel.
SDK-79320		88660_A0	When traverse exits with error the flush DB is cleaned the pause is disabled and `current_index` is set back to 0. No leftovers from the previous flush are left
SDK-79339		88660_A0	Changed FEC addition sequence from Super->Working->Protection to Working->Protection->Super to prevent packet loss when updating an existing FEC.
SDK-79374		56450_B1	Customer Problem Statement : In cold boot, modify the `cos_mapping` from default for ex: ((cos0,0)(cos1,0),(cos2,0).(cos7,0)) Now do warmboot and try to modify `cos_mapping` to ((cos0,0)(cos1,1) (cos2,2) (cos3,3) (cos4,4) (cos5,5) (cos6,5) (cos7,7)) and observed `E_RESOURCE` Root cause: Default profile is getting initialized in the warmboot also, as a result there is inconsistency in the software profile and the hardware data, As a result profiles are not matching and returning `E_RESOURCE` Fix Description: 1. Disabling the initializing the default profile in the warmboot 2. Disabled the subport based profile, if no subports are present 3. Profile reference count is incremented for all ports including flex ports
SDK-79398		56260_B0	Issue: The device was failing during ddr phy initialization when external memory was not present. Solution: The DDR PHY initialization is skipped when external memory is not present.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-79407		56750_A0	In previous releases, the runt threshold of HG ports was incorrectly configured during initialization stage on BCM56750, which might cause short packets drop. This has been fixed by configuring the runt threshold appropriately in HiGig modes. For IEEE mode, the value is 64. For HiGig it is 72 and for HiGig2 it is 76.
SDK-79408		56460	In earlier releases, the VP value was not set when NETWORK PORT was configured with BCM_MPLS_PORT_WITH_ID. This is fixed.
SDK-79436		88660_A0	Add support for single fap mode for BCM88541 + BCM88660.
SDK-79437		56260_A0 56460_A0	Code has been added to check if label is port independent or not and use port also to search in L3 tunnel table if label is port dependent.
SDK-79452		56850_A0	Issue ==== Entry install when only action was changed was causing entry to be removed and then updated with the new value. This was causing traffic hit Fix === Added changes to not remove entry when only action is modified.
SDK-79462		56440_A0 56450_A0	In P2MP application vlan translation for second pass processing was not working in BUD LSR scenario. This is fixed for bcm56450,bcm56440 devices.
SDK-79482		All	In previous releases, when using API bcm_port_ifg_set() to set the new IPG with speed more than 10G, it didn't take effect. This issue has been fixed.
SDK-79508		56260_A0	Issue: The device was failing during ddr phy initialization when external memory was not present. Solution: The DDR PHY initialization is skipped when external memory is not present.
SDK-79520		56260_B0 56460	Issue: 8 indexes are allocated in MA_INDEX table for MPLS OAM endpoints (BHH and MPLS LM/DM) even though only one MDL (7) is supported for both of them. Fix: Get a soc property mpls_oam_num_sessions from user which indicates that only 1 index should be allocated per MPLS OAM session and also the number of MPLS OAM sessions for which MA_INDEX table indexes need to be reserved.
SDK-79568		All	Current Linkscan code had a missing state which is rarely encountered. Added code to handle the state and issue port_update command when the transition is encountered.
SDK-79575		56960_B0 56960_A0	This release addresses an issue with the port command where port abilities do not display support for 11G, 27G, 50G, 53G, 100G and 106G where applicable.
SDK-79578		88670_A0	solve memory leaks in the driver to prevent driver crash when looping tr 141 more than 100 times
SDK-79579		88670_B0	Added support for serdes_if_type (SoC property).

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-79611		56960_A0	In previous release, the document did not show the restricted behavior in AGM module. User should know it and use existing APIs in a proper manner when changing AGM attached aggregation group. Now this restriction and proper API sequence has been updated in document.
SDK-79614		56860_A0	Fixed a bug for cr4 and sr4 interface set for Td2+
SDK-79619		56640_A0	In some cases, for the following tables, there would be HASH calculation mismatch between hardware insert and software insert. This has been fixed. VLAN_XLATE_2 L2_ENTRY_2 L3_ENTRY_2 L3_ENTRY_4 MPLS_ENTRY_2
SDK-79688		88670_A0	Fixed APIs bcm_port_detach and bcm_port_probe to allow dynamically power on and off of fabric SerDes.
SDK-79709		88650_A0 88650_B0 88650_B1 88660_A0	cosq: The Arad support sending and receiving flow control indications using the ILKN In-Band.The calendar size is a configurable value which should support lengths of 1-256 entries. Due to HW limitation, calendar length 256 is not supported for ILKN in-band TX interface. In this fix, error indication is added if inband calendar size of 256 is used for tx direction on Arad.
SDK-79726		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-79732		54220_A0 56450_A0 56450_B0 56450_B1 54220_B0	Perform the Global Reset to BCM542XX PHY devices only on the first initialization.
SDK-79742		88670_A0	In BCM88670, for IPv4 Compatible MC packet, enabled <FID, G,S> search in TCAM for Bridge-IPv4MC under the soc property of ipmc_l2_ssm_mode.
SDK-79762		88650_A0 88650_B0 88650_B1 88660_A0	Simulation build (PCID) for BCM88650, BCM88660 target while BCM88640 target is not included in Makefile.

SDK-79781		56260_B0 56460	The following set/Get APIs have been added along-with the required structs and enums.
			int bcm_oam_opcode_group_set(int unit, bcm_oam_protocol_type_t protocol, bcm_oam_opcodes_t opcodes, uint8 opcode_group);
			int bcm_oam_opcode_group_get(int unit, bcm_oam_protocol_type_t protocol, bcm_oam_opcode_t opcode, uint8 *opcode_group);

SDK-79835		All	In earlier releases, if the API bcm_ipmc_add() was used to update an existing entry of L3_ENTRY_IPV4/6_MULTICAST tables, the associated flex counter data would be lost. In this release, the issue has been fixed.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-79836		`88670_A0`	Added LIF-Based-Counting suppport for BCM88670. When the counter-pointer if a LIF ID on the packet, we have the following problem: 1. There is a local-to-global-LIF-mapping. This mapping cannot directly controlled. 2. For LIF-based-counting is the counter pointer is the local-LIF. To define the range of counted LIF for ingress/egress we define two LIF-counting-profiles, for each profile we define: 1. A source and a command-ID to be counted. 2. A range of Local-LIFs to count from. When creating a LIF a field with indicating LIF-counting-profile can have one value from no-counting-profile, counting-profile0|1for egress counting-profile2|3 for ingress. That gives us control from what range a local LIF will be allocated from (the range of the profile), for each counting source (command-ID per ingress/egress). If two, or more, LIFs from the same counting source (command-ID per ingress/egress) appear on a packet only one can be send to the counter engine, For each counting source (command-ID per ingress/egress) and each LIF-stack-level, we define a priority. The LIF in the LIF-stack-level with the highest priority that is in a Usage can be found in the user manual.
SDK-79860		`56450_B1`	Added implementation ofWarmboot support for Broadsync for BCM56450.
SDK-79875		`56850_A0 56850_A1 56850_A2`	In previous releases, calling `bcm_vlan_control_port_set()` API with bcmVlanPortUseOuterPri for mpls/MIM port returned success though `TRUST_OUTER_DOT1P` was invalid for TD/TD2. This has been addressed by returning correct error code `BCM_E_PORT` to notify customers.
SDK-79940		`56260_A0 56460_A0`	Issue: The HW field which represents the queue number to be taken by outgoing CCMs from CCM engine was filled with Port based cosq base + cos index. It did not take care of special queuing structures such as vlan based queueing etc. Fix: User needs to input GPORT values of such special queues created (such as vlan based queues) using `tx_gport` variable in `bcm_oam_endpoint_info_t` structure. This `tx_gport` variable is handled and appropriate queue base indexes are resolved and HW programmed appropriately.
SDK-79954		`56960_A0`	Problem: When egress port is in cut-thru mode, IPMC replication on multiple vlans doesn't happen. Packets with a single logical replication per port are eligible for cut-through consideration. If the IPMC group has more than 1 switch copy per egress port, `DO_NOT_CUT_THROUGH` bit in `L3_IPMC` table must be set for the ipmc group.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80014		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-80018 SDK-80166 SDK-48472		56850_A0	In previous releases, there were no APIs to enable port based sflow and set sample rate separately. The result was that when setting sample rate for fp based sflow, the port based sflow had to be enabled at the same time. To support this, new separate port controls (bcmPortControlSampleIngressEnable , bcmPortControlSampleEgressEnable, bcmPortControlSampleIngressRate, bcmPortControlSampleEgressRate) have been added to enable/disable port based sflow and to set sample rate.
SDK-80051		56260_A0 56460	Issue: bcm_oam_endpoint_create API was taking OAM LSP endpoint via endpoint_info.intf_id variable. This was insufficient in case when there were multiple Pseudowires that needed to be associated with a single LSP endpoint. Fix: A new API bcm_oam_endpoint_gport_egress_attach is introduced with endpoint and gport information. Endpoint needs to be passed with already created LSP endpoint ID and Gport needs to be passed with the mpls gport got during PW creation. The API would then associate the same MEPID of the LSP endpoint for the passed Pseudowire egress interface. This API could be called for multiple Pseudowire egress interfaces to be associated with single LSP OAM endpoint.
SDK-80059		88670_A0	move mbist to perform before tables init to avoid start the chip with memories filled with garbage as a result of mbist
SDK-80073		88660_A0	OAM: When processing OAM events, some of the events may have been from RMEPs that have been destroyed before the event was processed.
SDK-80162		88670_A0	On some cases, RIFs were limited to 4K and not according to the soc property value.
SDK-80189		88660_A0 88670_A0 88670_B0	code now looks also for disabled members in trunk, changed to use SW DB instead of HW tables.
SDK-80218		56620_B0	In previous release,Customers can't use API bcm_cosq_control_set(unit, port, cos, bcmCosqControlBandwidthBurstMin, burst_min) to configure BandwidthBurstMin on BCM56620. In this release,Customers can use API bcm_cosq_control_set(unit, port, cos, bcmCosqControlBandwidthBurstMin, burst_min) to configure BandwidthBurstMin on BCM56620.
SDK-80227		88670_A0	init block id to prevent crash

