Software Development Kit
Release Notes
SDK 6.3.0

March 7, 2013

Broadcom
Network Switching

Section 1: About This Document

These are the Release Notes for the Broadcom Network Switching Software Development Kit Release 6.3.0.

This document provides a general description of the release and its new features. It also describes the chips supported by the release, BCM/BCMX 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-PG630-R	BCM and BCMX API Reference Guide. This manual describes the theory of operations of the API and all existing BCM and BCMX 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-PG815-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: Release Media

The Software Development Kit is released as a gzipped tar file on the Broadcom Customer Support Portal, http://support.broadcom.com. The Network Switching Software Platform Guide, also available on the Customer Support Portal, provides documentation on the various components, the source directory layout, how to build the release for various platforms, and how to customize and port the software to new platforms.

Section 4: Support

Questions, feedback, and/or suggestions should be sent to your Broadcom FAE.

Section 5: Firmware Compatibility Matrix

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

BCM56440 FIRMWARE COMPATIBILITY MATRIX

Table 2:

SDK	***FW 1.0***	***FW 1.1***	***FW 1.2***	***FW 1.3***	***FW 1.5***	***FW 2.0***	***FW 2.1***	***FW 2.2***	***FW 3.0***	***FW 3.0.1***
SDK-5.10.0	Yes	No	No	No	No	No	No	No	No	No
SDK-5.10.1	No	Yes	Yes	No	No	No	No	No	No	No
SDK-5.10.2	No	Yes	Yes	No	No	No	No	No	No	No
SDK-5.10.3	No	Yes	Yes	Yes	No	No	No	No	No	No
SDK-5.10.4	No	Yes	Yes	Yes	Yes	No	No	No	No	No
SDK-5.11.0	No	Yes	Yes	Yes	No	No	Yes	No	No	No
SDK-5.11.1	No	Yes	Yes	Yes	No	No	Yes	No	No	No
SDK-6.0.0	No	No	Yes	No	No	No	No	No	No	No
SDK-6.0.1	No	No	No	Yes	No	No	No	No	No	No
SDK-6.2.0	No	No	No	No	No	Yes	No	No	No	No
SDK-6.2.1	No	No	No	No	No	No	No	Yes	No	No
SDK-6.2.3	No	No	No	No	No	No	No	Yes	Yes	No
SDK-6.2.4	No	No	No	No	No	No	No	Yes	Yes	No
SDK-6.3.0	No	No	No	No	No	No	No	No	Yes	Yes

BCM56640 FIRMWARE COMPATIBILITY MATRIX

Table 3:

SDK	*FW 2.2.0*	*FW 3.0*	*FW 3.0.1*
SDK-6.2.1	Yes	No	No
SDK-6.2.3	Yes	Yes	No
SDK-6.2.4	Yes	Yes	No
SDK-6.3.0	Yes	Yes	Yes

BCM88650 FIRMWARE COMPATIBILITY MATRIX

Table 4:

SDK	*FW 2.2.0*	*FW 3.0*	*FW 3.0.1*
SDK-6.2.1	No	No	No
SDK-6.2.3	No	No	No
SDK-6.2.4	No	No	No
SDK-6.3.0	No	Yes	Yes

Section 6: New in this Release

This section describes feature and device support that is introduced in this release.

SUMMARY OF NEW FEATURES

TRIDENT2 (BCM56850)

- Enhanced VXLAN support to include default network SVP. The default is needed for the cases where we encounter SIP/SVP miss. This enhancement needed API changes (additions) and these are documented in Section 6.4.19.
- Enhanced VXLAN feature to allow mapping of multiple services to the same VXLAN Virtual Port (tunnel)

ARAD (BCM88650)

- Fine grained Trill has been added to ARAD. This is available as preview only.
- Support for Broadcom NN12K (Netlogic KBP) has been added. This allows customers to enhance the packet processing functionality.
- Applicable for PON only - Security feature to prevent IP spoofing has been added. This allows Source MAC and Source IP to be validated as bound pairs before traffic is forwarded.
- Applicable for PON only - Added ability to add up to 3 VLANs in the egress direction. This allows ARAD to support PON applications where there is a need to translate 2 VLAN tags 2 tags (S/C) with the specified tunnel ID as the additional tag (3rd tag) (in downstream egress direction).
- Arad is now capable of supporting EVB (Edge Virtual Bridge) application which is already supported by XGS devices.
- L2GRE IP tunnel termination has 2 options for IP lookup <sip, dip> in tcam, <sip>, <dip> each separately in SEM. The latter option was not supported earlier and is now supported in 6.3.0 release
- 6.3.0 SDK adds support for Ring Protection G.8032 on Arad. New APIs have been added and are documented in the API document. This feature adds support for ingress/egress block/unblock of a G.8032 virtual ring port (i.e. a group of LIFs), Fast flush, R-APS control traffic etc.
- FCOE support has also been added to Arad with 6.3.0
- Added Broadsync timing support

HELIX4 (BCM56340)

Support for legacy features is available with this release as Limited General Availability. Customers can integrate their application to the SDK and get ready for production. Please note that 6.3.1 will add additional functionality beyond the legacy features listed below.

- L2 Features
- L3 Features
- Field Processor related features
- Mirroring
- STP
- Trunking
- IP Multicast
- QoS related features (Meters, Rate Limits etc.)
- ECMP
- Flexible table sizing
- Mac in Mac
- OAM

BCM56340 base SKU is supported in this release. Additional SKU support is planned for SDK-6.3.1.

HURRICANE2/WOLFHOUND (BCM56150/BCM53344)

Support for legacy features is available with this release as Limited General Availability. Customers can integrate their application to the SDK and get ready for production. Please note that 6.3.1 will add additional functionality beyond the legacy features listed below.

- L2 Features
- L3 Features
- Mirroring
- STP
- Trunking
- IP Multicast
- QoS related features (Meters, Rate Limits etc.)
- ACL
- IPv4/v6 LPM
- VFP, EFP
- QoS
- Stats
- Meter/Rate Limits

BCM56150 and BCM53344 SKU are supported in this release. Additional SKUs support is planned for SDK-6.3.1.

NORTHSTAR+ (BCM53022)

Support for legacy and new features is available in preview mode to enable customers to integrate SDK with their application software. In addition to legacy features APIs for some key enhancements like PPPoE are also available. Please note that 6.3.2 will be General Availability release for Northstar+.

- PPPoE frame support
- Port Port rate control/shaping Per flow Per port MIB clear Per port IVL and SVL Ingress port shaper
- VLAN,P-VLAN
- L2 Features Mirroring ACL classification via CFP Double Tagging Egress VID remarking 802.1x port authentication
- WRED Tr TCM
- Per flow PCP/DEI remarking

PETRA-B (BCM88640)

- Support for VLAN range compression has been added. This allows many flows with a range of VLANs to be processed identically in the packet processor which greatly reduces resource utilization.
- Added the outbound mirror capability based on port or port-VLAN.

KATANA2 (BCM56450)

Support for legacy features is available in preview mode to enable customers to integrate SDK with their application software. Customers are required to migrate to later SDK 6.3.X releases for fully functionality that is production ready.

- Legacy Features such as L2, VLAN, STP ACL OAM L3 QoS Metering IP Multicast Flexible Statistics
- Channelization over Ethernet
- Flexible I/O
- OLP Interface
- MPLS Protection

- OAM UP MEP

FIREBOLT4 (BCM56545 B0)

SDK support for this new device is available.

APOLLO2 (BCM56544 B0)

SDK support for this new device is available.

BCM54240 PHY (QUAD 1000/100/10G BASET)

Added support for Y.1731 compliance.

THINGS TO NOTE

This section lists items that require special attention.

HELIX4 (BCM56340)

Port 49 may not function properly when SDK is running on the integrated CPU.

TRIDENT2 (BCM56850)

VXLAN is supported in preview mode only.

SOFT ERROR RECOVERY(SER)

- L2 mac limit counters are not synchronized on parity correction for the following tables: `PORT_OR_TRUNK_MAC_COUNT VLAN_OR_VFI_MAC_COUNT L2_ENTRY_1/L2_ENTRY_2` Observed on BCM56640 and BCM56540 device family
- Parity error correction is not available for the following tables: RMEP `MA_STATE` Observed on BCM56640 and BCM56540 device family

BCM56850 SERDES CONNECTIVITY

- BCM56850 ports may experience inconsistent link state at 10G and 40G speeds when configured for Ethernet or Higig/Higig2 encapsulation and connected over copper.
- The following speeds are supported via forced modes: 40G MLD, 20G MLD, 10G XFI, 10G XAUI, 25.455G XAUI, 42G MLD HG2, 1G CL36
- Clause 72 is supported
- Clause 73 AN (without CL72 enabled): 40G, `10G_KR, 10G_KX4`
- CL73 + Parallel Detect to 10G X4/1G supported
- Clause 37, BAM AN modes that are supported: 40G `Base_X4,` 24.455G `BASE_X4,` 20G `Base_X2,` 10G `Base_X2_CX4` , HP-BAM
- PRBS/Eyescan data generation is supported in this release, both 1D and 2D and parallel eyescan generation.
- SGMII working (with external PHY , and internal PHY in forced speed mode) 1000M/100M. SGMII autoneg is not supported in internal PHY
- Flexport is not supported in this release

BCM56850 HANDLING OF MODULE_64PORTS

The number of ports per module on BCM56850 is 64 instead of 32 on other devices, the `module_64ports` config variable is the opposite than that of other devices if interpreting it as single module mode indicator.

OCCASIONAL STACK ATTACH FAILURES

There may be occasional stack attach failures due to the stack master attaching a slave device before slave programming is complete. Use the stk.soc config variable stktopomasterdelay to increase the length of time the stack master will wait before attaching a slave.

BCM8483X PHY FIRMWARE

Status of F/W version 1.66 released with SDK is preview. Check support.broadcom.com for latest available validated firmware for the BCM8483X family devices. Consult F/W release notes for known issues.

BCM8484X PHY FIRMWARE

Status of F/W version 1.03 released with SDK is preview. Check support.broadcom.com for latest available validated firmware for the BCM8484X family devices. Consult F/W release notes for known issues.

UNBALANCED MUTEX WARNING

A potential issue with unbalanced mutexes has been uncovered in previous releases of SDK and special code has been added to automatically detect that condition. While we believe that we've identified all these issues in our regression testing, there is a slight probability that you can see the following message on the console:

WARNING: Mutex `"mutex_name"` has not been unlocked before being destroyed.

Current owner is `"thread_name".`

Should you see such a warning, please, copy it verbatim and contact Broadcom Support.

BCM88650/BCM88640 CHANGE OF SEQUENCE WHEN CREATING VLAN-PORT

LIF allocation policy has changed when calling `bcm_vlan_port_create:` Prior to SDK 6.3.0, `bcm_vlan_port_t.vsi` != 0 meant allocating a LIF only at the ingress. As of 6.3.0, `bcm_vlan_port_t.vsi` != 0 means allocating both ingress and egress LIF (ISEM + ESEM entries). New flags were introduced `(BCM_VLAN_PORT_CREATE_INGRESS_ONLY/BCM_VLAN_PORT_CREATE_EGRESS_ONLY)` to provide user control on resource allocation. These flags are not supported yet. See issue in "known issues" which provides a simple workaround to support previous behvior.

BCM88650 FINE-GRAINED TRILL SUPPORT

ARAD supports fine-grained trill only as a preview, for unicast traffic. `cint_trill.c` has been updated to support FGL unicast sequence.

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 integrate the version of the SDK that provides supported status for their use on actual development or production systems.

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 5: Supported Switch Devices

Family	*Devices*	*Description*
BCM56340	BCM56340 A0	48-port GbE Multilayer Switch with 4-port 10 GbE uplinks, stacking and Integrated CPU
BCM56150	BCM56150 A0	24-port GbE Managed Switch with 4-port 10 GbE uplinks, integrated CPU and 16 copper PHYs
BCM53344	BCM53344 A0	24-port GbE Unmanaged Switch with 16 copper PHYs.
BCM56544	BCM56544 B0	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
BCM56545	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

Table 6: Preview Switch Devices

Family	*Devices*	*Description*
BCM56450	BCM56450 A0	100 Gbps Integrated Multilayer Ethernet Switch and Traffic Manager
BCM53022	BCM53022 A0	Communications Processor with Network and Cryptographic acceleration

Table 7: Preview PHYS

Device	*Driver Family*	*Description*
BCM54290_A0	BCM54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Untested Preview)
BCM54292_A0	BCM54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Untested Preview)
BCM54294_A0	BCM54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Untested Preview)

Table 8: Reference Designs

Name	*Support Status*	*Description*
BCM956340K	Supported	48x1G with on-board external phy ports (QSGMII) + 4x10G + 2xHG[21] ports, with support for two 1G/100M/10M Ethernet management ports, DDR3, NAND, QSPI, 3xUART, PCIE, JTAG, I2C and LED.
BCM956150K	Supported	16x1G on-die phy ports + 8x1G on-board external phy ports (QSGMII) + 4x10G TSC ports, with support for 100M/10M Ethernet management port, DDR3, NAND, QSPI, 2xUART, PCIE, JTAG, I2C, parallel and serial LED support.
BCM953344K	Supported	16x1G on-die phy ports + 8x1G on-board external phy ports (QSGMII) + 4x1G TSC ports, with support for 100M/10M Ethernet management port, DDR3, NAND, QSPI, 2xUART, PCIE, JTAG, I2C, parallel and serial LED support.

Table 8: Reference Designs

Name	*Support Status*	*Description*
BCM953344R	Supported	16x1G on-die phy ports + 8x1G on-board external phy ports (QSGMII) + 4x1G TSC ports, with support for DDR3, QSPI, 1xUART, JTAG & serial LED support.

SUMMARY OF BCM AND BCMX API CHANGES

This section summarizes BCM and BCMX API changes in this release. Complete documentation is available in the Network Switching Software Programmer's Guide 56XX-PG630-R (see Section 2 earlier in this document for availability).

Unless otherwise mentioned, any newly defined or changed BCM API will have equivalent changes in the BCMX APIs.

BIDIRECTIONAL FORWARDING DETECTION

New BFD Endpoint Info flags have been added.

Table 9: BFD Endpoint Info Structure Flag Definitions

Flag	***Description***
`BCM_BFD_ENDPOINT_IN_HW`	Process the endpoint in HW.
`BCM_BFD_ENDPOINT_STACK`	Must be used for creating BFD session end point that are terminated on a remote device in a multi-device/stacked system.

The following fields have been added to the BFD endpoint info structure.

```
typedef struct bcm_bfd_endpoint_info_s {
    ...

    bcm_gport_t tx_gport;               /* TX gport associated with this
                                           endpoint. */
    bcm_gport_t remote_gport;           /* Gport identifier of trapping
                                           destination: OAMP or CPU, local or
                                           remote. */
    bcm_mpls_egress_label_t egress_label; /* The MPLS outgoing label
information. */
} bcm_bfd_event_type_t;
```

The following new BFD event types have been added.

Table 10: BFD Event Types

Event type	***Description***
bcmBDFEventEndpointRemote	A remote endpoint has been reported as down (Remote Defect Indication is marked high).
bcmBDFEventEndpointRemoteUp	A remote endpoint has been reported as up (Remote Defect Indication is marked low).
bcmBDFEventEndpointRemoteEarly	A remote almost timeout has been detected on an endpoint. This event enables the user to move traffic from the Primary paths to the Protection paths before official LOC.

CLASS OF SERVICE QUEUE CONFIGURATION

New CosQ discard flags have been added.

Table 11: gport Discard Color Flags

Flag	Description
BCM_COSQ_DISCARD_IFP	Configure the default WRED profile that is used for IFP action.
BCM_COSQ_DISCARD_OUTER_CFI	CFI value of the outer tag.

New CosQ controls have been added

Table 12: CoSQ Control Type Values

Value	Description	Arg value
bcmCosqControlEgressWredDropCancel	Bypass the WRED drop at egress side.	value: 1 Enable, 0: Disable.
bcmCosqControlMulticastPriorityIngressScheduling	Multicast packets traffic class mapping	high-low strict priority.

The documentation for a field inside the "bcm_cosq_egress_multicast_config_t" structure has been updated

```
/* Multicast configuration settings */
typedef struct bcm_cosq_egress_multicast_config_s {

    ...
    int unscheduled_sp;            /* service pool ID */
} bcm_cosq_egress_multicast_config_t;
```

FORWARDING FAILOVER PROTECTION

Added a new failover flag

Table 13: Failover Create Flags

FLAG	Meaning
BCM_FAILOVER_L2_LOOKUP	flag to create a failover object for L2 lookup (fast flush)

FIELD PROCESSOR

New flags for L3 Header Data format have been added.

Table 14: L3 Header Protocol

L3 protocol	Purpose
BCM_FIELD_DATA_FORMAT_IP4_WITH_OPTIONS	IPv4 packets with options
BCM_FIELD_DATA_FORMAT_IP6_WITH_OPTIONS	IPv6 packets with options

New Field qualifiers have been added.

Table 15: Field Qualifiers

Qualifier	***Purpose***
bcmFieldQualifyForwardingHeaderOffset	Distance of the forwarding header from the start-of-packet. Unit: bytes.
bcmFieldQualifyL3SrcBind	L3 Source-Bind lookup is done

An existing forwarding type qualifier has been renamed.

Table 16: Field Forwarding Type

Type	***Purpose***
bcmFieldForwardingTypeFCoE	Fiber Channel over Ethernet forwarding

New loopback type qualifier enumerations have been added.

Table 17: Field Loopback Reason

Type
bcmFieldLoopbackTypeVxlan

New tunnel type qualifier enumerations have been added.

Table 18: Field Tunnel Type

Type	***Purpose***
bcmFieldTunnelTypeVxlan	Vxlan Tunnel Packet.

FIELD QUALIFY OLP HEADER TYPE

`bcm_field_olp_header_type_t` elements allow the user to specify the OLP header type to be used. The allowed formats are enumerated in `bcm_field_olp_header_type_t`.

Table 19: Field Qualify OLP header type Format

Formats	***Purpose***
bcmFieldOlpHeaderTypeEthCC,	OLP header is of type ETH-CC.
bcmFieldOlpHeaderTypeEthLmDm,	OLP header is of type ETH-LM/DM.
bcmFieldOlpHeaderTypeEthOthers,	OLP header is of type ETH-Others(other than CC/LM/DM).
bcmFieldOlpHeaderTypeCount	Always last, not a usable value.

FIELD QUALIFY OAM DOMAIN TYPES

`bcm_field_oam_domain_t` elements allow the user to specify the OAM domains on which the MEP lookups have to be done. The allowed formats are enumerated in `bcm_field_oam_domain_t`.

Table 20: Field Qualify OAM domain type Format

Formats	*Purpose*
bcmFieldOamDomainCVlan,	Endpoints configured on C-VLAN.
bcmFieldOamDomainSVlan,	Endpoints configured on S-VLAN.
bcmFieldOamDomainDoubleVlan,	Endpoints configured on S+C-VLAN.
bcmFieldOamDomainPbbTe,	Endpoints configured on PBB-TE.
bcmFieldOamDomainCount	Always Last. Not a usable value.

New field actions have been added.

Table 21: Field Actions

Action	*Description*	*param0*	*param1*
bcmFieldActionLearnInVPortNew	Replace the learnt packet InVPort.	n/a	n/a
bcmFieldActionOamDomain	OAM domains on which the MEP lookups are done.	First domain for lookup	second domain for lookup
bcmFieldActionOamOlpHeaderAdd	Add OLP header for packet	OLP header type	n/a
bcmFieldActionOamSessionId	OAM session Id to be used in OLP Rx header.	OAM session id	n/a
bcmFieldActionOuterVlanCfiCancel	Disable the CFI remarking of the outer vlan tag CFI bit.	n/a	n/a
bcmFieldActionInnerVlanPriCancel	Disable the PCP remarking of the inner vlan tag priority bits.	n/a	n/a
bcmFieldActionPrioIntRemark	New Internal packet priority for PCP or CFI remarking.	new priority	n/a
bcmFieldActionUseDefaultWred	Use the discard settings from default WRED configuration.	n/a	n/a

The following flag is defined to get Max width value of All Actions present.

- `BCM_FIELD_ASET_WIDTH_MAX` - Max value to indicate the Total Width of all actions

New field control has been added.

Table 22: Field Control Values

Name	*Purpose*
bcmFieldControlMeterPoolCompress	Enable/Disable meter pool compress operation during group compression

bcm_field_group_expansion_enable_set

Enable or Disable auto expansion of entries for a group.

Syntax

```
#include <bcm/field.h>
int bcm_field_group_expansion_enable_set(int unit,
                                         bcm_field_group_t group,
                                         int enable);
```

Parameters

`unit`	BCM device number
`group`	Field group Id
`enable`	Auto group expansion Enable (1) or Disable (0)

Description

`bcm_field_group_expansion_enable_set` API is used to enable auto expansion of entries in a group. Group auto expands into a new slice when number of entries created in the group exceeds the slice size in `bcm_field_entry_create` API call. If this control is enabled the group will be allowed to allocate a new slice. If this control is disabled then allocation of new slice will fail and `bcm_field_entry_create` API returns `BCM_E_RESOURCE` error.

Returns

`BCM_E_NONE`	Update of auto expansion group status successful
`BCM_E_UNAVAIL`	Feature is not available
`BCM_E_NOT_FOUND`	Group Id does not exist.

bcm_field_group_expansion_enable_get

Get auto expansion status of a group

Syntax

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

int bcm_field_group_expansion_enable_get(int unit,
                                         bcm_field_group_t group,
                                         int *enable);
```

Parameters

`unit`	BCM device number
`group`	Field group Id
`enable`	(OUT) Group's auto expansion status Enabled (1) or Disabled (0)

Description

`bcm_field_group_expansion_enable_set` API is used to get the auto expansion status of a group.

Returns

BCM_E_NONE	Successfully retrieved group's auto expansion status
BCM_E_UNAVAIL	Feature is not available
BCM_E_NOT_FOUND	Group Id does not exist

An existing API bcm_field_qset_data_qualifier_get has been modified as described below.

bcm_field_qset_data_qualifier_get
bcmx_field_qset_data_qualifier_get

Get field data qualifiers included in a group qset.

Syntax

```
#include <bcm/field.h>
int
bcm_field_qset_data_qualifier_get(int unit, bcm_field_qset_t qset,
                                  int qual_max, int *qual_arr,
                                  int *qual_count);

#include <bcmx/field.h>
int
bcmx_field_qset_data_qualifier_get(bcm_field_qset_t *qset,
                                   int qual_max, int *qual_arr,
                                   int *qual_count);
```

Parameters

unit	(IN) BCM device number.
qset	(IN) Field qualifier set.
qual_max	(IN) Maximum number of elements to fill in qual_arr.
qual_arr	(OUT) Data qualifier IDs array.
qual_count	(OUT) Number of data qualifier IDs filled in qual_arr.

Description

Get field data qualifiers included in a group qset. Use bcm_field_qset_data_qualifier_add API to add data qualifiers to the qualifiers set.

Returns

BCM_E_XXX

bcm_field_qualify_ForwardingHeaderOffset
bcm_field_qualify_ForwardingHeaderOffset_get

bcm_field_qualify_L3SrcBind
bcm_field_qualify_L3SrcBind_get

Set/Get the match criteria for the specified qualifier in the field entry

Syntax

```
#include <bcm/field.h>
int bcm_field_qualify_ForwardingHeaderOffset(int unit, bcm_field_entry_t entry,
                                              bcm_field_data_offset_base_t base_header,
                                                        uint32 data, uint32 mask);
int bcm_field_qualify_ForwardingHeaderOffset_get(unit, bcm_field_entry_t
entry,
                                              bcm_field_data_offset_base_t base_header,
                                                        uint32 *data, uint32 *mask);
int bcm_field_qualify_L3SrcBind(int unit, bcm_field_entry_t entry,
                                uint8 data, uint8 mask);
int bcm_field_qualify_L3SrcBind_get(int unit, bcm_field_entry_t entry,
                                    uint8 *data, uint8 *mask);
```

Parameters

`unit`	BCM device number
`entry`	Field entry ID
`data`	(IN) for set; (OUT) for get operation
`mask`	(IN) for set; (OUT) for get operation

Description

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

LAYER 2 ADDRESS MANAGEMENT

bcm_l2_match_masked_traverse

Iterates over entries in the L2 table and executes user callback function for each entry matching both given addr and mask.

Syntax

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

int bcm_l2_match_masked_traverse(int unit, uint32 flags, bcm_l2_addr_t
*match_addr, bcm_l2_addr_t *mask_addr, bcm_l2_traverse_cb trav_fn, void
*user_data);
```

Parameters

`unit`	BCM device number
`flags`	The callback will be called on entries that match the given entry and flags (similar to l2_replace)
`match_addr`	l2 address to compare with
`mask_addr`	l2 mask to compare with
`trav_fn`	The callback will be called on entries that match the given entry and flags (similar to l2_replace)
`user_data`	Pointer to any data provided by API caller

Description

Iterates over matching entries in the L2 table and executes user provided call back function that defined as following: typedef `int (*bcm_l2_traverse_cb)(int` unit, `bcm_l2_addr_t` *info, void `*user_data);`

Returns

```
BCM_E_XXX
```

bcm_l2_replace_match

Replace destination (or delete) multiple L2 entries matching the given address and mask. Only bits that are 1 (turned on) in the mask will be checked when checking the match.

Syntax

```
#include <bcm/l2.h>
int bcm_l2_replace_match(int unit, uint32 flags, bcm_l2_addr_t *match_addr,
bcm_l2_addr_t *mask_addr,
                bcm_l2_addr_t *replace_addr, bcm_l2_addr_t *replace_mask_addr);
```

Parameters

`unit`	BCM device number
`flags`	Replace/delete flags BCM_L2_REPLACE_*
`match_addr`	L2 parameters to match on delete/replace
`mask_addr`	L2 mask to match on delete/replace
`replace_addr`	value to replace match entries (not relevant in case of delete)
`replace_mask_addr`	sets which fields/bits from replace_addr are relevant for replace. Unmasked fields/bit will not be affected. (not relevant for delete)

Description

Replace all entries that match `match_addrmask_addr` with values as set in `replace_add eplace_mask_addr.` Matching is done by setting fields in the `match_addr` and setting `BCM_L2_REPLACE_*` flags to match entries with those fields.

Returns

```
BCM_E_XXX
```

New BCM L2 Station flags have been added.

Table 23: BCM L2 Station Flags

Name	*Purpose*
`BCM_L2_STATION_OLP`	Subject packets to OAM-OLP header processing when entry is hit
`BCM_L2_STATION_XGS_MAC`	Configure source MAC address to be used in OLP header for packets sent to OLP.

LAYER 3 MANAGEMENT

Added a new field to `bcm_l3_intf_t` structure.

```
typedef struct bcm_l3_intf_s {

    ....

   uint32        l3a_intf_flags;       /* See BCM_L3_INTF_XXX flag definitions.
*/
} bcm_l3_intf_t;
```

Added a new field to `bcm_l3_host_t` structure.

```
typedef struct bcm_l3_host_s {

    ....

    bcm_if_t    native_intf;    /* L3 native interface (source MAC) */
} bcm_l3_host_t;
```

Removed an existing field `"native_intf"` from `bcm_l3_key_t` structure.

```
typedef struct bcm_l3_key_s {
    uint32        l3k_flags;            /* IPv6 or IPv4 */
    bcm_ip_t      l3k_ip_addr;          /* Destination IP address (IPv4) */
    bcm_ip6_t     l3k_ip6_addr;         /* Destination IP address (IPv6) */
    bcm_ip_t      l3k_sip_addr;         /* Source IP address (IPv4) */
    bcm_ip6_t     l3k_sip6_addr;        /* Source IP address (IPv6) */
    bcm_vlan_t    l3k_vid;              /* VLAN ID */
} bcm_l3_key_t;
```

MPLS MANAGEMENT

Added new MPLS Egress label flags.

Table 24:

Name	*Purpose*
BCM_MPLS_EGRESS_LABEL_REPLACE	Replace existing entry.
BCM_MPLS_EGRESS_LABEL_WITH_ID	Add using the specified ID.

Added new MPLS Tunnel Switch flags.

Table 25: MPLS Tunnel Switch Flags

Name	*Purpose*
BCM_MPLS_SWITCH_REPLACE	Replace existing entry.
BCM_MPLS_SWITCH_WITH_ID	Add using the specified ID.

MULTICAST CONFIGURATION

Added a new Multicast flag.

Table 26: Multicast Flags

Name	*Purpose*
BCM_MULTICAST_TYPE_PORTS_GROUP	Instead of creating a multicast group, return a multicast ID representing a multicast bitmap

OPERATIONS, ADMINISTRATION, AND MAINTENANCE

Added new OAM Endpoint flags.

Table 27: OAM Endpoint Structure Flag Definitions

Flag	*Description*
BCM_OAM_ENDPOINT_MATCH_OUTER_AND_INNER_VLAN	Selection of MEP based on S and C(S+C) VLAN.
BCM_OAM_ENDPOINT_LMEP_PKT_FWD	Local endpoint CCMs must be forwarded to a remote unit for processing.

Added new OAM Event enumeration.

Table 28: OAM Event Types

Event type	*Description*
bcmOAMEventEndpointCCMTimeoutEarly	A CCM almost timeout has been detected on an endpoint. This event enables the user to move traffic from the Primary paths to the Protection paths before official LOC.

Table 29: OAM Opcode Action Formats

Acion Format Type	*Description*
bcmOAMActionCountEnable	Enable counting.
bcmOAMActionMeterEnable	Enable metering.
bcmOAMActionDrop	Drop the packet.
bcmOAMActionCopyToCpu	Send the packet to CPU.
bcmOAMActionFwdAsData	Forward the packet as normal data packet.
bcmOAMActionFwd	Forward the packet to a specified destination.
bcmOAMActionUcDrop	Drop UC OAM packet.
bcmOAMActionUcCopyToCpu	Send UC OAM packet to CPU.
bcmOAMActionUcFwdAsData	Forward UC OAM packet as normal data packet.
bcmOAMActionUcFwd	Forward UC OAM packet to a specified destination.
bcmOAMActionMcDrop	Drop MC OAM packet.
bcmOAMActionMcCopyToCpu	Send MC OAM packet to CPU.
bcmOAMActionMcFwdAsData	Forward MC OAM packet as normal data packet.
bcmOAMActionMcFwd	Forward MC OAM packet to a specified destination.
bcmOAMActionLowMdlDrop	Drop low MDL packet.
bcmOAMActionLowMdlCopyToCpu	copy low MDL packets to CPU.
bcmOAMActionLowMdlFwdAsData	Forward low MDL packets as normal data packets.
bcmOAMActionLowMdlFwd	Forward low MDL packets to a specified destination.
bcmOAMActionMyStationMissCopyToCpu	Forward MY_STATION_MISS packets to CPU.
bcmOAMActionMyStationMissDrop	DROP MY_STATION_MISS packets.
bcmOAMActionMyStationMissFwdAsData	Forward MY_STATION_MISS packets as normal data packets.
bcmOAMActionMyStationMissFwd	Forward MY_STATION_MISS packets to a specified destination.
bcmOAMActionProcessInHw	Process in internal OAM/CCM engine.
bcmOAMActionLowMdlCcmFwdAsRegularCcm	Use same action as that of normal CCM.
bcmOAMActionCount	Always Last. Not a usable value.

Table 30:

bcm_oam_endpoint_action_set

Set actions for different OAM opcodes.

Syntax

```
#include <bcm/oam.h>
int bcm_oam_endpoint_action_set(int unit, bcm_oam_endpoint_t id;
                                bcm_oam_endpoint_action_t *action);
```

Parameters

`unit`	(IN) BCM device number
`id`	(IN) ID of the endpoint for which opcode actions are being set
`action`	(IN) Pointer to action structure.

Description

This API allows setting of opcode actions for different OAM opcodes on a per endpoint basis.

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_E_RESOURCE`	Unable to create new opcode profile as all the resources are already used.

POLICER CONFIGURATION

Added new enumerations to Policer Group mode.

Table 31: Policer Group Modes

Mode	*Description*
bcmPolicerGroupModeEgressSingle	A dedicated policer for all egress traffic
bcmPolicerGroupModeEgressIntPri	A dedicated policer for egress traffic for each cos level

PORT CONFIGURATION

Added a new Port Class enumeration.

Added new Port Controls.

Table 32: Port Controls

Port Control	*Description*
bcmPortControlIPTerminationOverlayRecycle	When set Recycle port support Overlay processing: IP termination and Native processing
bcmPortControlEvbType	Set EVB port type (enumerated value within `bcm_port_evb_type_t)`
bcmPortControlOamLookupWithDvp	Use DVP as one of the keys for OAM lookup.
bcmPortControlSubportTagEnable	Enable/Disable SubTag subport channelization over Ethernet. In this mode a third VLAN tag is used to create subports under this port.
bcmPortControlEgressModifyOuterPktPri	Enable/Disable outer tag priority modification
bcmPortControlEgressModifyInnerPktPri	Enable/Disable inner tag priority modification

Added new Port PHY Control enumerations.

Table 33: bcm_port_phy_control_t

Name	*Description*
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_TX_ENABLE	Enable the decoupled prbs tx
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_TX_INVERT_DATA	Set the decoupled prbs tx side invert data
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_TX_POLYNOMIAL	Set the decoupled prbs tx side polynomial
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_RX_ENABLE	Enable the decoupled prbs rx
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_RX_INVERT_DATA	Set the decoupled prbs rx side invert data
BCM_PORT_PHY_CONTROL_PRBS_DECOUPLED_RX_POLYNOMIAL	Set the decoupled prbs rx side polynomial
BCM_PORT_PHY_CONTROL_CL73_FSM_AUTO_RECOVER	Enable CL73 FSM(finite state machine) auto recovery feature
BCM_PORT_PHY_CONTROL_RX_PEAK_FILTER_TEMP_COMP	Enable receiver's peaking filter temperature compensation feature
BCM_PORT_PHY_CONTROL_DRIVER_SUPPLY	Configure the tx driver power supply

bcm_port_egress_policer_set

Set the Policer ID associated for the specified egress physical port.

Syntax

```
#include <bcm/port.h>
int
bcm_port_egress_policer_set(
    int unit,
    bcm_port_t port,
    bcm_policer_t policer_id);
```

Parameters

unit	(IN) Unit number.
port	(IN) Port
policer_id	(IN) Policer ID.

Description

Returns

BCM_E_XXX

Added new generic Port Match criteria.

Table 34: Generic Port Match Criteria

BCM_PORT_MATCH_PORT_PON_TUNNEL_VLAN_ETHERTYPE	*Mod/port/trunk + PON Tunnel Value + outer VLAN + Ethertype.*
BCM_PORT_MATCH_PORT_PON_TUNNEL_VLAN_STACKED_ETHERTYPE	Mod/port/trunk + PON Tunnel Value + outer VLAN + inner VLAN + Ethertype.

Added new Port PHY Timesync flags

Table 35: Port Phy timesync flags

BCM_PORT_PHY_TIMESYNC_CAP_MPLS	*PHY is 1588 over MPLS capable (read only)*
BCM_PORT_PHY_TIMESYNC_CAP_ENHANCED_TSFIFO	PHY has enhanced TSFIFO (read only)
BCM_PORT_PHY_TIMESYNC_CAP_INBAND_TS	PHY supports inband timestamping (read only)
BCM_PORT_PHY_TIMESYNC_CAP_FOLLOW_UP_ASSIST	PHY supports follow up assist (read only)
BCM_PORT_PHY_TIMESYNC_CAP_DELAY_RESP_ASSIST	PHY supports delay response assist (read only)
BCM_PORT_PHY_TIMESYNC_VALID_PHY_1588_SYNC_FREQ	sync freq. is valid
BCM_PORT_PHY_TIMESYNC_VALID_PHY_1588_INBAND_TS_CONTROL	phy 1588 inband ts control is valid
BCM_PORT_PHY_TIMESYNC_FOLLOW_UP_ASSIST_ENABLE	Enable follow up assistant
BCM_PORT_PHY_TIMESYNC_DELAY_RESP_ASSIST_ENABLE	Enable delay response assistant
BCM_PORT_PHY_TIMESYNC_64BIT_TIMESTAMP_ENABLE	Enable 64bit timestamping
BCM_PORT_PHY_TIMESYNC_1588_OVER_HSR_ENABLE	Enable 1588 over HSR
BCM_PORT_PHY_TIMESYNC_INBAND_TS_RESV0_ID_CHECK	Enable inband resv0 id check
BCM_PORT_PHY_TIMESYNC_INBAND_TS_SYNC_ENABLE	Enable inband timestamping for Sync
BCM_PORT_PHY_TIMESYNC_INBAND_TS_DELAY_RQ_ENABLE	Enable inband timestamping for Delay request
BCM_PORT_PHY_TIMESYNC_INBAND_TS_PDELAY_RQ_ENABLE	Enable inband timestamping for PDelay request
BCM_PORT_PHY_TIMESYNC_INBAND_TS_PDELAY_RESP_ENABLE	Enable inband timestamping for PDelay response

Added new Port PHY Timesync Control enumerations.

PORT PHY OAM APIS

bcm_port_config_phy_oam_set

Set OAM configurations for the PHY.

Syntax

```
#include <bcm/port.h>
int bcm_port_config_phy_oam_set(int unit, bcm_port_t port,
bcm_port_config_phy_oam_t *config);
```

Parameters

unit	(IN)BCM device number
port	(IN)Device or logical port number
config	(IN) Pointer to OAM configuration structure

Description

This function sets OAM configurations on the PHY for the specified port. The OAM configuration allows to configure OAM delay measurement on Tx and Rx independently. The OAM delay measurement configuration must select the mode of operation from the choices provided such as Y1731, BHH or IETF.

Table 36: bcm_port_config_phy_oam_t

Field	BCM Type	Description
tx_dm_config	bcm_port_config_phy_oam_dm_t	Delay Measurement Config for Tx bcm_port_config_phy_oam_dm_t (page 32)
rx_dm_config	bcm_port_config_phy_oam_dm_t	Delay Measurement Config for Rx bcm_port_config_phy_oam_dm_t (page 32)

```
typedef struct bcm_port_config_phy_oam_s {
    bcm_port_config_phy_oam_dm_t tx_dm_config; /* OAM delay measurement config
for Tx */
    bcm_port_config_phy_oam_dm_t rx_dm_config; /* OAM delay measurement config
for Rx */
} bcm_port_config_phy_oam_t;
```

Table 37: bcm_port_config_phy_oam_dm_t

Field	BCM Type	Description
flags	uint32	OAM Delay Measurement Config flags OAM Configuration Flags (page 33)
mode	bcm_port_config_phy_oam_dm_mode_t	OAM Delay Measurement Mode OAM Delay Measurement Modes (page 33)

```
typedef struct bcm_port_config_phy_oam_dm_s {
    uint32 flags;                            /* OAM Delay Measurement config flags */
     bcm_port_config_phy_oam_dm_mode_t mode; /* OAM Delay Measurement mode
Y.1731, BHH or IETF */
} bcm_port_config_phy_oam_dm_t;
```

Table 38: OAM Configuration Flags

Flags	*Description*
BCM_PORT_PHY_OAM_DM_MAC_CHECK_ENABLE	Enable MAC check for delay measurement
BCM_PORT_PHY_OAM_DM_CONTROL_WORD_ENABLE	Enable control word for delay measurement
BCM_PORT_PHY_OAM_DM_ENTROPY_ENABLE	Enable entropy for delay measurement
BCM_PORT_PHY_OAM_DM_TS_FORMAT	Select timestamp format for delay measurement. PTP(0)/NTP(1)

Table 39: OAM Delay Measurement Modes

Mode	*Description*
bcmPortConfigPhyOamDmModeY1731	OAM Delay Measurement Y.1731 Mode
bcmPortConfigPhyOamDmModeBhh	OAM Delay Measurement BHH Mode
bcmPortConfigPhyOamDmModeIetf	OAM Delay Measurement IETF Mode

Returns

BCM_E_NONE	Success
BCM_E_XXX	Error

bcm_port_config_phy_oam_get

Gets OAM configurations for the PHY.

Syntax

```
#include <bcm/port.h>
int bcm_port_config_phy_oam_get(int unit, bcm_port_t port,
bcm_port_config_phy_oam_t *config);
```

Parameters

unit	(IN)BCM device number
port	(IN)Device or logical port number
config	(OUT) Pointer to OAM configuration structure

Description

This function gets the OAM configurations set on the PHY for the specified port.Please refer bcm_port_config_phy_oam_t (page 32)

Returns

`BCM_E_NONE`	Success
`BCM_E_XXX`	Error

bcm_port_control_phy_oam_set

Set the value for a particular OAM control type.

Syntax

```
#include <bcm/port.h>
int bcm_port_control_phy_oam_set(int unit, bcm_port_t port,
bcm_port_config_phy_oam_t *config);
```

Parameters

`unit`	(IN)BCM device number
`port`	(IN)Device or logical port number
`type`	(IN)PHY OAM control type
`value`	(IN)Value to be set for the control 'type'

Description

This function sets the OAM control type to a value on the PHY for the specified port. The phy/port must be appropriately configured `(oam_config_set)` before this api can be used. Please refer `OAM Control Types (`page 34`)` for supported control types.

Table 40: OAM Control Types

Type	*Description*
bcmPortControlPhyOamDmTxEthertype	OAM Delay Measurement Tx Ethertype
bcmPortControlPhyOamDmRxEthertype	OAM Delay measurement Rx Ethertype
bcmPortControlPhyOamDmTxPortMacAddressIndex	Index to select the MAC address for Tx. Use index of 1 to select MacAddress1, index of 2 for MacAddress2 and so forth.
bcmPortControlPhyOamDmRxPortMacAddressIndex	Index to select the MAC address for Rx. Use index of 1 to select MacAddress1, index of 2 for MacAddress2 and so forth.
bcmPortControlPhyOamDmMacAddress1	OAM Delay measurement MAC address 1.
bcmPortControlPhyOamDmMacAddress2	OAM Delay measurement MAC address 2.
bcmPortControlPhyOamDmMacAddress3	OAM Delay measurement MAC address 3.

Returns

BCM_E_NONE	Success
BCM_E_XXX	Error

bcm_port_control_phy_oam_get

Get the value for a particular OAM control type.

Syntax

```
#include <bcm/port.h>
int bcm_port_control_phy_oam_get(int unit, bcm_port_t port,
bcm_port_config_phy_oam_t *config);
```

Parameters

unit	(IN)BCM device number
port	(IN)Device or logical port number
type	(IN)PHY OAM control type
value	(OUT)Value for control 'type'

Description

This function gets the value set for OAM control type on the PHY for the specified port. The phy/port must be appropriately configured (oam_config_set) before this api can be used. Please refer OAM Control Types (page 34) for supported control types.

Returns

BCM_E_NONE	Success
BCM_E_XXX	Error

bcm_port_tpid_get_all

Get all allowed outer tag protocol ID(TPID)s for the specified port.

Syntax

```
#include <bcm/port.h>
int bcm_port_tpid_get_all(int unit, bcm_port_t port, int size, uint16
*tpid_array,
                          int *color_array, int *count);
```

Parameters

`unit`	BCM device number
`port`	logical port number
`size`	number of elements of the next two array parameters
`tpid_array`	buffer array to hold a list of retrieved tpids
`color_array`	buffer array to hold a list of retrieved color selections
`count`	actual number of tpids retrieved

Description

Get all allowed VLAN tag protocol IDs of outer VLAN tags on devices which allow to configure more than one outer TPID for a port.

Returns

`BCM_E_NONE`	Operation completed successfully
`BCM_E_UNAVAIL`	Operation not supported
`BCM_E_XXX`	Operation failed

QUALITY OF SERVICE

Added a new `BCM_QOS_MAP_XXX` flag that can be passed in `bcm_qos_map_*` APIs.

`BCM_QOS_MAP_SUBPORT`

BCM PACKET, TRANSMIT, AND RECEIVE APIS

Added new Rx Trap code.

Table 41: Rx Trap Codes.

Trap Code	*Description*
bcmRxTrapArplookupFail	For ARP packet, lookup for VRF and TPA (IP) failed

STATISTICS

Removed the below mentioned Statistic type enumerations.

Table 42: type

snmpFcmPortClass3RxFrames	***RFC 4044***
snmpFcmPortClass3TxFrames	RFC 4044
snmpFcmPortClass3Discards	RFC 4044
snmpFcmPortClass2RxFrames	RFC 4044
snmpFcmPortClass2TxFrames	RFC 4044

Table 42: type

snmpFcmPortClass3RxFrames	*RFC 4044*
snmpFcmPortClass2Discards	RFC 4044
snmpFcmPortInvalidCRCs	RFC 4044

SUBPORT CONFIGURATION

Added a new Subport group flag.