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80253 SDK-80318		56260_A0 56460	In response to a query "Can BCM56460 BHH CC on Trunk port? the response is "bcm_oam_endpoint_get will return trunk_index if endpoint is configured on LAG port."
SDK-80270		88660_A0	Support OAM/BFD with warm-boot.
SDK-80284		56854_B0	In previous releases, HSP_SCHED_GLOBAL_CONFIG was not configured correctly in some compilers when mmu port was in Y-pipe and greater than 32. This has been fixed in this release.
SDK-80291		56640_A0	In previous releases, the VLAN_VFI field was missed when initializing the EXT_L2_IPV4_ACL, which resulted in mismatching for some qualifiers. In this release, the VLAN_VFI field has been added back and EXT_L2_IPV4_ACL can work correctly.
SDK-80305		56840_A0 56846_A0	The restriction to qualify VlanFormat before adding VLAN related actions is removed. Now, it is up to the user to decide the valid combinations going to work for the incoming Double-tagged or Single-tagged packets. The restriction is removed for the following actions: bcmFieldActionOuterVlanCopyInner, bcmFieldActionOuterVlanPrioCopyInner, bcmFieldActionOuterVlanCfiCopyInner, bcmFieldActionInnerVlanCopyOuter, bcmFieldActionInnerVlanPrioCopyOuter and bcmFieldActionInnerVlanCfiCopyOuter.
SDK-80319		88670_B0	Added a new flag to set the learn distribution messages to the DMA fifo in the olp. BCM_L2_ADDR_DIST_SET_CPU_DMA_DISTRIBUTER flag to be used in bcm_l2_addr_msg_distribute_set
SDK-80332 SDK-80330 SDK-79274		56850_A0	In previous releases, the VPWS packet encapsulation modifications could not be done in the system ingress chip. In this release, this has been fixed by extending MPLS interface flags to support packet modification on HG port for VPWS stacking mode.
SDK-80352		88660_A0	Existence of RXAUI port in the system will cause to Stat Interface no to work properly, unless it is also defined as RXAUI. Fixed.
SDK-80358		88670_A0	Enable new PRGE selection and configuration to support inject Packet with ITMH only (No Egress PP).
SDK-80371		88650_B1 88660_A0 88670_A0	NULL pointer check was performed on tos value. Deleting this check solves the issue.
SDK-80387		56260_A0	Added support for far_loss in bcm_oam_pm_stat_t and to fetch the same from ukernel.
SDK-80388		56260_A0 56460	Issue: SDK host was unable to get the status of Individual BHH fault info from uKernel, if fault occurs in ukernel. Fix: Individual BHH fault status retrieve Support has been added.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80407		56540_B0	In earlier releases, SDK does not check the key type of the table VLAN_XLATE_EXTD after warmboot in BCM56640/BCM56540. In this release, added to check the key type of the table VLAN_XLATE_EXTD in BCM56640/ BCM56540.
SDK-80452		56850_A0 56850_A1 56850_A2 All	In earlier releases, ALPM sanity check command would erroneously report issue for route patterns or some configurations. Now, these false alarms have been fixed and sanity check routine has been improved to cover more software or hardware inconsistency, and it scans faster.
SDK-80453		56260_A0 56460	Problem:Lower maintenance domial level(MDL) BHH OAM packets are not sent to CPU for processing. Solution:Allowed lower MDL BHH packets to copy to CPU from data pipeline.
SDK-80465 SDK-77694 SDK-80463		56850_A0 56860_A0 56960_A0	In previous releases, there was no API to configure registers FLEXIBLE_IPV6_EXT_HDR/ EGR_FLEXIBLE_IPV6_EXT_HDR. This has been addressed by providing bcmSwitchFlexIP6ExtHdr and bcmSwitchFlexIP6ExtHdrEgress.
SDK-80479		56643_B0	In previous releases, speed of internal PHY was configured before Internal PHY initialization. The issue has been fixed.
SDK-80494		56640_B0	In the previous release cl37 and cl73 could not co-exist on the same port. This has been resolved so that user can do either 1G cl37 or cl73 10G KR(1G KX) without changing the config.
SDK-80503		56340_A0 84848_A0	In previous releases, the PHY master mode could be not right by bcm(_esw)_port_master_get API at PHY chip BCM84848. In this release, this issue was fixed.
SDK-80505		56850_A0 56850_A1 56850_A2 All	In previous releases, when knet receive the packets with error bit, the packet was not handled correctly. Now if customer creates a filter and matches error bit, this kind of packet will be processed.
SDK-80510		56460	Issue: -------------- In the earlier release, in the cpu_cos_map table ETH LM/DM had lower index then BHH LM/DM. As both were having the same reason code OamLmDm and ETH LM/DM is qualifying only in reason code OamLmDm , so the incoming BHH LM/DM packets were sent to the cos corresponding to ETH LM/DM. The search in cpu_cos_map happens from lower index only. In absence of ETH LM/DM, BHH LM/ DM packets are going to BHH LM/DM cos only. Fix: -------------- have put ETH LM/DM related cos in higher index in cpu_cos_map table then BHH LM/DM. So, packets will be matched for lower index of BHH LM/DM and will be put to BHH LM/DM cos though ETH LM/DM is present.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80522		56460_A0	Some CMIC interrupts that should have been handled in kernel mode and did not need to report to user mode were not being masked properly for iproc hosts. This issue has been fixed.
SDK-80528		88670_A0	ILKN TDM is supported.
SDK-80549		88660_A0	In the previous release, there was no way to disable/enable IPMC (S, G) lookup based on PON port in PON application. In this release, this issue has been addressed by adding a customer soc property (custom_feature_pon_ipmc) which is used to disable/enable IPMC lookup based on PON port.
SDK-80558		88660_A0 88670_A0	Between soc_ppd_frwrd_mact_get_block calls the flush machine stops. The way it was stopped was by setting current_index to 0. Pausing the flush machine using current_index=0 might cause an error if entries in the LEM move during the flush. Use PPDB_B_MACT_FLU_MACHINE_CONFIGURATION.MACT_FLU_MACHINE_PAUSE to overcome the problem.
SDK-80575		88660_A0	For previous BFD firmware, there was no way to know when can update the bfd endpoint if the poll bit was set. In this release, this issue has been addressed by adding new endpoint flags of bcm_bfd_endpoint_get() BCM API to get change event when the local flags changed.
SDK-80584		56460_A0	Problem: OAM APIs were not validating endpoint ID. Solution: Handled invalid endpoint ID passed and returned appropriate error.
SDK-80611		88670_A0	Fixed memory leak (existed in 6.4.7) of memory allocated when calling to bcm_field_group_create/ bcm_field_entry_create and was not freed when calling to bcm_field_group_destroy/ bcm_field_group_compress/ bcm_field_entry_remove.
SDK-80619		88670_A0	Diagnostic command "fabric link all" doesn't display links status. Fixed.
SDK-80632		88670_A0 88670_B0	In FEC, an IP-FEC which uses Strict RPF protection couldn't be used as the second-level FEC (marked with the CASCADED flag). No restrictions on IP-FEC as second-level FEC are currently enforced by the SDK.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80633		56860_A0	In previous releases, the reference count in VLAN action profile table entry was not updated correctly in below two sequences: 1) When adding a port to a trunk which is already part of the VPN, and using api bcm_vlan_port_default_action_set to align port property; 2) When removing a port from a trunk which is already part of the VPN, and using api bcm_vlan_port_default_action_delete to align port property. This has been fixed in this release.
SDK-80636		88660_A0	BFD: The procedure of updating BFD endpoints no longer requires the OAMP to freeze transmission of BFD frames, as long as additional profiles are available. It is the users responsibility to verify that there are available profiles. In the unlikely event that new bugs were introduced by this JIRA set the soc property custom_feature_bfd_original_update_procedure=1 This soc property will only be available for the next release.
SDK-80642		All	In previous releases, deadlock might happen if the operation of enabling flexible Statistics resulted in errors in some condition. This has been resolved.
SDK-80665		56440_A0 56450_B0 56450_B1	Adding 1+1 protection support for MSP containing egress queues. So far this was supported only on MSP containing egress objects.
SDK-80682		56460	Issue ==== 64 bit stat value was printing on cli using two 32bit values. Lower 32 bit value was not formatted properly so that when the value occupied more than 32bit, it was printed incorrectly. Fix === Corrected the lower 32bit print format, so that the value printed is proper even when it more than 32bit.
SDK-80701		88660_A0 88670_A0	MPLS PORT: bcm_mpls_port_get fails to get object which was just created using bcm_mpls_port_add
SDK-80720		56850_A0 56860_A0	In previous releases, API document hadn't described the purpose of adding bcmVlanVPMcControlAuto clearly. This has been fixed in this release by adding a note to provide the reason of adding bcmVlanVPMcControlAuto in API document. i.e. bcmVlanVPMcControlAuto was added for backwards compatibility, and it will be the default mode for VP Multicast replication. Users need to configure the VP Multicast replication mode to bcmVlanVPMcControlEnable/ bcmVlanVPMcControlDisable instead of bcmVlanVPMcControlAuto in their applications.
SDK-80733		53406_A0	Fix the endian information from pci_bus_feature function for 53xxx devices in kernel mode SDK. In previous release, the endian information was incorrect in kernel mode SDK.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-80737		`56860_A0`	bcmPortControlPFCClasses was not implemented. Added support for PFC control bcmPortControlPFCClasses.
SDK-80738		`88670_A0`	When Linkscan enabled wrong AN status might cause unexpected NIF configuration in PM4x25 ports. Fixed.
SDK-80739		`88670_A0`	Linkscan is now enabled in default application (jer.soc).
SDK-80789		`56260_A0`	Problem: In the earlier releases, there was no default Target MEP TLV support for BHH LB as per G8113.1 specification Solution:Added support of BHH LB to send Target MEP TLV as default.
SDK-80803		`56860_A0`	Fixed 1G sgmii autoneg on td2+
SDK-80811		`88650_B1 88660_A0 88670_A0 88670_B0`	DNX: Fixed an issue that caused lif banks not be cleared properly once the last issue has been deallocated, causing the bank to be unusable.
SDK-80817		`56840_A0 56850_A1`	In previous releases, when Linkscan thread was turned on, it didn't check PHY and MAC status of all ports. Now, when Linkscan thread is restarted, the thread will operate ports according to its PHY and MAC status.
SDK-80829		`56840_A0`	In previous releases, it was not allowed to add a NIV port created with `BCM_NIV_PORT_MATCH_NONE` into a VPLAG. This issue has been fixed.
SDK-80839		`56460_A0 56460_B0`	Earlier SDK was limiting the user to specify OAM RMEP and LMEP endpoint name unique among RMEP and LMEP database. But in fact RMEP endpoint identifier name should be unique among RMEP database and it can be common name in LMEP database. Similarly OAM LMEP endpoint identifier name should be unique among LMEP database but it can be common name in RMEP database. Now SDK allows users to configure same name for OAM RMEP and LMEP endpoint (but unique for same type of endpoint) via `bcm_oam_endpoint_create()`.
SDK-80871		`56643_A0`	Issue ==== Triumph3 has only one view in `FP_SLICE_MAP`, but SW was maintaining 3 different views due to this external tcam priorities was not getting updated properly. Fix == Added changes to update Default view for all external tcam groups.
SDK-80887		`56540_A0 56540_B0`	Port Configuration 48 x GbE + 1 GbE 4 x XFI/ HGs[11] 2 x HG[42] was not supported in earlier releases. The port configuration has been implemented in the current release.
SDK-80903		`56450_A0 56450_B0 56450_B1`	CosQ settings are done during creation of egress object. But settings were not removed during destroying that egress object. As part of this fix, SDK will remove CosQ setting during `bcm_l3_egress_destroy()` which were created during `bcm_l3_egress_create()` for that egress object.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-80909		88670_A0 88670_B0	In L3 IPv4, it is now possible to delete default routes in BCM88670.
SDK-80912		88670_A0	In BCM88670 local-port (0-511) is mapped to {core-id (0-1) and pp-port (0-255)}. In BCM88660 PP-port is equal to local-port between 0-255 Updated the Mac-in-Mac entry-management accordingly.
SDK-80917		88670_A0 All 88670_B0	In port control, a new enum member, bcmPortControlMplsSpeculativeParse, was added to bcm_port_control_t. This option enables the speculative parsing of IPv4/6 headers over MPLS headers.
SDK-80920		56850_A0	Support has been added for increasing the robustness of bcm_switch_pkt_info_hash_get() API in invalid configuration.
SDK-80940		88660_A0	In the previous release, LIFs from 0 to 4K could be allocated base on the FEC group. In this release, this issue has been addressed by adding the support via a new soc property "custom_feature_pon_wihout_rif_support = 1",
SDK-80960		56850_A0 56860_A0	1k stack was being used by internal function which used to move field entries in BCM56850/860, due to which stack issues were seen. Moved variables to heap and reduced stack usage for entry move function
SDK-80964		56640_A0	In previous releases, the HG port link status will flip if the port receive runt packets. In this release, this issue was fixed.
SDK-80979		56850_A0 56850_A1 56850_A2	The trill lock using in _bcm_esw_trill_trunk_member_delete/add is unnecessary. So remove it.
SDK-80988		88660_A0	In Ethernet Policer bcm_rate_bandwidth api, when Packet mode was used, the rate wasn't treated as PPS, leading to wrong handling of the minimal PPS check. This is now fixed. In addition, in metering, when packet mode meter was configured, the correct flag wasn't returned (see SDK-82362 for reference).
SDK-81000		56640_A0 56640_A1 56640_B0	From this release, MAC aging timer is backed up and restored during warm boot.
SDK-81008		56960_A0	HW counter offset modes used for VxLAN, were incorrectly shared with stat groups created for IFP rules that match on VxLAN header fields. Code changes were made to avoid sharing IFP created counter modes (created using custom stat API) with other stat groups created using legacy stats API.
SDK-81026		88670_A0	BCM88670 init issue on 64 bit linux system fixed.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-81043		88670_A0	Egress Protection: The egress_failover_port_id field that is returned by bcm_mpls_port_get(), indicates whether the object is a Working or Protecting port of an Egress Protection scheme. '1' indicates Working, while '0' indicates Protecting. The returned value was incorrect, so that '0' indicated Working and '1' indicates Protecting. The returned value was fixed to return the expected value.
SDK-81053		88660_A0 88670_A0	MPLS PORT: Using bcm_mpls_port_get will not set falgs2 with BCM_MPLS_PORT2_LEARN_ENCAP even if it was used when calling bcm_mpls_port_add.
SDK-81057		88670_A0	MPLS PORT: Using bcm_mpls_port_get will not set falgs with BCM_MPLS_PORT_NETWORK even if it was used when calling bcm_mpls_port_add.
SDK-81060		88670_A0	bcm88670: The diagnostic "diag pp fdt" will now display the correct outlif for "Ingress decision" row.
SDK-81063		56842_A0 56846_A1 56850_A0	In previous releases, bcm_cosq_cpu_cosq_enable_set might not work in some cases. This has been addressed by taking dynamic threshold into account to prevent potential API malfunction.
SDK-81075		56850_A0 56850_A1 56850_A2	In previous releases, RQE guaranteed cells for lossless mode was configured as 0. This has been addressed by setting the guarantee to 8 to align with MMU configuration specification.
SDK-81101		56960_A0	In previous releases, DSCP value in the outer tunnel header was always set to 0 when using bcmTunnelDscpAssign as the selection method. In this release, the issue has been fixed by configuring correct DSCP value through bcm_tunnel_initiator_set.
SDK-81104		56860_A0	Definition for Meter_Gran 5,6,7 in IFP has changed in BCM56850/BCM56860 and BCM56960. Added support to change rates and burst calculation for these granularity levels in IFP Metering logic.
SDK-81107		All	Added weighted load balance support for non-unicast packets in trunk group.
SDK-81135		88660_A0	Learn profiles are updated when vlan/vswitch is destroyed. Please use bcm_vlan_control_vlan_get and update only the relevant fields before using bcm_vlan_control_vlan_set.
SDK-81136		All	This release addresses a CLI issue where entering the command 'log file=', leads to an assertion. With this fix in place, the above indicated command can be run succesfully with logging output destined to the log file with name specified by any earlier log file command or the default log filename 'bcm.log'.
SDK-81161		56260_A0 56460	Added support for action OamLmDmSampleEnable for BCM5645x, BCM56260 and BCM56460.