Table 43: Subport Group Flags

Name	*Purpose*
BCM_SUBPORT_GROUP_TYPE_SUBPORT_TAG	Create a SubTag subport group.

The bcm_subport_config_t structure has been enhanced as follows:

```
typedef struct bcm_subport_config_s {
    bcm_gport_t   group;    /* Subport group          */
    bcm_vlan_t    pkt_vlan; /* Packet 16-bit VLAN tag */
    int           int_pri;  /* Internal Priority      */
    uint32        prop_flags; /* Property flags for this subport */
    uint16        stream_id_array[BCM_SUBPORT_CONFIG_MAX_STREAMS]; /* LinkPHY
                              fragment header Stream ID array associated with
the LinkPHY subport. */
    int          num_streams; /* Number of streams associated with the LinkPHY
subport. */
    bcm_color_t   color;       /* (internal) color or drop precedence for this
subport.*/
} bcm_subport_config_t;

typedef bcm_subport_config_t       bcmx_subport_config_t;
```

bcm_subport_group_linkphy_config_t_init

Initialize a LinkPHY port config structure.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_group_linkphy_config_t_init(bcm_subport_group_linkphy_config_t
*config);
```

Parameters

config	(OUT) Pointer to the struct to be initialized

Description

Initialize a LinkPHY port config structure.

Returns

None.

bcm_subport_group_linkphy_config_set

Set LinkPHY configuration parameters for a given port.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_group_linkphy_config_set(int unit,
                                     bcm_gport_t port,
                                     bcm_subport_group_linkphy_config_t
*linkphy_config);
```

Parameters

`unit`	(IN) BCM device number
`port`	(IN) LinkPHY port gport
`linkphy_config`	(IN) Pointer to the LinkPHY config struct

Description

Set LinkPHY configuration parameters for a given port. For BCM5645x the LinkPHY configuration parameters for a physical port applies to all the logical LinkPHY subport groups on that port.

The structure below is used to configure LinkPHY properties of a port.

```
typedef struct bcm_subport_group_linkphy_config_s {
    uint32    action_mask;    /* BCM_SUBPORT_LINKPHY_xxx flags */
    uint16    base_stream_id; /* External Stream ID base */
    bcm_subport_group_linkphy_header_mode_t header_mode; /* LinkPHY fragment
header mode */
    uint32    rx_check_flags; /* flags BCM_SUBPORT_LINKPHY_RX_XXX_CHECK for
                                 comparing RXed fragment header parameters */
   uint16   rx_vlan_tpid;   /* Expected Vlan TPID in header of LinkPHY fragments
RXed */
    bcm_mac_t rx_nearend_mac; /* Expected near end MAC address in header of
LinkPHY fragments RXed */
   bcm_mac_t rx_farend_mac;  /* Expected far end MAC address in header of LinkPHY
fragments RXed */
   bcm_mac_t rx_dfc_dest_mac; /* Expected Destination MAC address in header of
DFC fragments RXed */
    uint16    rx_dfc_opcode;    /* Expected opcode in header of DFC fragments
RXed */
   uint16    rx_dfc_time;     /* Expected TIME in header of DFC fragments RXed */
   uint16    rx_dfc_mac_type; /* Expected Ethertype in header of DFC fragments
RXed */
    uint16    tx_vlan_tpid;     /* Vlan TPID to be set in header of LinkPHY
fragments TXed */
    bcm_mac_t tx_nearend_mac;   /* Near end MAC address to be set in header of
LinkPHY fragments TXed */
    bcm_mac_t tx_farend_mac;   /* Far end MAC address to be set in header of
LinkPHY fragments TXed */
} bcm_subport_group_linkphy_config_t;
```

The `action_mask` is used for indicating the specific configurations that need to be done.

Table 44: LinkPHY port configuration action mask flags

Name	***Purpose***
BCM_SUBPORT_LINKPHY_BASE_STREAM_ID	Configure the Base Stream ID
BCM_SUBPORT_LINKPHY_HEADER_MODE	Configure the LinkPHY header mode
BCM_SUBPORT_LINKPHY_RX_CHECK_FLAGS	Configure LinkPHY Rx parameter checks
BCM_SUBPORT_LINKPHY_RX_VLAN_TPID	Set the LinkPHY Rx VLAN TPID
BCM_SUBPORT_LINKPHY_RX_NEAREND_MAC	Set the LinkPHY Rx near end MAC address
BCM_SUBPORT_LINKPHY_RX_FAREND_MAC	Set the LinkPHY Rx far end MAC address
BCM_SUBPORT_LINKPHY_RX_DFC_DEST_MAC	Set the LinkPHY Rx DFC destination MAC address
BCM_SUBPORT_LINKPHY_RX_DFC_OPCODE	Set the LinkPHY Rx DFC opcode
BCM_SUBPORT_LINKPHY_RX_DFC_TIME	Set the LinkPHY Rx DFC time
BCM_SUBPORT_LINKPHY_RX_DFC_MAC_TYPE	Set the LinkPHY Rx DFC MAC address type
BCM_SUBPORT_LINKPHY_TX_VLAN_TPID	Set the LinkPHY Tx VLAN TPID
BCM_SUBPORT_LINKPHY_TX_NEAREND_MAC	Set the LinkPHY Tx near end MAC address
BCM_SUBPORT_LINKPHY_TX_FAREND_MAC	Set the LinkPHY Tx far end MAC address
BCM_SUBPORT_LINKPHY_TX_PENULTIMATE_MODE	Set the LinkPHY Tx penultimate mode
BCM_SUBPORT_LINKPHY_ALL	Set all the above LinkPHY actions

If the flag `BCM_SUBPORT_LINKPHY_TX_PENULTIMATE_MODE` is selected in `action_mask` then "transmit penultimate mode" is enabled. In this mode the penultimate LinkPHY fragment payload of the Ethernet frame would be 64-bytes to avoid padding in the last fragment. If the flag is deselected in `action_mask` then "transmit penultimate mode" is disabled.

The received LinkPHY fragment's parameter check flags are given below.

Table 45: LinkPHY Rx parameter check flags.

Name	***Purpose***
BCM_SUBPORT_LINKPHY_RX_NEAREND_MAC_CHECK	enable/disable nearend MAC comparison
BCM_SUBPORT_LINKPHY_RX_FAREND_MAC_CHECK	enable/disable far end MAC comparison
BCM_SUBPORT_LINKPHY_RX_DFC_DEST_MAC_CHECK	enable/disable DFC destination MAC comparison
BCM_SUBPORT_LINKPHY_RX_DFC_OPCODE_CHECK	enable/disable DFC opcode comparison
BCM_SUBPORT_LINKPHY_RX_DFC_TIME_CHECK	enable/disable DFC time comparison

The `header_mode` parameter of `bcm_subport_group_linkphy_config_t` is defined by `bcm_subport_group_linkphy_header_mode_t`.

```
/* LinkPHY fragment header mode */
typedef enum bcm_subport_group_linkphy_header_mode_e {
    bcmSubportLinkphyHeaderModeEthAdapt, /* Ethernet-Adaptation LinkPHY header
mode */
    bcmSubportLinkphyHeaderModeTciLength /* TCI+Length LinkPHY header mode */
} bcm_subport_group_linkphy_header_mode_t;
```

For BCM5645x only the Ethernet-Adaptation LinkPHY header mode is supported.

Returns

```
BCM_E_XXX
```

bcm_subport_group_linkphy_config_get

Get LinkPHY configuration parameters for a given port.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_group_linkphy_config_get(int unit,
                                     bcm_gport_t port,
                                     bcm_subport_group_linkphy_config_t
*linkphy_config);
```

Parameters

`unit`	(IN) BCM device number
`port`	(IN) Port gport
`linkphy_config`	(OUT) Pointer to the LinkPHY config structure

Description

Get LinkPHY configuration parameters for a given port.

Returns

```
BCM_E_XXX
```

bcm_subport_port_stat_set

Set the statistics value for a given subport and statistics type.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_port_stat_set(int unit,
                          bcm_gport_t port,
                          int stream_id,
                          bcm_subport_stat_t stat_type,
                          uint64 val);
```

Parameters

`unit`	(IN) BCM Device number
`port`	(IN) Subport port gport
`stream_id`	(IN) Stream ID array index if LinkPHY subport
`stat_type`	(IN) Support statistics type
`val`	(IN) Statistics value

Description

Set the statistics value for a given subport and statistics type. The `stream_id` parameter is used only if `port` parameter is a LinkPHY subport gport and more than one streams are associated with the subport. The `stream_id` is used to point to the index of stream ID array. If `stream_id=-1` then set the statistics for all streams associated with the LinkPHY subport.

The `stat_type` that can be configured is defined by `bcm_subport_stat_t.`

```
/* Subport statistics */
typedef enum bcm_subport_stat_e {
    bcmSubportLinkPhyStatRxFragmentBytes, /* LinkPHY subport Rx fragment bytes
statistics */
    bcmSubportLinkPhyStatRxFragments,     /* LinkPHY subport Rx fragments
statistics */
    bcmSubportLinkPhyStatRxFrameBytes,    /* LinkPHY subport Rx frame bytes
statistics */
   bcmSubportLinkPhyStatRxFrames,       /* LinkPHY subport Rx frames statistics
*/
    bcmSubportLinkPhyStatTxFragmentBytes, /* LinkPHY subport Tx fragment bytes
statistics */
    bcmSubportLinkPhyStatTxFragments,     /* LinkPHY subport Tx fragments
statistics */
    bcmSubportLinkPhyStatTxFrameBytes,    /* LinkPHY subport Tx frame bytes
statistics */
   bcmSubportLinkPhyStatTxFrames        /* LinkPHY subport Tx frames statistics
*/
} bcm_subport_stat_t;
```

Returns

```
BCM_E_XXX
```

bcm_subport_port_stat_get

Get the statistics value for a given subport and statistics type.

Syntax

```
#include <bcm/subport.h>
int
bcm_subport_port_stat_get(int unit,
                          bcm_gport_t port,
                          int stream_id,
                          bcm_subport_stat_t stat_type,
                          uint64 *val);
```

Parameters

`unit`	(IN) BCM Device number
`port`	(IN) Subport port gport
`stream_id`	(IN) Stream ID array index if LinkPHY subport
`stat_type`	(IN) Subport statistics type
`val`	(OUT) Statistics value

Description

Get the statistics value for a given subport and statistics type. The `stream_id` parameter is used only if `port` parameter is a LinkPHY subport gport and more than one streams are associated with the subport. The `stream_id` is used to point to the index of stream ID array. If `stream_id=-1` then get the statistics for all streams associated with the LinkPHY subport.

Returns

```
BCM_E_XXX
```

SWITCH CONTROL

Added new Switch Controls.

Table 46: Switch Type Values

Value	*Description*	*Arg Value*
`bcmSwitchDosAttackIcmpV4`	Enable DoS attack checking for ICMPv4 packet size	`TRUE/FALSE`
`bcmSwitchDosAttackIcmpV6`	Enable DoS attack checking for ICMPv6 packet size	`TRUE/FALSE`

Table 46: Switch Type Values

Value	*Description*	*Arg Value*
bcmSwitchOamVersionCheckDisable	Disable version check for OAM packets in pipeline.	TRUE/FALSE
bcmSwitchOamOlpChipEtherType	Ether Type to be used in OLP Chip communication.	Ethertype
bcmSwitchOamOlpChipTpid	TPID to be used for OLP Chip communication.	TPID
bcmSwitchOamOlpChipVlan	VLAN ID to be used for OLP Chip communication.	VLAN ID
bcmSwitchPPPoESessionPktDscpRemarkEnable	Enable the DSCP remark of PPPoE session packets.	TRUE/FALSE
bcmSwitchEgressKeepSystemHeader	Keep system headers on Tx packets for a given port.	TRUE/FALSE
bcmSwitchMacroFlowEcmpHashConcatEnable	Enable hash concatenation for ecmp macro flow.	TRUE/FALSE
cw bcmSwitchMacroFlowLoadBalanceHashConcatEnable	Enable hash concatenation for Load balance macro flow.	TRUE/FALSE
cw bcmSwitchMacroFlowTrunkHashConcatEnable	Enable hash concatenation for Trunk macro flow.	TRUE/FALSE
cw bcmSwitchMacroFlowHigigTrunkHashConcatEnable	Enable hash concatenation for Higig Trunk macro flow.	TRUE/FALSE
cw bcmSwitchMacroFlowECMPHashSeed	Set Hash seed for ecmp macro flow.	Random seed value 0x1-0xFFFFFF
cw bcmSwitchMacroFlowLoadBalanceHashSeed	Set Hash Seed for Load balance macro flow.	Random seed value 0x1-0xFFFFFF
cw bcmSwitchMacroFlowTrunkHashSeed	Set Hash Seed for Trunk macro flow.	Random seed value 0x1-0xFFFFFF
cw bcmSwitchMacroFlowHigigTrunkHashSeed	Set Hash Seed for Higig Trunk macro flow.	Random seed value 0x1-0xFFFFFF
cw bcmSwitchHashTrillPayloadSelect0	Set hash control to select TRILL payload L2/L3 fields for Hash Block A.	• BCM_HASH_SELECT_INNER_L2 : Inner L2 fields • BCM_HASH_SELECT_INNER_L3 : Inner L3 fields
cw bcmSwitchHashTrillPayloadSelect1	Set hash control to select TRILL payload L2/L3 fields for Hash Block B.	• BCM_HASH_SELECT_INNER_L2 : Inner L2 fields • BCM_HASH_SELECT_INNER_L3 : Inner L3 fields
cw bcmSwitchHashTrillTunnelSelect0	Set hash control to select TRILL Tunnel L2/L3 fields for Hash Block A.	• BCM_HASH_SELECT_OUTER_L2 : Outer L2 fields • BCM_HASH_SELECT_TUNNEL_INNER_L2: Tunnel and inner l2 fields • BCM_HASH_SELECT_INNER_L3 : Inner L3 fields

Table 46: Switch Type Values

Value	Description	Arg Value
cw bcmSwitchHashTrillTunnelSelect1	Set hash control to select TRILL Tunnel L2/L3 fields for Hash Block B.	• BCM_HASH_SELECT_OUTER_L2 : Outer L2 fields • BCM_HASH_SELECT_TUNNEL_INNER_L2: Tunnel and inner l2 fields • BCM_HASH_SELECT_INNER_L3 : Inner L3 fields
cw bcmSwitchHashIP6AddrCollapseSelect0	Hash Control to collapse IPV6 addresses for hash block A.	● BCM_HASH_IP6_COLLAPSE_XOR ● BCM_HASH_IP6_COLLAPSE_LSB
cw bcmSwitchHashIP6AddrCollapseSelect1	Hash Control to collapse IPV6 addresses for hash block B.	● BCM_HASH_IP6_COLLAPSE_XOR ● BCM_HASH_IP6_COLLAPSE_LSB
bcmSwitchHashL2VxlanField0	BCM56850 Hash A Selection of L2 Payload for Vxlan Packets	• BCM_HASH_FIELD_DSTMOD - Destination module ID • BCM_HASH_FIELD_DSTPORT - Destination port ID • BCM_HASH_FIELD_SRCMOD - Source module ID • BCM_HASH_FIELD_SRCPORT - Source port ID • BCM_HASH_FIELD_VLAN - L2 payload outer VLAN ID • BCM_HASH_FIELD_ETHER_TYPE - L2 payload Ether type • BCM_HASH_FIELD_MACDA_LO MAC destination address lower 16 bits • BCM_HASH_FIELD_MACDA_MI MAC destination address middle 16 bits • BCM_HASH_FIELD_MACDA_HI MAC destination address upper 16 bits • BCM_HASH_FIELD_MACSA_LO MAC source address lower 16 bits • BCM_HASH_FIELD_MACSA_MI MAC source address middle 16 bits • BCM_HASH_FIELD_MACSA_HI MAC source address upper 16 bits
bcmSwitchHashL2VxlanField1	BCM56850 Hash B Selection of L2 Payload for Vxlan Packets	Same as bcmSwitchHashL2VxlanField0

Table 46: Switch Type Values

Value	*Description*	*Arg Value*
`bcmSwitchHashL3VxlanField0`	BCM56850 Hash A Selection of L3 Payload for Vxlan Packets	• `BCM_HASH_FIELD_DSTMOD` - Destination module ID • `BCM_HASH_FIELD_DSTPORT` - Destination port ID • `BCM_HASH_FIELD_SRCMOD` - Source module ID • `BCM_HASH_FIELD_SRCPORT` - Source port ID • `BCM_HASH_FIELD_PROTOCOL` - Protocol ID • `BCM_HASH_FIELD_DSTL4` - Destination L4 port • `BCM_HASH_FIELD_SRCL4` - Source L4 port • `BCM_HASH_FIELD_VLAN` - VLAN ID • `BCM_HASH_FIELD_IP4DST_LO` - IP destination address lower 16 bits • `BCM_HASH_FIELD_IP4DST_HI` - IP destination address upper 16 bits • `BCM_HASH_FIELD_IP4SRC_LO` - IP source address lower 16 bits • `BCM_HASH_FIELD_IP4SRC_HI` - IP source address upper 16 bits
`bcmSwitchHashL3VxlanField1`	BCM56850 Hash B Selection of L3 Payload for Vxlan Packets	Same as bcmSwitchHashL3VxlanField0

Table 46: Switch Type Values

Value	*Description*	*Arg Value*
`bcmSwitchHashL2L2GreField0`	BCM56850 Hash A Selection of L2 Payload for L2Gre Packets	• `BCM_HASH_FIELD_DSTMOD` - Destination module ID • `BCM_HASH_FIELD_DSTPORT` - Destination port ID • `BCM_HASH_FIELD_SRCMOD` - Source module ID • `BCM_HASH_FIELD_SRCPORT` - Source port ID • `BCM_HASH_FIELD_VLAN` - L2 payload outer VLAN ID • `BCM_HASH_FIELD_ETHER_TYPE` - L2 payload Ether type • `BCM_HASH_FIELD_MACDA_LO` MAC destination address lower 16 bits • `BCM_HASH_FIELD_MACDA_MI` MAC destination address middle 16 bits • `BCM_HASH_FIELD_MACDA_HI` MAC destination address upper 16 bits • `BCM_HASH_FIELD_MACSA_LO` MAC source address lower 16 bits • `BCM_HASH_FIELD_MACSA_MI` MAC source address middle 16 bits • `BCM_HASH_FIELD_MACSA_HI` MAC source address upper 16 bits
`bcmSwitchHashL2L2GreField1`	BCM56850 Hash B Selection of L2 Payload for L2Gre Packets	Same as bcmSwitchHashL2L2GreField0

Table 46: Switch Type Values

Value	*Description*	*Arg Value*
bcmSwitchHashL3L2GreField0	BCM56850 Hash A Selection of L3 Payload for L2Gre Packets	• BCM_HASH_FIELD_DSTMOD - Destination module ID • BCM_HASH_FIELD_DSTPORT - Destination port ID • BCM_HASH_FIELD_SRCMOD - Source module ID • BCM_HASH_FIELD_SRCPORT - Source port ID • BCM_HASH_FIELD_PROTOCOL - Protocol ID • BCM_HASH_FIELD_DSTL4 - Destination L4 port • BCM_HASH_FIELD_SRCL4 - Source L4 port • BCM_HASH_FIELD_VLAN - VLAN ID • BCM_HASH_FIELD_IP4DST_LO - IP destination address lower 16 bits • BCM_HASH_FIELD_IP4DST_HI - IP destination address upper 16 bits • BCM_HASH_FIELD_IP4SRC_LO - IP source address lower 16 bits • BCM_HASH_FIELD_IP4SRC_HI - IP source address upper 16 bits
bcmSwitchHashL3L2GreField1	BCM56850 Hash B Selection of L3 Payload for L2Gre Packets	Same as bcmSwitchHashL3L2GreField0

Added new flags for RTAG7 Hash field Control Selection

Table 47:

BCM_HASH_SELECT_INNER_L2	*Select the inner L2 fields for hash fields selection*
BCM_HASH_SELECT_OUTER_L2	Select the outer L2 fields for hash fields selection
BCM_HASH_SELECT_INNER_L3	Select the inner L3 fields for hash fields selection
BCM_HASH_SELECT_OUTER_L3	Select the outer L3 fields for hash fields selection
BCM_HASH_SELECT_TUNNEL_INNER_L2	Select tunnel header and inner L2 fields for hash fields selection

Added new flags for IPv6 Collapse Methods using RTAG7 Hashing

Table 48:

BCM_HASH_IP6_COLLAPSE_XOR	*Select XOR method to collapse IPv6 Addresses for RTAG7 hashing in field selection block*
BCM_HASH_IP6_COLLAPSE_LSB	Select LSB method to collapse IPv6 Addresses for RTAG7 hashing in field selection block

Added new flags for header selection in trunk hashing

Table 49:

BCM_HASH_HEADER_FORWARD	*Select the forward header as starting header for hashing*
BCM_HASH_HEADER_TERMINATED	Select the last terminated header as starting header for hashing

TRILL MANAGEMENT

bcm_trill_vpn_create

Create a TRILL VPN.

Syntax

```
#include <bcm/trill.h>
#include <bcm/multicast.h>
int bcm_trill_vpn_create(int unit, bcm_trill_vpn_config_t *info);
```

Parameters

unit	(IN) BCM device number
info	(IN/OUT) VPN info

Description

Create a TRILL VPN. The bcm_trill_vpn_config_t structure is described below:

```
typedef struct bcm_trill_vpn_config_s {
    uint32 flags;                        /* Configuration flags */
    bcm_vpn_t vpn;                       /* VPN ID. */
    bcm_vlan_t high_vid;                 /* Outer vlan */
    bcm_vlan_t low_vid;                  /* Inner vlan */
    bcm_multicast_t broadcast_group;     /* Broadcast group */
    bcm_multicast_t unknown_unicast_group; /* Unknown unicast group */
    bcm_multicast_t unknown_multicast_group; /* Unknown multicast group */
    bcm_policer_t policer_id;            /* Policer ID to be used */
} bcm_trill_vpn_config_t;
```

A VPN is assigned a unique ID. which is returned in the vpn field. The multicast group(s) for broadcast and unknown unicast/multicast must first be created using the bcm_multicast_* APIs.