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-81173		88660_A0	BFD: Slight optimization of endpoint update procedure.
SDK-81174		88670_A0	Added support for ILKN 5 lanes. Note that the currently the shared quad must be without lane swap.
SDK-81188		88670_A0	The APIs bcm_oam/bfd_detach are not available to the user.
SDK-81189		88660_A0	In the previous release, creating a lif with a specific ID failed if the ID had been created and destroyed before. In this release, this issue has been addressed by correcting the 3 tags logic of bcm_vlan_port_destroy function.
SDK-81191		88670_A0	Fixed drops when sending high rate traffic trough fabric.
SDK-81201		88670_A0	Missing support for stage configuration data for BCM88670 was added.
SDK-81202		88670_A0	Tuned default thresholds for OCB DBs reject test - API threshold.flags = BCM_COSQ_THRESHOLD_DROP | BCM_COSQ_THRESHOLD_INGRESS | BCM_COSQ_THRESHOLD_OCB; threshold.type = bcmCosqThresholdBufferDescriptors; threshold.value = <number-of-ocb-bds> * 0.97; BCM_COSQ_GPORT_VSQ_GL_SET(gl_vsq , BCM_CORE_ALL); for (threshold.dp = bcmColorGreen; threshold.dp <= bcmColorBlack; threshold.dp++){ print bcm_cosq_gport_threshold_set(unit, gl_vsq, 0, &threshold); } Fixed VSQs default tail-drop configuration - to be maximal value. Mask out irrelevant reject tests in loose congestion mode (SOC property: ingress_congestion_management_guarantee_mode=LOOSE). Fixed VSQs default tail-drop configuration: uint32 flags = BCM_COSQ_GPORT_SIZE_BUFFER_DESC | BCM_COSQ_GPORT_SIZE_COLOR_BLIND ; gport_size.size_max = 8355840; for (is_ocb_only = 0; is_ocb_only < 2; is_ocb_only++) { for (category = 0; category < 4; category++) { BCM_COSQ_GPORT_VSQ_CT_SET(vsq_gport, BCM_CORE_ALL, is_ocb_only, category); bcm_cosq_gport_color_size_set(unit, vsq_gport, 0, 0, flags , gport_size); }}
SDK-81213 SDK-83663		56960_A0	Problem: l3_bitmap in L3_IPMC table was not updated after bcm_multicast_egress_delete API call in BCM56960. Fixed the code to update l3_bitmap in L3_IPMC table after bcm_multicast_egress_delete API call in BCM56960..

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-81216		56260_A0 56260_B0 56460	Available api bcm_coq_mapping_set and bcm_cosq_port_mapping_set maps one priority to one cosq. So each priority cos map update consumes a profile entry leading to profile entry exhaustion. Providing a new api to map multiple priority with corresponding cosq in one go.
SDK-81217		88670_A0	Fixed VOQ WRED configuration in BCM88670.
SDK-81248		56334_B0	The response messages received from keystone were ignored by the host because of a wrong VLAN tag check. This check is now corrected to compare only VLAN ID instead of VLAN tag (i.e., mask out VLAN priority and CFI).
SDK-81259		88670_A0	When calling bcm_cosq_gport_bandwidth_set with an E2E port and kbits_sec_max as 0, a DIVISON_BY_ZERO exception was thrown. This has been fixed, and user will be able to set bandwidth to 0 for an E2E port using the previously mentioned function.
SDK-81269		56260_A0 56450_A0 56850_A0 56960_A0	On Multiple XGS devices, the default behavior for VFP is to override the internal priority assigned by L2/L3 lookups. A new field control bcmFieldControlOverrideStageLookupPhb is introduced in this release, and when this is set, VFP doesnot override the internal priority assigned by L2/L3 lookups.
SDK-81276		56860_A0	Added support to use all LPM(TCAM) table for IPv6 128-bit mode purpose.
SDK-81301		56440_A0	In previous releases, MMU parity errors were seen on BCM56440 devices. In this release the bucket burst calculation has been fixed addressing the MMU parity error problem.
SDK-81302		56860_A0	In previous releases, when the API bcm_l3_egress_multipath_max_create was invoked with BCM_L3_REPLACE, the passed parameter max_paths would be treated as the new MaxGrpSize of the given ECMP group and the API would not return an error for this case, which would lead to chaos in the ECMP member table. This has been addressed by allowing the dynamic change of MaxGrpSize of the given ECMP group.
SDK-81304		88670_A0 88670_B0	SPLIT HORIZON: fixed VLAN 2 bit split horizon behavior in certain scenarios.
SDK-81309		56260_A0	Support has been added in this release for LED port scan order for BCM56460.
SDK-81312		56960_B0 56960_A0	dsc dump will dump all 4 lanes of that core, however for the management ports on BCM56960, only two lanes are active, while the other two lanes are not active, so attempts to dump the inactive lane will cause time out error. Workaorund will be Only dump active lane PMD info for tsce on BCM56960 to avoid the error message.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-81339		88660_A0 88670_A0	SLM support: All OAM packets of type SLM, SLR, LMM, LMR can either be counted or not counted together (counting is done per LIF). The result is The user may not configure an SLM session on one LIF and an LM session on another. In order to enable SLM counting, call `bcm_oam_endpoint_action_set()` for an endpoint with the actions bcmOAMActionSLMEnable, bcmOAMActionCountEnable and the opcodes set to 55, 54 (SLM, SLR). For an example, refer to `slm_set()` in `cint_oam.c` To disable SLM/SLR/LMM/LMR counting (globally) call `bcm_oam_endpoint_action_set()` with the opcodes set to 55, 54 and without any actions. Note that whenever `bcm_oam_endpoint_action_set()` is called with the opcodes set to SLM or SLR, counting of SLMs/SLRs/LMMs/LMRs will be enabled or disabled, depending on whether or not the action bcmOAMActionCountEnable is set.
SDK-81349		88670_A0	Fix discards when peer FE device is shut down by bcmFabricShutdown.
SDK-81364		56960_B0 56960_A0	Meter and Counter bitmaps sync and recovery procedure was updated to point to right values and set right length of field for IFP warm boot in BCM56960.
SDK-81369		88660_A0 88670_A0	MPLS: When encapsulating with entropy label at BOS, it's possible to also add ELI label by setting the soc property `mpls_egress_label_entropy_indicator_enable=1` and using the flag `BCM_MPLS_EGRESS_LABEL_ENTROPY_INDICATION_ENABLE` for the first label in the tunnel initiator array.
SDK-81391		56040_A0 56060_A0	Added support for the ability to clear SRC/DST Hit-bit respectively in `bcm_l2_replace` function for some devices without supporting L2 bulk feature.
SDK-81398		88670_A0	- Lif info access is fixed to use the `local_lif_id` instead of `global_lif_id` - Diagnostic modified to display both global and local IDs
SDK-81414		56960_B0 56850_A0 56850_A1 56850_A2 56960_A0	This release optimizes the 'dump socmem' command in order to reduce execution time by about 120 seconds when compared to the prior release.
SDK-81422		56960_B0 56960_A0	added the support to allow dynamic switch of cl37 and cl37 sgmii
SDK-81480		56850_A0 56860_A0	Implemented new field qualifier "bcmFieldQualifyTrunkMemberSourceModuleId" to qualify on Trunk member source module ID.
SDK-81483		56850_A0 56860_A0	Implemented new qualifier "bcmFieldQualifySrcModuleGport" to provide a support in qualifying module ID on a trunk port.
SDK-81526		88670_A0 88670_B0	Added support for inserting the snoop fhei in copied packets.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-81541		88670_A0	The diagnostics command "diag pp DB_LEM_lkup_info" output was wrong, in case of SA authentication lookup (lkup_num=2). The command's output indicated "no hit", even if a hit has occurred. The command output is now correct.
SDK-81548		88660_A0 88670_A0	AC-LIF: when bcmVlanPortDoubleLookupEnable is enabled on incoming port, PWE/MPLS/IP/IP-GRE/VXLAN (etc) tunnel termination is not supported by default. The following soc property needs to be set to terminate tunnels: bcm886xx_mpls_termination_database_mode value 1/3/5/6/8 (see PG document BCM MPLS support -> General configuration)
SDK-81550		56450_B0	Packet Drop was observed when dynamic modification of replication list is done with traffic.Packet drops were avereted by enabling the atomic update for replication list in hardware .
SDK-81560		88670_A0	OAM: Fixed a bug in bcm_oam_endpoint_create() For endpoints of type MPLS-TP: When mpls_out_gport !=BCM_GPORT_INVALID, the field gport only needs to be defined as an inLIF. Note that for other cases the gport field must be defined as an outLIF as well as an inLIF.
SDK-81562		88660_A0	BFD: Support pre-allocated BFD transmission periods (required due to HW bug). The transmission rates are as defined in RFC-7419. These will be pre-allocated if the soc property custom_feature_bfd_fixed_tx_rate_mgmt is set. Additionally, it is possible to override the values of 3.33, 10, 20, 50, 100, 1000 (defined in RFC-7419) with: custom_feature_bfd_fixed_tx_rate_mgmt_rateX where X=[0,..,5], respectively.
SDK-81568		56260_A0 56460	Added support for MPLS-label-based BHH lookup for PW at ingress.
SDK-81569		56540_A0 56540_B0	In earlier releases SDK did not check the count of profile information when deleted mpls information. In this release, added to check the count.
SDK-81571		88670_A0	In BCM88670 Metering, the PCD map used both in Serial and Parallel modes, was configured incorrectly in cases in which only one meter was applied on a packet, this is now fixed. In addition, in Serial metering, when two meters are applied on a packet and the following applies: 1. first meter has green credits but doesn't have any yellow credits 2. second meter has yellow credits but doesn't have any green credits The output color would be yellow (but green credits will be reduced from the first meter). The previous configuration was to mark the packet as red.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-81586		56850_A0 56850_A1 56850_A2	In previous releases, SDK couldn't update mirror MTP reference count correctly if mirroring destination was added to a new port after mirror_dest.flags was set to BCM_MIRROR_DEST_REPLACE. This issue has been fixed in this release.
SDK-81598 SDK-84807		56462_B0 56266_B0 56467_B0 56466_A0 56261_A0 56467_A0 56462_A0 56263_B0 56266_A0 56263_A0 56466_B0 56261_B0 56260_A0 56260_B0 56262_A0 56262_B0 56265_A0 56265_B0 56267_A0 56267_B0 56268_A0 56268_B0 56460 56460_A0 56460_B0 56461_A0 56461_B0 56463_A0 56463_B0 56465_A0 56465_B0 56468_A0 56468_B0	Queues associated to ports which are not linked up were kept enabled and as a result SOBMH (CPU generated) packets were kept in these queues and were never drained out. Fix is putting the queue in disabled state to avoid any enqueing of packets to those queues which are bound to linked down interfaces.
SDK-81601		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-81609		56850_A0 56850_A1	Issue: TCAMs had a HW issue of no hits happening during an entry modification/update time interval. To achieve hitless update , we had to provide a SW WAR. Solution: Create an internal backup entry for hits to happen during the time when original entry is being updated. Once update is complete, delete the original entry.
SDK-81649		88650_B1 88670_A0	Cosq: bcm_cosq_fc_path_add with bcmCosqFlowControlReception can used to set flow control path in RX direction. When calling bcm_cosq_fc_path_add for NIF interface without flag BCM_COSQ_FC_ENDPOINT_PORT_IMPLICIT, BCM_E_PARAM will always be returned due to wrongly checking for paramter target cosq. After the fix, paramter target cosq can be 0 or -1.
SDK-81655		56850_A0 56850_A1 56850_A2	In previous releases, when a host entry was in embedded view in the l3 host table, the flag BCM_L3_DEREFERENCED_NEXTHOP could not be got in the API bcm_l3_host_add and bcm_l3_host_traverse. This has been fixed.
SDK-81668 SDK-1191		56260_A0	Added S/w swap option for CMIC register access from SDK running on iProc internal A9 CPU in BE mode (CMIC_SOFT_BYTE_SWAP flag), in order to work with Firmware applications which run on internal R5 cores in LE mode.
SDK-81673		88660_A0 88670_A0	PRGE variables like TM and PP profiles were not stored in WarmBoot causing a failures to restore after WB recovery - this issue is now fixed.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8