Table 50: TRILL VPN Config Flags

Name	*Purpose*
BCM_BCM_TRILL_VPN	TRILL Fine Grained VPN.
BCM_BCM_TRILL_VPN_WITH_ID	Create VPN with specified ID.
BCM_BCM_TRILL_VPN_REPLACE	Used when modifying the attributes of an existing VPN.
BCM_BCM_TRILL_VPN_INVALID	Invalid TRILL VPN .

Returns

```
BCM_E_XXX
```

bcm_trill_vpn_destroy
bcm_trill_vpn_destroy_all

Destroy a TRILL VPN.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_vpn_destroy(int unit, bcm_trill_vpn_t vpn);
int bcm_trill_vpn_destroy_all(int unit);
```

Parameters

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

Description

Destroy a TRILL VPN.

Returns

```
BCM_E_XXX
```

bcm_trill_vpn_config_t_init

Initialize the TRILL VPN config structure.

Syntax

```
#include <bcm/trill.h>
void
bcm_trill_vpn_config_t_init(
    bcm_trill_vpn_config_t *trill_vpn_config);
```

Parameters

`trill_vpn_config(IN /OUT)`	Pointer to the struct to be initialized

Description

Initialize the TRILL VPN config structure.

Returns

None

bcm_trill_vpn_get

Get a TRILL VPN.

Syntax

```
#include <bcm/trill.h>
int bcm_trill_vpn_get(int unit, bcm_trill_vpn_t vpn,
                      bcm_trill_vpn_config_t *info);
```

Parameters

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

Description

Get an TRILL VPN

Returns

`BCM_E_XXX`

bcm_trill_vpn_traverse

bcm_trill_vpn_traverse

Syntax

```
#include <bcm/trill.h>
int
bcm_trill_vpn_traverse(
    int unit,
    bcm_trill_vpn_traverse_cb cb,
    void *user_data);
```

Parameters

`unit`	(IN) Unit number.
`cb`	(IN) Callback routine, called per VPN entry.
`user_data`	(IN) User data.

Description

Traverse all valid TRILL VPNs and call the supplied callback routine.

Returns

`BCM_E_XXX`

VLAN MANAGEMENT

Added a new VLAN Port Control.

`bcmVlanPortOamUseXlatedInnerVlan`

bcm_vlan_egress_policer_get

Retrieve the egress Policer ID accociated with the specified VLAN

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_egress_policer_get(int unit,
                            bcm_vlan_t vlan,
                            bcm_policer_t *policer_id);
```

Parameters

`unit`	(IN) Unit number
`vlan`	(IN) VLAN ID.
`policer_id`	(OUT) Policer ID.

Description

Retrieve the egress Policer ID accociated with the specified VLAN

Returns

```
BCM_E_XXX
```

bcm_vlan_egress_policer_set

Set the egress Policer ID for the specified VLAN

Syntax

```
#include <bcm/vlan.h>
int
bcm_vlan_egress_policer_get(int unit,
                            bcm_vlan_t vlan,
                            bcm_policer_t policer_id);
```

Parameters

`unit`	(IN) Unit number
`vlan`	(IN) VLAN ID.
`policer_id`	(IN) Policer ID.

Description

Associate the egress Policer ID with the specified VLAN.

Returns

```
BCM_E_XXX
```

Added new VLAN Port Match Criteria.

Table 51: bcm_vlan_port_match_t

Value	*Meaning*
BCM_VLAN_PORT_MATCH_PORT_TUNNEL_VLAN_ETHERTYPE	Mod/port/trunk + Tunnel Value + outer VLAN + Ethertype.
BCM_VLAN_PORT_MATCH_PORT_TUNNEL_VLAN_STACKED_ETHERTYPE	Mod/port/trunk + Tunnel Value + outer VLAN + inner VLAN + Ethertype.

Added a new member to `bcm_vlan_port_t` structure.

```
typedef struct bcm_vlan_port_s {
    ....

    bcm_policer_t egress_policer_id;/* Egress Policer ID */
} bcm_vlan_port_t;
```

bcm_vlan_gport_info_t_init

Initialize a VLAN Gport information structure.

Syntax

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

Parameters

info	(OUT) Pointer to VLAN Gport information structure to initialize.

Description

Initializes a VLAN Gport information structure to default values.

VXLAN MANAGEMENT

Added a new member field to bcm_vxlan_vpn_config_t structure

```
typedef struct bcm_vxlan_vpn_config_s {

    ....
    uint32   vnid;                    /* VXLAN VNID */

} bcm_vxlan_vpn_config_t;
```

Added a new VXLAN Port flag.

Table 52: VXLAN port flags

Name	*Purpose*
BCM_VXLAN_PORT_DEFAULT	Create Default VXLAN Network Port

Removed a member field vnid from bcm_vxlan_port_t structure.

```
typedef struct bcm_vxlan_port_s {
    bcm_gport_t vxlan_port_id;          /* GPORT identifier. */
    uint32 flags;                       /* BCM_VXLAN_PORT_xxx. */
    uint32 if_class;                    /* Interface class ID. */
    uint16 int_pri;                     /* Internal Priority */
    uint8 pkt_pri;                      /* Packet Priority */
    uint8 pkt_cfi;                      /* Packet CFI */
    uint16 egress_service_tpid;         /* Service Vlan TPID Value */
    bcm_vlan_t egress_service_vlan;     /* SD-TAG Vlan ID. */
    uint16 mtu;                         /* MTU */
    bcm_gport_t match_port;             /* Match port / trunk */
    bcm_vxlan_port_match_t criteria;    /* Match criteria */
    bcm_vlan_t match_vlan;              /* Outer VLAN ID to match. */
    bcm_vlan_t match_inner_vlan;        /* Inner VLAN ID to match. */
    bcm_gport_t egress_tunnel_id;       /* Tunnel Initiator ID */
    bcm_gport_t match_tunnel_id;        /* Tunnel Terminator ID */
    bcm_if_t egress_if;                 /* VXLAN egress object. */
} bcm_vxlan_port_t;
```

Section 7: Resolved Issues for 6.3.0

The following issues are resolved in version 6.3.0 of the SDK.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-25090		All	Updated `bcm_field_entry_remove()` API to check entry installed status before performing remove from hardware operation.
SDK-28441		All	Updated field grog documentation for `bcm_field_group_create()` API, priority input parameter restrictions.
SDK-30761		`88230_B0 88230_A0`	FIC mode no longer generates bitmap entries for requeue ports.
SDK-32674		`56334_B0`	Fixed the reference count of VLAN range profiles when creating, updating, or deleting VLAN range translation on a trunk group.
SDK-32951		`56634_A0 56624_B0`	Fixed ESM hardware counter mode to software STAT type bitmap translation routines for Triumph and Triumph2 devices.
SDK-34493	424317	All	Resolved removing FP mirror actions mirrors unwanted packets to modport(0,0), i.e CPU in case of single chip unit, for a short while issue.
SDK-36480		`88650_A0 88640_A0`	Added support for System Red in BCM SDK. For more details please reference System RED section in UM.
SDK-36962	470991	All	Release note is not required
SDK-37680		All	The default value of the configuration property `trunk_extend` has been changed from 0 to 1, since the devices that do not support extended trunk mode are no longer supported in the SDK.
SDK-38133		All	When adding more than 256 FQ scheduling elements, the driver will start using already allocated FQs. Fixed by changing the order of execution so the `pool_base` and `pool_offset` will be calculated correctly. After this change, using more than 256 FQs will not use already allocated FQs.
SDK-38185		All	`bcm_field_data_packet_format_t_init()` API initializes input parameter structure in a backwards-compatible manner.
SDK-38826	442465	`88025_A0`	Added to 5.0.0-exa branch
SDK-38960		All	`topo_board_program()` was calling `_topo_stk_ports_update()` twice.
SDK-40304		`56845_B0 56725_A0 56720_A0 56700_A0 56685_B0 56680_A1 56639_A0 56538_B0 56841_A3 56841_B0 56526_B0`	Fixed `bcm_trunk_failover_get` API to return correct Higig trunk failover ports.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-40893		88650_A0 88640_A0	88650: Mac-in-Mac packets that arrive at egress with B-tag, are not recognized as tagged packets. Fixed the miss-configuration of Mac-in-Mac TPID profile, to allow Mac-in-Mac packets with B-tag, to be recognized as tagged packets in egress.
SDK-41286		88650_A0	The packet trap print has been adjusted to routing for ARAD. The VRF field was added and the fields in_ac & rif are assigned depending only when relevant.
SDK-41307		88650_A0	Add new SOC property for padding runt packets. The property is per port, it's avaliable for NIF ports (except ILKN). Property name: packet_padding_size Values range:0-127 Default:0 - means disable padding
SDK-41357	469082	All	6.2.2 release has SER correction implementation from HEAD already. The more recent fixes for mems implemented as regs and SER will be ported to the 6.2.x branch in the future.
SDK-41581		All	Removed unused STATIC routines from the ~bcm/field.c file: 1. _field_qual_stage_name() 2. _field_qual_IpType_name()
SDK-41865		88650_A0	Stop counter processor on TR 6. Note: the counter processor should be manually activated after running TR 6.
SDK-41870		88640_A0	In PetraB, when configuring the port header type to be TDM_RAW, the user must set the static destination of each source TM-Port. The encoding translation of this destination was not correct.
SDK-41989		88650_A0	L2GRE IP: IP tunnel termination can be done by one lookup of <SIP,DIP> or two separated lookups of <DIP>, <SIP> this can be control by soc-property bcm886xx_ip4_tunnel_termination_mode.
SDK-42018	563672	88640_A0	"phy info" command is not supported by Petra-B. Fix the current command to return failure in case of Petra-B.
SDK-42124		88650_A0	Background: RX packets parsing would fail if rx port_header_type_out is not CPU. Fix: Port is now checked for having CPU header type. Otherwise parsing is skipped.
SDK-42685		88650_A0	88650: traffic fails when changing port configuration dynamically from 100G to 10G.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-42777		`88640_A0`	Implement outbound mirror for Petra-B based on port or port-vlan. Note: outbound mirror in Petra-B allocate only one mirror profile per port. The Petra-B mirror can support two mirror modes. Use `bcm_mirror_mode_set/get` to set mirror mode. If the mode equals to 0, then device supports only Inbound mirroring using `bcm_mirror_port_set` API. else, device support both inbound and outbound mirroring using `bcm_mirror_destination_create` API. Default mode was changed and it is set to 1 `(bcm_mirror_destination_create).` Basic sequence to support mirror in Petra-B will be as below: 1. `bcm_mirror_destination_create` to create the inbound or outbound mirror profile. 2. `bcm_mirror_port_vlan_destination_add` or `bcm_mirror_port_destination_add` to attach the mirror profile to inbound mirror port or outbound mirror port. 3. `bcm_mirror_port_vlan_destination_destroy` or `bcm_mirror_port_destination_destroy` to dis-attach the mirror profile to inbound mirror port or outbound mirror port.
SDK-42855	575758	`88640_A0`	Minor fix, no functional change. Fixed tables database issue with `pcb_link_tbl` attributes being overwritten. Both `irr.rsq_fifo_tbl` and `pcb_link_tbl` databases are configured correctly. These tables are not accessed by the driver, and therefore the bug did not have any functional impact
SDK-42907		`88750_A0`	Added missing counters in DCM.
SDK-42933		`88650_A0`	A new TM application allows TM packets to go through two rounds (via recycle port), where the packet format consists in a double-ITMH-Tag. This application requires a specific microcode at egress editor, and is set per port at init via `tm_port_otmh_outlif_ext_mode_<port-id>=DOUBLE_TAG`
SDK-43023	577999	`88650_A0`	The IVE size in the FHEI for IVEC-IDs that are associated with IVE Profiles 0 & 1 was changed to 0 & 3B respectively (instead of 5B).
SDK-43024	581020	All	Add support for `bcm_port_queues_count_get` API on BCM5684x.
SDK-43139	575201	All	PetraB port names were changed to "prefix, local port num" (now port names are xe1, xe2 .. and not xe0, xe1..)
SDK-43175	580345	`56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0`	`bcm_mpls_port_stat_counter_get()` API instead of `bcm_mpls_label_stat_counter_get()` API should be used to retrieve counters associated with the MPLS gport.
SDK-43299	584719	`56334_A0`	Remove unnecessary checking when setting `bcm_rate_mcast_set().`

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-43300		88650_A0 88650_B0 88650_B1	88650: turn off WC4 in case not in use for power consumption efficiency (previously: was always on in case of ILKN0 with more than 12 lanes).
SDK-43354	567520	88650_A0	Fixed info get for COSQ VOQs that were created using TM_FLOW_ID flag.
SDK-43404		88650_A0	Added support for APIs: bcm_port_match_add/delete/set/replace in order to have multiple match criteria. Sequence is to create first vlan port by calling bcm_vlan_port_create then call bcm_port_match_* APIs with Ingress only or egress only flags and specify the additional match lookups required for the same Logical interface. Supported APIs: bcm_port_match_add/delete/replace/set. Limitation: User can't remove / replace the original match that was specified by bcm_vlan_port_create. In order to remove it call bcm_vlan_port_destroy. Note: Logical interface with multiple match criteria learn information should be disabled.
SDK-43522	589162	88230_C0	Fixed bug in parity_enable support.
SDK-43572	583285	88650_A0	A new cosq control type bcmCosqControlMulticastPriorityIngressScheduling was added to map ITMH->TC to HP-MC and LP-MC. The new control will be called from bcm_cosq_control_set API. Where cosq is the traffic class to be mapped is the traffic class to be mapped, a and arg is the priority the traffic class is mapped to: BCM_COSQ_HIGH_PRIORITY or BCM_COSQ_LOW_PRIORITY. bcm_petra_cosq_control_set(int unit, bcm_gport_t port, bcm_cos_queue_t cosq, bcm_cosq_control_t type, int arg) Where: Port=0 Cosq=TC Type=bcmCosqControlMulticastPriorityIngressScheduling Arg=BCM_COSQ_HIGH_PRIORITY / BCM_COSQ_LOW_PRIORITY
SDK-43574		88650_A0 88650_B0 88650_B1	BCM diag commands: "diag pp Ing_Vlan_Edit_info" and "diag pp PKT_associated_TM_info" where updated to Arad settings. For the latter command, new TM info fields were added: ETH_METER_PTR, INGRESS_SHAPING_DEST, ETH_ENCAPSULATION, ETH_DA_TYPE, ST_VSQ, LAG_LB_KEY & IGNORE_CP.
SDK-43589	589004	88650_A0	Added support for multicast id offset, Please use SOC property multicast_id_offset as such: multicast_id_offset_<port-id>.BCM88650=<offset> For example multicast_id_offset_2.BCM88650=10000 Or multicast_id_offset_rcy.BCM88650=10000 Etc... So, if a multicast packet will be received through <port-id> its multicast-id will be changed to <offset+multicast-id> This fix is relevant for Arad only.
SDK-43599		All	gcc 4.7.2 compilation support.
SDK-43604		All	fix gcc 4.7.1 warnings(unused-but-set-variable).

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-43605		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43606		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43607		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43608		88650_A0	BFD API enhancements: 1) Flag in endpoint_info BCM_BFD_ENDPOINT_IN_HW, to indicate whether the endpoint is handled in HW or not. 2) Timein/ Timeout events: bcmBDFEventEndpointRemote, bcmBDFEventEndpointRemoteUp flags. 3) bcm_bfd_endpoint_t.remote_gport field - remote destination of BFD packets.
SDK-43612		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43613		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43614		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43616		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43617		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43618		All	fix gcc 4.7.1 warnings(unused-but-set-variable).
SDK-43619	589674	88650_A0	88650: fixed "show patches" bcm shell command (was missing on 6.2.1-hotfix1)
SDK-43623 SDK-44728	586564	56850_A0	LED support is added for TD2. 'led init' in sdk will work as long as the physical - logical port mapping is sorted. random mapping between physical and logical port will require CMIC_REMAP registers to be programmed through a soc script.
SDK-43631	590287	88650_A0	88650: BCM Diagnostics Shell command "diag cosq non_empty_queues" doesn't work for FMQs, In case of non-empty FMQs with SOC property voq_mapping_mode=DIRECT, the diagnostics reports errors for FMQs, but still prints correct information for VoQs.
SDK-43706	591471	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1	Fixed incorrect routing behavior due to flag bit overloading.
SDK-43713	589448	88650_A0	The OTMH formats were not correct in case of IF_MC OTMH-CUD-Extension mode: in this case, the extension is from now on added only for Multicast packets, according to the latest CUD known on the egress pipe (similarly to the ALWAYS mode)
SDK-43716	591808	56440_A0	The stat value for GT16383 and GR 16383 can be read from API bcm_stat_ge_get
SDK-43740	592383	88650_A0	DNX: Fixed help text for "dump" command
SDK-43770		88650_A0	Added CGM Counters to BCM shell diagnostics diag counters graphical diag counters packet_flow
SDK-43807		88650_A0	BCM88650: "diag retransmit" command added. when called, presents relevant retransmit parameters for ILKN0 and ILKN1.
SDK-43837		88650_B0	Vxlan: in encapsulated packet IP protocol was set to GRE instead of UDP

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-43872		88650_A0	The BCM Meter Action (bcmFieldActionPolicerLevel*) was changing both Meter-Pointer0/1 and the DP-Meter-Command. A double HW action is impossible when using the TM & FP functionality. Besides, the DP-Meter-Command has a dedicated BCM action (bcmFieldActionUsePolicerResult), and the default value (0) is the value changing both at ingress and egress. Thus the DP-Meter-Command HW action is not needed and removed from the meter actions.
SDK-43883		88650_A0 88650_B0 88650_B1	Added support for EVB application. Please see more information in the CINT example src/examples/dpp/cint_evb_example.c
SDK-43921		88650_B0	In the Counter processor, in Arad-B0, different packet statistics can be set with 1 counter per Counter processor line (i.e. per Counter-ID): FWD, DROP, and ALL. In case the user sets one of them, but requires the counters of another (e.g., he sets DROP and requires the FWD counters), the Driver should fail instead of returning the counter of DROP.
SDK-43949		88650_A0	Support a new feature - Ring Protection Fast Flush. CINT example - cint_l2_fast_flush.c
SDK-43956	595868	All	WC B0: KR2(brcm) link does not resolve in SDK branches after SDK_6_2_0_EA2
SDK-43957		88650_A0 88650_B0	The egress does not terminate large headers correctly. Thus, a walk-around (WA) is built to terminate headers at ingress FP stage when the Forwarding-Header is too far (more than 32 Bytes). The WA removes up to the 14 Bytes after the Ethernet header location.
SDK-43977	595227	56850_A0	Fixed inaccuracy in shaper programming on TD2
SDK-43992	581119	All	Fixed.
SDK-43993	583971	88030_A0	TPID getting/setting now supported
SDK-43994	560768	88030_A0	OCM table access methods will now work correctly for entries that are not 32bit aligned.
SDK-43996		56640_A0 56641_A0 56642_A0 56643_A0 56644_A0 56645_A0 56648_A0 56640_A1 56643_A1 56644_A1 56640_B0 56644_B0 56643_B0 56648_B0 56649_B0 56649_A0	In RCPU scenario, to redirect incoming packet from CMIC to a specific port in the same unit, we have to configure prepare DCB and that DCB information should be configured in CMIC_PKT_RMHx CMIC registers. On earlier chips, to redirect a packet (SOBMH) from CMIC to a specific port, configuring that port information in DCB would be sufficient, but on latest devices where we can configure cosq's flexibly, we have to specify the cosq number in DCB. Otherwise all redirected packets will go to 0th cosq. This patch configures the cosq number of the egress port in dcb, so that CMIC will inject SOBMH packets to that cosq.
SDK-44001	571844	88650_A0	88650, 88750: Add show temp-PVT command to help message
SDK-44010	597002	All	fixed the bug in the port enable function for QSGMII core
SDK-44016		88650_A0 88650_B0 88650_B1	Outbound mirror functionality is now working correctly.
SDK-44021		88650_A0 88640_A0 88650_B0 88650_B1	VSQ discard set and color size set should not take into account VSQ Global - added type checking
SDK-44022		88650_A0 88650_B0 88650_B1	Set VSQs rate class default values for Tail Drop to be maximum as defined by hardware.

Table 53:

Number	CSP #	Chips	Release Notes For 6.3.0
SDK-44027	594874	88650_A0	In case ARAD port is in XGS-MAC-EXT mode, PP port is derived according to FRC.Source-Port[7:0]. It is user responsible to configure system ports in ARAD to be as follows: [XGS.Modid][ARAD.LocalPort]. Example: In case ARAD local port 41 is faced to XGS Modid 1 then ARAD system port is 297.
SDK-44036		88650_A0	The ITMH Destination has different modes. The modes where the Destination Extensions are used, are setting the Destination incorrectly - mainly the Out-LIF mode and the Ingress Shaping mode.
SDK-44038	597121	88650_A0 88650_B0 88650_B1	bcm_mpls_tunnel_switch_create() with the BCM_MPLS_SWITCH_ACTION_POP action now returns the tunnl_id created. In addition, it can be called with tunnel_id != 0 and flags = 0x02000000 (temporary, a flag BCM_MPLS_SWITCH_REPLACE will be added to API) to update existing switch.
SDK-44045		88650_A0	The Counter Processor uses the DMA FIFO mechanism to collect the counters from HW. In HW, the counters of the Counter Processors are buffered in a FIFO with approx. 60 entries. This FIFO is read repeatedly by the FIFO DMA and the results are saved in a buffer. The size of this buffer was 1K, and is increased now to its maximum, 16K. Besides, the timers are changed in BCM level (SW timers of CPU access to DMA FIFO buffer that can increase the CPU usage) to minimize the CPU usage from ~50 ms to ~0.5 sec. These timers are dynamic and their values are adapted according to the load of the DMA FIFO.
SDK-44047		88650_B0 88650_B1	Multicast Overlay: Added support for overlay IPMC Recycle port use: bcmPortControlIPTerminationOverlayRecycle
SDK-44070		All	Removed non-ASCII characters from register description.
SDK-44074		56850_A0	Code reorganized to handle mem clear issues in simulations.
SDK-44078		88650_A0 88650_B0 88650_B1	DNX: Packet that is trapped to CPU can be parsed either in non-interrupt context (default), or in interrupt-context (using compilation flag BCM_ARAD_PARSE_PACKET_IN_INTERRUPT_CONTEXT). In the second case the device is not accessed and fields src_port src_mod will be set to 0.
SDK-44086		88650_A0	In Counter Processor, one of the statistics was documented as: - FWD_COLOR: forward green, forward not green counters Actually, the HW supports the following mode, which replaces FWD_COLOR: -SIMPLE_COLOR: green, not green counters
SDK-44105		All	Removed -Wp per-processor option from Kernel flags as options are not passed.
SDK-44118		88750_A0 88650_A0	API bcm_stk_module_enable has a parameter where it can disable/enable fabric connectivity. The API was implemented that traffic enable sequence was run, without referring to enable parameter. The issue is fixed.
SDK-44123	598867	88650_A0 88650_B0 88650_B1	TDM optimize : Maximum destination ports in FTMH for TDM traffic is fixed
SDK-44170	597142	56850_A1 56850_A0	Support for UDF qualifiers for VCAP/VFP stage.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44171		`88650_A0`	In Petra-compatible header mode, the program parsing packets arriving at ingress with an FTMH header (e.g. for stacking, or after recycling / outbound mirroring) was not implemented and is implemented now.
SDK-44180		`88650_A0`	In Field Processor, up to 32 simple action macros (called FESes) can be used per PMF-Program. In general, the user tries to allocate the first 16 actions to the 2nd FES group: FES 16-31. Once this FES group is full, the Driver tries to allocate in the 1st FES group: 0-15. In case there is no Direct Extraction Database, and there is a less important Database in the 2nd FES group, the Driver moves this FES to the 1st FES group and allocates the new FES at its place. A bug was not considering the FES-group-index correctly.
SDK-44191		All	TC 2 TC mapping for 8 priorities mode, did not map traffic classes 3-7 properly. TC 2 TC mapping 8 priorities mode did not consider global offset of flow2voq mapping, hence - when called the gport value should have been minus the offset, after the fix use the gport value should be the value retrieved by `BCM_COSQ_GPORT_UCAST_QUEUE_GROUP` as usual.
SDK-44194		`88650_A0`	The LAG ranges of each port are defined in the HW table EGQ.PPCT which is accessed according to the Queue-Pair. This table was accessed only according to the Base-Queue-Pair of each port, and not according to all the port Queue-Pairs. Fixed now EGQ.PPCT init for all Queue-Pairs.
SDK-44205	595775	`88750_A0`	`bcm_stk_modid_set` activated reachability messages for each link. This operation took 128 readwrite operations and 128 sleep(20ms). API performance improved: Reduced to one readwrite to register operation and one sleep for all 128 links.
SDK-44212		`88650_A0`	`bcm_port_force_forward_get` returned and error for multicast id as GPORT
SDK-44216		`88650_B0`	Register access optimization for ARAD (redundant verification was removed). Speeds up Negev initialization by 10%.
SDK-44228	597091	`56334_B0`	Fix erroneous counter status in `bcm_field_group_status_get` API.
SDK-44233		`56840_A0 56634_A0 56440_A0 56850_A0 56640_B0`	Start automatic TCAM memscan only if parity is enabled.
SDK-44238	598928	`88650_A0`	Fixed `bcm_cosq_port_mapping_set` for system ports, Please use `BCM_GPORT_SYSTEM_PORT_ID_GET` to get a handle to system ports
SDK-44239		`88640_A0`	hub/spoke orientation setting for out-AC using `ppd_eg_filter_split_horizon_out_ac_orientation_set` failed.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44246		88650_A0	In Field Processor, data qualifiers allow the user to extrapolate bits either from the packet headers or from common qualifiers. In case of packet headers, the user can control the number of bits, and the location which is composed of the base-header and how many bits to jump. In 88650, the jump can be done in both directions (i.e. the offset can be positive or negative): inside the base-header or from the previous header. For example, extracting EtherType without considering the number of VLAN tags is done by taking the base-header as Header-after-Ethernet, and jumping 2 bytes backward. The implementation of the negative offset (BCM_FIELD_DATA_QUALIFIER_OFFSET_NEGATIVE flag) was not done
SDK-44270	599851	88650_A0	Statistics counter flag: all fabric counters flags updated to High
SDK-44273	600308	56640_A0 56640_A1 56640_B0	Fix L2 mem locks w.r.t freeze and thaw.
SDK-44299		56540_A0 56340_A0 56640_A1 56643_A1 56640_B0 56643_B0 56648_B0	Add per unit global variable instances for port and flex config.
SDK-44321		88650_A0	Both ingress & egress Field Processors can match on IpType, which corresponds to a parsed EtherType. For MPLS, at egress the parsing of bcmFieldIpTypeMplsUnicast was not correct. Besides, bcmFieldIpTypeMplsMulticast was added both at ingress and egress.
SDK-44324		88650_A0	In the Field Processor & ITMH (aka PMF-Extension-Headers) application, Field groups are preselected with particular preselectors, when Forwarding-Type = bcmFieldForwardingTypeTrafficManagement. The release of these preselectors was not correct. Thus, when using the same preselector-id for an Ethernet-based Field group, an internal error was appearing.
SDK-44325		88650_A0	In Field Processor, both: - Implement bcmFieldQualifyTranslated* qualifiers at ingress - and fix the parsing of bcmFieldQualifyTranslated* at egress
SDK-44336		88650_A0 88650_B0 88650_B1	Implemented BCM diag command "diag pp ENCAP" for 88650
SDK-44341		56850_A0	Resolved under SDK-44074 .
SDK-44344		88640_A0	Vlan editing: Change Egress vlan editing operation of NONE on tagged vlan from Ignore to Remove and Add again. This will let Transmit tag/untag and Outbound mirroring features to work.
SDK-44348		88650_A0	Due to incorrect allocation of reserved packet descriptors during init, some EGQ resources were wasted. Wasting EGQ resources can limit the device max supported traffic rate. The reason for this wrong allocation was that the formula for Service Pools in the EGQ was counting the reserved resources per port twice, instead of counting per service pool only the ports that used it. The fix was to calculate the reserved packet descriptors per Service Pool by only counting the resources of the ports using the service pool.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44352		`88650_A0 88650_B0`	CrpsActCntrsCnt counters were removed from "diag counters", due to the fact that these counters should not be read when counter mode is QSIZE and "diag" is general to all modes.
SDK-44357		`88650_A0`	In Field Processor, the user can assign a new trap with the action bcmFieldActionTrap. In previous implementation, this action sets the new {Trap-strength; Trap-Id} according to the assumed HW abilities. 2 options were given to the user when setting the action value with `bcm_field_action_add:` 1. param0 is a Trap GPORT, encoding the strength and the new trap-code. param1 is not used 2. param0 is the new trap-code, param1 is the trap strength. In practice, the HW assigns {Trap-qualifier, Trap-strength and Trap-Id}. Thus when setting the action value with `bcm_field_action_add,` the use must indicate: - param0 is a Trap GPORT, encoding the strength and the new trap-code - param1 is the trap qualifier The user must pay attention that this action size goes from 11 bits to 27 bits.
SDK-44359		`88650_A0`	Solved OAM event not being generated by the OAMP.
SDK-44363		`88650_A0`	The detach feature in Field Processor was not working properly in case that Direct Extraction Field groups with entries were installed to the HW. This is fixed.
SDK-44379		`88650_A0`	OAM mirroring did no work if OAM init was called before calling outbound mirroring API. Now OAM allocates the highest mirror profiles for mirroring, so order does not matter.
SDK-44380		`88650_A0 88650_B0 88650_B1`	Release to customers trunk example `cint_trunk.c`
SDK-44395		`88650_A0`	In Field Processor, the SOC property `field_class_id_size` sets the User-Header size that can be located between the end of System-Headers (FTMH + PPH + their extensions) and the beginning of the Network headers (Ethernet and following header stack). This SOC property is used for example in the Cascaded-Ingress-Egress Field Processor application. The location of the beginning of the Network headers was not correct in 2 cases: - in the HW, because of an internal misconfiguration at egress - in the parsing of the trapped packets. Both cases are solved.
SDK-44396		`88750_A0 88650_A0 88750_B0 88650_B0 88650_B1`	88650, 88750: `bcm_port_phy_control_get(BCM_PORT_PHY_CONTROL_PREEMPHASIS)` returns lane 0 taps for all lanes.
SDK-44399		`56546_A0 56542_A0 56541_A0 56545_A1`	Bringup and sanity fixes done.
SDK-44423		All	LDK-3.0.3 software for iProc is integrated into SDK
SDK-44425	600706	`56850_A0 56850_A1`	Fixed E2ECC message not being transmitted on TD2
SDK-44436		All	Created src/soc/common/ser.c file
SDK-44439		`88650_A0`	When closing MAC loopback on the NIF side, MAC loopback FIFOs are sometimes out of sync. This occasionally results in partial traffic loss. Fixed.
SDK-44444		`88650_A0`	Added a soc property `"mim_num_vsis".` Values are 4096 (default) or 32768. If it is set to 32768, then 32K different I-SIDs may be configured, but ingress VLAN editing is disabled for access facing (UNI) In-ACs.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44458		`88650_A0`	Background: The ITMH parsing is done through microcode. Some of the ITMH formats (Out-LIF, Ingress-Shaping, MC-Flow) are using the Destination extension. In the case of Ingress-Shaping, the microcode is supposed to indicate that the forwarding decision (i.e. the destination) is taken according to the ITMH-Destination-extension field. Limitation: A bug was found on the microcode of the Ingress-Shaping parsing. This bug is fixed. WA: None
SDK-44487		`88650_B0 88650_B1`	ECN using queue size in bytes to determine congestion is now supported in 88650 B0 and up. But (88650 B0/B1) due to a hardware erratum, only values of up to 0x7e00000 bytes (126MB) can be supported (the limitation is only when using bytes and not descriptors!) A value of 0x80000000 (2GB) can be used to disable the bytes limit after it is set. If the bytes limit is used in B0/B1, be sure to also configure tail drop of less than 128MB on the same queues. If values above 126MB are needed in B0,B1 the same affect can be achieved using the ECN WRED configuration.
SDK-44494		All	The CINT API wrapper for `bcm_l2_addr_t_init()` would fail to compile with the Wind River Diab Compiler dcc.
SDK-44510	606330	`88650_A0`	XGS Diffserv: Added support for new system port encoding where System port is being extracted according to [FRC.MODID 8 bits , 0, FRC.PORT 7 lsbs]. Default mode was taking [0,FRC.MODID 7 bits, FRC.PORT 8 bits].
SDK-44519		All	Updated license information for APIMODE, BIGDIGITS, CINT, ED Editor and VxWorks
SDK-44528		`56640_A0`	Fixed `bcm_tr3_cosq_port_sched_set` return `BCM_E_PARAM` if port gport is passed as argument.
SDK-44536	595066	`56850_A0 56850_A1`	In TD2, for Adv Flex Counter, set operation on stats should set both X and Y pipe.
SDK-44554		`88650_A0`	Add to tr 140 NoReset parameter, DRAM bist perform soft reset inthe end of the test unless this parameter set to 1.
SDK-44556		`88650_A0 88650_B0 88650_B1`	Command added to api: DIAG rates <OPTION> <parameters> OPTIONS: - EGQ - to calculate EGQ rate. Requires: `port=<port_id> tc=<traffic_class>` - PQP - to calculate PQP rate. Requires: `port=<port_id> tc=<traffic_class>` - EPEP- to calculate EPE port rate. Requires: `port=<port_id> tc=<traffic_class>` - EPEI- to calculate EPE interface rate. Requires: `if=<interface_id>` - EPNI- to calculate EPNI interface rate. Requires: `scheme=<measure_scheme> [bw=<bw_id>]` Schemes available: 0-measure total, 1-bw on interface, 2-bw on port, 3-bw on Q-pair, 4-bw on channel, 5-bw for mirror/not mirror
SDK-44581	607129	`88650_A0`	APIs `bcm_port_frame_max_setget` were fixed to configure `fram_max` on MAC. These APIs are supported for all NIF ports except ILKN.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44583		56820_A0 56725_A0 56680_A0 56624_B0 56624_A0	Added new reg files and ran code generation script.
SDK-44592		All	bcm_stk_port_set() now avoids updating hardware tables if the stack port membership has not changed.
SDK-44595		88650_A0 88650_B0 88650_B1	TDM Optimized mode, bcm_fabric_tdm_editing_set/get: Fixed getting user defined field from TDM editing in case of TDM optimized should always return 0. Please notice that this field can be configured only when using standard TDM mode. Note: When getting the user define count field in TDM standard mode, the return value will always be the maximum number of bits allowed for user define field.
SDK-44598		88650_A0 88650ACP_A0 88650_B0 88650_B1	PON 3 tags manipulation is supported completely. Tunnel tag is processed in bcm_port_vlan_create. And other 2 VLAN tags are processed by PON egress VLAN translation.
SDK-44600		88650_A0 88650_B0 88650_B1	Fixed: When setting port_init_speed=-1 on the NIF side, default rate is set to a wrong value.
SDK-44617	608209	88650_A0	MAC based VLAN assignment: In order to enable the functionality user needs to set soc property sa_auth_enabled = 1.
SDK-44636		88650_A0 88650_B0 88650_B1	BCM_VLAN_PORT_MATCH_PORT_TUNNEL_VLAN_ETHERTYPE is now supported for AC creation on Tunnel_ID + SVLAN. BCM_VLAN_PORT_MATCH_PORT_TUNNEL_VLAN_STACKED_ETHERTYPE is now supported for AC create on Tunnel_ID + SVLAN + CVLAN.
SDK-44645		88650_A0	88650: Added warm boot support for dynamic ports change.
SDK-44646	606367	56850_A0	Fixed BUD/LEAF loopback port init/deinit issue.
SDK-44672	604494	56850_A0 56850_A1	bcm_cosq_gport_add can allocate more than NUM_COS UC queues for td2 and tr3.
SDK-44688		88650_B0 88650_B1	88650: Added new SOC property to indicate which implementation of ECN for MPLS is used: mpls_ecn_mode. Valid values are 1 (1-bit mode), or 2 (2-bits mode).
SDK-44740	609484	56546_A0 56545_A0 56544_A0 56542_A0 56541_A0 56540_A0 56545_A1 56540_B0	Fixed LPM memory sizes for various configs/SKUs.
SDK-44766	610133	88650_A0 88650_B0 88650_B1	Port command was fixed to support also Fabric ports.
SDK-44800		88650_A0 88640_A0	QOS: Changed logic of WITH_ID flag in bcm_qos_map_create to handle correctly the value of qos_id.
SDK-44803		All	Fix function by Adding check to BCM state - init/deinit
SDK-44806		88650_A0 88650_B0 88650_B1	New option: diag cosq voq Displays all of the non empty VOQs and their current size in bytes. Two filtering option are enabled: diag cosq voq most=x - displays only the <x> most congested VOQs. diag cosq voq id=x - displays only VOQ <x>
SDK-44813		88750_A0	88750: Multi-thread support: Missing bcm_lock to bcm_dfe_init was added.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-44839	607723	88650_A0 88650_B0 88650_B1	Background: When the port header type set to STACKING, its internal LB_PROFILE is set to ONE. Limitation: The port internal LB_PROFILE was not set to 0 when the header type was set to Ethernet for example, and not stacking. WA: None.
SDK-44857	606402	56540_A0 56540_B0	Corrected String length for strncat, such as not to over run buffer in corner cases.
SDK-44912	612131	88640_A0	Resolved: Internal indexing error could result in failure to enable control cells during initialization when using FEC.
SDK-44918	605584	88650_A0	Background: The ITMH parsing of the Mirror-Enable bit (aka IN_MIRR_FLAG) should disable mirroring if the bit is unset. Limitation: If this bit is unset, the mirror profile should be 0. It was in practice dependent on packet's content. This is fixed WA: None
SDK-44926		88650_A0	In Field processor, a CINT called cint_field_dir_ext_counter_inlif.c has been added to illustrate how to attach a Counter-Pointer = In-LIF to the packet via Direct Extraction Field group.
SDK-44930	612050	88650_A0	MIM: Added support WITH_ID flag for bcm_mim_port_add API.
SDK-44947		All	Improve performance of API bcm_tunnel_initiator_set() when called before any next hop entries are set for a given interface (e.g. L3 egress objects).
SDK-44968	606209	88650_A0 88650_B0 88650_B1	Bug in MC on stacking system caused by error in FTMH header programming that was fixed.
SDK-44984		88650_A0	OAM: When sending upmep LM RX packet in is trapped to the CPU but in addition the counter with the index of the stamped value is increased.
SDK-44991		88650_A0 88650_B0 88650_B1	When external phy is connected, TX parameters for lane 0 in a quad not always set correctly (depends on ext phy implementation). This issue was fixed.
SDK-45132		All	New switch controls bcmSwitchDosAttackIcmpV4, bcmSwitchDosAttackIcmpV6 added to enable/disable ICMPV4 and ICMPV6 size check respectively
SDK-45148		56725_A0 56720_A0	SOC Port Valid check is applied to avoid the segmentation fault as it exceeds the MAX limit and corrupts the stack. The issue is seen only in case of CONQUEROR.
SDK-45151		88750_A0 88650_A0	88650, 88750: RX los application improvements - Updated application notes will be supplied.
SDK-45158		88650_A0 88650_B0 88650_B1	Setting port_init_speed_<p>=-1 should disable port p. However this configuration influence other ports. This was fixed.
SDK-45179		88650_A0 88650_B0 88650_B1	Allow setting FW mode by SOC property for 10G ports. Use serdes_firmware_mode=0(default) to reserve current behavior.
SDK-45239	612680	56640_A0 56640_A1 56640_B0	Adjusted default hash offset configs to take into account the scenario when all banks are used by a single memory type.
SDK-45249	611829	88030_A0	Add "QueueInfo" command for C3
SDK-45256	615804	56846_A1	Link flap on the port associated with lane0 if port associated with lane2 is disabled/enabled in KR2 mode

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-45261		All	Skip mem cache use in test mode in all memory ops. Enable test mode in tr tests like cpu benchmark tests tr 21 etc.
SDK-45289	576151	88640_A0	Background: In Field Processor module, Field groups (aka Databases) do not have necessarily pre-selectors. If not set, for the BCM88640 device, an implied pre-selector is selected according to the Field group qualifiers. Issue: Databases without explicit pre-selectors were not selected because the implied pre-selector was not set correctly. WA: None
SDK-45295	617162	88750_A0	88750: Software Reset should not isolate the chip. Fixed.
SDK-45331		88650_A0 88650_B0 88650_B1	Change (without any additional configuration) the division to 1/64 instead of 1/16. Add the ability to use "virtual stack" (will be limited to 4 TMD connection). User can define any set of stacking ports as virtual stack.
SDK-45340		88650_A0	88650: ARAD supports changing port interfaces dynamically. The feature supports the following interfaces: XFI, XLAUI, CAUI. Support for the following interfaces added: ILKN, RXAUI, XAUI and SGMII.
SDK-45342		88650_A0	Egress same-interface filter was disabled on init, by mistake. Fixed the code to enable the same-interface filter. User can control per port enable/disable the filter by calling API `bcm_port_control_set` with `control_type` = bcmPortControlBridge
SDK-45353		88650_A0 88650_B0 88650_B1	Force all incoming traffic from given port to invalid destination in the IRE. The default configuration is not change, meaning no additional configuration needed for one which is not using this feature (Panini for example). In order to discard the traffic, the API should be called. Please refer to the following example: Driver init: 1. Call to `bcm_stk_my_modid_set()` to set the mod-id. 2. Disable all the TDM ports by calling the API. 3. Call to `bcm_stk_module_enable()` . Provisioning example 1. Configure the Ingress and Egress by calling to `bcm_fabric_tdm_editing_set()` 2. Enable the traffic by calling the `bcm_port_control_set()` De-provisioning sequence example 1. Disable incoming traffic by calling to `bcm_port_control_set()` Bug
SDK-45475		88650_A0	Background: IPv4 MC program may do RPF check as well as MC entry search. In that case the search is done in the IPv4 UC tables. Limitation: When using ELK for IPv4 MC tables - then IPv4 UC tables should use ELK as well for the RPF check to succeed. Same happens when IPv4 MC doesn't use ELK. The driver forces the use of ELK for both tables or none (will produce an error if MC table uses ELK but UC table doesn't or the opposite).
SDK-45491	616124	All	Jumps in time provided by `sal_time()` no longer cause discrepancies in signaling message times, given that the SAL has a monotonic `sal_time_usecs()` function.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-45521		56850_A0	Following qualifiers are now supported in Trident2 Egress Stage with the JIRA fix bcmFieldQualifySrcClassL3 bcmFieldQualifySrcClassField bcmFieldQualifySrcClassL2 bcmFieldQualifyDstClassL3 bcmFieldQualifyDstClassField bcmFieldQualifyDstClassL2 bcmFieldQualifyInterfaceClassL2 bcmFieldQualifyInterfaceClassL3
SDK-45576		88650_A0 88650_B0	wrong warmboot data restore that can mainly effect following APIs after warmboot: bcm_cosq_gport_threshold_set/get bcm_cosq_control_set/get
SDK-45613		88650_A0 88650_B1	Fixed the FC init function to properly handle SAFC settings due to SOC Properties.
SDK-45624		88650_A0	Fixed an issue that when calling bcm_vlan_translate_egress_action_set to change outer_tpid_action and then bcm_vlan_translate_egress_action_get, the returned inner tpid was the new outer_tpid.
SDK-45643		53600_A0 53288_A0 53286_A0 53284_A0 53283_A0 53282_A0 53262_B1 53242_B1 53242_A0 53001_A0	Fix the issue that removing FE0 port from the bcm config pbmp_valid causes the initialization fail for FE+GE switches of ROBO FE family
SDK-45727	619823	56850_A1	56850: Fixed issue with group create on EFP with CpuQueue(bcmFieldQualifyCpuQueue) as qualifier.
SDK-45729		88650_A0 88650ACP_A0 88650_B0 88650_B1	For PON 2 tags manipulation, bcm_vlan_translate_egress_action_add of PON ports just needs to do the outer VLAN translation. It's not necessary to take care of tunnel_id anymore.
SDK-45768	587055	88650_A0 56640_A0 56640_B0	Always return SOC_E_FUNC_NOT_FOUND for function searchPCSFuncTable

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-45796	607348	`88650_A0`	Description: At egress Field Processor, the HW correlates the counting action (bcmFieldActionStat) with the redirection (bcmFieldActionRedirect). An improper SW implementation was forcing the user to set a stat-id related to the entry-id for simplicity. Since the number of stat-ids is limited, an error was occurring for high entry ids. Fix: The sequence of using both actions at egress (bcmFieldActionStat & bcmFieldActionRedirect) is changed: - when an entry must redirect and count, the user must indicate in bcmFieldActionRedirect the destination port, and in bcmFieldActionStat the stat-id (in param0). From now on, the user sets also in param1 of bcmFieldActionStat the destination port again. The stat-id must be between 1024 and 3839 (Counter-ID value). Refer to `cint_field_egress_modify_tc_per_port.c` for example. - when an entry must only redirect, the user must call both actions (bcmFieldActionStat & bcmFieldActionRedirect) similarly to the previous, with stat-id = 0 to indicate the Counter-ID is not to be changed. - An entry cannot only change the Counter-ID without redirecting due to an HW limitation Besides, the user cannot use `bcm_field_stat_create[_id]` and `bcm_field_entry_stat_attach` at egress. WA: None
SDK-45798	620819	`88650_B1`	Background: When calling entry install, a first attempt is performed to insert the entry and if it fails because there is no bank or bank is full, a new bank is allocated to the field group. The problem was that there was no validation that this bank has free entries. Then a second attempt is performed and if the allocated bank had no free entries the action fails. Fix: Add validation that bank is not full when allocating it.
SDK-45807	616113	`88650_B1`	Background: In Field processor configuration, PMF programs are HW entities not directly handled by the user. The PMF program is unique by its set of supported Databases. When removing a FP database, the Driver looks if another PMF program exists with the same set of Databases. If not, the resources taken by this database in this program are freed. If so, the whole PMF program is freed by copying a clean program to it. In case of an Egress database, the PMF program erase was not correct. Fix: Added a fix in the copy method of Egress programs.
SDK-45851	617348	`56544_A0 56542_A0 56540_B0 56541_B0`	Memory sanity scripts are included in the FILES.esw package.
SDK-45908		`56854_A0 56850_A0 56855_A0`	bcmFieldQualifyColor qualifier is now supported in Trident2 device Ingress Stage.
SDK-45943		`88650_A0 88650_B0`	ARAD Field warm boot, following was not restored after : 1. preselectors. 2. tcam actions. 3. entry flags (less critical, has effect only when doing WB in the middle of field API sequence).
SDK-45944	622458	`88130_B0 88130_A1`	`bcm_crossbar_enable_set()` fix for BM9600 - when BAG rate is 0, do not scale result to avoid divide by 0 issue.