SDK-81674		`56462_B0 56266_B0`
		`56467_B0 56466_A0`
		`56261_A0 56467_A0`
		`56462_A0 56263_B0`
		`56266_A0 56263_A0`
		`56466_B0 56261_B0`
		`56260_A0 56260_B0`
		`56262_A0 56262_B0`
		`56265_A0 56265_B0`
		`56267_A0 56267_B0`
		`56268_A0 56268_B0`
		`56452_B0 56450_A0`
		`56450_B0 56450_B1`
		`56455_A0 56456_A0`
		`56456_B0` 56460
		`56460_A0 56460_B0`
		`56461_A0 56461_B0`
		`56463_A0 56463_B0`
		`56465_A0 56465_B0`
		`56468_A0 56468_B0`
		`56457_B0 56455_B1`
		`56458_B0`
			When enabling COE/Linkphy on a port having live traffic, there were instances when the customer observed that the LLS scheduler hung and would not process packets from that port. This was root caused to internal API sequencing problem. The SDK code was corrected by changing the subport sequence such that subport enable is called as the last step.

SDK-81688		`56860_A0`	`phymod_phy_firmware_lane_config_get/set` funtions have been implemented to show DFE/DFE LP MODE in phy control ce0
SDK-81707		`88670_A0 88670_B0`	Support micro BFD (RFC-7130). This may be done by setting the soc property `micro_bfd_support_mode=IPv4` and calling `bcm_bfd_endpoint_create()` with the flag `BCM_BFD_ENDPOINT_MICRO_BFD.` Notes: 1. Currently micro BFD is supported only over IPv4 2. When using micro BFD, micro BFD frames as well as single hop BFD frames will be classified through the LEM.
SDK-81711		`56860_A0`	Issue seen because TX/RX Lane mapping is not read from config and applied for core1 and core 2 in TSC 12 With this fix "lane mapping is now read from config.bcm file for core 0/1/2 for the first logical port in the core 0/1/2 and is applied accordingly, if no lanemapping is set for core0/1/2 default tx/rx lanemapping "0x3210" is used"
SDK-81712 SDK-58521		`56547_A0`	In previous releases, the `SERDESDIGITAL_CONTROL1000X` register bit 0 was programmed on the API `bcm_port_medium_config_set` when customer want to set SGMII mode. This has been fixed by programming the `SERDESDIGITAL_CONTROL1000X` register bit 0 on the API `bcm_port_medium_config_set.`
SDK-81721		`56640_A0`	In earlier releases the user could not use the memory `L3_DEFIP_PAIR_128` when enabled LPM scaling function. In this release, supported to use the memory `L3_DEFIP_PAIR_128` when enabled LPM scaling function.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-81741		56860_A0	In previous releases, access MIM logical port SD TAG encapsulation could only base on VPN. In this release, access MIM logical port can be shared across multiple access VPNs, and the access SD TAG encapsulation bases on VPN and MiM port. API bcm_vlan_translate_egress_action_add has been extended to support this feature.
SDK-81754		88670_A0	Fixed bcm_port_interface_set for fabric SerDes.
SDK-81755		88660_A0	BFD: when using workaround for lifting the limit on SIP in generated BFD traffic, if a packet needed editing deeper than 58B (encapsulation/ stamping/etc.) the MSB had been truncated. This was fixed and the current limit is 92B of editing depth.
SDK-81765		88660_A0	BFD: When updating the Detection Time of a BFD endpoint (done with bcm_bfd_endpoint_create()), The timers indicating when the last BFD packet was recieved will remain untouched except for the following cases: 1. When going from low Detection Time to high Detection Time. 2. When one Detection Time is above ~1s and the other is below ~1s. These are due to the implementation in the HW.
SDK-81795		88650_A0 88660_A0 88670_A0	When Mac-in-Mac is disabled, Mac-in-Mac multicast traffic can now pass through.
SDK-81841		All	Fixed the issue in the code for RCPU address operation.
SDK-81850		56860_A0 56960_A0	Added the fix for hg 40G/42G MLD support, if in Hg mode, set if=KR4, then it will use MLD(IEEE) based encoding, if=xgmii will use brcm 64/66 encoding.
SDK-81851		88660_A0	MPLS PORT: CPU Based Learning protected MPLS Port fix
SDK-81857		88670_A0 All 88670_B0	Fixed dynamic counter configuration, to update a configured engine to enabled it when calling bcm_stat_counter_config_set().
SDK-81864		88670_A0	SoC property "serdes_fiber_pref" fixed to be read per port_name ("ce", "xe", etc.)
SDK-81871		88660_A0 88670_A0	Support for MPLS ingress protection has been added to SDK.
SDK-81914		56960_B0 56960_A0	In previous releases, Visibility API could fail if port untagged vlan was changed by bcm_port_untagged_vlan_set API. In this release, this bug has been fixed.

Table 73:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.8***
SDK-81924 SDK-83513 SDK-83511		`56960_B0 56850_A0 56850_A1 56850_A2 56960_A0`	Adding the supports of following L3 objects: bcmSwitchObjectL3HostV4Max, bcmSwitchObjectL3HostV6Max, bcmSwitchObjectL3RouteV4RoutesMinGuaranteed, bcmSwitchObjectL3RouteV6Routes64bMinGuaranteed, bcmSwitchObjectL3RouteV6Routes128bMinGuaranteed, bcmSwitchObjectL3EgressMax, bcmSwitchObjectIpmcV4Used, bcmSwitchObjectIpmcV6Used, bcmSwitchObjectIpmcV4Max, bcmSwitchObjectIpmcV6Max. Adding the supports of following L3 objects for ALPM tables: bcmSwitchObjectL3RouteV4RoutesMax, bcmSwitchObjectL3RouteV6Routes64bMax, bcmSwitchObjectL3RouteV6Routes128bMax.
SDK-81962		`88670_A0 88670_B0`	There was a bug in the code that configured the flush machine of BCM88670 in a wrong way. The wrong configuration left a lot of filtering work to the driver which made the traverse very slow. Now the filtering is done in the flush machine and it is match faster.
SDK-81983		`88660_A0 88670_A0`	OAM: Update of endpoints of type MPLS-TP may fail when `mpls_out_gport` != `BCM_GPORT_INVALID.`
SDK-81985		`88670_A0`	Traffic drop occurs at full rate. Fixed
SDK-82001		`56640_A0 56640_A1 56640_B0`	In previous releases, autoneg fast timers was set for WCMOD. This has been fixed by deleting autoneg fast timers config for WCMOD.
SDK-82005		All	In existing BCM KNET implementation, packet streams from virtual network interfaces and BCM Tx API are multiplexed by the common low-level driver on the transmitting side. Accordingly, a flag is utilized to indicate whether current packet streams are belonging to virtual network interfaces or BCM TX API. In previous releases, this flag was not protected properly under multi-core CPU SMP circumstances and there existed a race condition in which packet transmission from virtual network interface would conflict with BCM Tx API. In this release, this issue has been resolved by protecting this flag using the appropriate lock mechanism to eliminate the race condition.
SDK-82032		`88660_A0 88670_A0`	`cint_l2_cpu_learning` was updated. Test for `BCM_L2_EGRESS_CENT` failed. The learn mode should be set before changing the distribution of learn messages for a specific vsi. `bcm_switch_control_set` to set the learn mode and then (if needed) `bcm_l2_addr_msg_distribute_set.` `bcm_switch_control_set` is setting default values according to the learn mode and `bcm_l2_addr_msg_distribute_set` is updating these value
SDK-82039		`56440_A0`	In previous releases, `FP_TCAM` table was not in TCAM protection list, this table has been added into this list in this release.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82042		56860_A0	Allowed higher priority value (DSCP_TABLE:PRI) to be configured depending on asic's capability (PRI field length). DSCP_TABLE:PRI is 4bits in Trident 2+ but the API allowed the max value only as 7 to be programmed. Changed the checks in the API to allow for high priority value depending on the asic's capability (length of priority field).
SDK-82043		56260_A0 56460	Issue --------- lm_ctr_pool_id, as reported, are being returned from SDK-6.4.7.onwards. So no fix for these. flags2 field was not returned during endpoint_get. Fix ------- Returning flags2 field during endpoint_get.
SDK-82071		56860_A0	Enzo driver returns incorrect status and causes upper layers code to print out error messages when there are no errors.
SDK-82083		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	Added support for Actions IntCosUcastQueueNew & UcastQueueNew.
SDK-82110		56260_A0 56460	Issue: bcm_oam_endpoint_action_set did not have a reset functionality(i.e) a way a to reset already set actions. Fix : A new flag "BCM_OAM_ENDPOINT_ACTION_DISABLE " has been added for this purpose. This flag needs to be passed via flags field of bcm_oam_endpoint_action_t structure with the relevant actions set in the opcode_actions field to the bcm_oam_endpoint_action_set API. This flag is now handled in SDK to actually reset the action set to 0 (default).
SDK-82144		56867_A0 56860_A0	Reduced the timeout value in the sched_set for BCM56860.
SDK-82151		88660_A0	In the previous release, customer use "vlan show" command to get the port member of VLAN, which will return the wrong port number.In this release,this issue has been addressed by modifying the port number mapping, user can get the correct port number by vlan show.
SDK-82182		56260_A0	In this release lpm scaling feature is enabled for BCM56260 device. So the 64b IPV6 entries will also be supported in paired TCAM.
SDK-82184		56960_A0	This release addresses an issue where an attempt to set IFG for a port using bcm_port_ifg_set() would fail for HG ports. With this fix in place, IFG can be reliably set using bcm_port_ifg_set() for HG ports.
SDK-82220		88650_A0 88670_A0	Default application configured compensation only for base VOQs, instead it should set it for all priorities. Fixed

Table 73:

Number	*CSP #*	*Chips*	*Release Notes For 6.4.8*
SDK-82230		88660_A0	In Metering, in BCM88660, there was a bug in the call to enable IPG compensation, this is now fixed
SDK-82246		56540_A0 56640_A0	In former release , MMU_REPL_GROUP couldn't be corrected , because here was a hardware issue in OVQ_ECC_BITMAP ; In this release , MMU_REPL_GROUP can be corrected with a software walkaround .
SDK-82250		56460 56460_A0 56460_B0	When an upmep/downmep is created after a downmep/upmep with exactly same service parameters, since there was an entry created in egr_mp_group/mp_group for passive bitmap, we are assuming that the entry already has active bitmap and not reserving sw indexes for MA_INDEX table. Now active bitmap is checked to see if it is 0. If it is 0,we know that this is the first mdl for which the entry is created and sw indexes are allocated thereby solving the no resource error during delete. Also for Intermediate points, when there is no mdl to be configured that is mdl = 0 for passive bitmap, we do not write anything into the mp_group/egr_mp_group entry. And if we do not find a mp_group/egr_mp_group_entry for intermediate point and mdl_field = PASSIVE, we do not raise an error since we do not program the entry as mentioned above.
SDK-82260		88660_A0	oamp_fifo_dma_enable and related soc properties are supported only for BCM88670.
SDK-82262		88660_A0	OAMP DMA endianess is configured according to CPU's endianess.
SDK-82282		56850_A0 56850_A1 56850_A2	In previous releases, if TD2 use Limited Storage Recovery(1.5 WarmBoot), command "sc controlsync=1" cannot be executed successfully. Now this issue was fixed.
SDK-82321		88670_A0	Cosq: API bcm_cosq_fc_path_add with bcmCosqFlowControlReception can be used to map FC indication in PFC frame or in calendar FC to 512 EGQ PFCs by using PFC generic bitmap. After this fix, dynamically PFC generic bitmap allocation for NIF and calendar interface is supported.
SDK-82326		56850_A2	In previous releases, when the API bcm_trunk_set was called, the system might crash occasionally because a local variable in the API was not initialized. This has been addressed by initializing the variable.
SDK-82332		88670_A0	Dynamically adding many channels to a port fail. Fixed
SDK-82368		56450_B0	For 5645X chips the latency of the priority queue will spike up if scheduling change is effected in other queues.Setting min bandwidth for the priority queue will resolve this issue.
SDK-82380		56960_B0	Fixed a bug in terms of B0 lane swap internal workaround
SDK-82392		56340_A0	In earlier releases the device BCM56346 did not support URPF function. In this release, the device BCM56346 supports URPF function.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82394		`56640_B0`	In earlier releases BCM56640 used global system CML control, but it caused some ports' CML controls became useless. In this release, BCM56640 uses port CML control instead of global system CML control.
SDK-82405		`88670_A0`	MPLS PORT: Fix create after deleting same id in case of protection
SDK-82424 SDK-81204		`56860_A0`	In previous releases, the data packets dropping for 1+1 linear protection switching on receive direction of PBB-TE could not be controlled . In this release, this has been fixed. And the protection switching on receive direction or transmit direction has been controlled separately.
SDK-82428		`88670_A0`	DMA L2 learning: Check the CPU's endianness and configure the DMA accordingly. Also fixed the parsing of the OLP events to the DMA.
SDK-82433		`88660_A0` All	New custom `soc_property` `"oam_multiple_event_count_enable"` added. This `soc_property` enable counting of multiple identical events for same endpoint on the same interrupt. The count resides in `BCM_OAM_EVENT_FLAGS_MULTIPLE` field(16 MSB bits). Note that counting starts from 0, so if N events of the same type occur for the same remote endpoint on the same interrupt and the above soc property is on then the value of the counter will be (N-1)
SDK-82443		`56850_A0` `56860_A0` `56960_A0`	In releases starting from SDK-6.2.9/SDK-6.3.2, the virtual port bitmap used to store and recover configuration during warmboot synchronization and recovery was incorrect. The pointer increment for `vxlan_vfi_bitmap` was incorrect which caused the previous VPs configuration was overwritten during warmboot synchronization, which could affect the `VP_LAG,` PE, L2GRE, TRILL or NIV bookkeeping. This issue has been fixed in SDK-6.4.8/SDK-6.5.1. As a result, some SDK-6.2.9/SDK-6.3.2 -> SDK-6.4.8/SDK-6.5.1 warmboot test cases would fail since the virtual port bitmap value doesn't match after the warmboot. Releases starting from SDK-6.4.8/SDK-6.5.1 have no such problem, and the code has been restructured in such a way that pointer increment happens right after the memory copy operation to prevent such a bug.
SDK-82454		`56960_A0`	Cap the `key_type` for `L2_LEARN_INSERT_FAILUREm` & `L2_BULKm` in BCM56960. This prevents `key_type` values from accessing illegal view values during `soc_mem` dump.
SDK-82465		`56460_A0` `56460_B0`	Issue: TPID entries were deleted even when ccm tx was not enabled since `outer_tpid` related software variable was non zero. Fix: Delete TPID entries only when ccm tx is enabled during endpoint delete/destroy call.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82505		56850_A0 56850_A1 56850_A2 All	In previous releases, flag BCM_L3_COPY_TO_CPU could not be used for setting embedded L3 host entries. This support has been added.
SDK-82566		56450_A0 56450_B0 56450_B1	The issue seen was that bcm_cosq_control_set for shaper failed because of abnormally high values retrieved from the chip. This was root caused to some kind of timing issue between port flush operation and shaper get API call where the port flush increased the max shaper to maximum value for a temporary period of time. The fix in the SDK was to check for the flush status before retrieving or modifying the shaper values. A lock is also introduced to make sure that port flush does not begin while shaper get/set is in progress.
SDK-82597		56260_A0 56460_A0	In previous releases the RQE in BCM56450 does not make use of mutilple queues. In this release, all of these queues are used.
SDK-82613		56960_A0	Removed the hardcoding of unit number in BCM56960/field.c
SDK-82620		56860_A0	In earlier releases, vlan port vp could not be destroyed when associated gport was a trunk and this trunk did not have members. This has been resolved.
SDK-82636 SDK-82401		All	In L3, in-rif profiles are no longer allocated both in bcm_l3_intf_create and bcm_l3_ingress_create. Instead bcm_l3_intf_create allocates and bcm_l3_ingress_create exchanges the in-rif profile if necessary.
SDK-82665		56860_A0	In the previous releases, bcm_port_match_multi_get counted the number of all the entres and got the content of all the entries when parameter size's value was greater than zero, so the returned count's value may greater than the size's value, which was incorrect. In this release, when parameter size's value is greater than zero, bcm_port_match_multi_get only gets the customer specified number of entries(size's value), so the numbers and contents of the entries will be not more than parameter size's value.
SDK-82673		88670_A0	Tuned global threshold configuration.
SDK-82702		56640_A0 56640_A1 56640_B0	In previous releases, a SBUS DMA error may occur when TRIUMPH3 HG port receives runt packets. This issue was fixed in this release.
SDK-82740		88660_A0	solved a problem in warm reboot in a system where Interlaken is enabled. problem was reported on 6.4.7 but cannot be reproduced in 6.4.8
SDK-82746		88670_A0	The relevant functions were changed to support this feature. Feature should work as expected after this fix.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82770		All	When OAM error packets are trapped at the egress, trapped packets may include 2 sets of system headers. Such packets can be recognized by the trap code and the `PPH.FWD_HEADER_OFFSET` field. If the former is set to bcmRxTrapOamLevel or bcmRxTrapOamPassive and the latter is at 0 then an additional set of system headers follows. When such a packet is recognized by the RX parser, the `bcm_pkt_t` struct trapped to the CPU will have the field `dnx_header_count` equal to 4 and the headers in the `dnx_header_stack` will be the external FTMH, PPH followed by the internal FTMH, PPH. This is set when `custom_feature_oam_additional_FTMH_on_error_packets` is set to 1.
SDK-82772		`56150_A0`	Fixed CLI command 'phy eee ge' display issue. The status variable should be initialized for each round of EEE status retrieving.
SDK-82785 SDK-82783 SDK-80986		`56850_A0 56860_A0 56960_A0`	In previous releases, there was no API to configure `IPV6_SIP_AND_DIP_LINK_LOCAL_DO_NOT_DROP` field of register `ING_MISC_CONFIG2.` This has been addressed by providing bcmSwitchV6L3SrcDstLocalLinkDropCancel.
SDK-82795		`56640_A0`	In previous releases, the ESM recovery thread always recovered the external Tcams on unit 0, even though it was a multi-units system. In this release, the ESM recovery thread can work correctly.
SDK-82800		All	Fixed WRED configuration for VOQs and VSQs.
SDK-82812		`88670_A0`	In case of port match AC which is the SEM result was not calculated correctly out of the lif-id.
SDK-82815		`88670_A0`	MPLS PORT: Creation of MPLS PORT protected with only FEC caused failure.
SDK-82819		`56340_A0 56340M_A0`	In previous releases, QSGMII ports affected other ports' link status in SGMII mode. in this release, this issue was fixed.
SDK-82832		`88670_A0`	When using external TCAM, the application initialization command used to support only a single KBP device connected to the BCM88670 device. The "kbp `init_appl"` command has been expanded to include an option of 2 KBP devices connected to a single BCM88670 device (KBP device per BCM88670 core). A new argument was added to the command: `"second_elk_enable".` When set, the ROP test which is part of the application init, will be performed on both KBP devices. The argument's default value is '0'.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82840		All	Issue and Fix: For multiple choice elements marked with PQ_MULTI(which are dest_type and reason) during parse_table_add, we need to set a valid value in the default, since this is used as an index into the multiple choice string list. Since we were using -1, it caused a segmentation fault when interactive mode tries to fetch the default string to be printed. Fixed it to zero (index) which would resolve to NULL interface value for desttype and reason = invalid. Also reason = 0 is invalid. So only when reason is not equal to zero (invalid), we need to send the reasons flag set to KNET module. Fixed the check.
SDK-82842		All	Ports with id bigger than 192 can drop packets at ingress due to reassembly errors in case there are more than 192 ports in the system. Fixed
SDK-82850		88670_A0 88670_B0	ILKN and Ethernet port on adjuscent quads can operate on different cores.
SDK-82851		88660_A0	OAM: The API bcm_oam_endpoint_action_set() failed to update the destination when it was set to the recycle port (for LBM response).
SDK-82865		56960_B0 56960_A0	Fixed a bug in phy control command to allow preemphasis set properly for tsce.
SDK-82867		88670_A0 88950_a0	falcon_tsc_uc_lane_config_st structure element initialized to 0
SDK-82893		56850_A0 56850_A1 56850_A2	Extended support for FP qualifier bcmFieldQualifyDstMulticastGroup and FP action bcmFieldActionRedirectIpmc for multicast VPLS group.
SDK-82914		56860_A0	We modified BCM56860 to assign queues based upon MMU port number rather than using a running number. This ends up with an inaccurate count of the number of queues used since we may be skipping some queues. Therefore we also base the count of our number of queues on the MMU port number as well.
SDK-82915		88670_A0 88670_B0	In the previous release, system port isn't set correctly for port vlan COE. In this release, this issue has been addressed by adding mask to virtual port table offset.
SDK-82928		56260_A0 56260_B0 56450_A0 56450_B0 56450_B1 56460_A0 56460_B0	Issue: When a tpid entry is deleted, its reference count is decremented. When reference count becomes 0, the entry should be freed back to the list of available tpid entries by making it 0. Instead it was wrongly made as 0x8100 (DEFAULT_TPID). Fix: Now when reference count becomes 0, the tpid entry is updated with 0 instead of 0x8100 (DEFAULT_TPID).