Table 53:

Number	*CSP #*	*Chips*	*Release Notes For 6.3.0*
SDK-45968		`88650_A0 88650_B0`	compilation error when compiling for ARAD only with warm boot support: `BCM_PTL_SPT=1 BCM_88650_A0=1 BCM_88650_B0=1 # BCM_88640_A0=1` (No definition) CFGFLAGS += `-DBCM_WARM_BOOT_SUPPORT`

Section 8: Unresolved Issues for 6.3.0

The following issues are unresolved in version 6.3.0 of the SDK.

Table 54:

Number	*CSP #*	*Chips*	*Release Notes*
SDK-24798		All	Because of limitations in CINT's type checking, use of the "(auto)" cast directive is needed for the MAC address parameter when using the BCM_PKT_HDR_DMAC_SET and BCM_PKT_HDR_SMAC_SET macros to configure packet header. Example: BCM_PKT_HDR_DMAC_SET(p, (auto) m);
SDK-30856		All	When mirror-to-port is reside on a different than local unit, mirror packet may not egress out correctly.
SDK-32461		56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0	On BCM5684x devices one can observe inaccurate packet discard, when it is based on packet color (CNG bits).
SDK-32617	382975	56840_A0	The counter rate values reported by show counters are sometimes incorrect when the packet rates are very high. The absolute packet and bytes counts are still correct.
SDK-32676	381244	All	The switch control bcmSwitchL2PortBlocking is not correctly preserved for Level 2 warmboot.
SDK-33609	403444	56840_A0	RFC 1757 says that the etherStatsDropEvent counter should be incremented when any sort of resource shortage within the device causes a packet to be dropped. For the BCm56840 however, this statistics counter is not functional and is always 0.
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-35755	411572	56820_A0 56820_B0	Compared to older releases, L2 Notification thread (bcmL2X) requires more CPU bandwidth to run in polling mode (l2xmsg_mode=0), due to the requirement for more thorough entry comparisons. The recommendation, however, is to run L2 notification thread using L2MOD_FIFO DMA mechanism, which is much more efficient and provides more functionality. To do that, please, set the configuration variable (property) l2xmsg_mode to 1.
SDK-35951	455501	All	Compile-time flag (preprocessor variable) BROADCOM_DEBUG enables a lot of useful debugging feature and in generally recommended even for production builds, since it allows the customers to collect and display additional information about the internal SDK state, which is very useful in case of unexpected problems. Generally, the feature was designed to have negligible (though non-zero) performance impact. In one instance, however, the impact might be significant. The reason is that setting BROADCOM_DEBUG sets another flag, SAL_ALLOC_DEBUG, which has much more significant performance impact. Unless specifically needed, it is recommended to remove the code that sets SAL_ALLOC_DEBUG automatically from src/sal/core/<your_os>/alloc.c

Table 54:

Number	*CSP #*	*Chips*	*Release Notes*
SDK-36453	463348	All	The automatically generated grammar file for CINT (cint_c.tab.c) may produce compiler warnings when compiled with certain compilers (including the vxWorks DIAB compiler). The default makefile elevate warnings to errors causing the build process to halt prematurely. Users can remove the "-Werror" compiler switch from the build options to get past this issue.
SDK-37274		56640_A0 56440_A0 56850_A0	An API to attach flexible counters (stats), such as bcm_port_stat_attach() allows to attach multiple flexible counters to a port. For example, both Ingress and Egress counters can be attached that way. However, the API bcm_port_stat_detach() detaches all the flexible stats at once and there is no ability to detach stats selectively. For the time being the only option is to use bcm_port_stat_detach() and then re-attach the stat that is still needed.
SDK-37556	472599	56840_A0	All tables with the VINTF_CTR_IDX field are supported in the SDK except the L3_IIF table.
SDK-37821		56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56440_A0 56843_B0 56841_A3 56846_A1 56841_B0	bcm_cosq_config_set() had traditionally been used to set the system wide number of COSQs. This does not apply to devices with hierarchical schedulers. For these devices, the hierarchy constructed at device initialization time is dependent upon the number of COSQs defined in the system configuration at the time of initialization. Changing the queue count after the hierarchy has been constructed has no effect.
SDK-39180		88650_A0	88650: for each SerDes quad in use, first lane in quad must be associated with a port.
SDK-39878	530531	56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0	On certain devices, when the traffic rate on the ports get to about 80% of 40Gbps and above, byte counters might be collected and accumulated incorrectly, which results in the overflows.
SDK-40268	529728	56334_B0 56334_A0	Calling bcm_port_control_set with bcmPortControlDoNotCheckVlan parameter sets the DISABLE_VLAN_CHECKS field of both the PORT_TABLE and IPORT_TABLE table. However setting the IPORT_TABLE.DISABLE_VLAN_CHECKS drops the DMA packets from cpu with higig2 header and OpCode=2.
SDK-40472		56846_A0 56845_A2 56844_A0 56842_A0 56639_A0 56636_A0 56634_A0 56538_B0 56534_B0 56526_A0 56524_A0 56521_A0 56843_B0 56841_A3 56846_A1 56841_B0 56634_B0 56524_B0	On Chariot class of devices, for L3 packets, the L3 VRF counters do not get incremented, if the packet is redirected to CPU. On Katana and Triumph3 where the ingress vrf stats internally use flex stats, even those packets get counted causing difference in behavior among the two architecture families.
SDK-41113	548536	88650_A0	When creating a RIF by calling bcm_l3_intf_create() and/or when calling bcm_l2_station_add(), MyMac MSB is set. Whenever global MyMac MSB is set, the LSB of the given Mac address is used to also to set High-VSI global MyMac LSB. High-VSI MyMac LSB is not related to MyMac MSB and should not be set when calling bcm_l3_intf_create(). A separate API should be used to set High-VSI MyMac LSB.
SDK-41509		88640_A0	Limited API / Missing implementation for the requested action.
SDK-42027	566253	56640_A0	The implementation of low-level BSC access functions (e.g. soc_i2c_stat) has not yet been updated to support BCM5664x devices. As a result, BSC related APIs might fail on these devices.

Table 54:

Number	CSP #	Chips	Release Notes
SDK-42051	566159	All	On certain devices, the API call bcm_cosq_control_set(unit, port, cosq, bcmCosqControlCongestionManagedQueue, 0); requires the third parameter (cosq) to be a physical queue number, rather than a COS Queue GPORT.
SDK-42140	566499	56840_A0	Currently, only green packets are redirected with bcmFieldActionRedirectPbmp. All packets should be redirected.
SDK-42259		56440_A0 56440_A1	Spurious error messages may be seen when executing Rx/TX tests TR90 and TR91 when the KNET modules is loaded.
SDK-42355		88650_A0	88650: When adding an IPv4 tunnel remark, and then calling bcm_qos_map_destroy with the relevant qos map, after the call you can still use bcm_qos_port_map_get and get the same map back.
SDK-42769	576019	56840_A0	BCM API provides an informational API , bcm_switch_pkt_info_hash_get() that allows the customers to obtain hash values for a given packet (as they are calculated by the switch hardware). It was noticed, that for the packets that have IP Ethertype (0x0800 or 0x86DD), but no real IP header this API returns BCM_E_PARAM, while the real device treats such packets as pure L2 ones. Note, that this erratum does not affect the switching of the said packets.
SDK-42805	462642	56820_A0 56820_B0	On certain devices pause frames might be double counted, which results in snmpDot3OutPauseFrames stat to report twice as many pause frames as there were actually sent.
SDK-42835	470240	All	On BCM56440, the CES framer PRBS test command may fail with error "LIU device failed to initialize".
SDK-42867		88650_A0 88640_A0	88650, 88640: bcm_l3_egress_get currently does not return correct information on some cases: 1. field intf returned value is incorrect. 2. Information retrieved from LL (encap_id object) sometimes returns an empty structure.
SDK-42898	547697	All	It was noted that bcm_port_encap_set() and bcm_mirror() init functions can deadlock if called simultaneously, due to the fact that they take PORT_LOCK and MIRROR_LOCK in different order. In practice, it is recommended to call bcm_mirror_init() in the very beginning, typically in the following sequence: bcm_attach(unit, NULL, NULL, unit); bcm_switch_control_set(unit, bcmSwitchDirectedMirroring, TRUE); bcm_mirror_init(unit); /* other initialization, including bcm_port_encap_set() */
SDK-43094		88650_A0	MLD incorrect configuration when working ILKN+XMAC under same MLD
SDK-43102	582201	56820_A0 56820_B0	On earlier XGS4 devices, such as BCM5662x and 5682x FP-based mirroring might conflict with port-based mirroring, when bcmFieldActionMirrorIngress uses a MODPORT as an argument as opposed to Mirror GPORT. It is recommended to always use DirectedMirroring API mode and with that mode always rely on mirror destinations and corresponding Mirror GPORTs.
SDK-43263	583650	All	Under certain conditions, typically characterized by heavy ATP protocol traffic internal data structures might become corrupted that might causes RX thread to dereference a NULL pointer accidentally.
SDK-43312	585887	88650_A0	MyMac MSB is constant and cannot be configured using BCM APIs.
SDK-43334	584814	56640_A0 56542_A0 56540_A0 56640_A1 56640_B0	When Sync-E mode is enabled under certain circumstances (sc SynchronousPortClockSource=20 for example), linkscan operations will cause ports to behave incorrectly.

Table 54:

Number	*CSP #*	*Chips*	*Release Notes*
SDK-43361		All	Diag Shell commands "fp stat get" and "fp stat set" always have to be entered with the arguments, such as StatId and Type. Forgetting to enter the parameters (i.e. simply typing "fp stat get" or "fp stat set") might result in a Segmentation Fault on certain platforms. Simply, avoid this incorrect usage to prevent any problems.
SDK-43459		88650_A0	When a user saves SDK state, and runs the application in a new process, state of bcmSwitchControlAutoSync is lost, and user will need to re-configure it. When a user tests warmboot without running a new process, but rather using a de-init and init sequence, if AutoSync was previously enabled, it will cause bcm_init to fail. User should disable AutoSync prior to calling deinit, and re-enable it after warm initialization is done.
SDK-43523	588139	56544_A0 56542_A0 56541_A0	OAM LM and DM implementation at IFP is not completed.
SDK-43602		All	The latest versions of GCC (4.6 and later) produce the following warnings, when used to compile the SDK. For example: Compiling src/appl/test/loopback.c /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c: In function 'lb_ppce_initval': /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:330:12: error: variable 'num_cells_per_port_cos' set but not used [-Werror=unused-but-set-variable] /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:329:20: error: variable 'cos' set but not used [-Werror=unused-but-set-variable] /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:329:14: error: variable 'port'set but not used [-Werror=unused-but-set-variable] These warnings are innocuous, but since the default is to treat all warnings as errors, SDK compilation fails. The following GCC command-line option can be used to not treat these specific warnings as errors: -Wno-error=unused-but-set-variable. The easiest way to add it is to call make the following way: make DEBUG_CFLAGS+=-Wno-error=unused-but-set-variable
SDK-43603		All	The latest versions of GCC (4.6 and later) produce the following warnings, when used to compile the SDK. For example: Compiling src/appl/test/loopback.c /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c: In function 'lb_ppce_initval': /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:330:12: error: variable 'num_cells_per_port_cos' set but not used [-Werror=unused-but-set-variable] /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:329:20: error: variable 'cos' set but not used [-Werror=unused-but-set-variable] /tmp/sdk-xgs-robo-6.2.1/src/appl/test/loopback.c:329:14: error: variable 'port'set but not used [-Werror=unused-but-set-variable] These warnings are innocuous, but since the default is to treat all warnings as errors, SDK compilation fails. The following GCC command-line option can be used to not treat these specific warnings as errors: -Wno-error=unused-but-set-variable. The easiest way to add it is to call make the following way: make DEBUG_CFLAGS+=-Wno-error=unused-but-set-variable
SDK-43841		56850_A0	Issues with HiGig Stacking ports currently prevent stacking on BCM56850 systems.
SDK-43899	592482	88650_A0	DNX: Compilation error when DUNE_UI is added to FEATURE_LIST, long size in the system is 8 Bytes and int size is 4 Bytes.
SDK-44346		88650_A0	As in-port type is not Ethernet so LB key is not calculated.

Table 54:

Number	*CSP #*	*Chips*	*Release Notes*
SDK-44383		`88750_A0 88650_A0`	`bcm_port_speed_set()` API overrides CL72 configurations (equalizer). Workaround: After calling `bcm_port_speed_set()` API, call `bcm_port_control_set()` to configure CL72.
SDK-44406		`88650_A0`	OAM 1588 Delay-measurement is not supported
SDK-44416		`88640_A0`	1. API is reading the wrong register from the device. 2. API is missing the parameter of ResetLoad, so this value cannot be configured.

SDK-44418		`88750_A0 88650_A0`
		`88640_A0 88750_B0`
		`88754_A0 88650_B0`
		`88650_B1`
			Build failure when building for Warm Boot but without `BCM_WARM_BOOT_SUPPORT_SW_DUMP` flag.

SDK-44471	599747	`56544_A0`	BCM56544 XAUI ports support single lane GE operation via lane 0 (at boot time). The applicable config is `bcm56544_4x10_12x10=1.` However, current software has not supported this yet. Modifying the `src/soc/esw/triumph3.c->port_speed_max_94` as following can support GE operation: static const `int port_speed_max_94[]` = { -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, 1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 1/* 10 */, -1, -1, -1, -1, -1, -1, -1, 10, 10, 10, 10, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, -1, 10, 10, 10, 10, 10, 10, 10, 10, -1, -1, -1, -1, -1 }; However, there should be more decent way to achieve this feature.
SDK-44485		`88650_A0`	bcm88650 - CL73 (autoneg) is not functional for XLAUI port.
SDK-44506	593957	`56842_A0`	If an L3 interface entry is in the my station TCAM instead of the usual L2 table it will not be restored upon warmboot.
SDK-44602		`88650_A0`	Warmboot not supported for Mac in Mac.

SDK-44659		`88650_A0 88650_B0`
		`88650_B1`
			Background: `bcm_port_link_state_get` returns current link indication and `latch_down` indication. latch down indicates whether the link was down since the last call to this API `(latch_down=1` - the link was down).
			Limitations: In CAUI port the API `latch_down` indication is always 0.
			Workaround: none.

SDK-44726	607522	`56846_A0 56845_B0`
		`56843_B0 56841_B0`
			For unicast traffic, bcmCosqStatDroppedPackets stays 0 for known unicast traffic. This is a problem even if one does not configure any EUC ports.

SDK-44734	606960	`56846_A0 56845_B0`
		`56845_A2 56844_A0`
		`56842_A0 56840_A0`
			After adding more and more L3 unicast configuration entries continuously, the CLI command "dump chg `l3_entry_only"` does reflect the newly added entries. Using the CLI command "dump chg `l3_entry_ipv4_unicast",` which works correctly, can serve as a workaround

SDK-44789	607598	`56640_A0 56640_A1`
		`56640_B0`
			Using `bcm_policer_envelop_create()` API to configure a macro flow rate limiting fails when the configured Committed Information Rate is above 4.6 gbps. The metering below 4.6G works as expected

Table 54:

Number	CSP #	Chips	Release Notes
SDK-44855	610191	All	Current XGS devices provide a common pool of policers (meters) that can be addressed by any entry in the ICAP (Ingress FP). This addressability potentially allows arbitrary sharing of policers by entries, belonging to various groups in the field processor. However, the current hardware imposes a restriction, that micro-flow meters cannot be shared by two entries that can be hit by the same packet. The API enforces this restriction quite vigorously, by simply not allowing entries, belonging to different groups to share the same policer without actually checking whether the entries might still be mutually exclusive. This check cannot be currently disabled.
SDK-44863	609656	56840_A0	During the API initialization, the internal function soc_counter_attach() happens to get called twice, allocating memory for counter collection both times. The memory allocated the first time is simply wasted and is never used again.
SDK-44906	612059	All	This is due to missing include file directives. If this mode is desired, please add #include <bcm_int/api_xlate_port.h> In file $SDK/src/bcm/esw/ces.c and #include <sal/core/libc.h> in file $SDK/include/bcm_int/api_xlate_port.h
SDK-44922		88650_A0	Workaround: Compile PCID without INTR in feature list, or comment the definition if DIAG_LIBS in systems/sim/pcid/Makefile
SDK-44953	611183	56850_A1	The BCM command "phy diag <pbm> dsc" can actually flap the link on all ports in the device. The reason is "phy diag <pbm> dsc command" disables linkscan prior to reading registers and then re-enables linkscan on all ports. However the expected behavior is to affect only the tested ports. Therefore this command should only disable and later on enable linkscan only on the port being read without affecting other ports.
SDK-44982		88650_A0	OAM: Adding bcm_oam_group_traverse and bcm_oam_endpoint_traverse apis
SDK-44988		88650_A0	Warmboot is not support for OAM module
SDK-45090	607085	88650_A0	An error is received while trying to create a composite HR.
SDK-45217		88650_A0	When using bcm_port_loopback_setget APIs and external phy is connected the driver assumes that remote loopback is supported. If remote loopbck is not supported on external phy, BCM_E_UNAVAIL will be returned.
SDK-45263	615806	56850_A0 56855_A0 56854_B0 56854_A0 56850_A1	bcmSwitchL2OverflowEvent is not supported on BCM56850 devices
SDK-45339	617523	88650_A0	The QDCT_TABLE PD thresholds are 15 bits wide. The hardware design is limited that there can only be a total of 4k PD's per queue, so these fields should only be 12 bits wide. To protect customers from doing an illegal configuration of these thresholds we should return an error if customers try to program values larger than 12 bits. Workaround: don't allow to configure the threshold above 12 bit
SDK-45366	611273	56440_A0	When the API bcm_cosq_port_bandwidth_set(...) is called on a particular port and COSq to enable egress rate limiting, sometimes the CLI command "show c" will show the incorrect dropping statistics on a irrelevant port.

Table 54:

Number	***CSP #***	***Chips***	***Release Notes***
SDK-45387	617450	`56640_A0`	When a matched L3 IPv4 packet is correctly routed by hitting a configured L3 IPv4 entry, the hit bit for the corresponding entry should be set. The CLI command "l3 l3table show" should reflect such hit status. However, due to the current erratum, this command does not show the correct hit status. For example: BCM.0> l3 l3table show Unit 0, free L3 table entries: 131071 Entry VRF IP address Mac Address INTF MOD PORT CLASS HIT 1 0 1.1.1.1 00:00:00:00:00:00 100002 0 0 0 n The workaround is to use the CLI command "dump chg `L3_ENTRY_1"` to read the real hit bit value. For example: BCM.0> d chg `L3_ENTRY_1 L3_ENTRY_1.ism0[47336]: <WIDE_ENTRY_BITS=0x100000080808080002,VALID=1,NEXT_HOP_INDEX=2,KEY=0x20202020000,IP_ADDR=0x1010101,IPV4UC:NEXT_HOP_INDEX=2,IPV4UC:KEY=0x20202020000,IPV4UC:IP_ADDR=0x1010101,IPV4UC:HASH_LSB=0x101,IPV4UC:DATA=0x20000,HIT_BITS=1,HIT_0=1,HASH_LSB=0x101,DATA=0x20000,>`
SDK-45393	618025	`56850_A0 56855_A0 56854_B0 56854_A0 56850_A1`	On certain devices, the default for newly created L3 interface is to collapse all DSP values to 0 instead of simply copying the DSCP of ingress packet.
SDK-45545	605346	`56620_B0`	If the source VP entry on the VLAN translation table is written after selecting a new inner VLAN action, the inner VLAN value may be incorrect.
SDK-45564		`56640_A0 56640_A1 56640_B0`	The service meter feature (the associated APIs are `bcm_policer_set`, `bcm_policer_action_create`, `bcm_policer_action_add`, etc.) can controls the packets rate based on VLAN, port, VLAN translate, SVP, VFI or VFP. However, the rate limiting should be independent of the packet size, but is not.
SDK-45622		`88650_A0`	88650: Dynamic ports allows changing the interface type on alive machine: RXAUI interface is not supported.
SDK-45648		`88650_A0`	88650: ILKN counter per channel mode - Statistic per ILKN channel should be done using `bcm_port_stat_get.` Current implementation requires logical port as input, However, according to API gport is required (local port).
SDK-45750		`88650_A0 88650_B0 88650_B1`	The functionality of associating BFMC0-2 GFMC with a subset of fabric congestion leaky buckets LB0-3 using the API `bcm_cosq_gport_flow_control_set` isn't implemented.
SDK-45800	618773	`56850_A0 56855_A0 56854_B0 56854_A0 56850_A1`	`bcm_port_phy_get()`, `bcm_port_phy_set()` and `bcm_port_phy_modify()` APIs do not work correctly on the PHYs with PHY IDs above 0xFF
SDK-45888	597040	`56440_A0`	In sdk-5.10.4 and BFD firmware 1.3.0, starting a MPLS-TP PW BFD session will cause the first 4 bytes of the standard L2 ethernet packets to be truncated after the L2 switching.
SDK-45918	619564	`56846_A0 56845_B0 56845_A2 56844_A0 56842_A0 56840_A0 56850_A0 56855_A0 56843_B0 56841_A3 56846_A1 56841_B0 56854_B0 56854_A0 56850_A1`	`bcm_rx_cosq_mapping_set()` incorrectly rejects certain valid settings in reason mask.
SDK-45957	611540	`56636_A0 56636_B0`	In the process of addition additional rules with associated qualifiers, the destination port qualifier from an earlier entry may be lost.