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-82941		56847_A1 56849_A1 56840_A0 56841_A3 56841_B0 56842_A0 56843_B0 56844_A0 56845_A2 56845_B0 56846_A0 56846_A1 All	In previous releases, feature soc_feature_led_data_offset_a0 was defined for BCM56840 device. It was used to determine the default offset of LED in RAM. But in fact this feature should not be defined for such device. It has been corrected in this release.
SDK-82966		88670_A0 All	ILKN remapping for ILKN 1/3/5, ILKN over fabric and when using SoC property ILKN_lane_map is fixed
SDK-82983		56850_A2 56860_A0	In previous releases, when bcm_l3_intf_create() was called with l3a_intf_flags=BCM_L3_INTF_INNER_VLAN_ADD only then bcm_l3_intf_get() would return incorrect l3i_intf_flags value for this interface. In this release the issue has been fixed.
SDK-83029		All	no risk in applying this JIRA.
SDK-83042 SDK-74301		56850_A0 56850_A1 56850_A2 56860_A0	In previous releases, there was no way to configure VNTAG/ETAG information on an l3 egress object for IFP usage. In this release, this has been fixed by extending the structure bcm_l3_egress_t.
SDK-83068		88660_A0	DNX - vlan: The API bcm_vlan_port_create was printing error messages to screen when it was called WITH_ID. These prints will no longer happen.
SDK-83069		56960_B0 56960_A0	In the previous SDK release for BCM56960, configuring/retrieving scheduling policy using bcm_cosq_gport_sched_get/ bcm_cosq_gport_sched_set APIs was not supported when the given gport is port gport and cosq 8 or above. With this fix, scheduling policy can be configured/retrieved using bcm_cosq_gport_sched_get/ bcm_cosq_gport_sched_set APIs with gport of port and cosq 8 or above.
SDK-83070		56540_A0 56540_B0	When configuring LED CPU0/CPU1 for BCM56640/BCM56540 chip in soc_ledproc_config, as BCM56640/ BCM56540 chips support soc_feature_cmicm, it will use the default case handling only LED CPU0, which results in LED CPU1 starting failed. Now,add BCM56640/ BCM56540 case in soc_feature_cmicm check statement so that it can support starting LED CPU0/CPU1 successfully for BCM56640/ BCM56540 chips which support cmicm feature.
SDK-83088		56960_B0	added the fix to clear the AN restart bit after AN is enabled
SDK-83126		All	In previous releases, the field VRF_ID in SOURCE_TRUNK_MAP_TABLEm might be cleared wrongly in some special situation. In this release, the condition of clearing that field has been restricted.

Table 73:

Number	***CSP #***	***Chips***	***Release Notes For 6.4.8***
SDK-83164		56860_A0	In this release, the bcm_port_ifg_set() API correctly programs the user specified interframe gap value to the hardware register
SDK-83210		56260_A0	The division operation is done in the SDK to get the DM-avg from accumulated DM value and number of sampled packets.
SDK-83222		88670_A0 All 88670_B0	Fixed egress-receive-queue/tm-port counting using dynamic-APIs only. Using dynamic-APIs did not configure the counter engine as enabled.
SDK-83224		88660_A0	OAM Up-MEP: Bugs related to resource leakage were fixed: When re-configuring up-MEPs and actions related to them, the SDK no longer runs unexpectedly out of resources.
SDK-83250		56640_A0 56640_A1 56640_B0	The member count of the table IM_MTP_INDEX is equal to 0 means the trunk port has one member when a mirroring port is a trunk port on BCM56640. In earlier releases the software did not process this situation after warmboot on BCM56640. In this release, added to check this situation on BCM56640.
SDK-83254		56860_A0	bcmPortControlPFCClasses was not implemented. Added support for PFC control bcmPortControlPFCClasses.
SDK-83255		56860_A0	In previous releases, the reference count in VLAN action profile table entry was not updated correctly in below two sequences: 1) When add a port to a trunk which is already part of the VPN, and use api bcm_vlan_port_default_action_set to align port property; 2) When remove a port from a trunk which is already part of the VPN, and use api bcm_vlan_port_default_action_delete to align port property. This has been fixed in this release.
SDK-83275		88660_A0	port diag shell command was broken for fabric port. The issue was fixed
SDK-83282		88650_B0	Dynamically added port with1 or 2 priorities drops traffic due to sch TCG shaper misconfiguration. Fixed
SDK-83296		88670_A0	ILKN<>KBP connection using ILKN #2-5 BCM8867X/8837X can be connected to 2 KBP devices via 2 ILKN interfaces. User can choose these interfaces using ":kbp" suffix at ILKN definition SOC property. e.g. "ucode_port_31.BCM88675=ILKN0:core_0.31:kbp". The problem described here is that ILKN link up not achieved while using ILKN #2-5.
SDK-83315		All	bcm_port_timesync_config_get() fixed to return the correct flags.
SDK-83330		56850_A0 56850_A1 56850_A2	In previous releases, the port may be not coming up after disable/re-enable in linkscan software accelerate mode. That was triggered by a race condition. In this release, this issue has been fixed.
SDK-83350		56960_B0 56960_A0	Fault handling is not added into link scan for CL MAC before this fix

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-83382		All	Fix the system crash issue with multiple devices with KNET enabled doing the `soc_shuttdown` by avoid doing the knet cmd request after clearing the knet related parameters in the shutdown procedure. In the previous release, the system crashed when multiple devices do the warmboot with the KNET enabled.
SDK-83389		`56960_A0`	Added 'checking for validity of index' to error injection routine.
SDK-83390		`88670_A0`	Fixed enabling counter engine when using dynamic APIs only.
SDK-83393		`88670_A0`	There was an inconsistency in trap initialization sequence. No functional impact was detected. The issue was fixed.
SDK-83400		`56960_B0 56960_A0`	In BCM56960, the `PERQ_DROP_PKT/BYTE` rate was not displayed as a part of "show c" output for the cpu port, when the packets enter from pipe1 or pipe 2 and egress on CPU Port. With this fix, the `PERQ_DROP_PKT/BYTE` CPU port rate is displayed correctly using "show c" when the packets enter from any pipe1 or pipe 2 and egress on CPU Port.
SDK-83438		`88660_A0 88670_A0`	MPLS PORT: In case of using optimized model, adding and deleting many thousands of mpls ports might fail because of push profile management
SDK-83473		`56640_A0`	In previous releases, `bitclock_hz` and `heartbeat_hz` are received as 0, which is an invalid configuration resulting in an assert. In this release, error checking has been introduced to prevent this configuration.
SDK-83475		All	In earlier releases port class was wrong to converted to gport id. In this release, deleted this conversion.
SDK-83561		`88650_A0 88660_A0`	In L3, BCM88660 and below IPV6 entries are now explicitly converted from `uint8` to `uint32` and vice versa, avoiding potential endianness issues. This change may affect existing workarounds in place that were meant to circumvent the issue.
SDK-83584		`56850_A2`	Updated warmboot code to recover Priority for all entry parts in EFP and VFP
SDK-83597		`88950_a0`	Fix bug in BCM88950 Warmboot causing warminit to fail between 6.4.7 and older versions.
SDK-83660		`88670_A0`	In L3, IPv4 MC `bcm_ipmc_find` and `bcm_ipmc_remove` now support vrf!=0.
SDK-83670		`56850_A0`	In previous releases, the API `bcm_l3_egress_get` and `bcm_l3_egress_traverse` would crash if there was an egress object associated with a niv port created by `bcm_niv_egress_add.` This has been addressed.
SDK-83674		`56860_A1 56860_A0`	Fixed to support for HardWare linkscan with BCM82792 PHY.

Table 73:

Number	CSP #	Chips	Release Notes For 6.4.8
SDK-83726		88660_A0 88670_A0	OAM: when registering an OAM event (using bcm_oam_event_register()), if an unsupported event was registered BCM_E_INTERNAL was returned instead of BCM_E_PARAM.
SDK-83744		All	This release addresses an issue with incorrect reference count accounting for entries in the PROTCOL PKT CTRL profile table, seen in prior releases.
SDK-83766		56860_A1	Interface change should apply to external PHY only. Fixed the code.
SDK-84056		All	Fixed an done status.
SDK-84144		88670_A0	In EVPN, IML label entries requires two different global lifs. Both include the same lif information but one handles the case of ESI label as well. bcm_mpls_tunnel_switch_get has been adjusted to EVPN requirements.
SDK-84181 SDK-85051		88660_A0 88670_A0	In Field Processor, when using external TCAM, entries removal was not handled correctly in some cases. This resulted in mismanaged software state and failure to remove the entries from the hardware. The external TCAM entries removal in Field Processor is now correct and properly managed.
SDK-84237		All	In Metering, in BCM88670, tables that are used to calculate the packet size were not initialized properly for core-1, this is now fixed.
SDK-84277		88670_A0	PWE traffic has been enabled with EVPN setting.
SDK-84316		88670_A0	Add support to bcm_port_control_set(bcmPortControlLinkFaultLocal / bcmPortControlLinkFaultRemote): Clear LSS Fault indication.
SDK-84365		56260_A0 56260_B0 56460	Problem: FW got stuck while processing lot of events generated due to packet processing Solution: Prevented filling the event queue with same endpoint event till ACK has been received from SDK. Instead updated the event mask with new event received by FW.
SDK-84424		88670_A0	An assertion for the out vsi range has been inserted into bcm_mpls_tunnel_initiator_create.
SDK-84515		56860_A0 56860_A1	The fix maps the values to the LED remap register based on the LED chain connection mentioned in the Theory of Operation which basically provides the port status ordering into the CMIC block.
SDK-84520		56860_A0	Fixed bcm_port_phy_control_set command to work with legacy phy.
SDK-84761		88660_A0 88670_A0	OAM/BFD: When the soc property counter_engine_source is not set bcm_oam/bfd_init() returns an error.
SDK-84976		88670_A0	Fixed enabling counter engine when using dynamic APIs only.

Section 10: Unresolved Issues for 6.4.8

The following issues are unresolved in version 6.4.8 of the SDK.

Table 74:

Number	CSP #	Chips	Release Notes
SDK-33686	389108	56634_A0	If a MiM virtual port has statistics enabled for it and if such MiM port is replaced using BCM API `bcm_mim_port_add( )` along with flag `BCM_MIM_PORT_REPLACE` then the statistics of that MiM port might be lost.
SDK-45075		88650_A0	When an interrupt occurs on different blocks of the same type (e.g. multiple FMAC blocks), the count will be accumulated in the same counter. For example RX-LOS interrupt may occur on different FMAC blocks, and counted as same recurring event, although it is in fact a different interrupt. This may affect corrective action in case it is different for a recurring event, in the case if recurring-threshold for this event is crossed.
SDK-46556	621213	88650_A0 88650_B0 88650_B1	`bcm_l2_cache_delete()` API does not delete `general_trap` entry configuration in HW
SDK-51978		88650_A0	In a device with channelized CPU ports, where some of the CPU ports are Higig and some not, the WB wont preserve the Higig indication correctly.
SDK-54219	744517	88650_B1	In BCM88650 B1, VxLAN/L2GRE packet size of 236B-299B will be dropped by EPNI if the packet needs to be terminated. Workaround is introduced in FP CINT: `cint_field_ingress_large_termination.c`
SDK-54488		88650_A0	In Field Processor, when creating a cascaded field group, a failure might occur in key allocation in case the key ID is already determined. To be investigated.
SDK-56851	767727	88650_B1 88660_A0	The guarantee hardware mechanism only guaranteed the resource (BDs). The number of BDs used is not smaller than the used buffers, but there are three buffer types and not each buffer type can be used for any packet.
SDK-66650		88660_A0	OAM: calls to `bcm_oam_endpoint_action_set()` affect all other non-accelerated MEPs defined on the MEPs LIF. Additionally, the following (incorrect) behavior has been observed: 1. Create MIP on a LIF 2. Create an accelerated Up endpoint on same LIF 3. Call `bcm_oam_endpoint_action_set()` for the MIP 4. Destroy the MIP 5. Create a new MIP on the LIF. After step 5 one half of the MIP will behave with the default configurations and the other half will behave with the configurations defined on the first MIP.
SDK-67436		88650_B1 88650_B0 88650_A0 88660_A0	Accessing Field Processor APIs with different configuration threads may fail due to wrong lock coding.
SDK-69747		88660_A0 88670_A0	For some cases, If the application initialization in `bcm_attach(),` fails after `bcm_dpp_counter_init().` The counter processor background thread is not disabled, Which can cause a segmentation fault. until there is a fix, in case of an error call: `bcm_dpp_counter_background_collection_enable_set(unit,` FALSE); to disable the background thread.

Table 74:

Number	CSP #	Chips	Release Notes
SDK-70015		88670_A0	The following soc properties are not supported in BCM88670: default_logical_interface_mpls_termination_explicit_null mpls_termination_pwe_vccv_type4_mode mpls_termination_explicit_null_label_lookup_mode mpls_entropy_label_indicator_enable fast_reroute_labels_enable Should return error but currently ignore.
SDK-72407		88670_A0	OAMP transmission fails to account for User-Header settings. The result is that Down MEP and BFD transmission will fail when the User-Header soc properties are set.
SDK-75191		88670_A0	BCM88670: Per port packet header compensation for Meter/Counter Processor is applied using bcmSwitchMeterAdjust switch, this call will fail on BCM88670.
SDK-76703		88660_A0 88670_A0	BCM88670: IPv6 entries are stored in the KAPS, where a FEC result is the only available action. When adding entries with FEC but without info->encap_id, an error occurs.
SDK-77553		88670_A0	BCM88670: using "diag oam ak" to diagnose MIP behavior may produce wrong results
SDK-78894		88650_B1 88660_A0	Bi-directional 1:1 Protection: The suggested configuration for Bi-directional 1:1 Protection is a Selector Bridge (FEC Protection) for the In coming traffic and a Selective Selector (Ingress Protection) for the out coming traffic. In order to achieve this scheme, bcm_vlan_port_create() should allow 1:1 and 1+1 configuration using the relevant fields failover_id & ingress_failover_id, for the same protection pair. The configuration of both types of protection on the same pair, currently fails due to validations.
SDK-80274		88670_A0	MPLS PORT: api bcm_mpls_port_delete_all does not delete ports configured with EGRESS_ONLY flag.
SDK-80349		88670_A0	BFD The field int_pri in bcm_bfd_endpoint_create() is ignored. The actual TC/DP on the ITMH of transmitted packets from the OAMP will be 0.
SDK-80490		56960_A0	Setup up IPMC with the BCM_IPMC_RANGE_PIM_BIDIR flag. We see a packet storm after sending just one packet with ports in loopback mode. This is observed only in balanced latency mode: switch_bypass_mode=1
SDK-80824		88670_A0	BCM88670: Mac-in-Mac Multicast fails to send the second pass copies towards the access ports from core1.
SDK-82183		88660_A0	Creating a trill port by bcm_trill_port_add, _bcm_dpp_trill_alloc_port_id will allocate an internal port id for the trill port. But if the internal port id is same as the trill name, the nick name cant be added to hash table.
SDK-83215		88670_A0	Egress Protection: Configuration of new AC, MPLS Tunnel & PWE objects may use Egress Protection when calling to bcm_vlan_port_create(), bcm_mpls_tunnel_initiator_create() and bcm_mpls_port_add() respectively. After definition, a REPLACE operation should be supported in all cases, for the Egress protection field egress_failover_id, in order to change the failover_id for an object that was defined as part of an Egress Protection scheme. The REPLACE operation isn't supported and in the case of MPLS Tunnel, an attempt to change egress_failover_id may cause an assertion.
SDK-83474		88660_A0	BFDoIPv6 implementation don't support for remote destination per endpoint for both multi-hop and single-hop.

Table 74:

Number	CSP #	Chips	Release Notes
SDK-83768		56860_A0	On 56860 devices, the BCM API bcm_port_speed_set() is not properly setting the new speed across the entire switch device logic. This may result in unexpected traffic rate. A workaround is to use the new FlexPort API bcm_port_resource_multi_set() for a port speed change.
SDK-83878		88660_A0 88670_A0	Port Extender: Egress VLAN Translation isn't functional for Port Extender packets. Packets can only be transmitted without an E-TAG with unchanged inner VLAN tag (whenever exists).
SDK-83946		88670_A0	88670 LEDs does not work in QSGMII mode.
SDK-83948		88670_A0	88670: LEDs of four 10G ports in Negev chassis front panel are not supported yet (only 40G, 100G ports LEDs are currently working)
SDK-84265		88670_A0	bcm shell command: diag alloc fec direct=1 continues to iterate over allocated FECs even after last requested FEC ("to" parameter) has been reached.
SDK-84346		56860_A0	In SDK-6.4.8 release application, the property configuration, "portmap_<port>=<physical port>:<speed>:i", for flex port will cause initialization failed when apply on external PHY. When this issue happen, "Parameter is out of range" warning will be reported from phymod_dispatch_type_t_validate function. Here is a configuration example: portmap_1=1:40 portmap_2=2:10:i portmap_3=3:10:i portmap_4=4:10:i port_phy_addr_xe0=0 port_phy_addr_xe1=1 port_phy_addr_xe2=2 port_phy_addr_xe3=3
SDK-84613		56860_A0	During flexing from 1x100 to 10x10, new ports are created. For each port on which ipmc is configured, we need to create an internal database. In the present implementation of ipmc per-port initialization in BCM56860's Flexport code, the ipmc database is created only for the existing port, and not for the newly added ports (which come in to existence during flexing operation). Also, programming of a few per-port ipmc registers and memories is not happening. When bcm_tr3_ipmc_egress_intf_get is called for a newly created port, segmentation violation happens, since the database for the new port does not exist (null pointer exception).
SDK-84637		88660_A0	Fixed mutex deadlock in interrupt handling function. Issue is reproduce only in multi threaded applications.
SDK-84840		88670_A0	RIFs higher than 4K do not handle well in bcm_l3_egress_create
SDK-85029		88670_A0	When the learn mode is BCM_L2_INGRESS_CENT| BCM_L2_LEARN_CPU there are cases where FECs are not learned correctly, in particular in VPLS models.

Section 11: Device and Platform Support

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

SWITCH DEVICES

Table 75: 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	

Table 75: Switch Devices

Family	Devices	Description
	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	
	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	

Table 75: Switch Devices

Family	Devices	Description
	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
	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	

Table 75: Switch Devices

Family	*Devices*	*Description*
	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
	BCM53724 B0	
	BCM53726 A0	Managed 24-port L2 Switch with Integrated CPU
	BCM53726 B0	
	BCM5675 A1	
	BCM5676 A0	4-Port, 96-Gbps Switch Fabric
	BCM5676 A1	
BCM56010	BCM56014 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56014 A1	
	BCM56014 A2	
	BCM56018 A0	48-Port Integrated Multilayer Switch and CPU
	BCM56018 A1	
	BCM56018 A2	
	BCM56018 A1	48-Port Integrated Multilayer Switch and CPU
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
BCM56100	BCM56100 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch
	BCM56100 A1	
	BCM56101 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56101 A1	
	BCM56102 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig Ports

Table 75: Switch Devices

Family	Devices	Description
	BCM56102 A1	
	BCM56105 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch
	BCM56105 A1	
	BCM56106 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56106 A1	
	BCM56107 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig Ports
	BCM56107 A1	
BCM56110	BCM56110 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch
	BCM56111 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56112 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig Ports
	BCM56115 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch
	BCM56116 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with One 10-Gigabit Ethernet/HiGig Port
	BCM56117 A0	24-Port Fast Ethernet and 2-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig Ports
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	
BCM56140	BCM56140 A0	24-Port Gigabit Ethernet/6-Port SGMII GbE Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56142 A0	24-Port Gigabit Ethernet Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56143 A0	24-Port Gigabit Ethernet Multilayer switch with combination of two/four 1G/2.5/HiGig2 Uplink Ports
	BCM56144 A0	16-Port Gigabit Ethernet Multilayer switch with four 1G/2.5HG Uplink Ports
	BCM56146 A0	24-Port Fast-Ethernet Multilayer switch with four 2.5HG Uplink Ports
	BCM56147 A0	24-Port Fast-Ethernet Multilayer switch with combination of one/two/four 1G/2.5G/10/12/13HG Uplink Ports
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

Table 75: Switch Devices

Family	Devices	Description
	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)
BCM56210	BCM56212 A0	
	BCM56212 A1	
	BCM56212 A2	
	BCM56213 A0	
	BCM56213 A1	
	BCM56213 A2	
	BCM56214 A0	BCM56214 Integrated Multilayer Switch and CPU
	BCM56214 A1	
	BCM56214 A2	
	BCM56215 A0	
	BCM56215 A1	
	BCM56215 A2	
	BCM56216 A0	BCM56216 Integrated Multilayer Switch and CPU
	BCM56216 A1	
	BCM56216 A2	
	BCM56217 A0	
	BCM56217 A1	
	BCM56217 A2	
	BCM56218 A0	BCM56218 Integrated Multilayer Switch and CPU
	BCM56218 A1	
	BCM56218 A2	
	BCM56219 A0	BCM56219 Integrated Multilayer Switch and CPU
	BCM56219 A1	
	BCM56219 A2	
BCM56220	BCM56224 A0	24 GbE + 4 x 1 Gb/2.5 Gb, L3/L2+
	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

Table 75: Switch Devices

Family	Devices	Description
BCM56300	BCM56300 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56300 A1	
	BCM56300 B0	
	BCM56300 B1	
	BCM56301 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56301 A1	
	BCM56301 B0	
	BCM56301 B1	
	BCM56302 A0	24-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig+ Ports
	BCM56302 A1	
	BCM56302 B0	
	BCM56302 B1	
	BCM56303 A0	24-Port Gigabit Ethernet Multilayer Switch with Three 10 Gigabit Ethernet/HiGig+ Ports
	BCM56303 A1	
	BCM56303 B0	
	BCM56303 B1	
	BCM56304 A0	24-Port Gigabit Ethernet Multilayer Switch with Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56304 A1	
	BCM56304 B0	
	BCM56304 B1	
	BCM56305 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56305 A1	
	BCM56305 B0	
	BCM56305 B1	
	BCM56306 A0	16 Port Gigabit Ethernet Switch
	BCM56306 A1	
	BCM56306 B0	
	BCM56306 B1	
	BCM56307 A0	24-Port GE L2 Switch with Two 10 GE/HiGig+ Ports
	BCM56307 A1	
	BCM56307 B0	
	BCM56307 B1	
	BCM56308 A0	24-Port GE L2 Switch with Three 10 GE/HiGig+ Ports
	BCM56308 A1	
	BCM56308 B0	
	BCM56308 B1	
	BCM56309 A0	24-Port GE L2 Switch with Four 10 GE/HiGig+ Ports
	BCM56309 A1	
	BCM56309 B0	