Table 54:

Number	*CSP #*	*Chips*	*Release Notes*
SDK-45965	623105	56446_B0 56440_A0 56445_A0 56440_A1 56445_A1 56444_A1 56449_B0 56445_B0 56440_B0 56447_B0 56443_B0 56441_B0	Due to an oversight, the fields `new_inner_pkt_priority` and `new_inner_cfi`, priority and `new_outer_cfi` from `bcm_van_action_set_t` are not programmed correctly for a default, port-based action.
SDK-45978	610640	56639_A0 56638_A0 56636_A0 56634_A0 56638_B0 56636_B0 56634_B0	If the specified QSET includes DstMac, DstIp6High and SrcIp6High, IP packets with destination MAC addresses that match the qualified MAC address will not match as they should.
SDK-45988	618997	56850_A0 56850_A1	After creating a VXLAN tunnel, attempts to map packet the DSCP to an internal priority using `bcm_qos_port_map_set()` API will fail with "parameter error"
SDK-46005	615704	56640_A0 56540_A0	When the local endpoint MEP ID is different from the remote endpoint MEP ID, the misconnectivity defect should/will be triggered and is indeed triggered. However, the behavior of both endpoint states toggling between Init, Up and Down states endlessly and repetitively does not seem compliant with the spec.
SDK-46009	620021	56640_A0 56540_A0	When the application software configures a per-VLAN TPID, the TPID in BFD packets do not necessarily match the TPID as configured.
SDK-46020		88650_A0 88650_B0	Background: The OAM trap engine can be configured to trap all traffic to external user port. In this case user doesn't have to add any endpoints. Bug description: This mode can not be combined with the regular mode (i.e. either all traffic is handled in the same way, or each endpoint is added separately).
SDK-46021		88650_A0 88650_B0	Background: When OAM LM or DM packet is trapped to CPU it has a OAM-TS header. The 'mep-type' field in this header should be 1 for up-mep and 0 for down-mep Bug description: 'mep-type' field in this header is always 0
SDK-46075		88750_A0 88650_A0 88750_B0 88650_B0 88650_B1	'phy measure' diag shell command returns wrong results, or no results at all for fabric ports.
SDK-46099		88650_A0 88650_B0 88650_B1	VLAN port create does not provide an ability to allocate local Ingress LIF. Up until now, the allocation of Ingress was according to `vlan_port_t.vsi` != 0. Now, `vlan_port_t.vsi` != 0 has the only meaning of identify if to allocate ISEM , ESEM entries. New flags introduce `BCM_VLAN_PORT_CREATE_INGRESS_ONLY` , `BCM_VLAN_PORT_CREATE_EGRESS_ONLY` to allocate local asymmetric LIF. Those flags are not implemented yet. Workaround in order to allocate local ingress LIF (previous behavior) change line 4109 in `src/bcm/dpp/gport_mgmt.c` from: `is_egress =` TRUE; to: `is_egress = !(vlan_port->flags & BCM_VLAN_PORT_CREATE_INGRESS_ONLY);`

Section 9: 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.

TEST RESULTS

ALL DEVICES

Table 56:

Suite	*6.3.0*	*6.2.4*	*6.2.3*
Golden	4.8%	7.4%	7.8%
bfd	2.6%	1.4%	2.4%
bhh	6.1%	24.4%	22.0%
cosq	3.0%	3.1%	3.0%
dvapi	4.3%	4.2%	4.8%
field	3.3%	4.0%	3.0%
l2	4.1%	3.7%	3.2%
l3	4.3%	5.6%	2.5%
mpls	4.7%	3.8%	1.9%
ptp	0.0%	0.0%	0.0%
stacking	0.7%	0.6%	0.5%
stat	0.6%	3.2%	2.8%
trill	6.7%	12.2%	22.2%
trunk	2.3%	2.7%	2.6%
tunnel	1.3%	1.0%	0.7%
virtual	34.7%	32.1%	27.1%
vlan	4.5%	3.4%	4.1%
vxlan	2.1%	2.3%	2.3%
Total	3.7%	4.0%	3.8%

TRIDENT2

Table 57:

Suite	*6.3.0*	*6.2.4*	*6.2.3*
Golden	9.1%	10.9%	8.9%
bfd	0.0%	0.0%	0.0%
bhh	0.0%	0.0%	0.0%
cosq	3.3%	3.8%	3.6%
dvapi	3.2%	3.9%	3.4%
field	1.4%	1.4%	1.1%
l2	0.7%	0.5%	0.5%
l3	1.4%	3.5%	0.5%
mpls	3.8%	12.0%	1.3%
ptp	0.0%	0.0%	0.0%
stacking	0.0%	0.0%	0.0%
stat	1.4%	3.1%	3.1%
trill	20.5%	39.5%	44.7%
trunk	2.5%	2.4%	2.9%
tunnel	0.0%	2.1%	0.0%
virtual	58.3%	50.0%	50.0%
vlan	1.3%	2.7%	2.4%
vxlan	37.7%	51.2%	51.2%
Total	2.7%	3.8%	3.5%

TRIUMPH3

Table 58:

Suite	*6.3.0*	*6.2.4*	*6.2.3*
Golden	4.3%	11.1%	7.8%
bfd	0.0%	0.0%	0.0%
bhh	0.0%	37.1%	37.1%
cosq	2.0%	3.0%	3.0%
dvapi	5.1%	4.6%	4.8%
field	4.3%	4.1%	3.4%
l2	1.9%	1.8%	1.9%
l3	5.0%	4.5%	0.7%
mpls	2.5%	0.3%	0.8%
ptp	0.3%	0.0%	0.0%
stacking	2.0%	2.3%	2.0%
stat	2.5%	13.6%	14.1%
trill	11.2%	39.8%	89.1%
trunk	1.1%	0.6%	2.1%
tunnel	0.4%	0.0%	0.0%
virtual	50.0%	41.7%	42.9%
vlan	4.1%	4.0%	4.3%
vxlan	0.0%	0.0%	0.0%
Total	3.8%	4.3%	4.8%

HURRICANE2

Table 59:

Suite	*6.3.0*
Golden	5.4%
bfd	0.0%
bhh	0.0%
cosq	0.7%
dvapi	5.0%
field	1.8%
l2	1.9%
l3	1.2%
mpls	1.3%
ptp	0.0%
stacking	3.1%
stat	0.0%
trill	0.0%
trunk	1.4%
tunnel	0.0%
virtual	0.0%
vlan	1.6%
vxlan	0.0%
Total	2.6%

HELIX4

Table 60:

Suite	*6.3.0*
Golden	6.1%
bfd	0.0%
bhh	0.0%
cosq	1.7%
dvapi	3.5%
field	1.7%
l2	5.4%
l3	2.9%
mpls	1.3%
ptp	0.0%
stacking	0.0%
stat	1.4%
trill	7.7%
trunk	0.4%
tunnel	0.0%
virtual	50.0%
vlan	3.5%
vxlan	0.0%
Total	3.0%

Section 10: Device and Platform Support

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

SWITCH DEVICES

Table 61: Switch Devices

Family	*Devices*	*Description*
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 A0	5-Port Gigabit Ethernet Managed Switch integrated with dual cores ARM Cortex-A9 processor and macsec cores
BCM53101	BCM53101 A0	5-Port Fast Ethernet Managed Switch + 1 Fast Ethernet WAN port
	BCM53101 B0	
BCM53115	BCM53115 A0	5-Port GbE Managed Switch + 1 Gigabit WAN port with integrated serdes
	BCM53115 A1	
	BCM53115 B0	
	BCM53115 B1	
	BCM53115 C0	
BCM53118	BCM53118 A0	8-Port Gigabit Ethernet Switch
	BCM53118 B0	
	BCM53118 B1	
BCM53125	BCM53125 A0	5-Port Gigabit Ethernet Switch with 1 Gigabit WAN port and 8051 processor
	BCM53125 B0	
BCM53128	BCM53128 A0	8-Port Gigabit Ethernet Switch with embedded 8051 processor
	BCM53128 B0	
BCM53242	BCM53242 A0	Managed Switch with 24 FE Ports + 2 GbE Interface
	BCM53242 B0	
	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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	BCM53283 A0	16-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53283 B0	
	BCM53283 B1	
	BCM53283 B2	
	BCM53284 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch
	BCM53284 B0	
	BCM53284 B1	
	BCM53284 B2	
	BCM53286 A0	24-Port Fast Ethernet + 4-Port Gigabit Ethernet Multilayer Switch
	BCM53286 B0	
	BCM53286 B1	
	BCM53286 B2	
	BCM53288 A0	24-Port Fast Ethernet + 2-Port Gigabit Ethernet Multilayer Switch with one 2.5GbE Uplink Port
	BCM53288 B0	
	BCM53288 B1	
	BCM53288 B2	
BCM53300	BCM53300 A0	Managed 24-port L2 Switch
	BCM53300 A1	
	BCM53301 A0	Managed 16-port L2 Switch
	BCM53301 A1	
	BCM53302 A0	Managed 24-port L2 Switch
	BCM53302 A1	
BCM53310	BCM53312 A0	BCM53312 Integrated Multilayer Switch and CPU
	BCM53312 B0	
	BCM53313 A0	BCM53313 Integrated Multilayer Switch and CPU
	BCM53313 B0	
	BCM53314 A0	BCM53314 Integrated Multilayer Switch and CPU
	BCM53314 B0	
BCM53320	BCM53322 A0	BCM53322 Integrated Multilayer Switch and CPU
	BCM53323 A0	BCM53323 Integrated Multilayer Switch and CPU
	BCM53324 A0	BCM53324 Integrated Multilayer Switch and CPU
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
BCM53710	BCM53714 A0	BCM56714 Integrated Multilayer Switch and CPU
	BCM53714 A1	
	BCM53714 A2	
	BCM53716 A0	BCM56716 Integrated Multilayer Switch and CPU

Table 61: Switch Devices

Family	Devices	Description
	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	
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
	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

Table 61: Switch Devices

Family	*Devices*	*Description*
	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	
	BCM56134 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb Uplink Ports
	BCM56134 B0	
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
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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	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+
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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	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	
	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	
	BCM56321 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56321 B0	
BCM56330	BCM56331 A0	24-Port GbE Multilayer Switch with Four 2.5GbE Uplink Ports
	BCM56331 B0	
	BCM56333 A0	16-Port GbE Multilayer Switch
	BCM56333 B0	
	BCM56334 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56334 B0	
	BCM56338 A0	8-Port GbE Multilayer Switch with two 10-GbE/HiGig2 Uplink Ports
	BCM56338 B0	
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

Table 61: Switch Devices

Family	*Devices*	*Description*
	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	
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	
	BCM56500 B0	
	BCM56500 B1	
	BCM56500 B2	
	BCM56501 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56501 A1	
	BCM56501 B0	
	BCM56501 B1	
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports
	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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	BCM56506 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56506 A1	
	BCM56506 B0	
	BCM56506 B1	
	BCM56506 B2	
	BCM56507 A0	24-Port GbE Layer 2 Switch with Two 10-GbE/HiGig+ Ports
	BCM56507 A1	
	BCM56507 B0	
	BCM56507 B1	
	BCM56507 B2	
	BCM56508 A0	24-Port GbE Layer 2 Switch with Three 10-GbE/HiGig+ Ports
	BCM56508 A1	
	BCM56508 B0	
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	
	BCM56509 B1	
	BCM56509 B2	
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch
	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	
BCM56540	BCM56541 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56541 B0	
BCM56540	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

Table 61: Switch Devices

Family	*Devices*	*Description*
BCM56540	BCM56544 A1	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch
BCM56540	BCM56545 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
BCM56540	BCM56546 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56546 B0	
BCM56580	BCM56580 A0	16 x 2.5 GbE + 4 x 10 GbE Ethernet Multilayer Switch
BCM56620	BCM56620 A0	
	BCM56620 A1	
	BCM56620 B0	
	BCM56620 B1	
	BCM56620 B2	
	BCM56624 A0	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56624 A1	
	BCM56624 B0	
	BCM56624 B1	
	BCM56624 B2	
	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	

Table 61: Switch Devices

Family	*Devices*	*Description*
BCM56640	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	
BCM56640	BCM56643 A1	48xGE + 4xXFI + 4xHG[42] + 1GE Multilayer Ethernet Switch
	BCM56643 B0	
BCM56640	BCM56644 A1	48xGE + 2xHG[25] + 2xHG[25] + 1GE Multilayer Ethernet Switch
	BCM56644 B0	
BCM56640	BCM56648 A1	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56648 B0	
BCM56640	BCM56649 A1	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch
	BCM56649 B0	
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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	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	
	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	

Table 61: Switch Devices

Family	*Devices*	*Description*
	BCM56841 B1	
	BCM56843 A0	480 Gbps Ethernet Multilayer Switch
	BCM56843 A1	
	BCM56843 A2	
	BCM56843 A3	
	BCM56843 A4	
	BCM56843 B0	
	BCM56843 B1	
	BCM56845 A0	640 Gbps Ethernet Multilayer Switch
	BCM56845 A1	
	BCM56845 A2	
	BCM56845 A3	
	BCM56845 A4	
	BCM56845 B0	
	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	
BCM56850	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56854 A1	
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 Queueing Engine with 49 SPI 4.2 subports
	QE-2000 A2	
	QE-2000 A3	
	QE-2000 A4	
BCM88230	BCM88230 A0	XGS Core (XCore/SBX) Fabric Queueing Engine with Integrated Traffic Management with 4 HiGig2 ports, 50Gbps
	BCM88230 B0	

Table 61: Switch Devices

Family	*Devices*	*Description*
	BCM88235 A0	XGS Core (XCore/SBX) Fabric Queueing 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	
BCM88750	BCM88750 A0	DNX 1600 GBps Switch Fabric
	BCM88750 B0	

Table 62: SER Supported Devices

Family	*Devices*
Trident	56841, 56842, 56843, 56844, 56845, 56846, 56850
Triumph	56640, 56643, 56644, 56648, 56649, 56540, 56541, 56542, 56544, 56545
Katana	All SKUs
Enduro2	All SKUs

Warm boot Supported devices

Note: There is no warm boot support for External table expansion in BCM56620, BCM56630 and BCM56640 device family.

Table 63: Switch Devices that support Warm boot

Family	*Devices*	*Description*
BCM5675	BCM5675 A0	8-Port, 192-Gbps Switch Fabric
	BCM5675 A1	
	BCM5676 A0	4-Port, 96-Gbps Switch Fabric
	BCM5676 A1	
BCM56020	BCM56024 A0	24-Port Integrated Multilayer Switch and CPU
	BCM56024 B0	
	BCM56025 A0	24-Port Integrated L2 Switch and CPU
	BCM56025 B0	

Table 63: Switch Devices that support Warm boot

Family	Devices	Description
	BCM56026 A0	24-Port Integrated L2 Switch and CPU
	BCM56026 B0	
BCM56130	BCM56132 A0	24-Port Fast Ethernet Multilayer Switch with Two 10-GbE/HiGig2 and Two 1G/2.5Gb Uplink Ports
	BCM56132 B0	
	BCM56134 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb Uplink Ports
	BCM56134 B0	
BCM56142	BCM56142 A0	24-Port Fast Ethernet Multilayer Switch with four 1G/2.5Gb/Higig2/HG Lite Uplink Ports
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+
BCM56330	BCM56331 A0	24-Port GbE Multilayer Switch with Four 2.5GbE Uplink Ports
	BCM56331 B0	
	BCM56333 A0	16-Port GbE Multilayer Switch
	BCM56333 B0	
	BCM56334 A0	24-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports
	BCM56334 B0	
	BCM56338 A0	8-Port GbE Multilayer Switch with two 10-GbE/HiGig2 Uplink Ports
	BCM56338 B0	
BCM56240	BCM56240 A0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
BCM56240	BCM56240 B0	2-Port 10GbE (OR 8 *2.5GbE) Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56241 A0	6-Port GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56242 A0	8-Port 2.5GbE Multilayer Switch with Two 2.5GbE Uplink ports
	BCM56243 A0	4-Port 2.5GbE Multilayer Switch
BCM56440	BCM55441 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56440 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56440 B0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports
	BCM56441 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56442 A0	16-Port GbE Multilayer Switch
	BCM56443 A0	8-Port 2.5GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports
	BCM56445 A0	24-Port GbE Multilayer Switch with Four 10-GbE/Hig2 Uplink ports pin compatible with BCM56334
	BCM56446 A0	8-Port GbE Multilayer Switch with Two 10-GbE/Hig2 Uplink ports pin compatible with BCM56338

Table 63: Switch Devices that support Warm boot

Family	*Devices*	*Description*
	BCM56447 A0	16-Port GbE Multilayer Switch pin compatible with BCM56333
	BCM56448 A0	24-Port GbE Multilayer Switch with Four 1GbE/ One 2.5G Uplink ports
BCM56500	BCM56500 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56500 A1	
	BCM56500 B0	
	BCM56500 B1	
	BCM56500 B2	
	BCM56501 A0	Four 10-Gigabit Ethernet/HiGig+ Ports
	BCM56501 A1	
	BCM56501 B0	
	BCM56501 B1	
	BCM56501 B2	
	BCM56502 A0	24-Port GbE Multilayer Switch with Two 10-GbE/HiGig+ Ports
	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	

Table 63: Switch Devices that support Warm boot

Family	*Devices*	*Description*
	BCM56507 B2	
	BCM56508 A0	24-Port GbE Layer 2 Switch with Three 10-GbE/HiGig+ Ports
	BCM56508 A1	
	BCM56508 B0	
	BCM56508 B1	
	BCM56508 B2	
	BCM56509 A0	24-Port GbE Layer 2 Switch with Four 10-GbE/HiGig+ Ports
	BCM56509 A1	
	BCM56509 B0	
	BCM56509 B1	
	BCM56509 B2	
BCM56510	BCM56510 A0	24-Port Gigabit Ethernet Multilayer Switch
	BCM56511 A0	Four-Port 10-GbE/HiGig+ Multilayer Switch
	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
BCM56620	BCM56620 A0	
	BCM56620 A1	
	BCM56620 B0	
	BCM56620 B1	
	BCM56624 A0	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
	BCM56624 A1	
	BCM56624 B0	
	BCM56624 B1	
	BCM56624 B2	
	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	

Table 63: Switch Devices that support Warm boot

Family	*Devices*	*Description*
	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	
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	
BCM56540	BCM56540 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56540 A1	
BCM56540	BCM56541 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56541 A1	
BCM56540	BCM56542 A0	28xGE + 2xF.XAUI/2x10GE + 2xF.HG[42] + 2xF.HG[21] + 1GE, 28xGE + 8xGE/ 8x2.5GE + 2xHG[42] + 2xHG[21] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56542 A1	
BCM56540	BCM56544 A0	10xF.XAUI + 4xHG[21] + 1GE, 10xF.XAUI + 4xXFI, 10xF.XAUI + 2xHG[42], 4xXAUI + 12xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56544 A1	
BCM56540	BCM56545 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56545 A1	
BCM56540	BCM56546 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
BCM56640	BCM56640 A0	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 (Preview)
	BCM56640 A1	
BCM56640	BCM56643 A0	48xGE + 4xXFI + 4xHG[42] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56643 A1	
BCM56640	BCM56644 A0	48xGE + 2xHG[25] + 2xHG[25] + 1GE Multilayer Ethernet Switch (Preview)
	BCM56644 A1	
BCM56640	BCM56648 A0	48xGE + 2xHG[42] + 2xHG[21] + 1GE, 48xGE + 4xXFI + 2xHG[42] + 1GE, 48xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
	BCM56648 A1	

Table 63: Switch Devices that support Warm boot

Family	Devices	Description
BCM56640	BCM56649 A0	28xGE + 2xHG[42] + 2xHG[21] + 1GE, 28xGE + 4xXFI + 2xHG[42] + 1GE, 28xGE + 8xXFI + 1GE Multilayer Ethernet Switch (Preview)
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
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	
	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
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	

Table 63: Switch Devices that support Warm boot

Family	*Devices*	*Description*
	BCM56843 A4	
	BCM56843 B0	
	BCM56843 B1	
	BCM56845 A0	640 Gbps Ethernet Multilayer Switch
	BCM56845 A1	
	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	
BCM56850	BCM56850 A0	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56854 A0	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56850 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
	BCM56854 A1	1.28Tbps I/O, 1Tbps Core Ethernet Switch
BCM88640	BCM88640 A0	80GBps DNX Traffic manager + Packet processor
	BCM88640 B0	
BCM88650	BCM88650 A0	200GBps DNX Traffic manager + Packet processor
	BCM88650 B0	
	BCM88650 B1	
BCM88750	BCM88750 A0	1600GBps DNX Switch fabric
	BCM88750 B0	

Table 64: Switch Device Codenames

Product Family	*Architecture*	*Codename*
BCM5650	StrataXGS	-
BCM5665	StrataXGS	-
BCM5670	StrataXGS	-
BCM5673	StrataXGS	-
BCM5674	StrataXGS II	-
BCM5675	StrataXGS II	-
BCM5690	StrataXGS	-
BCM5695	StrataXGS II	-
BCM53310	StrataXGS III	Hawkeye
BCM53710	StrataXGS III	Raptor
BCM53720	StrataXGS III	Raven
BCM56010	StrataXGS III	Raptor
BCM56020	StrataXGS III	Tropicana