Table 75: Switch Devices

Family	Devices	Description
	BCM56309 B1	
BCM56310	BCM56310 A0	BCM56310 Series 24-Port GbE Multilayer Switch with Four 10-GbE/HiGig+ Uplink Ports
	BCM56311 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56312 A0	24-Port Gigabit Ethernet Multilayer Switch with Two 10-Gigabit Ethernet/HiGig+ Ports
	BCM56313 A0	24-Port Gigabit Ethernet Multilayer Switch with Three 10-Gigabit Ethernet/ HiGig+ Ports
	BCM56314 A0	24-Port Gigabit Ethernet Multilayer Switch with Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56315 A0	BCM56310 Series 24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Uplink Ports
	BCM56316 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56317 A0	24-Port Gigabit Ethernet Layer 2 Switch with Two 10-Gigabit Ethernet/HiGig+ Ports
	BCM56318 A0	24-Port Gigabit Ethernet Layer 2 Switch with Three 10-Gigabit Ethernet/HiGig+ Ports
	BCM56319 A0	24-Port Gigabit Ethernet Layer 2 Switch with Four 10-Gigabit Ethernet/HiGig+ Ports
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

Table 75: Switch Devices

Family	*Devices*	*Description*
BCM56340	BCM56548 A0	7xF.QSGMII + 3xF.HG[42] + 1GE
BCM56440	BCM56440 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	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	

Table 75: Switch Devices

Family	*Devices*	*Description*
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports
	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	

Table 75: Switch Devices

Family	Devices	Description
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch
	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	

Table 75: Switch Devices

Family	Devices	Description
	BCM56624 B2	
	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

Table 75: Switch Devices

Family	*Devices*	*Description*
BCM56640	BCM56546K	28-port GE switch + 4x10GE + 4xHG[42] / 40GE
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	
BCM56800	BCM56800 A0	20-Port 10-Gigabit Ethernet Multilayer Switch
	BCM56801 A0	10-Port 10-Gigabit Ethernet and 8-Port HiGig2/10GbE Multilayer Switch
	BCM56802 A0	16-Port 10-GbE/HiGig2 Multilayer Switch
	BCM56803 A0	12 Port 10GE/HiGig2 Multilayer Switch
BCM56820	BCM56820 A0	24 x 10-GbE + 4 x 1-GbE Multilayer Ethernet Switch
	BCM56820 B0	

Table 75: Switch Devices

Family	Devices	Description
	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	
	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	

Table 75: Switch Devices

Family	Devices	Description
	BCM56845 A2	
	BCM56845 A3	
	BCM56845 A4	
	BCM56845 B0	
	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
BCM56860	BCM56860 A0	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switc
	BCM56867 A0	104x 10GbE/32x 40GbE/8x 100GbE Multilayer Switch
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	

Table 75: Switch Devices

Family	Devices	Description
	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
	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	

Table 76: SER Supported Devices

Family	Devices
BCM53400/BCM56060	All SKUs
BCM56150	All SKUs
BCM56260	All SKUs
BCM56440	All SKUs
BCM56445	All SKUs
BCM56450	All SKUs
BCM56620	All SKUs
BCM56840	All SKUs
BCM56850	All SKUs
BCM56860	All SKUs
BCM56960	All SKUs

PHYS

Table 77: PHYs

Device	*Driver Family*	*Description*
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

Table 77: PHYs

Device	*Driver Family*	*Description*
BCM54340 B0	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
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

Table 77: PHYs

Device	Driver Family	Description
BCM8073 A0	8072	Dual-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version d502.
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	12port Gallardo28 supporting 12x10G, 3x40G, 1x100G. 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

Table 77: PHYs

Device	Driver Family	Description
BCM84823 B0	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84823 B1	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
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 with CAUI4 to CAUI4 / Dual 40GbE XLPPI to XLAUI/ Octal 10GbE SFI to XFI port. Firmware version D00E Driver supports 100G CR4, 40G SR4/LR4/CR4 and 10G SR/LR Functionality Supported: PHY Reset(Hard/soft), Link status, Speed and Interface configuration, port enable/disable, AN, Tx Training, remote/digital loopback, Polarity inversion, PRBS, Eyescan, DSC, PLL sequencer restart, Phy dump, FEC and Rx PMD lock. Driver Limitation : 1. Chip reset issues reset to all the ports that shares same PHY-ID(MDIO ADDR). Dual 100GbE CAUI4-to-CAUI4 PHY/Octal 25G Retimer.

Table 77: PHYs

Device	Driver Family	Description
BCM82764 A2	82764	BCM82764 Dual 40GbE channels with Dual 40GbE XLPPI to XLAUI / Quad 40GbE PMA reverse bit MUX to convert 2x20G to XLPPI(4x10G) /CAUI4 to CAUI4 / Octal 10GbE SFI to XFI port. Firmware version D008. Driver supports 40G SR4/LR4/CR4/KR4 and 10G SR/LR/KR/CR Functionality Supported: Link status, Speed and Interface configuration, port enable/disable, Polarity inversion, Power set/get, PRBS, Eyescan, PLL sequencer restart, Phy dump, Rx PMD lock. remote/digital loopback set/get, PHY Reset(Hard/soft), DSC, Phy dump, Tx lane control set/get Tx set/Get, Rx set/Get and Tx Training Driver Limitation : 1. Speed change is only supported through config.bcm. Speed switch through flex port is not supported in this release. 2. Chip reset issues reset to all the ports those shares same PHY-ID(MDIO ADDR). 3. Tx Training may not finish on some 10G lanes(Line side). 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

Section 12: Compatibility

BROADCOM TASK ENGINES (BTE) FIRMWARE COMPATIBILITY MATRIX

The following table shows compatibility between different versions of SDK and Firmware releases.

Table 78:

	SDK-6.4.8	*SDK-6.4.7*	*SDK-6.4.6*	*SDK-6.4.5*	*SDK-6.4.4*	*SDK-6.4.3*	*SDK-6.4.2*	*SDK-6.4.1*	*SDK-6.4.0*
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					
4.2.1									

Table 78:

	SDK-6.4.8	SDK-6.4.7	SDK-6.4.6	SDK-6.4.5	SDK-6.4.4	SDK-6.4.3	SDK-6.4.2	SDK-6.4.1	SDK-6.4.0
4.2.0				BCM53400 BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM56260	BCM53400 BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM56260 BCM88650				
4.1.0						BCM53400* BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM88650*	BCM56150 BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850
4.0.5				BCM88030	BCM88030	BCM56150 BCM56440 BCM56450 BCM56640 BCM56850* BCM88030	BCM56150 BCM56440 BCM56450 BCM56640 BCM56850 BCM88030	BCM56440 BCM56450 BCM56640 BCM56850 BCM88030	BCM56440 BCM56450 BCM56640 BCM56850 BCM88030
4.0.4						BCM56150 BCM56440 BCM56450 BCM56640 BCM56850*	BCM56150 BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850

Table 78:

	SDK-6.4.8	SDK-6.4.7	SDK-6.4.6	SDK-6.4.5	SDK-6.4.4	SDK-6.4.3	SDK-6.4.2	SDK-6.4.1	SDK-6.4.0
4.0.3						BCM56150 BCM56440 BCM56450 BCM56640 BCM56850*	BCM56150 BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850
4.0.2						BCM56150 BCM56440 BCM56450 BCM56640 BCM56850*	BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850	BCM56440 BCM56450 BCM56640 BCM56850
4.0.1						BCM56150 BCM56440 BCM56450** BCM56640 BCM56850*	BCM56440 BCM56450** BCM56640 BCM56850	BCM56440 BCM56450** BCM56640 BCM56850	BCM56440 BCM56450** BCM56640 BCM56850
4.0.0						BCM56150 BCM56440 BCM56450** BCM56640 BCM56850*	BCM56440 BCM56450** BCM56640 BCM56850	BCM56440 BCM56450** BCM56640 BCM56850	BCM56440 BCM56450** BCM56640 BCM56850

* Patch Required

** Firmware prior to 4.0.2 is not compatible with B0 chip revision

BMACSEC SDK COMPATIBILITY MATRIX

Table 79: BMACSEC SDK Compatibility Matrix

Switch SDK Release	BMACSEC SDK Release
6.4.0	4.5
6.3.5	4.6
6.3.6	4.7
6.3.7	4.8

Table 79: BMACSEC SDK Compatibility Matrix

Switch SDK Release	BMACSEC SDK Release
6.4.1	4.8
6.3.8	4.9
6.3.9	4.10
6.4.2	4.10
6.4.3	4.11
6.4.4	4.12
6.4.5	4.13
6.4.6	4.14
6.4.7	4.14
6.4.8	4.15

PHY FIRMWARE COMPATIBILITY MATRIX

The following table identifies changes in PHY firmware for newer PHY devices.

Table 80: PHY Firmware Compatibility Matrix

PHY Core	6.4.6 Firmware Versions	6.4.7 Firmware Versions	6.4.8 Firmware Versions
BCM84861	00.00.09	00.00.10	00.00.10
BCM84864	00.00.09	00.00.10	00.00.10
BCM84868	00.00.09	00.00.10	00.00.10
BCM84858	01.02.02	01.02.02	01.02.10
BCM84856	01.02.02	01.02.02	01.02.10
BCM84744	0xD105(A0)/0x0128(B0/C0)	0xD105(A0)/ 0x0128 (B0/C0)	0x0132 (B0/C0)
BCM84757	0xD105(A0)/0x0128(B0/C0)	0xD105(A0)/ 0x0128 (B0/C0)	0x0132 (B0/C0)
BCM84328	R027	R027	R027
BCM82322/28 B1	0xD	0xF	0xF
BCM82780	0x22	0x22	0x23
BCM82752	0x22	0x22	0x23
BCM82758	0x22	0x22	0x23
BCM82764	D00B	D00F	D011
BCM82790	D00B	D00F	D011
BCM82792	D00B	D00F	D011
BCM82796	D00B	D00F	D011
BCM82381	D00E	D011	D011
BCM82209	D00E	D011	D011
BCM82073	D00E	D011	D011
Eagle	D10E_04	D10F_03	D10F_03
Falcon	D10A_02	D10A_02	D10A_06

Section 13: 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 81: 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 148)
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c	See (ED Editor License terms and conditions) (page 150)
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]	See (CINT parser license terms and conditions) (page 150)
CES Driver	BATM Advanced Communications Ltd	src/soc/ces/nemo_driver/*.[ch], src/soc/ces/clsbuilder/*.[ch]	See (Circuit Emulation Service (CES) Driver terms and conditions) (page 151)
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 152)
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]	See (APIMODE parser license terms and conditions) (page 152)
VxWorks	Wind River Systems, Inc.	systems/vxworks	See (Wind River Systems license terms and conditions) (page 152)

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.  */
```

CIRCUIT EMULATION SERVICE (CES) DRIVER TERMS AND

CONDITIONS

The Circuit Emulation Services (CES) driver code provided herewith is provided by BATM Advanced Communications Ltd (BATM) and is subject to licensing agreement between BATM and Broadcom Corporation.

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 150) for the Bison licence.

WIND RIVER SYSTEMS LICENSE TERMS AND CONDITIONS

See WRS_LICENSE.pdf contained in each systems/vxworks subdirectory.