Table 64: Switch Device Codenames

Product Family	***Architecture***	***Codename***
BCM56100	StrataXGS III	Felix
BCM56110	StrataXGS III	Felix+
BCM56140	StrataXGS IV	Hurricane
BCM56150	StrataXGS IV	Hurricane2
BCM56210	StrataXGS III	Raptor
BCM56220	StrataXGS III	Raven
BCM56300	StrataXGS III	Helix
BCM56310	StrataXGS III	Helix+
BCM56320	StrataXGS IV	Helix3
BCM56340	StrataXGS IV	Helix4
BCM56330	StrataXGS IV	Enduro
BCM56130	StrataXGS IV	Stardust
BCM56440	StrataXGS IV	Katana
BCM56445	StrataXGS IV	Enduro2
BCM56500	StrataXGS III	Firebolt
BCM56510	StrataXGS III	Firebolt2
BCM56520	StrataXGS IV	Apollo
BCM56530	StrataXGS IV	Firebolt3
BCM56540	StrataXGS IV	Apollo2
BCM56580	StrataXGS III	Goldwing
BCM56600	StrataXGS III	Easyrider
BCM56620	StrataXGS IV	Triumph
BCM56629	StrataXGS IV	Triumph
BCM56630	StrataXGS IV	Triumph2
BCM56640	StrataXGS IV	Triumph3
BCM56680	StrataXGS IV	Valkyrie
BCM56685	StrataXGS IV	Valkyrie2
BCM56700	StrataXGS III	Humv
BCM56720	StrataXGS IV	HUMV+
BCM56725	StrataXGS IV	Conqueror
BCM56740	StrataXGS IV	Titan
BCM56744	StrataXGS IV	Titan+
BCM56800	StrataXGS IV	Bradley
BCM56820	StrataXGS IV	Scorpion
BCM56825	StrataXGS IV	Sco320G
BCM56840	StrataXGS IV	Trident
BCM56840_PLUS	StrataXGS IV	Trident+
BCM56850	StrataXGS IV	Trident2
BCM88732	StrataXGS IV	Shadow
BCM88020	XGS Core	Caladan FE-2000
BCM88025	XGS Core	Caladan2
BCM88030	XGS Core	Caladan3
BCM88130	XGS Core	Polaris
BCM88230	XGS Core	Sirius
BCM88235	XGS Core	Sirius+
BCM88231	XGS Core	Sirius TM
BCM88236	XGS Core	Sirius+ TM
BCM56931	XGS Core	Sportster
BCM56936	XGS Core	Sportster+
BCM53010	ROBO	Northstar
BCM53018	ROBO	Costar

Table 64: Switch Device Codenames

Product Family	***Architecture***	***Codename***
BCM53101	ROBO	Lotus
BCM53115	ROBO	Vulcan
BCM53118	ROBO	Blackbird
BCM53125	ROBO	Starfighter
BCM53128	ROBO	Blackbird2
BCM53242	ROBO	Harrier
BCM53280	ROBO	Thunderbolt
BCM53600	ROBO	Voyager
BCM88X4X	SAND	Petra-B
BCM88650	SAND	Arad
BCM88750	SAND	FE1600

PHYS

Table 65: 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
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 (Untested Preview)
BCM54292_A0	54280	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Untested Preview)
BCM54294_A0	54280	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Untested Preview)
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)

Table 65: PHYs

Device	*Driver Family*	*Description*
BCM54340_C1	54380	Quad 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54380_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54382_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_B0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_C0	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54385_C1	54380	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver (Needs additional software component)
BCM54616_A0	54616	Single-Chip 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54640	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E_A1	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54640E B0	54640	Quad-Port Gigabit Copper Transceiver with Copper/Fiber Media Interface
BCM54680_A0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E_A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54680E_B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54682E_A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54682E_B0	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with 2 Copper/Fiber Media Interface
BCM54684_D0	54684	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54684E B0	54682	10/100/1000 Octal (65nm) QSGMII-Copper/Fiber(2) with EEE
BCM54685	54682	Octal QSGMII to 10/100/1000BaseT or Fiber Ethernet Transceiver
BCM54685E_A1	54682	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with Copper/Fiber Media Interface
BCM54810_A0	54880	BroadR-Reach Single-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880_A0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880_B0	54880	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver with BroadR-Reach support
BCM54880E_A1	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54880E_B0	54680	Octal-Port 10/100/1000BASE-T Gigabit Ethernet Transceiver
BCM54881_B0	54880	Octal 10/100Base/Tx Ethernet BroadReach Transceiver
BCM54942 A0	84728	Quad-Channel 10GbE XAUI-to-XFI PHY. Firmware version 0124
BCM54980_B2	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980_C0	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM54980_C1	54980	Octal 1000/100/10BASE-T Gigabit Ethernet Transceiver
BCM8040_A2	8040	Eight-Channel Multirate 1-Gbps - 3.2-Gbps Retimer/Switch
BCM8073_A0	8072	Dual-Channel Serial 10-GbE BASE-KR to XAUI Transceiver. Firmware version d502.
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

Table 65: PHYs

Device	*Driver Family*	*Description*
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. Preview)
BCM8728_A0	8706	Dual-Channel 10-GbE SFI-to-XAUI(TM) Transceiver with EDC. Firmware version 0511. (Preview)
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)
BCM84328_A0	84328	Dual 40 GbE/Octal 10 GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version D018
BCM84334_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.66 (Preview) (Needs additional software component)
BCM84336_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.66 (Preview) (Needs additional software component)
BCM84812_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 2.13
BCM84821_A0	8481	10GBASE-T Transceiver. Firmware version 2.13 (Preview)
BCM84822_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 3.02
BCM84823_B0	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84823_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 4.02
BCM84833_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.66(Driver support for IEEE 1588 features are preview)
BCM84834_B1	8481	Quad 10GBASE-T Transceiver. Firmware version 1.66(Driver support for IEEE 1588 features are preview)
BCM84836_B1	8481	Dual 10GBASE-T Transceiver. Firmware version 1.66(Driver support for IEEE 1588 features are preview)
BCM84844_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.03(Driver support is preview)
BCM84846_A0	8481	Dual 10GBASE-T Transceiver. Firmware version 1.03(Driver support is preview)
BCM84848_A0	8481	Quad 10GBASE-T Transceiver. Firmware version 1.03(Driver support is preview)
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 preview)
BCM84740 A0	84740	40 GbE PPI-to-XLAUI PHY with EDC. Firmware version D102.
BCM84747_A0	84728	Quad SFI to XAUI with 1588 (Firmware version 0x124. Preview)
BCM84748_A0	84728	Quad SFI to XAUI with WAN/1588 (Firmware version 0x124. Preview)
BCM84749_A0	84749	Quad SFI to XAUI with Macsec, 1588 (Firmware version 0x124. Driver support for IEEE 1588 features are preview)
BCM84752 A0	84740	Dual-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D105. (Preview)
BCM84753 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D102.
BCM84754 A0	84740	Quad-Channel 10 GbE SFI-to-XFI PHY with EDC. Firmware version D102.

Table 65: PHYs

Device	Driver Family	Description
BCM84756 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105. (Needs additional software component)
BCM84759 A0	84756	Quad SGMII/XFI to SGMII/SFI Transceiver Firmware version D105.
BCM84780_A0	84740	Octal-Channel 10 GbE SFI-to-XFI PHY with 1588. Firmware version 0x11c (Preview)
BCM84784_A0	84740	Dual 40GbE/Octal 10GbE QSFP+ XLPPI-to-XLAUI PHY. Firmware version 0x125 (Preview)
BCM84764_A0	84728	Quad SFI to RXAUI with 1588 (Firmware version 0x124. Preview)
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. Preview)

OPERATING SYSTEMS

The SDK provides the SAL and BDE abstraction implementations necessary for running the SDK on the following operating systems. See the Platform Guide (56XX-PG810-R) for instructions on porting the SDK to another platform.

Table 66: Operating Systems

Operating System
VxWorks 5.5
VxWorks 6.2
VxWorks 6.4
VxWorks 6.5
VxWorks 6.6
Linux 2.6.21 User Mode
Linux 2.6.21 Kernel Resident Mode
Linux 2.6.25 User Mode
Linux 2.6.25 Kernel Resident Mode
Linux 2.6.27 User Mode
Linux 2.6.27 Kernel Resident Mode
Linux 2.6.35 User Mode
Linux 2.6.35 Kernel Resident Mode
POSIX Compliant (SAL ONLY)

CPU SUBSYSTEMS

Table 67: CPU Subsystems

CPU Subsystem	*Description*
BCM98245	CPCI 32-bit PPC with Motorola 8245 Processor
BCM98548XMC	XMC 32-bit PPC with Freescale 8548 Processor
BCM953003C	XMC 32-bit MIPS74Kc with BCM53003 Processor
BCM5300X	32-bit MIPS74Kc with BCM5300X Processor
BCM5301X	Integrated ARM Cortex-A9 CPU on BCM5301X Switch Devices
BCM5621X	Integrated MIPS CPU on BCM5621X Switch Devices
BCM5622X	Integrated MIPS CPU on BCM5622X Switch Devices
BCM5331X	Integrated MIPS CPU on BCM5331X Switch Devices
BCM5360X	Integrated MIPS74Kc CPU on BCM5360X Switch Devices

CPU AND OPERATING SYSTEM COMBINATIONS

The following CPU and Operating System combinations are supported by the SDK (in addition to the above):

Table 68: CPU and Operating System Combinations

CPU Subsystem	*Operating System*	*Description*
BCM98245	VxWorks 6.2	BSP Provided
BCM98245	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5621X	VxWorks 6.4	BSP Provided
BCM5621X	Linux 2.6.21	Available through WindRiver Linux 2.0 `bcm_ntswics`
BCM5331X	VxWorks 6.4	BSP Provided
BCM5331X	Linux 2.6.21	Available through WindRiver Linux 2.0 `bcm_ntswics`
BCM98548XMC	VxWorks 6.5	BSP Provided
BCM98548XMC	Linux 2.6.27	Available through WindRiver Linux 3.0. Note: Additional patches for issues WIND00172598 and WIND00161649 are required. Contact your WindRiver support personnel for these patches and other WindRiver information.
BCM5300X	VxWorks 6.6	BSP Provided
BCM5300X	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5300X	Linux 2.6.27	Available through WindRiver Linux 3.x
BCM5301X	Linux 2.6.35	Available through Broadcom Customer Support Portal
BCM5360X	VxWorks 6.6	BSP Provided
BCM5360X	Linux 2.6.21	Available through WindRiver Linux 2.0
BCM5360X	Linux 2.6.27	Available through WindRiver Linux 3.x
Generic X86	Linux 2.6.25/2.6.27	

REFERENCE DESIGNS

The following Switch Reference Designs are available from Broadcom and are supported in the SDK.

Table 69: Reference Designs

Platform	*Description*
BCM953001R24M	24-port FE + 2-port GE 53242 SW Ref. Design with BCM53001 Processor
BCM953115R5GM	5-port GE + 1-port serdes 53115 Ref. Design
BCM953125RM	5-port GE 53125 Ref. Design
BCM953242R24M	24-port FE + 2-port GE 53242 SW Ref. Design
BCM953262R24M	24-port FE + 4-port GE 53262 SW Ref. Design
BCM953284R	24-port FE + 2-port GE 53284 SW Ref. Design
BCM953284MDU	24-port FE 53284 SW Ref. Design with TK3715 EPON ONU MAC/Serdes
BCM953286R	24-port FE + 4-port GE 53286 SW Ref. Design
BCM953300	24-port GE 53300 Switch Ref Design
BCM953302	48-port GE 53302 Switch Ref Design
BCM953314K	24-port GE - 53314 System Verification Kit
BCM953314R24GS	24-port GE - 53314 Switch Ref Design
BCM953604R	24-port FE + 1-port 1/2G EPON ONU MAC/SerDes Reference Design
BCM956018K48T	48-port FE + 2-port GE + 2-port HGL(CAT 7) - 56018 SVK
BCM956024K24T	24-port FE + 4-port HGL(CAT 7) - 56024 SVK
BCM956102R48XS	48-port FE + 4 port GE 56102 SW Ref Design w/2-HiGig/10GE
BCM956112R48XS-02	48-port FE + 4 port GE 56112 SW Ref Design w/2-HiGig/10GE - PPC8245
BCM956132K	24-port FE 56132 SW SVK Design w/ two 10GE/HiGig2 and two 1G/ 2.5Gb Uplink Ports
BCM956214R26T	26-port GE (2 TX/SX) + 2-port HGL(CAT 7) - 56214 Reference Design
BCM956219K50T	50-port GE + 2-port HGL(CAT 7) - 56218 - PPC8245 SVK
BCM956218K50T	50-port GE + 2-port HGL(CAT 7) - 56218 System Verification Kit
BCM956224K24T	24-port GE + 4-port HGL(CAT 7) - 56224 SVK
BCM956224R24F	24-port GE + 4-port GE SFP - BCM56224 Reference board.
BCM956300R24	24-port GE 56300 Switch Ref Design
BCM956304R24XS	24-port GE (2 TX/SX) 56304 SW Ref Des w/2-HiGig + 2-10GE
BCM956314R24ST	24-port GE + 4 HiGig/2.5GE(CAT 7) 56314 Ref Design
BCM956314R24XST	24-port GE + 4 10GE/HiGig/2.5GE(CX4) - 56314 Ref Design
BCM956334K_02/BCM956334K_03	24xGE + 4x10G/13HG (iPass) with BCM56334 switch
BCM956500R24	24-port GE 56500 Switch Ref Design
BCM956504R24XS	24-port GE (2 TX/SX) 56504 SW Ref Des w/2-HiGig + 2-10GE
BCM956504R48XSP	48-port GE (12 w/POE) 56504 Switch Ref Design 4 - HiGig/10GE
BCM95650K24	24-port FE + 4 port GE Switch Development Kit
BCM95650R24	24-port FE + 4 port GE (TX or SFP) Reference Design
BCM956514R24XST	24-port GE + 4 10GE/HiGig/2.5GE(CX4) - 56514 Ref Design
BCM956514R48XSP	48-port GE (12 w/POE) 56514 Switch Ref Design 4 - HiGig/10GE
BCM956580K16TXS	16-port 2.5G SFP Fibre + 4 HiGig/10GE 56580 SDK
BCM956601K12D	12-port GE + 1-HiGig 56601 DDR SDRAM SDK
BCM956601K12N	12-port GE + 1-HiGig 56601 Netlogic TCAM SDK
BCM956602KXSN	1-HiGig + 1-10GE 56602 Netlogic TCAM SDK
BCM95665K48	48-port FE + 4 port GE TX/SX + 1HiGig Switch Development Kit
BCM956700K16S	16-port HiGig CX4 56700 SDK
BCM95670K8	8-port 5670 GE Switch Development Kit
BCM95673K2S	2 x 5673 10-GE + HiGig Switch Development Kit

Table 69: Reference Designs

Platform	*Description*
BCM95673R8	8-port 5673 10 GE XFP Switch Reference Design
BCM95673R8CX4	8-port 5673 10 GE CX4 Switch Reference Design
BCM95675K8	8-port 5675 GE Switch Development Kit
BCM95675K8U	8-port 5675 GE Switch Development Kit - PPC8245
BCM956800K20X	20-port 10 GE CX4 56800 SDK
BCM95690K24S	24-port 5690 GE Switch Development Kit w/2HiGig
BCM95690K24	24-port 5690 GE Switch Development Kit
BCM95690P24REF	24-port 5690 GE + 5671 w/2HiGig Ports Reference Design
BCM95690R24	24-port 5690 GE Ports Reference Design
BCM95690R24S	24-port 5690 GE + 5671 w/2HiGig Ports Reference Design
BCM95690R48S	48-port 5690 GE + 5670 w/4HiGig Ports Reference Design
BCM95690R48X2S	48-port 5690 GE + 5670 w/2-HiGig Ports + 2-10-GE Ports Ref. Design
BCM95691K12	12-port 5691 GE Switch Development Kit
BCM95695K24	24-port 5695 GE Switch Development Kit
BCM95695R24S	24-port 5695 GE + 5671 w/2HiGig Ports Reference Design
BCM95695R24X2S	24-port 5695 GE + 2-port 5675 HiGig + 2-port 5674 10GE CX4
BCM95695R48X2S	48-port 5695 GE + 5670 w/2-HiGig Ports + 2-10-GE Ports Ref. Design
BCM91125CFM16	BCM956010CS Dual 5675 Fabric + 1125H CPU
BCM91125CFM8	BCM956006CS Single 5675 Fabric + 1125H CPU
BCM956501LM	12-port 10GE CX4 56501/5675 Line Module
BCM956504LM	48-port GE 56504 Line Module
BCM956700CFM16	16-HiGig 56700 Fabric + BCM1125 CPU Module
BCM95674LM	6-port 10GE CX4 5674/5675 Line Module
BCM956802LM	12-port 10GE CX4 56802 Line Module
BCM95695LM	48-port GE 5695/5675 Line Module
BCM956802CFM8	BCM956006CS 56802 Fabric + 8 10GE + 1125H CPU
BCM956680K24TS_02/BCM956680K24TS_05	25 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports SVK
BCM956624K49TS_02/ BCM956624K49TS_05	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports and External Table Expansion SVK
BCM956624R49S_02	49 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE SFP+ Uplink ports BCM56624 reference board
BCM956634K49S_02	49xGE + 4 x XAUI/HG (iPass) with BCM56634 switch
BCM956636K25S_02	24x1GE + 2x12HG + 4x16HG (iPass) with BCM56636 switch
BCM956638K8XS_02	4x12HG + 4x16HG (iPass) with BCM56638 switch
BCM956639K25S_02	24x1GE + 8x10G (iPass) with BCM56639 switch
BCM956526K29S_02	28x1GE + 6x12HG (iPass) with BCM56526 switch
BCM956685K24TS_02	24 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports SVK
BCM956820K24XG_02/BCM956820K24XG_05	24 x 10-GbE + 4 x 1-GbE Multilayer Ethernet Switch SVK
BCM956820R24XG_02	24 x 10-GbE + 4 x 1-GbE BCM56820 Multilayer Ethernet Switch Reference board with SFP+ interface.
BCM956825K24XG_02	16 x 10-GbE + 8 x 20-Gbps HG2 + 1 x 1-GbE Multilayer Ethernet Switch Reference board.
BCM956720K16S_02/BCM956720K16S_05	16-Port, 256-Gbps Switch Fabric + 4 x 1-GbE SVK
BCM956725K16S_02/BCM956725K16S_05	8-Port (20Gbps) + 4-port (16Gbps) Switch Fabric + 4 x 1-GbE SVK
BCM988020QSK24X2	Carrier Ethernet 24-port GE + 2-port 10GbE Reference Design (also known as Metrocore)
BCM988130FK24X2	Carrier Ethernet 24-port GE + 2-port 10GbE Reference Design (also known as Polaris Line card)
BCM988025QSK24X2	Carrier Ethernet 24-port GE + 2-port 10GbE Reference Design (also known as C2 SVK)

Table 69: Reference Designs

Platform	***Description***
BCM988130K_02	BCM88130 SVK with 96 fabric serdes connections (24 iPass ports)
BCM988235K_02	BCM88235 SVK with 4 HiGig2 ports (4 iPass), 2 flow control ports (2 iPass)
BCM953724R26WS	26-Port, 26-Gbps Integrated Multilayer Switch and CPU
BCM956628K8TS	8 port 10-GbE/HiGig2 Multilayer Ethernet Switch with External Table Expansion
BCM956620K24TS	24 port 1-GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
BCM956684K24TS	24 port 1-GbE/2.5GbE Multilayer Ethernet Switch with 4 x 10-GbE/HiGig2 Uplink ports
BCM956725K16S	8 Port, 20-Gbps + 4 Port, 16-Gbps HiGig2 Switch Fabric
BCM956626K8TS	25 port 1-GbE Multilayer Ethernet Switch with 6 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
BCM956629K24S	25 port 1-GbE Multilayer Ethernet Switch with 8 x 10-GbE/HiGig2 Uplink ports and External Table Expansion
BCM956224R50T	50-port GE + 2-port HGL(CAT 7) - 2 X BCM56224
BCM956024R50T	48-port FE + 2-port GE + 2-port HGL(CAT 7) - 2 X BCM56024
BCM956524K24S_02	24xGE + 4 x XAUI/HG (iPass) with BCM56524 switch
BCM956521K_02	24-Port GbE Multilayer Switch with 10 GbE/HiGig2 Uplink Ports
BCM956740K_02	480/640 Gbps Switch fabric
BCM956743K_02	480/640 Gbps Switch fabric
BCM956840K_02	320/480/640 Gbps Ethernet Multilayer Switch
BCM956845K_02	320/480/640 Gbps Ethernet Multilayer Switch
BCM956846KQ	320/480/640 Gbps Ethernet Multilayer Switch
BCM98727MC	16 port Ipass to SFP+ Media Converter
BCM956534K24TS	24xGE + 4 x XAUI/HG (iPass) with BCM56534 switch
BCM956538K49S	48-Port GbE Multilayer Switch with Four 10-GbE/HiGig2 Uplink Ports SVK
BCM956640K_02	BCM56640 SVK
BCM956643K_02	BCM56643 SVK
BCM956644K_02	BCM56644 SVK
BCM9NEGEV	Two BCM988640TMM line cards (3x10Gb SFP+ phys, 6x20Gb QSFPs) + FE600 switch fabric.
BCM9NEGEVII	BCM988750FEM fabric card
BCM956850K	1.2Gbps Ethernet Multilayer Switch

Note: The flash sizes of some old BCM53XX platforms are 4 MBytes only. As the code size of SDK increases, the 4 MB flash is not enough for this release. Replace the flash to 8 MB or above for those reference designs.

Section 11: 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 70: 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 120)
ED Editor	USENET comp.sources.misc Volume 9, Issue 36	src/appl/diag/edline.c	See (ED Editor License terms and conditions) (page 122)
CINT	http://www.gnu.org/software/bison/	src/appl/cint/cint_parser.[ch]	See (CINT parser license terms and conditions) (page 123)
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 124)
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 125)
APIMODE	http://www.gnu.org/software/bison/	src/appl/diag/api/api_grammar.tab.[ch]	See (APIMODE parser license terms and conditions) (page 126)
VxWorks	Wind River Systems, Inc.	systems/vxworks	See (Wind River Systems license terms and conditions) (page 127)

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: 1.8 $

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>

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 123) for the Bison licence.

WIND RIVER SYSTEMS LICENSE TERMS AND CONDITIONS

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